Nonlinear Estimation Techniques for Navigation

Michael J. Veth
Veth Research Associates, LLC
Niceville, FL 32578
USA

michael.veth@vethresearch.com

ABSTRACT

Optimal estimation techniques have revolutionized the integration of multiple sensors for navigation applications. These estimation techniques typically make assumptions about the sensor measurements, namely the sensor measurements and errors are well modeled as linear, Gaussian systems. Unfortunately, there is a large class of potential navigation sources which are non-linear, non-Gaussian or both. This motivates the development and exploration of nonlinear estimation techniques suitable for integrated navigation systems.

This paper presents an overview of estimation techniques suitable for systems with nonlinearities which are not well-suited to traditional linear or extended Kalman filter algorithms. The paper begins with a description of the generalized recursive estimation problem and associated notation and conventions. Next, the limitations of applying linear theory to nonlinear problems are addressed, along with techniques for compensating for these adverse effects, including a brief overview of the traditional extended Kalman filter. In addition, the mathematical effects of system nonlinearities on random processes are presented along with computational techniques for efficiently capturing this information, which serves as the foundation for the development of many nonlinear estimators. Next, the unscented Kalman filter (UKF) and particle filters (PF) are presented and analyzed. Common limitations of nonlinear estimators are addressed and hybrid solutions are discussed. The paper concludes with a discussion and qualitative comparison of the strengths and weaknesses of various recursive estimation techniques from linear Kalman filtering to particle filtering, and their applicability to various problem spaces related to military navigation requirements.

1.0 INTRODUCTION AND BACKGROUND

Military leaders have long understood the need for reliable, accurate navigation information. The ability to answer the simple question of "Where am I?" is fundamental to the classical Clausewitzean principles of offensive, maneuver, economy of force, and mass. As such, nations have invested significant resources into the development of robust navigation solutions.

In order to be effective, navigation systems must be designed to meet mission requirements in four primary areas: accuracy, integrity, continuity, and availability. Accuracy is the ability of the system to estimation position and/or orientation relative to the true values. Integrity is the ability of the system to identify when accuracy is not meeting desired levels. Continuity is a measure of the system's capability to provide uninterrupted, quality measurements. Availability is the overall percentage of time the system is able to provide accurate navigation information. Meeting all of these requirements can be very challenging, depending on the mission.

There are two primary decisions a navigation system designer must make in order to meet performance specifications, namely, which (and what number of) sensors to use and how to integrate the sensors into one solution. One common example is using updates from a Global Positioning System (GPS) receiver to aid an inertial navigation unit via a Kalman filter algorithm. In many cases, additional complementary sensors can be added to improve the system robustness including: radar measurements, Doppler velocity sensors, barometric altimeters, sonar, and celestial navigation sensors, to name a few.

The continually increasing requirements for cooperative engagement in dense urban environments and in areas where global navigation satellite system (GNSS) coverage is degraded or denied has spurned the development of novel navigation sensors. Some examples of these include: optical aiding, laser scanners, magnetic field sensors, acoustic sensors, pedometry, etc. Many of these sensors utilize signals of opportunity, i.e., signals not designed for navigation. As a result, these sensors tend to have nonlinear error statistics.

While the standard Kalman filter works very well when system errors are well-described by Gaussian probability density functions (pdf), the algorithm can suffer performance degradation or failure when these assumptions are not met. This paper will focus on alternative recursive estimation algorithms that are able to address nonlinear and non-Gaussian navigation problems. The article is arranged as follows. First, a basic overview of the stochastic estimation problem is presented along with the background of the linear Kalman filter. Next, the discussion is expanded to include nonlinear stochastic system models which sets the stage for the discussion of nonlinear estimators. Finally, three classes of nonlinear estimators are presented including: Multiple Model Adaptive Estimators (MMAE), Unscented Kalman Filters (UKF), and particle filters (PF). Conclusions are drawn regarding strengths and weaknesses of the various classes of nonlinear estimators.

2.0 BACKGROUND ON STOCHASTIC ESTIMATION

The foundation of estimation theory is the development of a stochastic model for a given system of interest. This is motivated by the observation that most real-world systems experience some degree of random behavior [1]. Sensors never make perfect measurements. Manufacturing variances cause random variation in the performance of components. Even if a system could be created perfectly, it must still interact with uncontrollable, random disturbances in the environment. Fortunately, stochastic models can be developed that treat these unknowns in a statistically-rigorous manner.

A typical recursive estimation algorithm consists of two distinct operations: propagation through time and measurement updates. The estimator is tasked with maintaining an estimate of the state pdf conditioned on the collection of measurements. A sample sequence from time *k*-1 to time *k* is

$$p(\mathbf{x}_{k-1}|\mathbb{Z}_{k-1}) \xrightarrow[propagate]{} p(\mathbf{x}_k|\mathbb{Z}_{k-1}) \xrightarrow[update]{} p(\mathbf{x}_k|\mathbb{Z}_k) \xrightarrow[propagate]{} \tag{1}$$

where $\mathbf{x}_k$ is the system state vector and $\mathbb{Z}_k$ is the collection of observations given by

$$\mathbb{Z}_k = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k\} \tag{2}$$

2.1 Linear Stochastic Systems

A typical linear system model consists of both a dynamic model and a model of the measurement devices. The dynamics model can be expressed as a linear stochastic difference equation given by

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathsf{w}_{k-1} \tag{3}$$

where $\mathbf{x}_k$ is the state vector at time *k*, $\mathbf{\Phi}_{k-1}$ is the state transition matrix at time *k-1*, $\mathbf{x}_{k-1}$ is the state vector at time *k-1*, $\mathbf{B}_{k-1}$ is the control influence matrix at time *k-1*, and $\mathbf{u}_{k-1}$ is the control vector at time *k-1*, and w_{k-1} is a random vector representing the uncertainty in the dynamics model. The additive noise vector is zero-mean and Gaussian with

$$E[\mathsf{w}_j\mathsf{w}_k^T] = \mathbf{Q}_k\delta_{jk} \tag{4}$$

where $E[\cdot]$ is the expectation operator, $\mathbf{Q}_k$ is the process noise covariance matrix and δ_{jk} is the Kronecker delta function.

The measurement model can be expressed as:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathsf{v}_k \tag{5}$$

where $\mathbf{z}_k$ is the observation vector, $\mathbf{H}_k$ is the observation matrix, and v_k is a random vector representing the uncertainty in the measurement model. The measurement noise vector is zero-mean and Gaussian with

$$E[\mathsf{v}_j\mathsf{v}_k^T] = \mathbf{R}_k\delta_{jk} \tag{6}$$

where $\mathbf{R}_k$ is the measurement noise covariance matrix.

In the most general case, the probability density function (pdf) of the state vector, conditioned on the prior measurements, provides all possible statistics that could be used to develop a state estimate. The conditional pdf is given by

$$p(\mathbf{x}_k|\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \tag{7}$$

where $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k$ represents the collection of observations up to, and including, time *k*. This is also known as the *a posteriori* state pdf. An example of a typical Kalman filter application is shown in Figure 1.

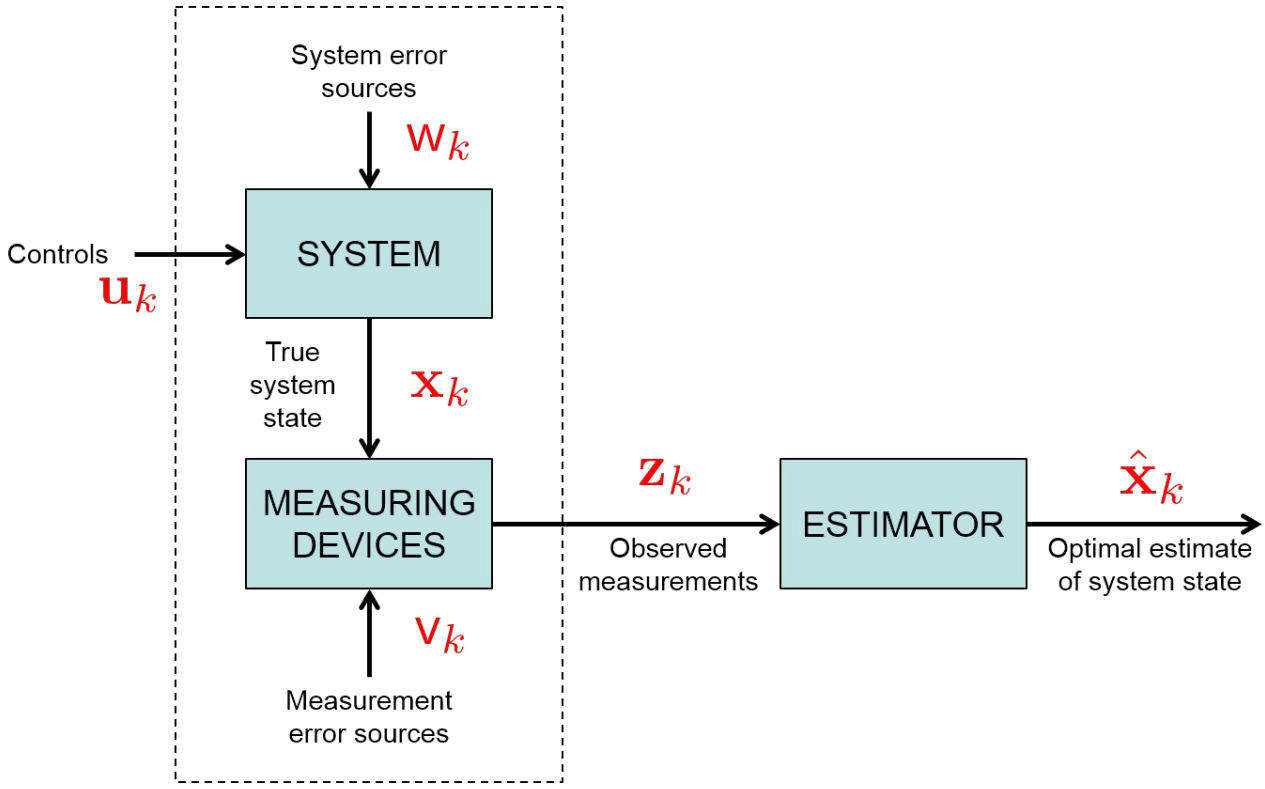

Figure 1. Typical Kalman Filter Application. The estimator produces an optimal estimate of the system state based on observed measurements and known control inputs.

2.2 Linear Kalman Filter

The *a posteriori* pdf is guaranteed to be Gaussian under the following conditions:

- The system dynamics and measurement components consist exclusively of linear models
- The system and measurement noises are white and Gaussian
- All initial conditions can be represented as Gaussian random variables

In this case, the entire statistics of the pdfs of interest can be expressed as a mean vector and covariance matrix. This unique property is exploited by the linear Kalman filter.

The Kalman filter has two distinct steps: propagation of the mean and covariance between time epochs (i.e., propagation), and incorporating a discrete observation (i.e., update). Given the linear stochastic system model driven by white Gaussian noise sources shown in (3) and (5), the Kalman filter propagation equations are given by

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{8}$$

and

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \tag{9}$$

where $\hat{\mathbf{x}}_k^-$ is the *a priori* state estimate at time *k*, $\mathbf{\Phi}_{k-1}$ is the state transition matrix at time *k-1*, $\hat{\mathbf{x}}_{k-1}^+$ is the *a posteriori* state estimate at time *k-1*, $\mathbf{P}_k^-$ is the *a priori* state covariance matrix at time *k*, $\mathbf{P}_{k-1}^+$ is the *a posteriori* state covariance matrix at time *k-1*, and $\mathbf{Q}_{k-1}$ is the process noise covariance matrix at time *k-1*.

The Kalman filter measurement update equations are:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathrm{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right) \tag{10}$$

and

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^- \tag{11}$$

The Kalman gain, $\mathbf{K}_k$, is

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \tag{12}$$

The linear Kalman filter algorithm is optimal by every conceivable measure, is easily implemented on a digital computer, and has served as the benchmark standard of recursive estimation algorithms for decades [4].

