

Relative Track Metrics to Determine Model Mismatch

Erik Blasch

Air Force Research Lab
2241 Avionics Cir
WPAFB, OH 45433
erik.blasch@wpafb.af.mil

Andrew Rice

Air Force Research Lab
2241 Avionics Cir
WPAFB, OH 45433
andrew.rice@wpafb.af.mil

Chun Yang

Sigtem Technology, Inc.
113 Clover Hill Lane
Harleysville, PA 19438
chunyang@sigtem.com

Ivan Kadar

Interlink Sys. Sciences., Inc.
Lake Success, NY
ikadar@SystemsSciences.com

Abstract – Tracking performance is a function of data quality, tracker type, and target maneuverability. Many contemporary tracking methods are useful for various operating conditions. To determine nonlinear tracking performance independent of the scenario, we wish to explore metrics that highlight the tracker capability. With the emerging relative track metrics, as opposed to root-mean-square error (RMS) calculations, we explore the Averaged Normalized Estimation Error Squared (ANEES) and Non Credibility Index (NCI) to determine tracker quality independent of the data. This paper demonstrates the usefulness of relative metrics to determine a model mismatch, or more specifically a bias in the model, using the probabilistic data association filter, the unscented Kalman filter, and the particle filter.

Keywords: Relative track metrics, NCI, PF, UKF, PDAF

1 Introduction

In a dynamic targeting scenario, there are host of algorithms that affect performance: sensor registration, measurement-to-track assignment, track-to-track association, sensor management, and ultimately, the user. A key determinant of target-tracking success is being able to choose the right sensor with the correct measurement-to-track algorithm. Typically, a tracking algorithm is evaluated by the absolute position error (i.e. Root Mean Square) [1]. Using the RMS represents the sensor-algorithm performance, as opposed to evaluating the tracker independent of the sensor. In deployable systems, it is important to assess the tracker capability over different operational conditions (target dynamics, sensor quality, and environment obstacles). To evaluate tracker capability independent of operating conditions (OCs), we utilize relative metrics [1].

The goal of target tracking is to associate measurements of moving objects. There are many approaches such as the multiple model estimator (MME) [2], joint probabilistic data association (JPDAF) [2], multiple hypothesis tracker (MHT) [3], interactive multiple model (IMM) [4, 5], variable-structure IMM (VS-IMM) [5,6], unscented Kalman filter [7], particle filter [8] and its variants [9-10]. Others have used identification information to improve tracker performance [11-14]. One difficulty for a user is to determine the best approach for a given scenario. Such an example is tracker performance in clutter [9, 15]. To determine the tracker

algorithm performance for various OCs, there is a need to study track metrics [16].

Many authors have explored various tracking metrics to better determine tracker performance [19, 20] for sensor management [17, 18]. Drummond investigated various approaches and has compiled a list of absolute metrics [16]. Chong and Mori looked at track association metrics [21]. The IMM is investigated through model probability switching [2]. A fourth approach is the Cramér-Rao lower bound [22, 23]. One emerging set of metrics from X. Rong Li and coauthors [1] is called *relative metrics* [1]. Li promotes the usefulness of relative metrics as statistical estimators to isolate tracker performance independent of the data. In this paper, we explore relative metrics to determine the applicability to nonlinear track evaluation.

The relative track metrics include *ANEES*, *Non-Credibility Index*, and *Credibility Index* [1]. In a classic example of estimating the position of a target moving in a circle, Figure 1 shows the value of relative metrics. The two credibility measures (NCI and ANEES) differ with respect to equitable treatment of optimism versus pessimism. The NCI is a geometric average, with the mean near the mode. Hence optimism and pessimism are treated equitably. The ANEES is an arithmetic average, and hence severely penalizes optimism.

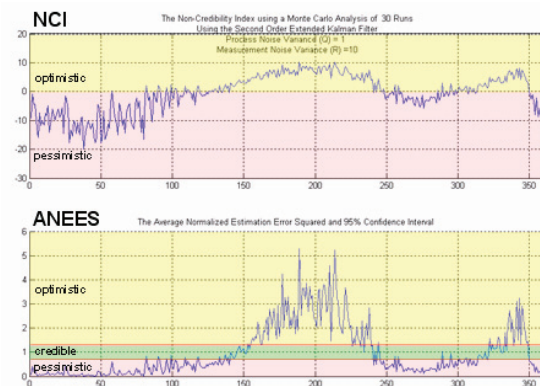


Figure 1. Tracking example - Circle data, no bias, 2nd order extended Kalman filter, $Q=1$, $R=10$, 30 Monte Carlo runs

In data association tracking approaches, typical tracking is provided through position measurements. For linear Gaussian cases, this is acceptable. However, for nonlinear cases the Gaussian assumption is not viable. Of

recent interest is the particle filter method which uses a series of particles to estimate the likelihood of a target location. The sampling approaches approximate the posterior density by a set of samples: the unscented Kalman filter (UKF) uses a small number of deterministically chosen samples, while the particle filter uses a large number of random (Monte Carlo) samples as shown in **Figure 2**.

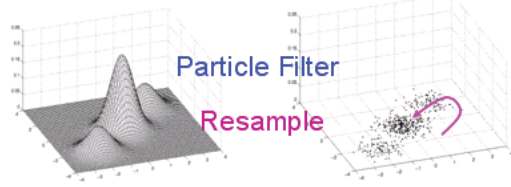


Figure 2. Resampling.

We are concerned with assessing track quality and sensor pointing using performance metrics. Many metrics are absolute metrics that are focused on the RMS position error. To better assess how well a nonlinear tracker [24] does based on a given sensor, we focused on the relative track metrics, as presented by Li. These metrics can be perceived as obvious, however, that does not preclude their usefulness and furthermore, since the documentation of nonlinear relative track metrics is not presented in the literature, it is something that the tracking community could seek more clarification.