2.3 Effects of Nonlinearities on the Linear Kalman Filter

While many systems are well-modeled by linear stochastic equations, most real-world applications are nonlinear at some level. There are many types of nonlinearities to consider. Some examples include non-Gaussian noise sources, saturation effects, nonlinear dynamics or measurement models, and jump discontinuities. All of these effects ultimately result in the true conditional state pdfs being non-Gaussian in nature, which violates the fundamental assumptions of the linear Kalman filter [3].

If the degree of nonlinearity is relatively small, the extended Kalman filter (EKF) can provide acceptable results [3]. The EKF filter design is based on linearization of the system and measurement models using a first-order Taylor series expansion.

Although the EKF is very widely used in navigation applications, there are limitations that should be understood. First, the EKF is subject to linearization errors. These linearization errors result in incorrect state estimates and covariance estimates and can lead to unstable operation, known as filter divergence. EKFs can be extremely sensitive to this effect during periods of relatively high state uncertainty such as initialization and start-up. The second issue is the inherently unimodal assumption of the EKF. In cases where multi-modal densities are known to exist, the EKF would not be a good choice.

In the following sections, various types of nonlinear estimation approaches will be addressed. Each approach will be presented along with a typical navigation application. The first approach to address non-Gaussian densities is the class of Gaussian Sum Filters, namely the Multiple Model Adaptive Estimation technique.

2.4 Multiple Model Adaptive Estimation (MMAE)

The Multiple Model Adaptive Estimation (MMAE) filter extends the linear Kalman filter to address the situation of unknown or uncertain parameters in the system model. MMAE is in the class of estimators known as

Gaussian Sum Filters. Others in the class include the Interacting Multiple Model (IMM) filter and the Rao-Blackwell particle filter. Some examples of applications that fall into this category include: system failure modes, unknown structural parameters, "jump" processes, etc. [3].

The MMAE filter represents the state pdf using a sum of individually-weighted Gaussian densities. This allows the filter to properly account for complex, non-Gaussian densities. In this way, the effects of the uncertain parameters on the overall state pdf are explicitly and directly maintained inside the estimator. An example of a sum of Gaussian density is shown in Figure 2.

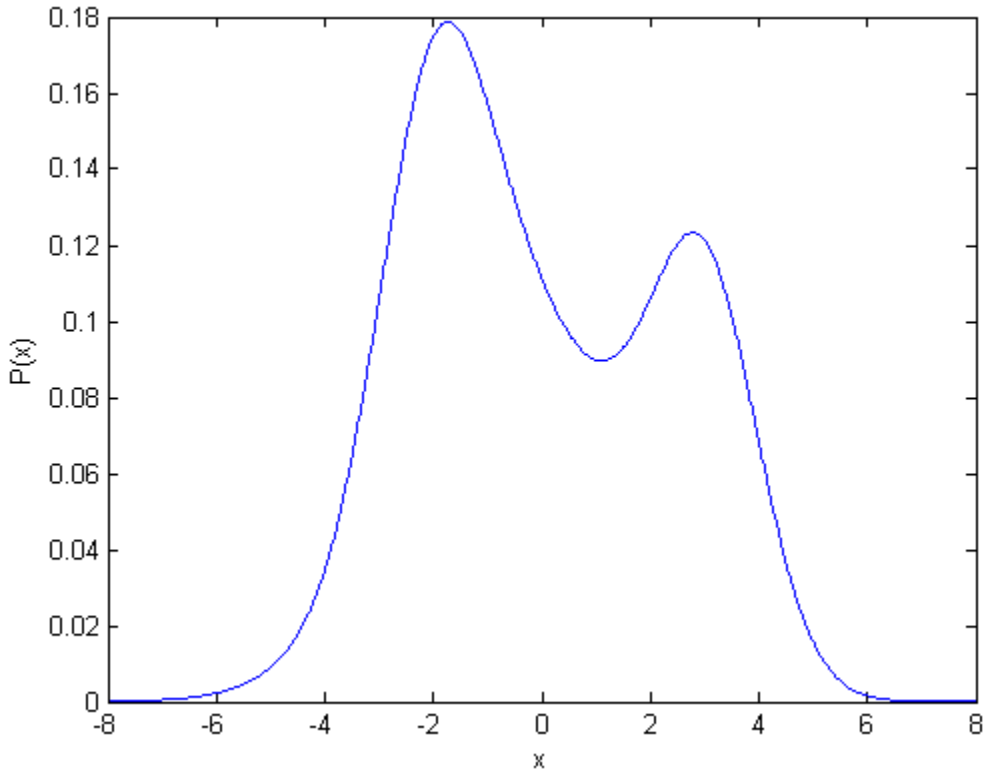

Figure 2: Example Sum of Gaussian Density. Gaussian sums can be used to represent a wide variety of densities, including multi-modal densities as shown in this example.

Consider the system represented by the linear stochastic difference equation (3)

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathsf{w}_{k-1} \tag{13}$$

and corresponding measurement equation (5)

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathsf{v}_k \tag{14}$$

which are repeated here for clarity. Similarly, the noise sources, w and v, are zero-mean and Gaussian with covariances

$$E[\mathsf{w}_j \mathsf{w}_k^T] = \mathbf{Q}_k \delta_{jk} \tag{15}$$

and

$$E[\mathsf{v}_j \mathsf{v}_k^T] = \mathbf{R}_k \delta_{jk} \tag{16}$$

respectively.

In contrast to the standard linear Kalman filter, consider the additional situation where a portion of the system model parameters are unknown. These unknowns could be in the structure of the dynamics model (i.e., $\mathbf{\Phi}$, $\mathbf{B}$), in the structure of the measurement model (i.e., $\mathbf{H}$), or in the statistics of the uncertainties (i.e., $\mathbf{Q}$, $\mathbf{R}$). Some examples of applications with these uncertainties include: systems with unknown or dynamic sensor biases or scale factors, system with changing operating conditions that affect the linearized model, and tracking applications where the characteristics of the target are unknown.

The estimation problem becomes that of a dual estimator. In other words, the unknown parameters must be included in the unknowns. The MMAE estimator derivation proceeds by defining a new random vector, $\mathbf{a}$, that contains the unknown parameters. The *a posteriori* pdf is augmented to create a new, joint *a posteriori* pdf of interest

$$p(\mathbf{x}_k, \mathbf{a} | \mathbb{Z}_k) \tag{17}$$

where $\mathbf{x}_k$ is the system state vector and $\mathbb{Z}_k$ is the collection of observations given by

$$\mathbb{Z}_k = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k\} \tag{18}$$

Equation (17) can be rewritten using Bayes' rule as

$$p(\mathbf{x}_k, \mathbf{a} | \mathbb{Z}_k) = p(\mathbf{x}_k | \mathbf{a}, \mathbb{Z}_k) p(\mathbf{a} | \mathbb{Z}_k) \tag{19}$$

Assuming the parameter vector is of finite dimension n (i.e., $\mathbf{a} \in \mathbb{R}^n$) the parameter density can be expressed as

$$p(\mathbf{a} | \mathbb{Z}_k) = p(\mathbf{a} | \mathbf{z}_k, \mathbb{Z}_{k-1}) \tag{20}$$

Invoking Bayes' rule once again yields

$$p(\mathbf{a} | \mathbb{Z}_k) = \frac{p(\mathbf{a}, \mathbf{z}_k | \mathbb{Z}_{k-1})}{p(\mathbf{z}_k | \mathbb{Z}_{k-1})} \tag{21}$$

Finally, invoking Bayes' rule on the numerator and expressing the denominator as equivalent integral of the marginal density on $\mathbf{a}$ gives the following relation

$$p(\mathbf{a} | \mathbb{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{a}, \mathbb{Z}_{k-1}) p(\mathbf{a} | \mathbb{Z}_{k-1})}{\int p(\mathbf{z}_k | \mathbf{a}, \mathbb{Z}_{k-1}) p(\mathbf{a} | \mathbb{Z}_{k-1}) da} \tag{22}$$

It should be noted that the term $p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1})$ is the measurement prediction density conditioned on a *known* parameter vector $\mathbf{a}$ and all prior measurements. Given the system model from (13) and (14), this pdf is given by the following Gaussian density

$$p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1}) := \mathcal{N}(\mathrm{z}_k; \mathbf{H}_k\hat{\mathbf{x}}_k^-, \mathbf{S}_k) \tag{23}$$

where $\mathbf{S}_k$ is the residual covariance given by

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k \tag{24}$$

Finally, the term $p(\mathbf{a}|\mathbb{Z}_{k-1})$ is the *a priori* density of the parameter vector. Unfortunately, the integral in the denominator of (22) is intractable in general. However, if the system parameters can be chosen from a finite set, i.e., $\mathbf{a} \in \left\{\mathbf{a}^{[1]}, \mathbf{a}^{[2]}, \ldots, \mathbf{a}^{[J]}\right\}$ then the parameter density can be expressed as the sum of the individual probabilities of the finite set

$$p(\mathbf{a}|\mathbb{Z}_{k-1}) = \sum_{j=1}^{J} w_{k-1}^{[j]}\delta(\mathbf{a} - \mathbf{a}^{[j]}) \tag{25}$$

where $w_{k-1}^{[j]}$ is the probability weight of the *j-th* parameter vector, $\mathbf{a}^{[j]}$.

Substituting (25) into (22) yields

$$p(\mathbf{a}|\mathbb{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1})\sum_{j=1}^{J} w_{k-1}^{[j]}\delta(\mathbf{a} - \mathbf{a}^{[j]})}{\int p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1})\sum_{n=1}^{J} w_{k-1}^{[n]}\delta(\mathbf{a} - \mathbf{a}^{[n]})da} \tag{26}$$

Because the probability weights are independent of the integration, the order of integration can be reversed which results in

$$p(\mathbf{a}|\mathbb{Z}_k) = \frac{\sum_{j=1}^{J} w_{k-1}^{[j]}p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1})\delta(\mathbf{a} - \mathbf{a}^{[j]})}{\sum_{n=1}^{J} w_{k-1}^{[n]}\int p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1})\delta(\mathbf{a} - \mathbf{a}^{[n]})da} \tag{27}$$

which can be further reduced using the sifting property on the denominator

$$p(\mathbf{a}|\mathbb{Z}_k) = \frac{\sum_{j=1}^{J} w_{k-1}^{[j]} p(\mathbf{z}_k|\mathbf{a}, \mathbb{Z}_{k-1}) \delta(\mathbf{a} - \mathbf{a}^{[j]})}{\sum_{n=1}^{J} w_{k-1}^{[n]} p(\mathbf{z}_k|\mathbf{a}^{[n]}, \mathbb{Z}_{k-1})} \tag{28}$$

The *j*-th probability weighting at time *k* can be defined as the following recursive relationship

$$w_k^{[j]} = \frac{p(\mathbf{z}_k|\mathbf{a}^{[j]}, \mathbb{Z}_{k-1})}{\sum_{n=1}^{J} w_{k-1}^{[n]} p(\mathbf{z}_k|\mathbf{a}^{[n]}, \mathbb{Z}_{k-1})} w_{k-1}^{[j]} \tag{29}$$

In a similar fashion to (23), the term $p(\mathbf{z}_k|\mathbf{a}^{[j]}, \mathbb{Z}_{k-1})$ is the measurement prediction density based on the *j-th* model. This pdf is given by the following Gaussian density

$$p(\mathbf{z}_k|\mathbf{a}^{[j]}, \mathbb{Z}_{k-1}) := \mathcal{N}(\mathrm{z}_k; \mathbf{H}_k^{[j]} \hat{\mathbf{x}}_k^{-[j]}, \mathbf{S}_k^{[j]}) \tag{30}$$

where z_k is the measurement realization at time *k*, $\mathbf{H}_k^{[j]}$ is the *j-th* observation matrix at time *k*, $\hat{\mathbf{x}}_k^{-[j]}$ is the *j-th a priori* state estimate, and $\mathbf{S}_k^{[j]}$ is the *j-th* residual covariance. Once a measurement is realized, the *j-th* measurement likelihood (30) can be directly evaluated and the associated weight scaled.