Most track methods utilize covariance information to gate measurements and determine tracker quality. Track error might have better representations than just covariance [25]. The goal of this work was to explore the use of the relative metrics to non-linear tracking methods. We chose to explore the use of relative metrics to the linear *particle data association filter*, suboptimal nonlinear *unscented KF*, and the optimal nonlinear *particle filter*. Assessing the relative metrics for nonlinear approaches required some modifications to “estimate” the density information. As shown in the paper, useful results were gained in using the relative metrics in assessing tracker quality.

This paper develops the relative metrics for nonlinear assessment. Section 2 describes tracking metrics. Section 3 and 4 describes linear and nonlinear tracking. Section 5 and 6 show a performance analysis and draws conclusions.

2 Tracking Metrics

Tracking methods include many opportunities for analysis. Some metrics are listed below:

Metric	Description
Absolute Track Quality	Mean square position, velocity, acceleration error
Relative Track Quality	Mean square kinematic error relative to sensor covariance
Track Life-Time	Total time target is in track
Relative Track Life-Time	Total time target in track, relative to length of track-lets

Track Length	Distance over which target is tracked
Relative Track Length	Distance over which target is tracked relative to maneuverability
Track Purity	Percent of associations of dominant track over lifetime [3]
Track Coverage	Area of tracking
Track Density	Number of targets track per area
Track Continuity	Number of individual targets associated with a given track

An objective function for measurement-to-track assignment can be a composite function of the track factors. Blackman [3] has described the use of marginal utility as the assignment objective function. *Utility* is defined as the ratio of current performance (in the reference application it is tracking performance, measured as estimation error standard deviation) to desired performance. The desired performance is standard criteria established by the tracker. *Marginal utility* is defined as the difference between current utility and the *expected utility* that would be achieved if the sensor is allocated to the target. Computation of expected utility requires a model of tracker performance if the sensor is assigned to each candidate target. Thus, we are interested in the case of the *relative* track metrics to determine tracker performance of the sensor-tracker performance model.

The Averaged Normalized estimation error squared (ANEES) is defined by [1]:

$$ANEES = \frac{1}{Nn} \sum_{i=1}^N (\mathbf{x}_i - \hat{\mathbf{x}}_i)^T \mathbf{P}_i^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i) \quad (1)$$

where $(\mathbf{x}_i - \hat{\mathbf{x}}_i)$ is the state estimation error, \mathbf{P}_i is the error covariance provided by the estimator in the i^{th} run, n is the state dimension, and N is the number of runs in the MC test. If the estimation error and estimated covariance are the same, then the ANEES ~ 1 , and the filter is credible [1].

The credibility test is an assessment of the Chi-square capability of the estimator. The determinant is a check for the Non-credibility Index as to whether the sensor system is credible.¹ When the ANEES is outside the 95% probability interval, the estimator is not credible. For the vector case, the NCI is the sample average $10 \log_{10}(\rho)$ (analogous to the SNR).

$$\text{Credibility} = \rho = \frac{(\mathbf{x}_i - \hat{\mathbf{x}}_i)^T \mathbf{P}_i^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i)}{(\mathbf{x}_i - \hat{\mathbf{x}}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i)} \quad (2)$$

where $\mathbf{\Sigma}_i$ is the actual MSE covariance of the sensor.

$$\text{Non-Credibility Index} = \frac{10}{N} \sum_{i=1}^N \log_{10}(\rho_i) \quad (3)$$

If $NCI \gg 0$, \mathbf{P}_i is low (optimistic)

¹ For an “isotropic” filter, \mathbf{P} is proportional to $\mathbf{\Sigma}$. In this case, the credibility as defined in Eq. (2) remains to be constant all the time. Personal communication with John Sudano.

If $\text{NCI} \ll 0$, \mathbf{P}_t is high (pessimistic)

Many estimators / filters provide self assessment of estimation errors – *error covariance*. However, are the self-assessments (error-covariance) close to the actual MSE, and if so how close? A credibility test determines how much we can trust these self-assessments. An estimator is: *Credible* if its actual error and self-assessment are statistically equal (e.g. their difference is statistically insignificant). It is *Optimistic* if its self-assessment is statistically smaller than the actual error and *Pessimistic* if its self-assessment is statistically larger than the actual error. An estimator is not credible usually because its assumptions / model / approximations are not accurate.

3 Linear Tracking

Kalman filters are the baseline for tracking and are optimal if the process and measurement equations are linear and the noise is Gaussian. To enhance the tracker performance, a bank of Kalman filters can be used where different state transition matrices are varied based on the kinematics of the target being tracked. The state and measurement equations for all targets are:

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{F}_t \hat{\mathbf{x}}_{t|t} \quad (4)$$

$$\hat{\mathbf{z}}_{t+1|t} = \mathbf{H}_{t+1} \hat{\mathbf{x}}_{t+1|t} \quad (5)$$

The covariance of the predicted state is,

$$\mathbf{P}_{t+1|t} = \mathbf{F}_t \mathbf{P}_{t|t} \mathbf{F}_t^T + \mathbf{Q}_t \quad (6)$$

All measurements received from a sensor at time k are then compared to the predicted measurements to ascertain whether or not the measurements fall within the kinematic gate. The validation region is the elliptical region:

$$\mathcal{V}(t, \gamma) = \{z: [z - \hat{\mathbf{z}}_{t+1|t}]^T \mathbf{S}^{-1} [z - \hat{\mathbf{z}}_{t+1|t}] \leq \gamma\} \quad (7)$$

where γ is the *gate threshold*. Using the gate and the relative metrics we can determine a model mismatch.

Figure 3 shows the case of a target maneuvering. The credibility metrics, as shown in Figure 4, show behavior hidden by absolute metrics. NCI indicates track confidence is dependent of target direction. The performance analysis reveals un-modeled system dynamic and sub-optimal covariance estimates.

4 Nonlinear Tracking

4.1 Suboptimal KF – UKF

The UKF is a recursive *minimum mean square error estimator* that often provides an improvement over the extended Kalman filter (EKF) for nonlinear models. The EKF linearizes the nonlinear process and measurement models using the first-order terms of a Taylor series

expansion. The UKF, on the other hand, does not approximate the nonlinear process and measurement models. It uses the actual models and instead approximates the distribution of the state variable as a Gaussian. The Gaussian approximation is specified using a minimal set of deterministically chosen samples called *sigma points*. Each sigma point is independently propagated through the process and measurement models, and the set of propagated sigma points is analyzed to provide a posterior Gaussian approximation.