Substituting (29) into (28) yields an updated expression of the *a posteriori* pdf

$$p(\mathbf{a}|\mathbb{Z}_k) = \sum_{j=1}^{J} w_k^{[j]} \delta(\mathbf{a} - \mathbf{a}^{[j]}) \tag{31}$$

Substituting (31) into (19)

$$p(\mathbf{x}_k, \mathbf{a}|\mathbb{Z}_k) = p(\mathbf{x}_k|\mathbf{a}, \mathbb{Z}_k) \sum_{j=1}^{J} w_k^{[j]} \delta(\mathbf{a} - \mathbf{a}^{[j]}) \tag{32}$$

and finally using the sifting property of the delta function yields a tractable expression for the *a posteriori* pdf

$$p(\mathbf{x}_k, \mathbf{a}|\mathbb{Z}_k) = \sum_{j=1}^{J} p(\mathbf{x}_k|\mathbf{a}^{[j]}, \mathbb{Z}_k) w_k^{[j]} \delta(\mathbf{a} - \mathbf{a}^{[j]}) \tag{33}$$

which, given the linear, stochastic system model given in (13) and (14) is simply the weighted sum of the output of *J* Kalman filters, each based on the parameter vector $\mathbf{a}^{[j]}$. This can be visualized as a bank of *J* independent Kalman filters as shown in Figure 3.

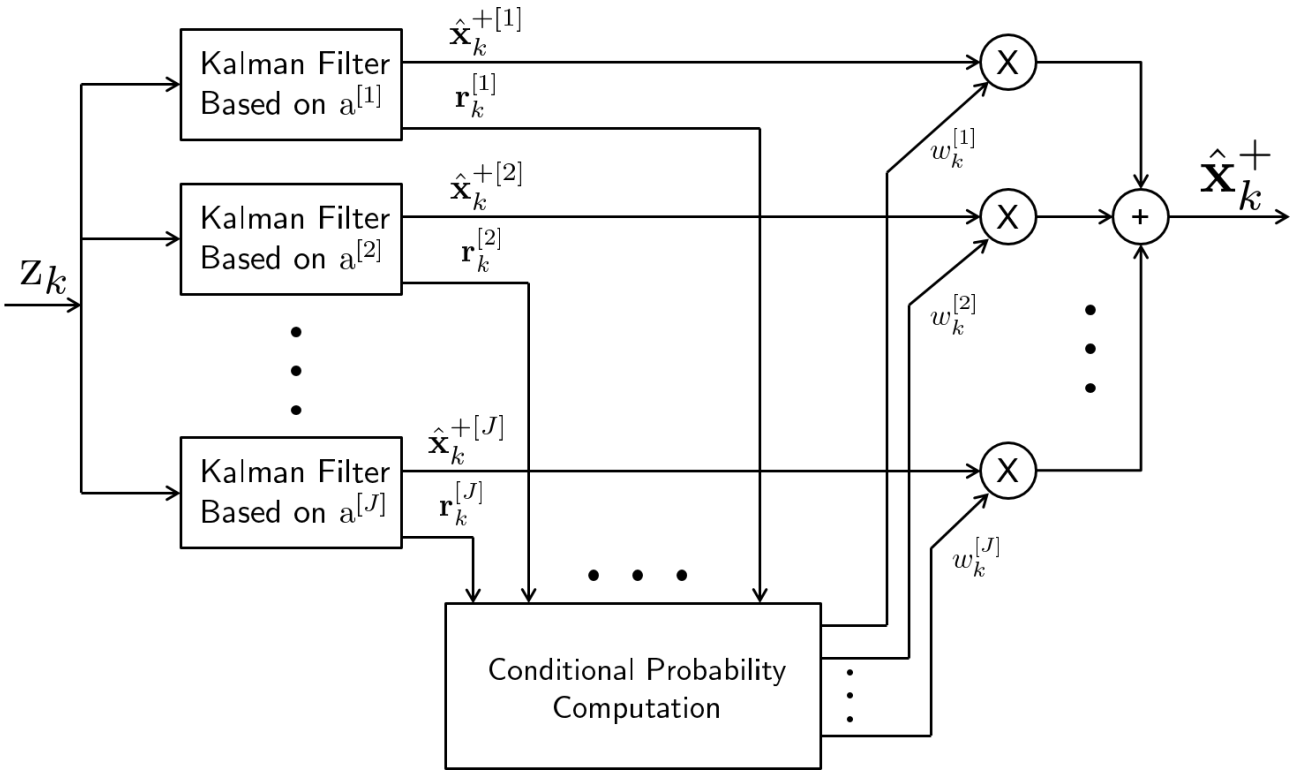

Figure 3: Typical MMAE block diagram. The MMAE estimator consists of a bank of independent Kalman filters, each based on a discrete parameter vector.

The blended state estimate and covariance is given by

$$\hat{\mathbf{x}}_k^+ = \sum_{j=1}^{J} w_k^{[j]} \hat{\mathbf{x}}_k^{+[j]} \tag{34}$$

and

$$\mathbf{P}_k^+ = \sum_{j=1}^{J} \left[\left(\hat{\mathbf{x}}_k^{+[j]} - \hat{\mathbf{x}}_k^+ \right) \left(\hat{\mathbf{x}}_k^{+[j]} - \hat{\mathbf{x}}_k^+ \right)^T + \mathbf{P}_k^{+[j]} \right] w_k^{[j]} \tag{35}$$

respectively.

The MMAE filter operating sequence is as follows. When an observation is made, the residual likelihood is evaluated for each filter and the corresponding model weight is scaled appropriately. Models that agree well with the residual sequence will be increased in weight. Models that do not agree with the residual sequence will decrease in weight.

2.6 Unscented Kalman Filter

As discussed previously, the recursive estimation problem is fundamentally concerned with estimating the conditional state density. When dynamic or measurement models are nonlinear, the linear, Gaussian assumptions of the linear Kalman filter are invalidated. The extended Kalman filter exploits a first-order Taylor series expansion to approximate the transformation of random vectors through the nonlinearity. Unfortunately, this first-order technique can fail when nonlinearities become large. In addition, the first-order Taylor series method requires calculation of a Jacobian matrix which is unnecessary in the unscented and particle filter algorithms.

The Unscented Kalman Filter (UKF) exploits the Unscented Transform (UT) to transform random vectors through nonlinearities. For Gaussian random variables, the UT captures the correct statistics equivalent to a third-order Taylor series [7]. In addition, the UT is computationally efficient compared to particle filtering algorithms.

2.6.1 Unscented Transform

The unscented transform represents a random vector's density as a collection of carefully-chosen "sigma points." The sigma points are chosen so that they can be directly transformed through a nonlinear (or linear) function while more accurately maintaining the statistics of true density. Consider the following generalized transformation, $\mathrm{y} = g(\mathrm{x})$ where x is a random vector with dimension L, y is a random vector with dimension M, and $g(\cdot)$ is a mapping function from $\mathrm{x} \rightarrow \mathrm{y}$. The mean and covariance of x is given by

$$E[\mathrm{x}] = \bar{\mathbf{x}} \tag{36}$$

and

$$E[(\mathrm{x} - \bar{\mathbf{x}})(\mathrm{x} - \bar{\mathbf{x}})^T] = \mathbf{P}_x \tag{37}$$

respectively.

The first step of the forward unscented transform is to calculate the collection of sigma points, $\mathcal{X}$, which is a deterministic function of $\bar{\mathbf{x}}$ and $\mathbf{P}_x$. The sigma point matrix consists of $2L + 1$ sigma points, $\mathcal{X}_k$, given by

$$\begin{aligned}
\mathcal{X}_0 &= \hat{\mathbf{x}} \\
\mathcal{X}_j &= \hat{\mathbf{x}} + \left[\sqrt[c]{(L+\lambda)\,\mathbf{P}_x}\right]_j \qquad \forall \;\; j \in \{1, \ldots, L\} \\
\mathcal{X}_j &= \bar{\mathbf{x}} - \left[\sqrt[c]{(L+\lambda)\,\mathbf{P}_x}\right]_{j-L} \quad \forall \;\; j \in \{L+1, \ldots, 2L\}
\end{aligned} \tag{38}$$

with the subscript notation representing the column number (e.g., $\mathbf{A}_j$ is the *j-th* column of the matrix $\mathbf{A}$), and $\sqrt[c]{\cdot}$ is the Cholesky square root. The scaling parameter, λ, is given by

$$\lambda = \alpha^2(L + \kappa) - L \tag{39}$$

The sigma point weights are given by the collection

$$\begin{aligned} W_0^m &= \frac{\lambda}{L+\lambda} \\ W_0^c &= \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta) \\ W_j^m &= W_j^c = \frac{1}{2(L+\lambda)} \end{aligned} \tag{40}$$

The scalars α, β, and κ are tuning parameters that vary the spread of the sigma points. Default values for Gaussian densities are $\alpha \approx 1 \times 10^{-3}, \beta = 2, \kappa = 0$.

Given a collection of sigma points, the mean and covariance of the random vector is calculated via

$$\begin{aligned} \hat{\bar{\mathbf{x}}} &= \sum_{j=0}^{2L} W_j^m \mathcal{X}_j \\ \hat{\mathbf{P}}_x &= \sum_{j=0}^{2L} W_i^c \left(\mathcal{X}_j - \hat{\bar{\mathbf{x}}}\right)\left(\mathcal{X}_j - \hat{\bar{\mathbf{x}}}\right)^T \end{aligned} \tag{41}$$

and is referred to as the reverse unscented transform.

A transformation of random variable (e.g., $\mathrm{x} \rightarrow \mathrm{y}$) is accomplished by simply passing each sigma point of x, namely $\mathcal{X}_i$, through the mapping function

$$\mathcal{Y}_j = \mathbf{g}\left(\mathcal{X}_j\right) \quad \forall \;\; j \in \{0, \ldots, 2L\} \tag{42}$$

to produce a transformed collection of sigma points, $\mathcal{Y}$. Each sigma point has dimension M, as given by the mapping function $g(\cdot)$. The mean and covariance of y can be recovered using the same procedure given in (41), namely

$$\begin{aligned} \hat{\bar{\mathbf{y}}} &= \sum_{j=0}^{2L} W_j^m \mathcal{Y}_j \\ \hat{\mathbf{P}}_y &= \sum_{j=0}^{2L} W_j^c \left(\mathcal{Y}_j - \hat{\bar{\mathbf{y}}}\right)\left(\mathcal{Y}_j - \hat{\bar{\mathbf{y}}}\right)^T \end{aligned} \tag{43}$$

In the next section, the unscented transform will be used as the basis for a recursive estimation algorithm known as the Unscented Kalman Filter (UKF).

2.6.2 Unscented Kalman Filter

The UKF algorithm is conceptually similar to the linear Kalman filter. There are two distinct steps: propagation of the state vector density over time, and updating the state vector density based on a measurement observation.

Consider the nonlinear stochastic system model given by

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}\left[\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right] + \mathsf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{h}\left[\mathbf{x}_k\right] + \mathsf{v}_k \end{aligned} \tag{44}$$

where the process noise, w_k, and measurement noise, v_k, are both zero-mean, independent, white Gaussian random vectors with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. In addition, assume the *a posteriori* state estimate and covariance is available at time *k-1*, (i.e., $\hat{\mathbf{x}}_{k-1}^{+}$ and $\mathbf{P}_{x_{k-1}}^{+}$).

The first step is to propagate the state vector from time *k-1* to time *k*. This is accomplished using the forward unscented transform to create sigma points at time *k-1*, represented by $\mathcal{X}^{+}$. The sigma points are transformed through the nonlinear system state transition function

$$\mathcal{X}_j^{-} = \mathbf{f}[\mathcal{X}_j^{+}, \mathbf{u}_{k-1}] \;\; \forall \;\; j \in \{0, \ldots, 2L\} \tag{45}$$

The mean and covariance of $\mathcal{X}^{-}$ are calculated using the backward unscented transform and the effects of the process noise are incorporated to produce the *a priori* state estimate and covariance at time *k*

$$\begin{aligned} \hat{\mathbf{x}}_k^{-} &= \sum_{j=0}^{2L} W_j^m \mathcal{X}_j^{-} \\ \mathbf{P}_{x_k}^{-} &= \sum_{j=0}^{2L} W_j^c \left(\mathcal{X}_j^{-} - \hat{\mathbf{x}}_k^{-}\right)\left(\mathcal{X}_j^{-} - \hat{\mathbf{x}}_k^{-}\right)^T + \mathbf{Q}_{k-1} \end{aligned} \tag{46}$$

A block diagram of the UKF propagation algorithm is shown in Figure 4.

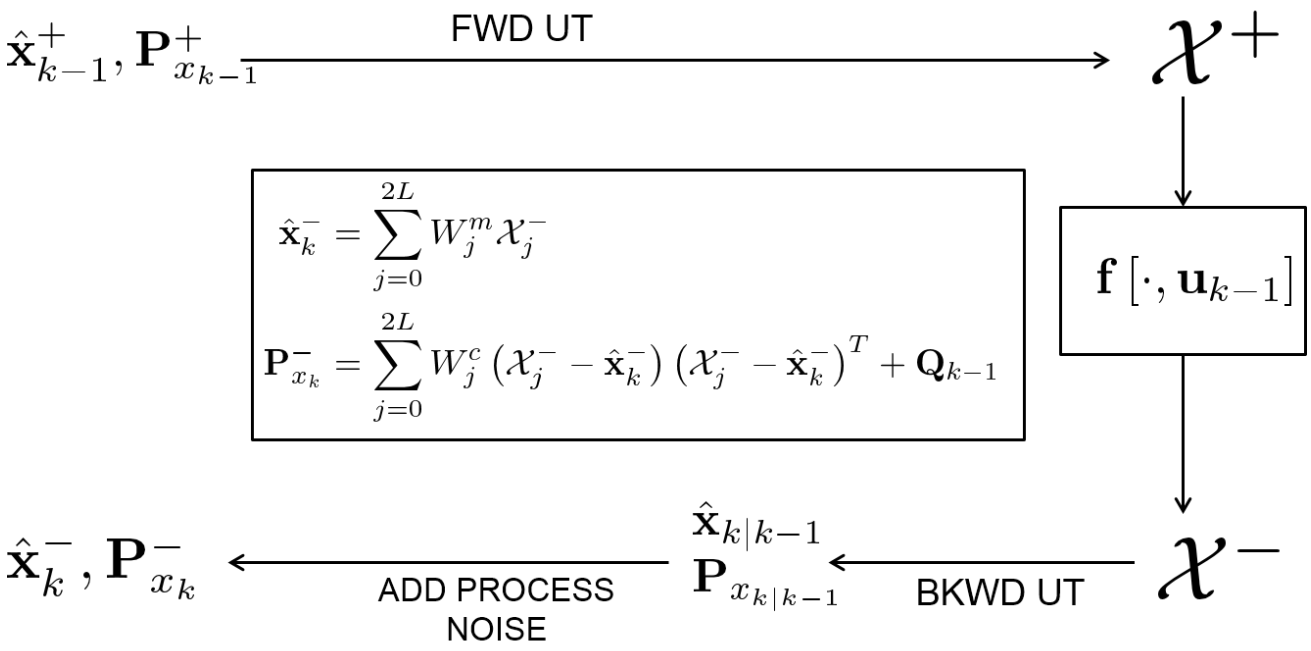

Figure 4: Unscented Kalman Filter Propagation.

The measurement update proceeds similarly, beginning with the transformation of the *a priori* mean and covariance into a collection of sigma points, $\mathcal{X}^-$. These sigma points are transformed into the measurement space via

$$\mathcal{Z}_j^- = \mathbf{h}[\mathcal{X}_j^-] \;\; \forall \;\; j \in \{0, \ldots, 2L\} \tag{47}$$

The measurement mean, $\hat{\mathbf{z}}_k$, covariance, $\mathbf{P}_{z_k}$, and cross-correlation matrix, $\mathbf{P}_{xz_k}$ are calculated as

$$\begin{aligned}
\hat{\mathbf{z}}_k &= \sum_{j=0}^{2L} W_j^m \mathcal{Z}_j \\
\mathbf{P}_{z_k} &= \sum_{j=0}^{2L} W_j^c \left(\mathcal{Z}_j - \hat{\mathbf{z}}_k\right)\left(\mathcal{Z}_j - \hat{\mathbf{z}}_k\right)^T + \mathbf{R}_k \\
\mathbf{P}_{xz_k} &= \sum_{j=0}^{2L} W_j^c \left(\mathcal{X}_j^- - \hat{\mathbf{x}}_k^-\right)\left(\mathcal{Z}_j - \hat{\mathbf{z}}_k\right)^T
\end{aligned} \tag{48}$$

Finally, the *a posteriori* statistics are calculated using the standard Kalman update algorithms, albeit in a slightly different, but mathematically equivalent, form. The Kalman gain is given by

$$\mathbf{K} = \mathbf{P}_{xz_k}\mathbf{P}_{z_k}^{-1} \tag{49}$$

and the state estimate and covariance are given by

$$\begin{aligned}\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}\left(\mathsf{z}_k - \hat{\mathbf{z}}_k\right)\\ \mathbf{P}_{x_k}^+ &= \mathbf{P}_{x_k}^- - \mathbf{K}\mathbf{P}_{z_k}\mathbf{K}^T\end{aligned} \tag{50}$$

A block diagram of the UKF measurement update process is shown in Figure 5.

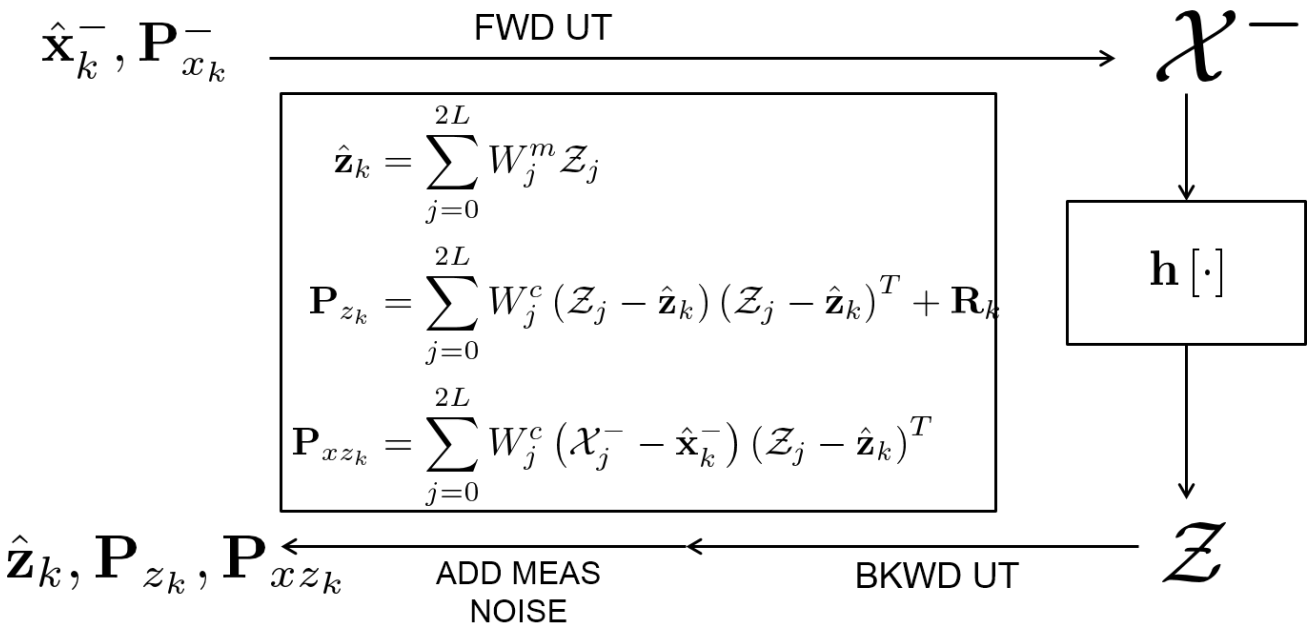

Figure 5: Unscented Kalman Filter Measurement Update.

In addition to the above algorithm, other UKF algorithms exist to address the case of non-additive noise sources. The reader is referred to the literature [2],[5] for more details.

As mentioned previously, the UKF has some important advantages over the linear or extended Kalman filter. First, the UKF algorithm captures higher order statistics than the linear or extended Kalman filter. In addition, the requirement to calculate Jacobians is completely eliminated resulting in less chance of errors, especially for complicated models. However, there are some disadvantages of the UKF that need to be considered. First, the UKF requires slightly more processing time than the equivalent EKF. Finally, the UKF is still unable to capture the statistics of multi-modal densities.

2.7 Particle Filter

The class of recursive estimators known as particle filters utilize a different approach for representing the state pdf, which has the potential to accurately maintain the statistics of interest at an arbitrary level. Two main classes of particle filter exist: grid-based or discrete methods, and continuous methods. Due to space limitations, only the grid-based particle filter algorithm will be developed in this article. For additional information on other types of particle filters, the reader is referred to [5].

2.7.1 Propagation

The propagation of the state pdf through time incorporates the stochastic effects of the system model. This requires knowledge of the transition pdf. To accomplish this mathematically, the property of density marginalization is required. Recall the marginalization integral

$$p(a) = \int p(a|b)p(b)db \tag{51}$$

Substituting our *a priori* density at time *k* for a and the *a posteriori* density at time *k-1* for b yields

$$p(\mathbf{x}_k|\mathbb{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbb{Z}_{k-1})p(\mathbf{x}_{k-1}|\mathbb{Z}_{k-1})d\mathbf{x}_{k-1} \tag{52}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbb{Z}_{k-1})$ is the transition density. Because we assume a first-order Markovian system model, the transition density is *independent* of the measurement observations, thus

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbb{Z}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{53}$$

Substituting (53) into (52) yields the well-known Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbb{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbb{Z}_{k-1})d\mathbf{x}_{k-1} \tag{54}$$

2.7.2 Measurement Update

The measurement update incorporates the state knowledge gained during the observation by exploiting the measurement model. The measurement update is derived by recalling Bayes' rule

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)} \tag{55}$$

The derivation begins by separating the current measurement from the measurement collection in the *a posteriori* pdf

$$p(\mathbf{x}_k|\mathbb{Z}_k) = p(\mathbf{x}_k|\mathbf{z}_k, \mathbb{Z}_{k-1}) \tag{56}$$

Applying Bayes' rule to the right-hand side of (56) yields

$$p(\mathbf{x}_k|\mathbb{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbb{Z}_{k-1})p(\mathbf{x}_k|\mathbb{Z}_{k-1})}{p(\mathbf{z}_k|\mathbb{Z}_{k-1})} \tag{57}$$

Because the conditional observation is density is independent of the prior observations, the following simplification can be applied

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbb{Z}_{k-1}) = p(\mathbf{z}_k|\mathbf{x}_k) \tag{58}$$

Substituting (58) into (57) yields the measurement update equation

$$p(\mathbf{x}_k|\mathbb{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbb{Z}_{k-1})}{p(\mathbf{z}_k|\mathbb{Z}_{k-1})} \tag{59}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the measurement likelihood, and $p(\mathbf{z}_k|\mathbb{Z}_{k-1})$ is referred to as the *evidence*. The evidence effectively serves as a normalization factor to ensure unit area of the *a posteriori* pdf. As a final note, applying the marginalization integral to the evidence yields an equivalent form that will aid our development of the particle filtering algorithms

$$p(\mathbf{z}_k|\mathbb{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbb{Z}_{k-1})d\mathbf{x}_k \tag{60}$$

2.8 Discrete Particle Filtering Algorithm

The discrete particle filter is conditioned on the assumption of partitioning the state space into a finite set of discrete states. A discrete random variable is characterized by the probability mass function (pmf). The pmf consists of a collection of states and corresponding probability weights. The sum of all weights must equal one.