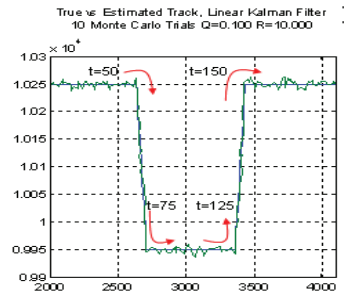


Figure 3. Linear case for the relative metrics.

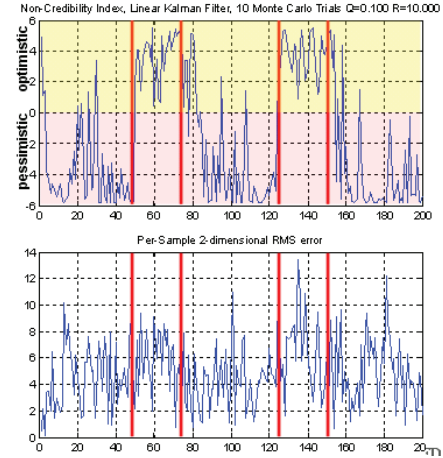


Figure 4. Linear case for the relative metrics.

The nonlinear density and state models are:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}) \quad (8)$$

$$\mathbf{x}_t = f(\mathbf{u}_{t-1}, \mathbf{x}_{t-1}, \mathbf{v}_{t-1})$$

$$\mathbf{x}_t \approx f(\mathbf{u}_{t-1}, \bar{\mathbf{x}}_{t-1}, \bar{\mathbf{v}}) + \mathbf{F}_t (\mathbf{x}_{t-1} - \bar{\mathbf{x}}) + \mathbf{G}_t (\mathbf{v}_{t-1} - \bar{\mathbf{v}})$$

where \mathbf{x}_t denotes the state of the system at time t , $\mathbf{u}_t \in \mathcal{R}^{\text{nu}}$ is some known input, $\mathbf{v}_t \in \mathcal{R}^{\text{nv}}$ is a noise term with the covariance matrix \mathbf{Q} , $f: \mathcal{R}^{\text{nx}} \times \mathcal{R}^{\text{nu}} \times \mathcal{R}^{\text{nv}} \rightarrow \mathcal{R}^{\text{nx}}$ is the deterministic nonlinear state transition model and $\mathbf{F}_t = [\partial f(\mathbf{x}_t) / \partial \mathbf{x}_t]_{\mathbf{x}_t = \bar{\mathbf{x}}_t}$ and $\mathbf{G}_t = [\partial f(\mathbf{v}_t) / \partial \mathbf{v}_t]_{\mathbf{v}_t = \bar{\mathbf{v}}_t}$ are the Jacobians of the state model linearized around the reference value $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{v}}_t$, respectively.

The nonlinear measurement model is

$$p(\mathbf{y}_t | \mathbf{x}_{t-1}) \quad (9)$$

$$\underline{y}_t = h(\underline{u}_t, \underline{x}_t, \underline{n}_t)$$

$$\underline{y}_t \approx f(\underline{u}_t, \underline{\bar{x}}_t, \underline{\bar{n}}) + H_t(\underline{x}_t - \underline{\bar{x}}) + H_t(\underline{n}_t - \underline{\bar{n}})$$

where $\underline{y}_t \in \mathbb{R}^{ny}$ is the measurement vector, $\underline{n}_t \in \mathbb{R}^{nn}$ the measurement noise with covariance matrix \mathbf{R} , which is assumed to be independent of \underline{y}_t , $h: \mathbb{R}^{nx} \times \mathbb{R}^{nu} \times \mathbb{R}^{nn} \rightarrow \mathbb{R}^{ny}$ is the deterministic nonlinear measurement, $H_t = [\partial h(x_t) / \partial x_t]_{x_t = \bar{x}_t}$ and $L_t = [\partial h(n_t) / \partial n_t]_{n_t = \bar{n}_t}$ are the Jacobians of the measurement model linearized around the reference values of $\underline{\bar{x}}_t$ and $\underline{\bar{n}}$, respectively.

For the UKF, consider a random variable $\underline{u} \in \mathbb{R}^n \sim \mathcal{N}(\underline{m}_x, \mathbf{P}_x)$. The covariance matrix is scaled up by a factor $(n + \kappa)$, thus spreading the standard deviation by a factor of $\gamma = \sqrt{n + \kappa}$. Now factorize the covariance matrix by the Cholesky decomposition $\sqrt{\mathbf{P}_x}$. Construct $2n + 1$ sigma points around the mean vector \underline{m} with the scaled spread square root $\gamma \sqrt{\mathbf{P}_x}$ as :

$$\mathcal{X} = [\underline{m}_x, \underline{m}_x + \gamma \sqrt{\mathbf{P}_x}, \underline{m}_x - \gamma \sqrt{\mathbf{P}_x}] \quad (10)$$

Let $[\mathcal{X}]_i$ be the i^{th} column of \mathcal{X} (i.e. the i^{th} sigma point vector with the count starting from 0). Propagating it through the nonlinear function $\underline{y} = f(\underline{x})$ produces the posterior sigma point vectors as $[\mathcal{Y}]_i$ for $i = 0, \dots, 2n$. The mean and covariance are calculated from the weighted mean and covariance as

$$\underline{m}_y = \sum_{i=0}^{2n} W_i^m [\mathcal{Y}]_i \quad (11)$$

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} W_i^c ([\mathcal{Y}]_i - \underline{m}_y) ([\mathcal{Y}]_i - \underline{m}_y)^T \quad (12)$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} W_i^c ([\mathcal{X}]_i - \underline{m}_x) ([\mathcal{Y}]_i - \underline{m}_y)^T \quad (13)$$

where W_i^m and W_i^c are the constant weights for the mean and covariance, given by

$$W_0^m = \frac{\lambda}{\lambda + n}, \quad W_i^m = \frac{1}{2(\lambda + n)}; \quad W_i^c = W_i^m \quad (14)$$

$$W_0^c = \frac{\lambda}{\lambda + n} + (1 - \alpha^2 + \beta); \quad i = 1, \dots, 2n. \quad (15)$$

where $\lambda = \alpha^2(n + \kappa)$. The parameter $\kappa \geq 0$ is chosen to ensure the positive-definiteness of the covariance matrix, which defaults to zero. The parameter $0 \leq \alpha \leq 1$ controls the size of the sigma point distribution (probabilistic spread in terms of covariance) and takes a small number to avoid sampling non-local effects when non-linearities are strong. β is a weighting term used to incorporate any knowledge of higher order moments of the distribution. For a Gaussian, the optimal choice is $\beta = 0$ [Van de Merwe, 26].