The delta function can be used to create a mathematical representation of a pmf, for example

$$p(\mathbf{x}_k) = \sum_{j=1}^{N} w^{[j]}\delta(\mathbf{x}_k - \mathbf{x}_k^{[j]}) \tag{61}$$

where $w^{[j]}$ is the probability weight for particle *j*, and $\mathbf{x}_k^{[j]}$ is the discrete state location of particle *j*. Thus, the *a posteriori* state pmf is given by

$$p(\mathbf{x}_{k-1}|\mathbb{Z}_{k-1}) = \sum_{j=1}^{N} w_{k-1|k-1}^{[j]}\delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{[j]}) \tag{62}$$

where $w_{k-1|k-1}^{[j]}$ represents the probability weight for particle *j* at time *k-1*, conditioned on measurements up-to, and including, time *k-1*.

2.8.1 Propagation Relations

Because the discrete states are at fixed locations, only the particle weights change as an effect of propagation and measurement updates. The propagation weighting function is derived using the Chapman-Kolmogorov Equation, presented in the previous section. Substituting the *a posteriori* pmf into (54) yields

$$p(\mathbf{x}_k|\mathbb{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) \sum_{j=1}^{N} w_{k-1|k-1}^{[j]} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{[j]}) d\mathbf{x}_{k-1} \tag{63}$$

Changing the order of integration and simplifying using the sifting property gives

$$p(\mathbf{x}_k|\mathbb{Z}_{k-1}) = \sum_{j=1}^{N} w_{k-1|k-1}^{[j]} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{[j]}) \tag{64}$$

Because the state vector is discrete, the transition density can be expressed as

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^{[j]}) = \sum_{l=1}^{N} p(\mathbf{x}_k^{[l]}|\mathbf{x}_{k-1}^{[j]}) \delta(\mathbf{x}_k - \mathbf{x}_k^{[l]}) \tag{65}$$

Substituting (65) into (64) and grouping terms

$$p(\mathbf{x}_k|\mathbb{Z}_{k-1}) = \sum_{l=1}^{N} \left[\sum_{j=1}^{N} w_{k-1|k-1}^{[j]} p(\mathbf{x}_k^{[l]}|\mathbf{x}_{k-1}^{[j]}) \right] \delta(\mathbf{x}_k - \mathbf{x}_k^{[l]}) \tag{66}$$

The bracketed term represents the particle weights that have been updated by the transition probability (i.e., *a priori* particle weights). Using the notation presented previously, the *a priori* particle weights are formally given by

$$w_{k|k-1}^{[l]} = \sum_{j=1}^{N} w_{k-1|k-1}^{[j]} p(\mathbf{x}_k^{[l]}|\mathbf{x}_{k-1}^{[j]}) \tag{67}$$

and the corresponding pmf is given by

$$p(\mathbf{x}_k|\mathbb{Z}_{k-1}) = \sum_{j=1}^{N} w_{k|k-1}^{[j]} \delta(\mathbf{x}_k - \mathbf{x}_k^{[j]}) \tag{68}$$

2.8.2 Measurement Update

The measurement update weighting function is determined by substituting the *a priori* pmf (68) and (60) into (59)

$$p(\mathbf{x}_k|\mathbb{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) \sum_{j=1}^{N} w_{k|k-1}^{[j]} \delta(\mathbf{x}_k - \mathbf{x}_k^{[j]})}{\int p(\mathbf{z}_k|\mathbf{x}_k) \sum_{l=1}^{N} w_{k|k-1}^{[l]} \delta(\mathbf{x}_k - \mathbf{x}_k^{[l]}) d\mathbf{x}_k} \tag{69}$$

Applying the sifting property and simplifying yields

$$p(\mathbf{x}_k|\mathbb{Z}_k) = \frac{\sum_{j=1}^{N} w_{k|k-1}^{[j]} p(\mathbf{z}_k|\mathbf{x}_k^{[j]}) \delta(\mathbf{x}_k - \mathbf{x}_k^{[j]})}{\sum_{l=1}^{N} w_{k|k-1}^{[l]} p(\mathbf{z}_k|\mathbf{x}_k^{[l]})} \tag{70}$$

Grouping the weighting terms and expressing using the notation presented previously results in

$$p(\mathbf{x}_k|\mathbb{Z}_k) = \sum_{j=1}^{N} w_{k|k}^{[j]} \delta(\mathbf{x}_k - \mathbf{x}_k^{[j]}) \tag{71}$$

with the *j-th a posteriori* particle weight given by

$$w_{k|k}^{[j]} = \frac{w_{k|k-1}^{[j]} p(\mathbf{z}_k|\mathbf{x}_k^{[j]})}{\sum_{l=1}^{N} w_{k|k-1}^{[l]} p(\mathbf{z}_k|\mathbf{x}_k^{[l]})} \tag{72}$$

3.0 EXAMPLE APPLICATION

Consider the following two-dimensional vehicle navigation scenario shown in Figure 6. The vehicle is equipped with a sensor that receives an RF, carrier-phase ranging signal from a constellation of transmitters. The transmitters are positioned in circular orbits around the area of interest. For simplicity, time synchronization is assumed between all devices.

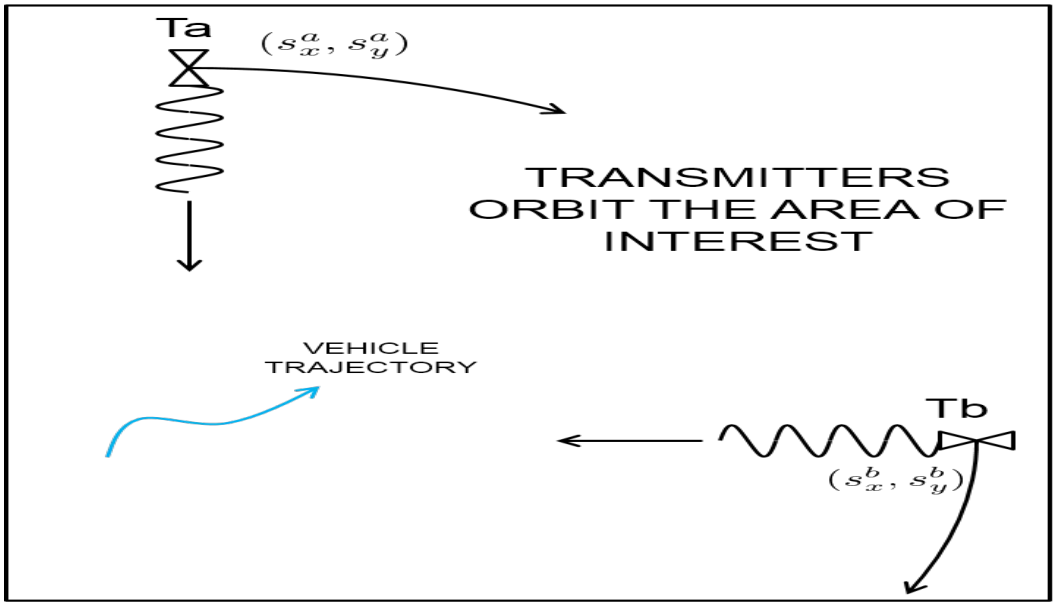

Figure 6: Simple Navigation Scenario. The transmitters orbit the area of interest while broadcasting a navigation signal.

The vehicle moves at a known, constant velocity. The heading can be represented as a first-order Gauss-Markov (FOGM) process. Thus, the vehicle stochastic dynamics model can be expressed as

$$\begin{aligned} x_k &= x_{k-1} + V\cos(\psi_{k-1})\,\Delta t \\ y_k &= y_{k-1} + V\sin(\psi_{k-1})\,\Delta t \\ \psi_k &= e^{-(\Delta t/\tau_\psi)}\psi_{k-1} + \mathsf{w}_{k-1} \end{aligned} \tag{73}$$

where x_k and y_k are the x-position and y-position at time k, respectively, ψ_k is the heading, V is the vehicle speed, Δt is the sampling interval, τ_ψ is the FOGM heading time constant, and w is the process noise. The process noise is a zero-mean, Gaussian process with

$$E[\mathsf{w}_j\mathsf{w}_k] = \sigma_\psi^2\left[1 - e^{-(2\Delta t/\tau_\psi)}\right]\delta_{jk} \tag{74}$$

where σ_ψ^2 is the variance of the FOGM heading process, and δ_{jk} is the Kronecker delta function.

The measurement equation for transmitter *a* is given by

$$z_k^a = \left[\left(s_{x_k}^a - x_k\right)^2 + \left(s_{y_k}^a - y_k\right)^2\right]^{1/2} + N^a\lambda^a + \mathsf{v}_k^a \tag{75}$$

where $s_{x_k}^a$ and $s_{y_k}^a$ are the *x* and *y* locations of transmitter *a* at time *k*, respectively. The carrier phase cycle number is represented by the integer N^a with wavelength given by λ_a. Finally, the measurement noise is

represented by the zero-mean, Gaussian random variable v_k with variance kernel

$$E\left[\mathsf{v}_j \mathsf{v}_k\right] = \sigma_{t_a}^2 \delta_{jk} \tag{76}$$

3.1 Monte Carlo Analysis

The scenario outlined in the previous section is used to evaluate the performance of the extended Kalman filter, unscented Kalman filter, multiple-model adaptive estimator, and particle filter. Six scenarios are chosen to demonstrate the strengths and weaknesses of the various algorithms. For each scenario, a 30-run Monte Carlo simulation is performed. Each filter is executed using identical input data and the results for estimation error are compared.

The scenarios are evaluated for two main cases: known and unknown integer cycle count. When the integer cycle count is known, the state estimate is unimodal and appropriate for the EKF, UKF, and particle filter. When the integer cycle count is unknown, only the MMAE and particle filter are evaluated as the EKF and UKF would require modifications to address integer ambiguity resolution methods.

The radius and number of transmitters is varied in order to change the level of nonlinearity and observability of the system. The scenarios are arranged in general order of difficulty and are listed in Table 1. The vehicle motion parameters are the same for all scenarios with $V = 1\,m/s$, $\tau_\psi = 10\,s$, and $\sigma_\psi = 25\,deg$.

Table 1: Transmitter characteristics for UAV navigation scenario example.

Scenario	Initial State Uncertainty Standard Deviation			Number of Transmitters	Transmitter Range (X, Y, Z)	Transmitter Geometry	Transmitter Measurement Standard Deviation (σ_t)	Transmitter Wavelength (λ)
	σ_{x_0}	σ_{y_0}	σ_{ψ_0}					
1	5 *m*	5 *m*	25 *deg*	3	1000 *m*	Excellent	0.2 *m*	1 *m*
2	5 *m*	5 *m*	25 *deg*	3	10 *m*	Excellent	0.2 *m*	1 *m*
3	5 *m*	5 *m*	25 *deg*	2	10 *m*	Good	0.2 *m*	1 *m*
4	5 *m*	5 *m*	25 *deg*	2	10 *m*	Poor	0.2 *m*	1 *m*
5	5 *m*	5 *m*	25 *deg*	1	1000 *m*	Very Poor	0.2 *m*	1 *m*
6	5 *m*	5 *m*	25 *deg*	1	10 *m*	Very Poor	0.2 *m*	1 *m*

3.1.1 Known Integer Ambiguity

In the case where the integer cycle number is known, the problem is reduced to a simple problem of multilateration. As the transmitter range is decreased, the observation becomes more nonlinear. These nonlinearities are exacerbated as the transmitter geometry results in reduced observability. The ensemble root-mean-square (RMS) error and estimator quality histograms for each scenario are shown in Figures 7 – 12.