The UKF is a scaled unscented transformation (SUT) [27] to recursive minimum mean-square estimator (MMSE) estimation. The propagation of a random variable through a nonlinear function with the SUT (as in the UKF) leads to a recursive estimation steps for the UKF.

- Step 1 – Construct an augmented state by concatenating the original state with the process noise and measurement noise.
- Step 2 – Calculate the covariance matrix for the augmented state and select the sigma points for the augmented state.
- Step 3 – Conduct time-update by propagating sigma points through exact nonlinear process and measurement equations.
- Step 4 – Calculate covariance matrix of predicted measurements per sigma points (\mathbf{P}_{YY}) and cross-covariance matrix between predicted measurements and augmented state (\mathbf{P}_{XY}).
- Step 5 – Conduct the measurement update using the Kalman filter gain as $\mathcal{K}_k = \mathbf{P}_{XY} (\mathbf{P}_{YY})^{-1}$.

Table 1. The Unscented Kalman Filter [Yang and Miller 2005]

Concatenation:	Augment the state vector to include noise terms: $\underline{x}_t^a = [\underline{x}_t^T, \underline{u}_t^T, \underline{n}_t^T]^T, n_a = n_x + n_y + n_n$
Initialization:	$\underline{\bar{x}}_0 = E\{\underline{x}_0\}, P_0 = E\{(\underline{x}_0 - \underline{\bar{x}}_0)(\underline{x}_0 - \underline{\bar{x}}_0)^T\}$ $\underline{\bar{x}}_0^a = E\{\underline{x}_0^a\} = [\underline{\bar{x}}_0^T, 0, 0]^T, P_0^a = E\{(\underline{x}_0^a - \underline{\bar{x}}_0^a)(\underline{x}_0^a - \underline{\bar{x}}_0^a)^T\} = \text{diag}\{P_0, Q, R\}$
Recursive over t	
Calculate Sigma Points	$\underline{x}_{t-1}^a = [\underline{x}_{t-1}^a, \underline{u}_{t-1}^a, \dots, \underline{n}_{t-1}^a]^T$ where $\underline{x}_{0,t-1}^a = \underline{\bar{x}}_{t-1}^a, W_0^{(m)} = \lambda/(n_x + \lambda), W_0^{(c)} = \lambda/(n_x + \lambda) + (1 - \alpha^2 + \beta)$ $\underline{x}_{i,t-1}^a = \underline{\bar{x}}_{t-1}^a + \sqrt{(n_x + \kappa)} P_{t-1}^a \mathbf{e}_i, i = 1, \dots, n_a$ $\underline{x}_{i,t-1}^a = \underline{\bar{x}}_{t-1}^a - \sqrt{(n_x + \kappa)} P_{t-1}^a \mathbf{e}_i, i = n_x + 1, \dots, 2n_a$ $W_i^{(m)} = W_i^{(c)} = 1/(2(n_x + \lambda)), i = 1, \dots, 2n_a$
Time Update	$\underline{x}_{t t-1}^a = f(\underline{x}_{t-1}^a, \underline{u}_{t-1}^a, \underline{n}_{t-1}^a)$ $\underline{\bar{x}}_{t t-1}^a = \sum_{i=0}^{2n_a} W_i^{(m)} \underline{x}_{i,t t-1}^a$ $P_{t t-1}^a = \sum_{i=0}^{2n_a} W_i^{(c)} [\underline{x}_{i,t t-1}^a - \underline{\bar{x}}_{t t-1}^a] [\underline{x}_{i,t t-1}^a - \underline{\bar{x}}_{t t-1}^a]^T$ $\underline{y}_{t t-1}^a = h(\underline{x}_{t t-1}^a, \underline{u}_{t t-1}^a, \underline{n}_{t t-1}^a)$ $\underline{\bar{y}}_{t t-1}^a = \sum_{i=0}^{2n_a} W_i^{(m)} \underline{y}_{i,t t-1}^a$
Measurement Update	$P_{y,y_t} = \sum_{i=0}^{2n_a} W_i^{(c)} [\underline{y}_{i,t t-1}^a - \underline{\bar{y}}_{t t-1}^a] [\underline{y}_{i,t t-1}^a - \underline{\bar{y}}_{t t-1}^a]^T$ $P_{x,y_t} = \sum_{i=0}^{2n_a} W_i^{(c)} [\underline{x}_{i,t t-1}^a - \underline{\bar{x}}_{t t-1}^a] [\underline{y}_{i,t t-1}^a - \underline{\bar{y}}_{t t-1}^a]^T$ $\mathbf{K}_t = P_{x,y_t} (P_{y,y_t})^{-1}$ $\underline{\bar{x}}_t = \underline{\bar{x}}_{t t-1}^a + \mathbf{K}_t (\underline{y}_t - \underline{\bar{y}}_{t t-1}^a)$ $P_t = P_{t t-1}^a - \mathbf{K}_t P_{y,y_t} \mathbf{K}_t^T$

Table 1 lists the algorithm that updates the mean and covariance of the Gaussian approximation to the posterior distribution of the state. As shown, there is no explicit calculation of Jacobians or Hessians. Since the covariance matrix can be expressed recursively in the square-root form, not only does the UKF outperform the EKF in accuracy and robustness, it does so at no extra computational cost. When the process and measurement noise terms are purely additive, there is no need to augment the system state with noise, further reducing the computational complexity. Both the EKF and UKF rely on the Gaussian assumption and its approximation. The particle filtering method described next; however, does

not require this assumption. But it has problems of its own.