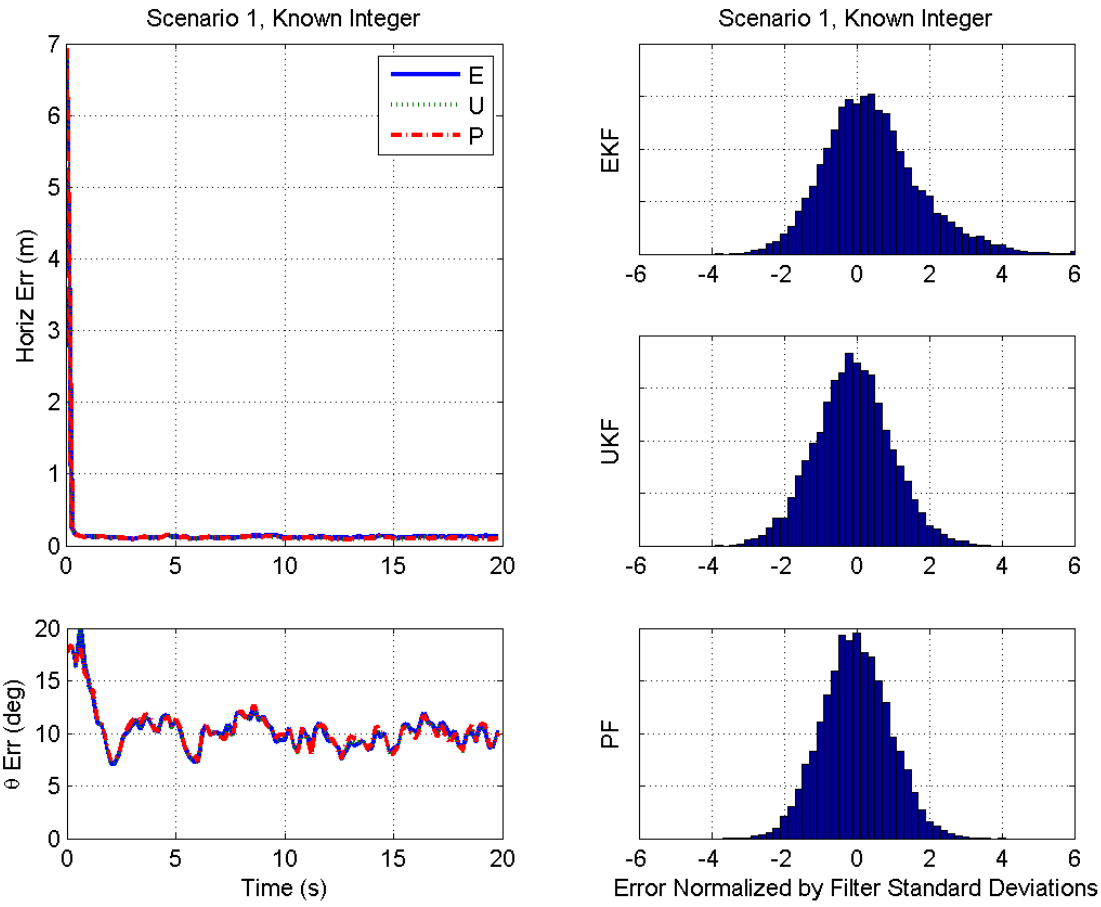

Figure 7: Estimator errors and consistency histograms for Monte Carlo Scenario 1. In this case, all filters provide accurate, consistent results.

Scenario 1 is the least challenging case as the three, well-separated transmitters provide excellent observability and their long range from the vehicle reduce the nonlinearity to a very low level. As expected, all of the estimators demonstrate consistent and accurate performance for this scenario. The consistency of the filter is a measure of the *a posteriori* state estimate errors relative to the filter predicted *a posteriori* standard deviation. In a well-tuned filter, the error statistics will be consistent with the filter predicted uncertainty. A filter that is not consistent is reporting an inaccurate confidence and is not operating optimally. In addition, the filter is susceptible to divergence. The next scenario reduces the satellite radius from 1000 *meters* to 10 *meters* in an effort to introduce some nonlinear measurements.

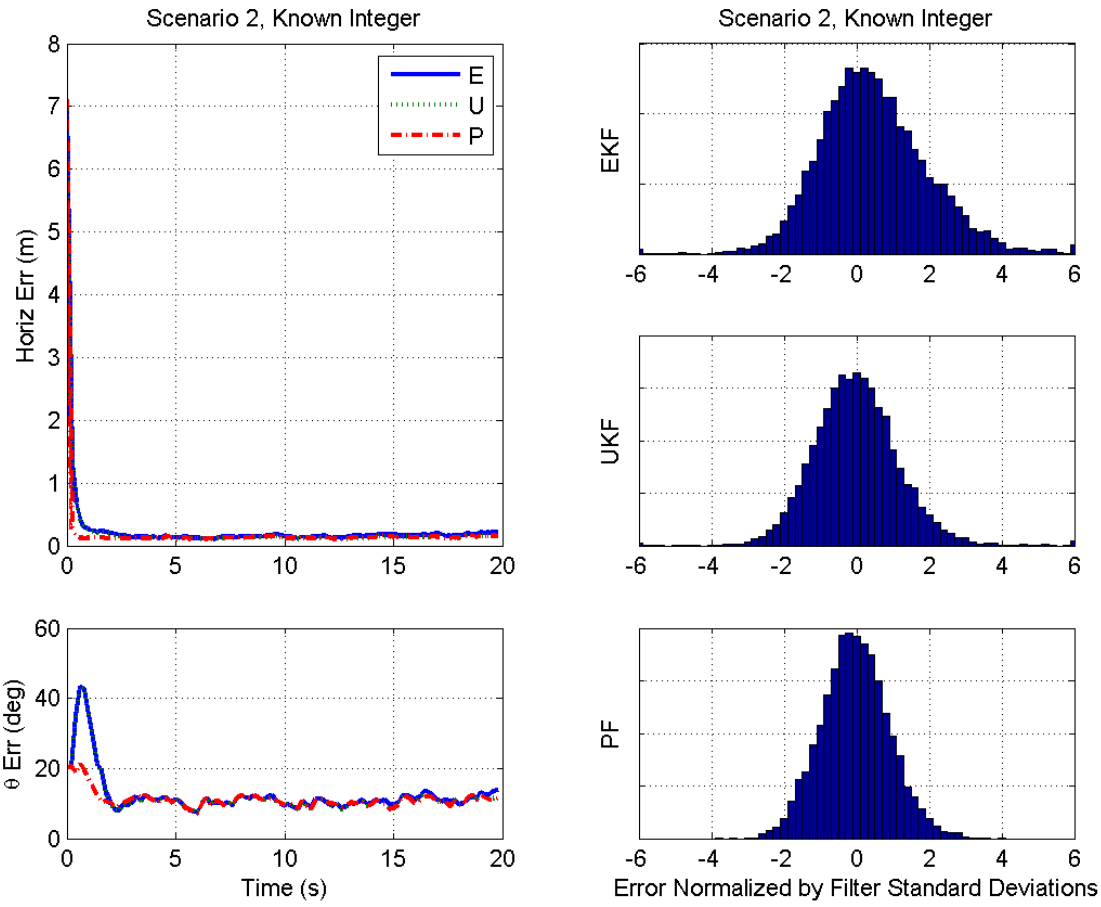

Figure 8: Estimator errors and consistency histograms for Monte Carlo Scenario 2. In this case, all filters provide accurate, consistent results, although the particle filter demonstrates slightly better accuracy and consistency.

In Scenario 2, shown in Figure 8,the increase in measurement nonlinearity is evidenced by an increase in errors for the EKF and UKF, especially during the first few seconds of operation. After the initialization period, both the EKF and UKF converge to performance similar to the particle filter. In addition, the EKF's error estimate shows signs of overconfidence as shown by the increase in the tails of the error ratio deviations. It is interesting to note that while the UKF shows almost identical estimation errors to the EKF, the UKF does maintain a more accurate filter covariance. This is one of the benefits of the UKF. In the next scenario, the number of transmitters is reduced from three to two.

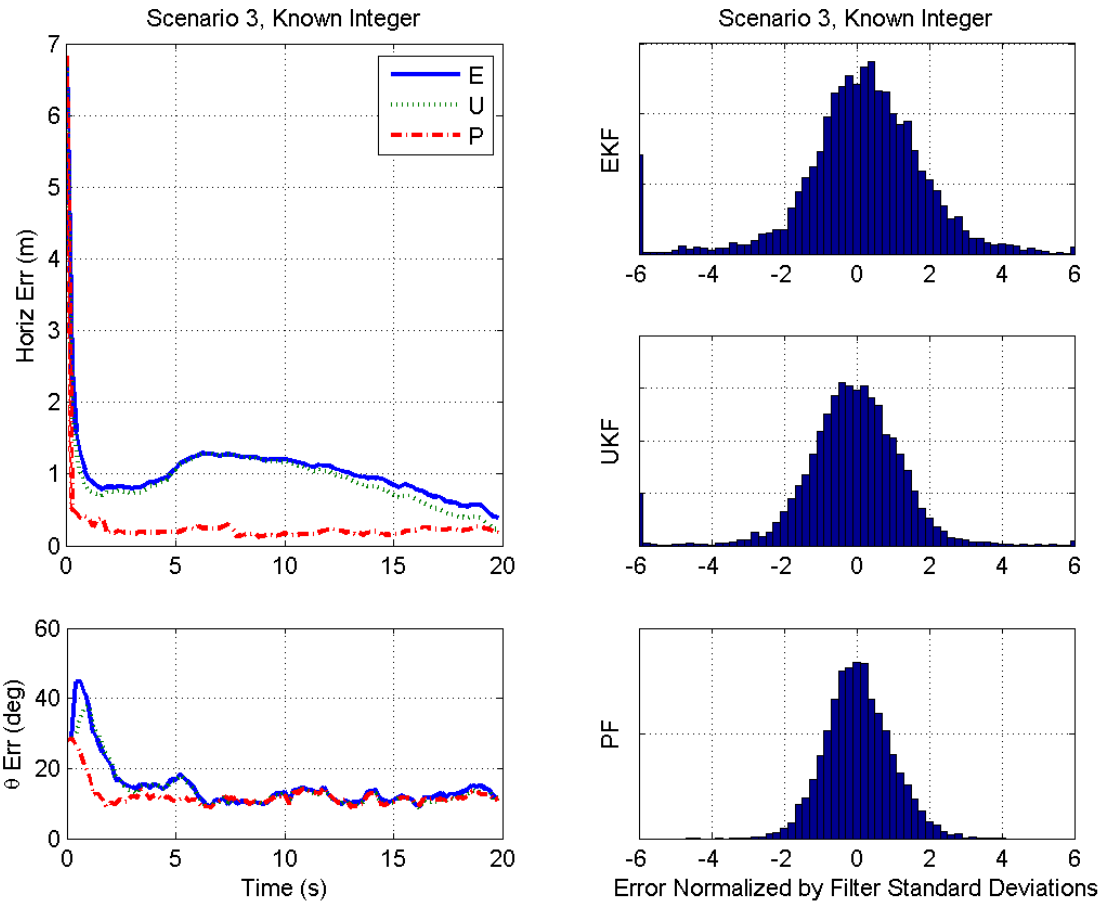

Figure 9: Estimator errors and consistency histograms for Monte Carlo Scenario 3. The particle filter is producing a more accurate result than the EKF and UKF. In addition, the EKF's error estimates are beginning to become overly optimistic.

In Figure 9, the results of Scenario 3 are shown. The reduction in transmitter observability results in a reduction in accuracy for all filters, although the EKF and UKF show the greatest increase in errors. Additionally, the consistency of both the EKF and UKF are reduced as shown by the increase in the histogram tails.