4.2 Optimal Nonlinear Filter - PF

Classical Monte Carlo methods for dynamic systems, such as particle filters, are capable of tracking complex nonlinear systems with noisy measurements. The problem is that estimates from a particle filter tend to have a high variance for small sample sets. Using large sample sets is computationally expensive and defeats the purpose.

Particle filters have been extensively used for Bayesian state estimation in nonlinear systems with noisy measurements. They approximate the probability distribution with a set of samples or particles. Particle filters have a number of characteristics that make them attractive: they are nonparametric (can represent arbitrary distributions), can handle hybrid state spaces, can handle noisy sensing and motion, and can easily be balanced to meet performance objectives where the number of particles (hence, estimation accuracy) can be adjusted to match available computation. The design of a particle filter is based on four basic ideas, which are discussed below.

4.3 Discrete Approximation of PDF

The first idea is the approximation of a continuous support distribution $p(x_{0:t} | y_{1:t})$ by N discrete samples $x_{0:t}^{(i)}$, “randomly” drawn from the distribution $p(x_{0:t} | y_{1:t})$, for $i = 1, \dots, N$:

$$p(x_{0:t} | y_{1:t}) \cong \frac{1}{N} \sum_{i=1}^N \delta(x_{0:t} - x_{0:t}^{(i)}) \quad (16)$$

where the subscript “0:t” or “1:t” indicates the observation interval from 0 or 1 to t and $\delta(\bullet)$ is the Dirac delta function. This is the so-called Monte Carlo method. With this approximation, the computation of expectation of any function of $x_{0:t}$, $g(\bullet)$, is reduced from a complicated integration to a simple summation as:

$$\begin{aligned} E\{g_t(x_{0:t})\} &= \int g_t(x_{0:t}) p(x_{0:t} | y_{1:t}) dx_{0:t} \\ &\cong \frac{1}{N} \sum_{i=1}^N g_t(x_{0:t}^{(i)}) \end{aligned} \quad (17)$$

4.4 Importance Sampling

The second idea is the importance sampling. In the estimation problem, the posterior distribution $p(x_{0:t} | y_{1:t})$ is in fact what we want to estimation from data, thus not available for sampling directly. One way to get around this is to approximate the expectation over the unknown distribution $p(x_{0:t} | y_{1:t})$ by another expectation taken over a known easy-to-sample distribution $q(x_{0:t} | y_{1:t})$, called importance function, also known as proposal distribution (which we will use interchangeably):

$$\begin{aligned} E\{g_t(x_{0:t})\} &= \int g_t(x_{0:t}) \frac{p(x_{0:t} | y_{1:t})}{q(x_{0:t} | y_{1:t})} q(x_{0:t} | y_{1:t}) dx_{0:t} \\ &= \int g_t(x_{0:t}) \frac{p(y_{1:t} | x_{0:t}) p(x_{0:t}) / p(y_{1:t})}{q(x_{0:t} | y_{1:t})} q(x_{0:t} | y_{1:t}) dx_{0:t} \\ &= \int g_t(x_{0:t}) \frac{w_t(x_{0:t})}{p(y_{1:t})} q(x_{0:t} | y_{1:t}) dx_{0:t} \end{aligned} \quad (18)$$

where the variable $w_t(x_{0:t})$ is the un-normalized importance weight defined by:

$$w_t(x_{0:t}) = \frac{w_t(x_{0:t})}{p(y_{1:t})} \quad (19)$$

The probability density function $p(y_{1:t})$ in the denominator can be evaluated as:

$$\begin{aligned} p(y_{1:t}) &= \int p(y_{1:t} | x_{0:t}) p(x_{0:t}) \frac{q(x_{0:t} | y_{1:t})}{q(x_{0:t} | y_{1:t})} dx_{0:t} \\ &= \int w_t(x_{0:t}) q(x_{0:t} | y_{1:t}) dx_{0:t} \\ &= E_q\{w_t(x_{0:t})\} \end{aligned} \quad (20)$$

where $E_q\{\bullet\}$ is the expectation taken over the importance function or proposal distribution $q(x_{0:t} | y_{1:t})$.

Following Eq. (20), $p(y_{1:t})$ can be taken out of the integral of Eq. (18), still in the denominator, though. By consequence, the numerator of Eq. (18) can be evaluated over $q(x_{0:t} | y_{1:t})$ in the same manner, leading to $E_q\{g_t(x_{0:t}) w_t(x_{0:t})\}$.

When these expectations are approximated as in Eq. (16) by drawing N samples from the importance function (proposal distribution) $q(x_{0:t} | y_{1:t})$, we have:

$$\begin{aligned} E\{g_t(x_{0:t})\} &= \frac{E_q\{g_t(x_{0:t}) w_t(x_{0:t})\}}{E_q\{w_t(x_{0:t})\}} \\ &\approx \frac{(1/N) \sum_{i=1}^N g_t(x_{0:t}^{(i)}) w_t(x_{0:t}^{(i)})}{(1/N) \sum_{i=1}^N w_t(x_{0:t}^{(i)})} \\ &= \sum_{i=1}^N g_t(x_{0:t}^{(i)}) \frac{(1/N) w_t(x_{0:t}^{(i)})}{(1/N) \sum_{j=1}^N w_t(x_{0:t}^{(j)})} \end{aligned} \quad (21)$$

$$\begin{aligned} &= \sum_{i=1}^N g_t(x_{0:t}^{(i)}) \tilde{w}_t^{(i)} \\ \tilde{w}_t^{(i)} &= \frac{w_t^{(i)}}{\sum_{j=1}^N w_t^{(j)}} \end{aligned} \quad (22)$$

The normalized importance weight $\tilde{w}_t^{(i)}$ is used in the last equality of Eq. (21) with $\sum_{i=1}^N \tilde{w}_t^{(i)} = 1$.

4.5 Sequential Importance Sampling (SIS)

The third idea is sequential importance sampling (SIS). Under the assumptions that the underlying state corresponds to a Markov process [i.e., $p(x_{0:t}) = \prod_{j=1}^t p(x_j | x_{j-1})$], the observations are conditionally independent given the state [i.e., $p(y_{1:t} | x_{0:t}) = \prod_{j=1}^t p(y_j | x_j)$], and the importance function or *proposal distribution* is factorable [i.e., $q(x_{0:t} | y_{1:t}) = q(x_{0:t-1} | y_{1:t-1}) q(x_t | x_{0:t-1} | y_{1:t-1})$ according to the Bayes' rule], then the un-normalized importance weight can be estimated recursively as:

$$w_t(x_{0:t}) = \frac{p(x_{1:t-1} | x_{0:t-1}) p(y_t | x_t) p(x_{0:t-1}) p(x_t | x_{t-1})}{q(x_{0:t-1} | y_{1:t-1}) q(x_t | x_{0:t-1} | y_{1:t-1})}$$

$$= w_{t-1}(x_{0:t-1}) \frac{p(y_t | x_t) p(x_t | x_{t-1})}{q(x_t | x_{0:t-1} | y_{1:t-1})} \quad (23)$$

Eq. (23) provides a mechanism to sequentially update the importance weight given the conditional proposal distribution $q(x_t | x_{0:t-1} | y_{1:t-1})$. Indeed, we can sample from this proposal distribution (i.e., generate N discrete samples $x_t^{(i)}$ according to $q(x_t | x_{0:t-1} | y_{1:t-1})$ and evaluate the likelihood and transition probabilities [i.e., $p(y_t | x_t)$ and $p(x_t | x_{t-1})$ given by the process and measurement models of Eqs. (1a) and (2a), respectively) for these samples. Doucet et al. [16] show that the proposal distribution is the “*optimal*” *proposal distribution*, that is, the distribution that minimizes the variance of the importance weights conditioned on x_{t-1} and y_t . Table 2 lists the algorithm for a generic particle filter with additional steps discussed below.

4.6 Resampling and Diversification

For any practical implementation, the number of samples that can be drawn from a distribution is limited. As such, the choice of importance function or proposal distribution becomes critical, creating other issues impeding the success of a particle filter.

An easy choice of importance function for a process model with additive Gaussian noise is the transition prior:

$$q(x_t | x_{0:t-1} | y_{1:t-1}) \approx p(x_t | x_{t-1})$$

$$\sim \mathcal{N}(f(u_{t-1}, x_{t-1}, 0), Q_{t-1}) \quad (24)$$

However, this proposal distribution does not incorporate the latest data available and it runs the risk to deplete the samples in the sense that after a few iterations, one of the normalized importance weights tends to 1 while the remaining weights tend to 0. This effectively removes a

large number of samples from the sample set because their importance weights are numerically insignificant.

Table 2. Generic Particle Filter [Yang and Miller, 2005]

Initialization: $t = 0$	Draw the initial state $x_0^{(i)}$ from the prior $p(x_0)$, $i = 1, \dots, N$
For $t = 1, 2, \dots$	
Importance Sampling	<p>Sample $\hat{x}_t^{(i)} \sim q(x_t x_{0:t-1}^{(i)}, y_{1:t})$ and set $\hat{x}_{0:t}^{(i)} = [\hat{x}_{0:t-1}^{(i)}, \hat{x}_t^{(i)}]$ for $i = 1, \dots, N$</p> <p>Evaluate the unnormalized importance weight:</p> $w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t \hat{x}_t^{(i)}) p(\hat{x}_t^{(i)} \hat{x}_{t-1}^{(i)})}{q(\hat{x}_t^{(i)} \hat{x}_{0:t-1}^{(i)}, y_{1:t})}$ <p>for $i = 1, \dots, N$</p> <p>Normalize the importance weights:</p> $\tilde{w}_t^{(i)} = w_t^{(i)} \left[\sum_{j=1}^N w_t^{(j)} \right]^{-1}$ <p>for $i = 1, \dots, N$</p>
Selection (Resampling):	<p>Multiply (or suppress) samples $\hat{x}_t^{(i)}$ with high (or low) importance weights $\tilde{w}_t^{(i)}$ to obtain N random samples $x_{0:t}^{(i)}$ approximately distributed according to $p(x_{0:t} y_{1:t})$</p> <p>Set $w_t^{(i)} = \tilde{w}_t^{(i)} = 1/N$ for $i = 1, \dots, N$</p>
Output:	<p>The output of the algorithm is a set of samples $\hat{x}_{0:t}^{(i)}$ with weights $\tilde{w}_t^{(i)} = 1/N$ for $i = 1, \dots, N$</p> <p>The samples can be used to approximate the posterior distribution as:</p> $p(x_{0:t} y_{1:t}) \approx \hat{p}(x_{0:t} y_{1:t}) = \frac{1}{N} \sum_{i=1}^N \delta(x_{0:t} - \hat{x}_{0:t}^{(i)})$ <p>The samples can also be used to calculate expectation for appropriate functions as:</p> $E(g_t(x_{0:t})) = \int g_t(x_{0:t}) p(x_{0:t} y_{1:t}) dx_{0:t} \approx \frac{1}{N} \sum_{i=1}^N g_t(\hat{x}_{0:t}^{(i)})$ <p>which is the marginal conditional mean of $x_{0:t}$ when $g_t(x_{0:t}) = x_{0:t}$ and the marginal conditional covariance of $x_{0:t}$ when $g_t(x_{0:t}) = x_{0:t} x_{0:t}^T - E_{p(x_{0:t} y_{1:t})}(x_{0:t}) E_{p(x_{0:t} y_{1:t})}^T(x_{0:t})$.</p>

To avoid degeneracy, the fourth idea is importance resampling, also called *selection*, in which samples with low importance weights are eliminated while samples with high importance are multiplied keeping the total population of samples at the same level. Techniques for resampling include sampling importance resampling (SIR), residual resampling, and minimum variance sampling.