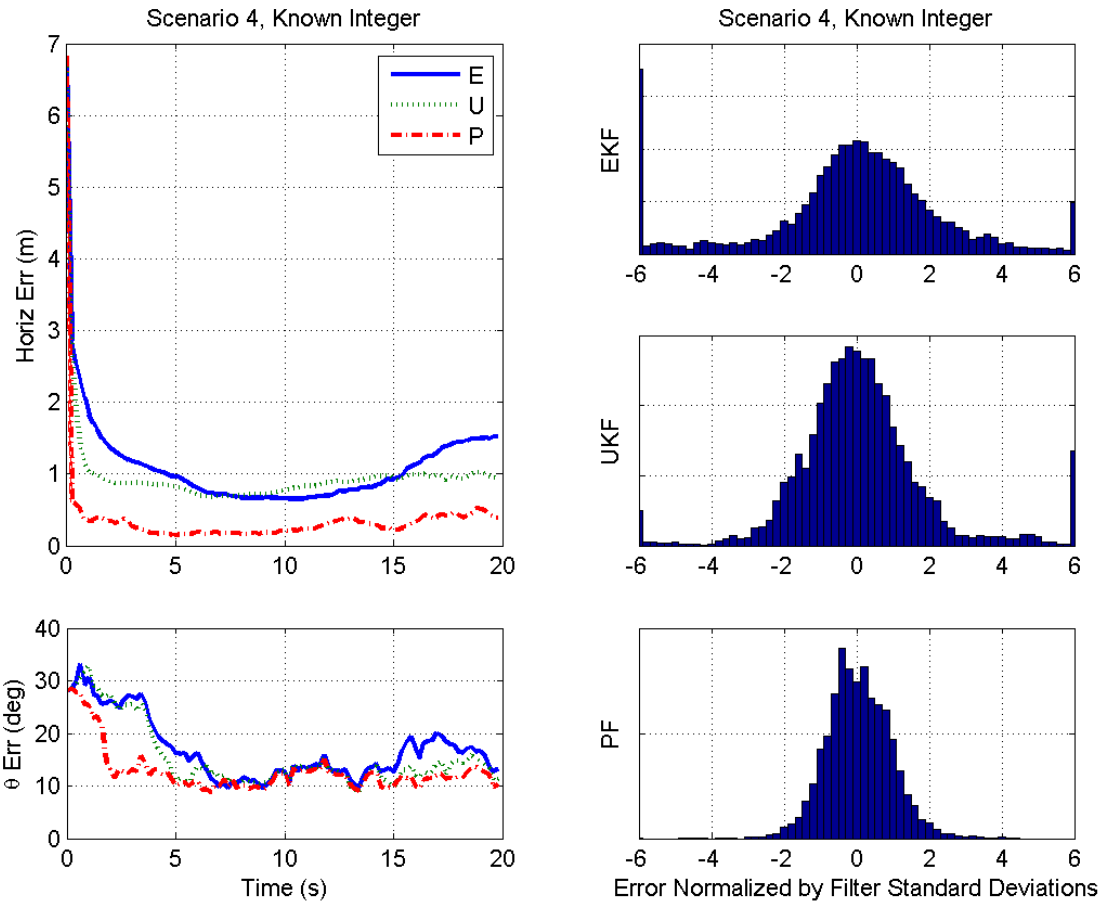

Figure 10: Estimator errors and consistency histograms for Monte Carlo Scenario 4. Similar to the Scenario 3, the particle filter is producing a more accurate result than the EKF and UKF. Both the UKF and EKF's error estimates are overly optimistic.

The results for Scenario 4 are shown in Figure 10. This scenario continues to reduce the observability of the transmitters by reducing the angular separation. As seen in the figure, the EKF is demonstrating clear signs of inconsistency via a markedly overconfident filter. As mentioned previously, this can lead to filter divergence.

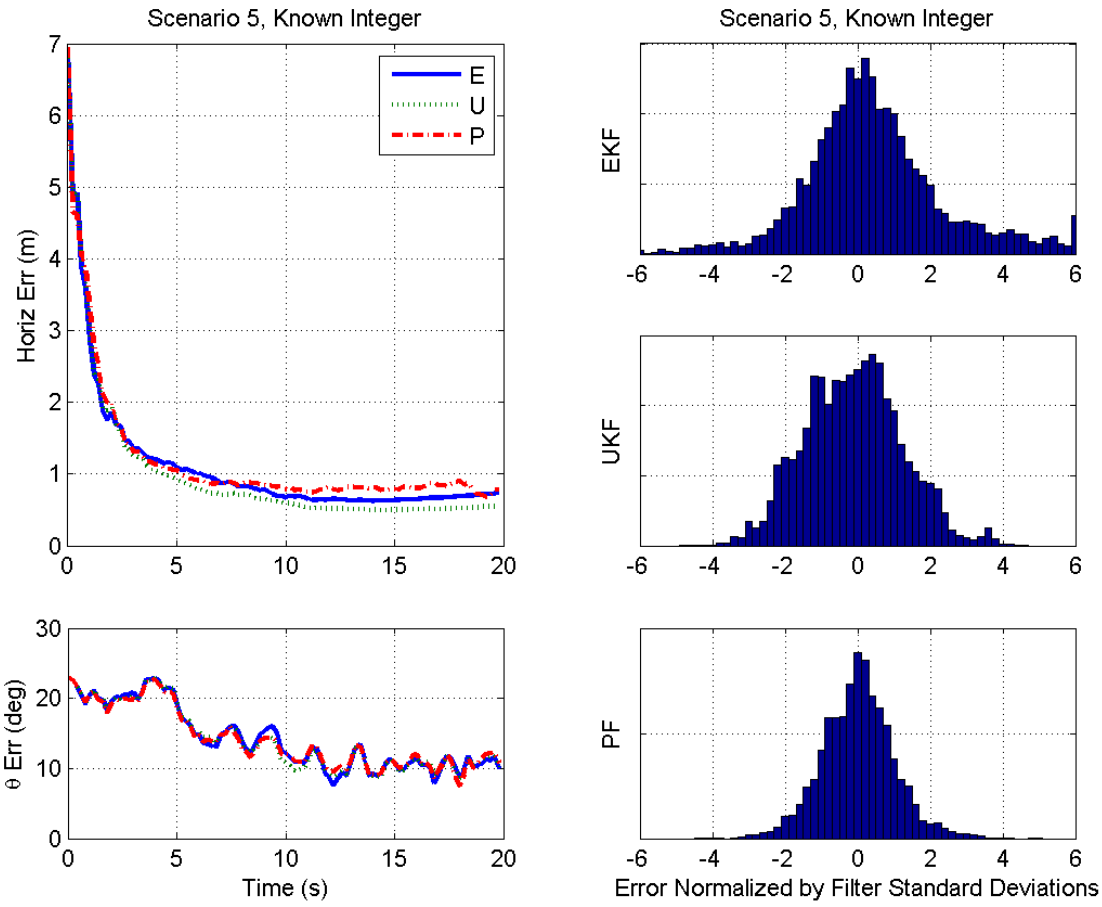

Figure 11: Estimator errors and consistency histograms for Monte Carlo Scenario 5. In this scenario, the filters perform similarly due to the relatively low level of nonlinearity.

Scenario 5 is a single satellite scenario with relatively low measurement nonlinearities. In this case, all filters show good accuracy. The EKF shows some signs of inconsistent operation, however they are relatively small. Overall, each filter handles this scenario well. In the next scenario, the nonlinearities will be increased.

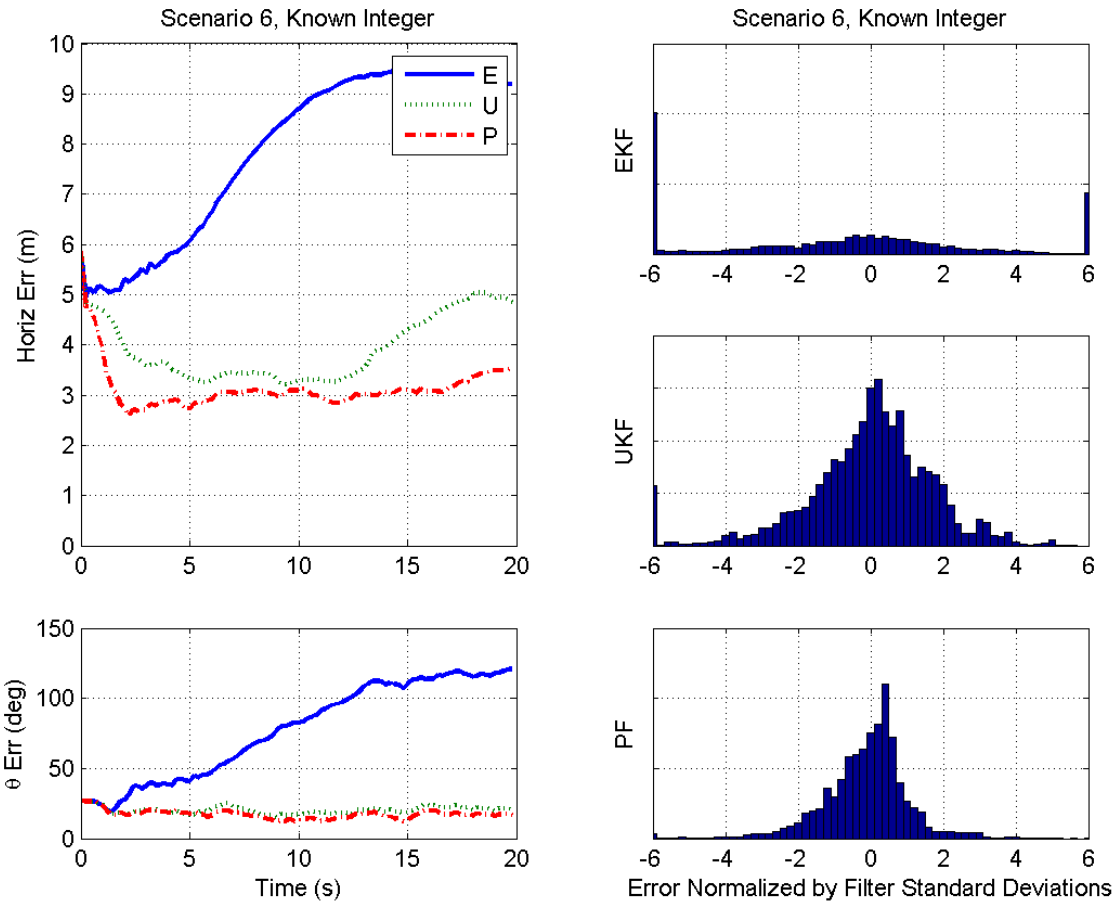

Figure 12: Estimator errors and consistency histograms for Monte Carlo Scenario 6. In this example, the EKF is beginning to diverge, producing inaccurate and overly confident estimates.

Scenario 6 is the same single transmitter configuration as in Scenario 5, with one important change. The range to the transmitter is reduced to greatly increase the level of nonlinearity. In this case, the EKF is clearly diverging. Not only are the errors increasing rapidly, the filter estimated covariance is extremely optimistic as evidenced by the high number of error greater than 6 sigma. The UKF handles the nonlinearities better than the EKF, although it is still outperformed by the particle filter.

3.1.2 Unknown Integer Ambiguity

In the case where the integer cycle number is unknown, the resulting state density function becomes multi-modal with each integer ambiguity combination representing a hypothesis. Because of this, only the MMAE and particle filter algorithms are able to properly represent the density. The MMAE filter is constructed to explicitly model every ambiguity combination within a 2-sigma bound of the initial state error.

The two algorithms are compared for Scenarios 1 through 3. The ensemble root-mean-square (RMS) error and estimator quality histograms for each scenario are shown in Figures 13 – 15.

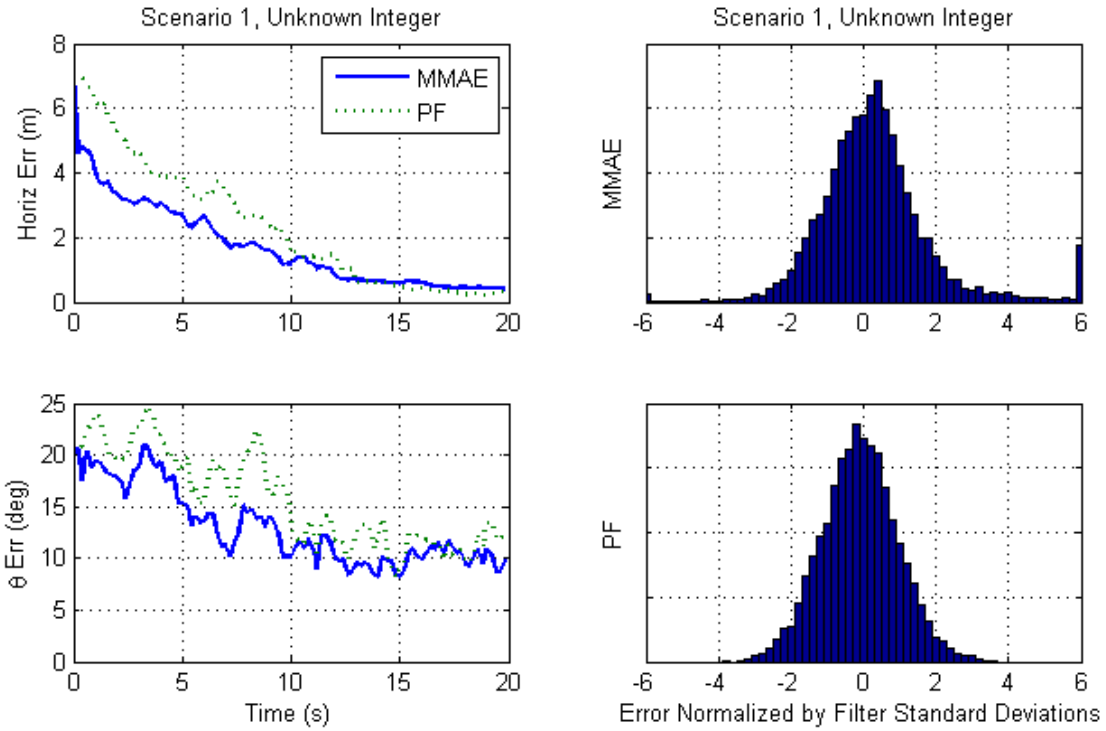

Figure 13: Estimator errors and consistency histograms for Monte Carlo Scenario 1. In this example, both the MMAE and particle filter are able to converge to an accurate state estimate.

Scenario 1 combines nearly linear measurements with an overdetermined number of satellites in an excellent geometric configuration (i.e., highly observable state). As such, both estimators converge to an accurate solution. There is some inconsistency observed in the MMAE error ratios, however this is likely due to the selection of the initial ambiguity hypothesis bound which has the effect of making the MMAE algorithm slightly overconfident. In the next scenario, the transmitter range will be reduced which increases the measurement nonlinearity.

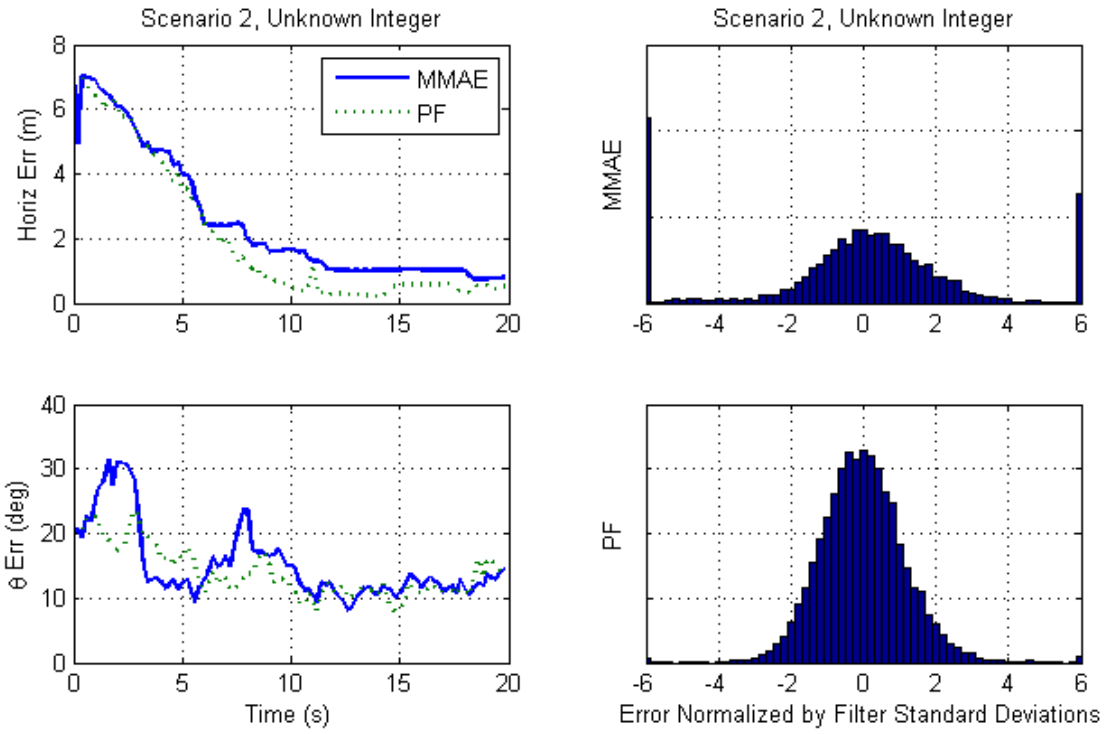

Figure 14: Estimator errors and consistency histograms for Monte Carlo Scenario 2. Both estimators converge to an accurate solution, however the uncertainty for the MMAE algorithm is beginning to show signs of inconsistency.

Scenario 2 increases the nonlinearity of the observations by reducing the transmitter range. Both estimators are still able to converge to an accurate solution. As a result of the nonlinearities, however, the MMAE algorithm's filter uncertainty is slightly optimistic as seen by the increase in the tails of the normalized error ratios.

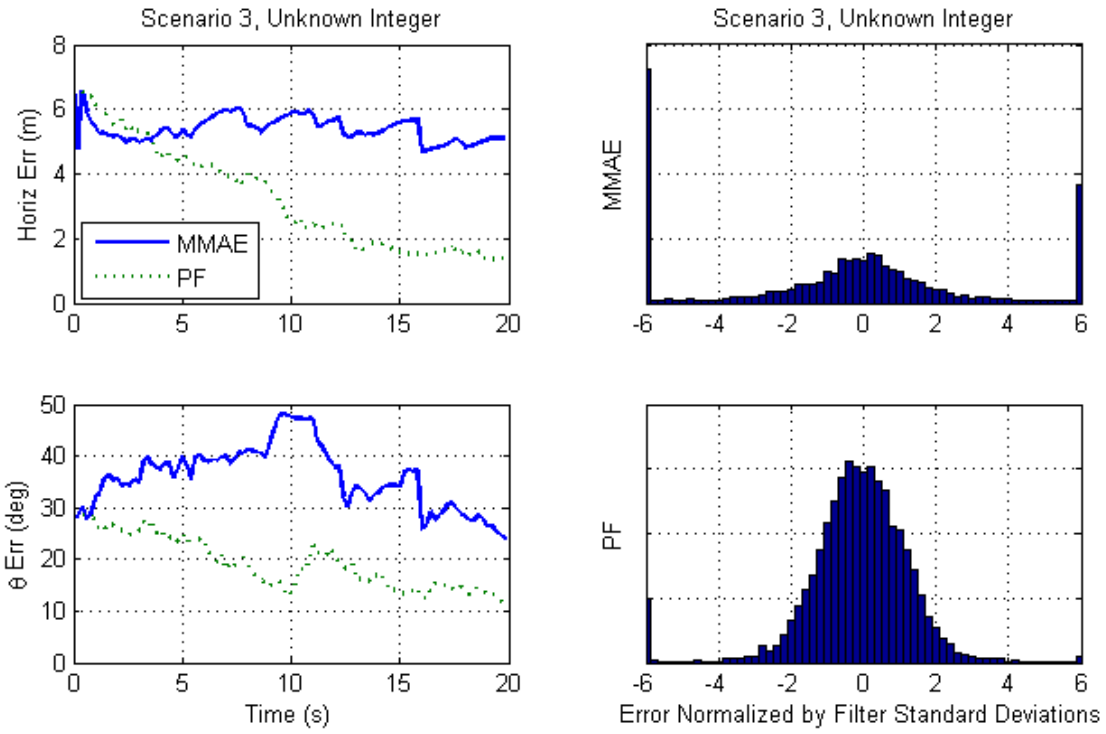

Figure 15: Estimator errors and consistency histograms for Monte Carlo Scenario 3. In this example, the particle filter is able to converge to an accurate solution while the MMAE algorithm is unsuccessful.

Scenario 3 maintains the measurement nonlinearity of Scenario 2, but reduces the number of transmitters from three to two. In this case, the MMAE algorithm is unable to converge to an accurate solution while the particle filter seems unaffected.

In the remaining scenarios (4 through 7), both estimators fail to converge to an accurate solution during the simulation time. The general trend of the consistency results remain the same, i.e., the MMAE algorithm shows a decrease in consistency when the nonlinearities increase.

In the next section, the computational burden of each algorithm is presented.

3.1.3 Processing Time Analysis

The nonlinear estimation algorithms presented in this article require varying levels of processing time. In the example, the algorithms were implemented in Matlab and all Monte Carlo simulations were timed to provide a rough measure of comparison. None of the algorithms were optimized or vectorized. The processing time results are shown in Table 2. All times are normalized to the time expended by the EKF algorithm.

	EKF	UKF	MMAE (1 transmitter, 21 hypotheses)	MMAE (2 transmitters, 441 hypotheses)	MMAE (3 transmitters, 9261 hypotheses)	Grid Particle Filter (~1.2 M cells)
Normalized Time (relative to EKF)	1	~1	7	15	400x	425x

Table 2: Comparison of processing time for various nonlinear estimation algorithms.

The results clearly show the computational burden of the particle filter and the MMAE filter as the number of hypotheses becomes large. One of the known drawbacks of particle filtering methods is computational complexity, especially in high dimensional state spaces. Optimization and parallelization techniques have demonstrated substantial improvements in processing time for particle filters. The significant computational burden of high-dimensional particle filters is a considerable drawback for real time environments.

4.0 CONCLUSIONS AND DISCUSSION

In this article, the foundations of nonlinear estimation theory is presented along with a number of nonlinear estimation algorithms. The ability of each algorithm to properly model the effects of various types of nonlinearities is shown by analysis and via a simple example. The results show that as the nonlinearities are increased, both the accuracy and consistency of the EKF are decreased. This is a well-known issue with the EKF and is one of the main reasons to choose a more advanced nonlinear estimator when moderate nonlinearities are expected.

For random vectors with unimodal densities, the UKF is a compelling choice. In all cases shown in the test scenarios, the UKF shows equal or better accuracy than the EKF. Moreover, the UKF maintains a much more consistent state covariance estimate. The importance of consistency cannot be overemphasized. The ability of an estimator to produce an accurate estimate of the *quality* of its performance is a key component in a system with integrity requirements. In addition, the UKF does not require significantly more processing resources than the EKF.

Given unlimited computational resources, the particle filter would always give the most accurate estimation performance. Unfortunately, this is oftentimes impractical, especially for high-dimensional state vectors. In the simplest construction, the memory and processing requirements rise geometrically with the number of states. This can quickly become untenable. To compensate for this known limitation a number of different particle filtering strategies have been presented in the literature [5],[6],[2]. The general approach is to focus the particles only in the areas of highest likelihood, thus maintaining the important modes of the density with a minimum of resources. This is currently an area of active research.

For military navigation systems, integrity and accuracy are paramount. While this application area has been dominated by the EKF to this point, the advent of non-traditional aiding sources motivates a more advanced filter. The UKF is well-suited to meet future needs due to its demonstrated ability to accurately and consistently work with low-to-medium nonlinearities while maintaining an acceptable computational burden.

5.0 BIBLIOGRAPHY

1. Brown, Grover and Patrick Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York: John Wiley & Sons, 1997.

2. Julier, S.J. and Uhlmann, J.K., "Unscented Filtering and Nonlinear Estimation". *Transactions of the IEEE*, 92(3):401 - 422, 2004.

3. Maybeck, Peter S., *Stochastic Models, Estimation, and Control Volume II*. Orlando, FL: Academic Press, Inc., 1979.

4. Maybeck, Peter S., *Stochastic Models Estimation and Control, Volume 1*. Orlando, FL: Academic Press, Inc., 1979.

5. Ristic, Branko and Sanjeev Arulampalam and Neil Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

6. Van Der Merwe, R. and Doucet, A. and De Freitas, N. and Wan, E., "The Unscented Particle Filter". *Advances in Neural Information Processing Systems*, 584-590, 2001.

7. Wan, E.A. and Ven Der Merwe, R.. "The Unscented Kalman Filter for Nonlinear Estimation". *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, 153-158, 2000.