Since the selection step favors the creation of multiple copies of the “fittest” particles (thus allows us to track the updated distributions), many “unfit” particles may end up with few or none copies, leading to sample impoverishment. To solve this problem, an additional step is therefore needed to introduce the sample diversification after the selection step without affecting the validity of the approximation. A brute force approach would increase the number of samples. But a refined technique is to implement a Markov chain Monte Carlo (MCMC) step, which moves new particles to areas of more interest in the state space by applying a Markov chain transition kernel. Figure 5 illustrates this process.

4.7 Formulation for Classical Particle Filter (CPF)

1) Draw N particles $x_t^{(i)} = 1, \dots, n$ based on the previous posterior, as follows:

$$\hat{x}_{1:t}^{(i)} \sim p(\hat{x}_t^{(i)}, x_{1:t-1}^{(i)}) \quad (25)$$

2) Assign/update **importance weight**, w_t to each particle:

$$w_t^{(i)} \sim p(y_t | x_t^{(i)}) \quad (26)$$

3) Resample, i.e., draw N particles from the weighted particle set with a probability proportional to the weights (26). Assign the particle set to $x_{1:t}^{(i)}$ with $w_t^{(i)} = 1$.

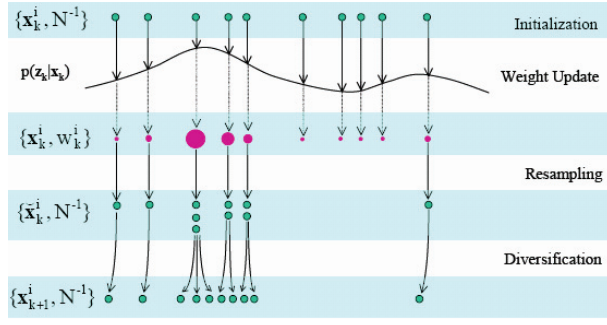


Figure 5. Re-sampling/selection and diversification

Particle filters have computational and representational advantages over other Bayesian techniques. The main problem is that a *large number of particles* are often needed to maintain a reasonable approximation of the state probability distribution to detect rapid maneuvers which requires maintaining and updating large numbers of particles. This is typically not practical due to limited computation. However, a *small particle sets* do not provide reasonable approximations because they are unlikely to represent quick maneuver changes and their estimates are likely to have a high variance. The balance between the number of particles and tracker performance is assessed using the NCI estimator. To overcome such problems, the particle filter may be combined with an EKF or UKF in a hybrid manner [24]. To address the sample size and tracker performance, we use the relative metrics by approximating \mathbf{P} in Eq. (1-3) as a function of the particles.

5 Relative Performance Analysis

We compared the PDAF, UKF, and PF using the relative metrics ANESS and NCI versus RMS. The scenario is presented in Figure 6 which includes long-run target maneuvering scenario to assess the difficulty of a tracker losing track. In the investigation the number of particles was varied between 200 and 5000.

Figure 7 presents the absolute metrics of the trackers. These absolute metrics mask the performance of the tracker relative to the data quality. Thus, we explored methods that look at the estimation quality of the tracker robustness.

Figure 8 shows the linear PDAF performance, with large errors, using the relative performance metrics. During the large track errors, NCI shows that the tracker is optimistic and can be used to ascertain a lost track (if no truth) or errors associated with the models (if truth is available).

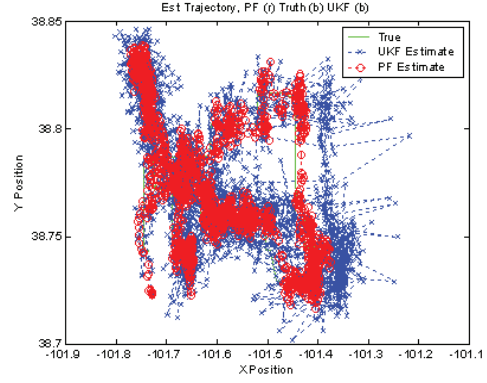


Figure 6. Sampling results

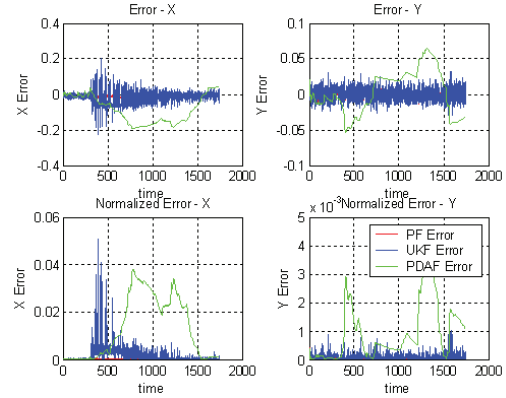


Figure 7. Absolute RMS errors

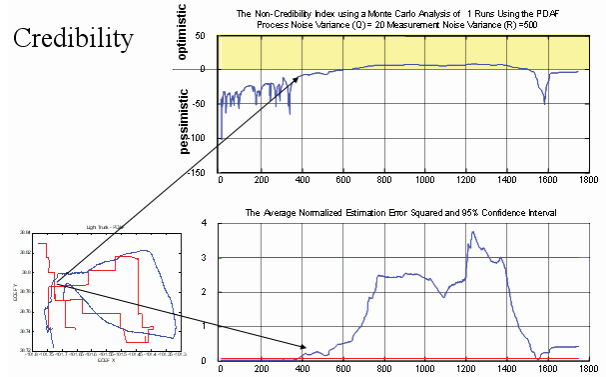


Figure 8. PDAF credibility analysis.

Next, we show performance results to determine a model mismatch or bias in the estimator for the tracking problem using nonlinear methods. Figures 9-10 demonstrate nonlinear tracking using the NCI and ANESS to determine the tracker performance independent of the data.

6 Discussion & Conclusions

We looked at the standard RMS track metrics which show the sensor-tracker accuracy; however, we desired to assess tracker performance independent of the data. To do this, we applied the Relative Track Metrics proposed by X.

Rong Li. In a series of simulation experiments, the NCI demonstrated the ability to detect a model mismatch or bias in the results. Also, we were able to implement the NCI in nonlinear tracking systems. By estimating the particle region, there is an issue with *PF re-sampling* number versus *relative metric analysis* in the MSE over estimator covariance. We showed sensitivity metrics that address tracker quality independent of measurements useful for nonlinear tracker evaluation. Regions of optimism could spawn particles and the regions of pessimism could reduce particles to save computation. Additionally, the NCI and ANESS could control the spread of particles.

Future work will explore (1) the tracker sensitivity metrics to multiple target tracking, (2) target maneuverability, and (3) addressing simultaneous track and identification. These tracker metrics will be developed for a sensor management cost function and user performance validation.

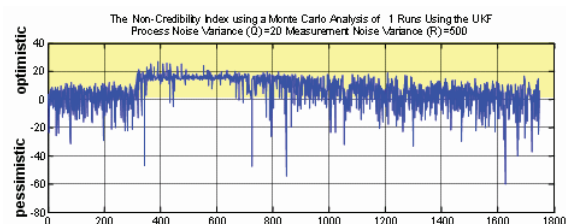


Figure 9. UKF credibility analysis.

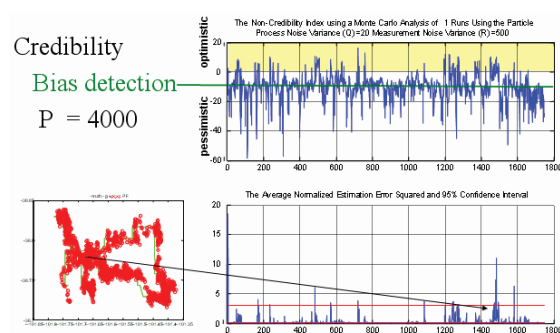


Figure 10. PF credibility analysis.

7 References

- [1] X. R. Li & Z. Zhao, "Evaluation of Estimation Algorithms – Part 1: Local Performance Measures," submitted to *IEEE AES*, 2004.
- [2] Y. Bar-Shalom & X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS, New York, 1995.

- [3] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House Publisher, Boston, 1999.
- [4] HAP Blom & Y Bar-Shalom, "The interacting Multiple Model algorithm for systems with Markovian Switching coefficients, *IEEE Tr. Auto. Cntrl*, Vol. 33, 1988, 780-783.
- [5] X. R. Li, *Probability, Random Signals, and Statistics*, Boca Raton, FL, CRC Press, 1999.
- [6] Y. Bar-Shalom, X. R. Li, & T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, New York Wiley, 2001.
- [7] S. J. Julier, J.K. Uhlmann, & H. Durrant-Whyte, "A New approach for filtering Nonlinear Systems," *Amer. Control Conference*, 1995.
- [8] B. Ristic, S. Arullampalam, & N. Gordon, *Beyond the Kalman Filter: Particle filters for tracking applications*, Artech House, 2004.
- [9] H. Blom & E. Bloem, "Joint IMM-PDA Particle Filter," *Fusion03*.
- [10] M. Mallick, S. Maskell, T. Kirubarajan, & N. Gordon, "Littoral Tracking using Particle Filter," *Fusion2002*, pp.935-942.
- [11] D. Salmond, D. Fisher, & N. Gordon, "Tracking and Identification for closely spaced objects in clutter," in *Proc. European Contr. Conf.*, Brussels, Belgium, 1997.
- [12] E. Blasch, and L. Hong, "Data Association through Fusion of Target Track and Identification Sets," *Fusion00*.
- [13] A. Tchamova, J. Dezert, *et al*, "Target Tracking with Generalized data association based on the general DSsm rule of combination," *Fusion04*.
- [14] D. Angelova & L. Mihaylova, "Joint Tracking and Classification with Particle Filtering and Mixture Kalman Filtering using Kinematic Radar Information," *Digital Signal Processing*, 2005.
- [15] D. Mušicki, S. Suvorova, S. Challa, "Multi Target Tracking of Ground Targets in Clutter with LMIPDA-IMM," *Fusion04*, Stockholm, SW, June 2004.
- [16] O.E. Drummond, "Methodologies for Performance evaluation of Multitarget Multisensor tracking," in *SPIE*, Vol. 3809, pp. 355-369, 1999.
- [17] T. Nguyen, V. Jilkov, and X. R. Li, "Comparison of Sampling-Based Algorithms for Multisensor Distributed Target Tracking," *Fusion03*.
- [18] A. I. El-Fallah, R. P. Mahler, T. Zajic, E. Sorensen, M. Alford & R. K. Mehra, "Scientific Performance evaluation for sensor management," in *SPIE*, Vol. 4053, pp. 183-194, 2000.
- [19] O.E. Drummond, X. R. Li, and C. He, "Comparison of Various Static Multiple-Model Estimation Algorithms," in *SPIE*, Vol. 3373, pp. 510-527, 1998.
- [20] R. Rothrock & O. Drummond, "Performance Metrics for Multiple-sensor, Multiple-Target Tracking," in *SPIE*, Vol. 4048, pp. 521-531, 2000.
- [21] C.Y. Chong, "Problem Characterization in Tracking/Fusion Algorithm Evaluation," *IEEE AES Sys. Mag.*, July 2001.
- [22] X. Zhang & P. Willet, "Cramér-Rao Bounds for Discrete Time Linear Filtering with Measurement Origin Uncertainties," *Tribute to YBS Conf.*, pp. 546-561. 2001.
- [23] M. Hernandez, "Performance bounds for GMTI Tracking," *Fusion03*.
- [24] C Yang & M Miller, "Nonlinear Filtering Techniques for GNSS Data Processing," *ION 61st Ann. Mtg/ MITRE Corp. & Draper Lab.*, 27-29 June 2005, Cambridge, MA, pp. 690-703.
- [25] J.J. Sudano, & Jones, M.H., "An Improved Representation of the True Tracking Errors as a Better Alternative to the Kalman Covariance," *2003 Core Tech. for Space Sys. Conf.*, 2003, CO. <http://www.spacecoretech.org/>
- [26] R. Van der Merwe, *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, PhD Dissertation, Oregon Health and Service Univ., 2004.
- [27] S. J. Julier, "The scaled Unscented Transformation" *American Control Conference*, 2002.