Nonlinear Estimation Techniques for Impact Point Prediction of Ballistic Targets

J. Clayton Kerce^a, George C. Brown^a, and David F. Hardiman^b

^aGeorgia Tech Research Institute, Georgia Institute of Technology, Atlanta, GA USA 30332-0857 ^bU.S. Army Research, Development, and Engineering Command, Redstone Arsenal, AL USA 35898

ABSTRACT

This paper considers three nonlinear estimation algorithms for impact point prediction (IPP) of ballistic targets. The paper assumes measurements are available from a 3D surveillance radar or phased array radar over some portion of the ballistic trajectory. The ballistic target (BT) is tracked using an extended Kalman filter (EKF), an unscented Kalman filter (UKF), and a particle filter (PF). With the track estimate as an initial condition, the equations of motion for the BT are integrated to obtain a prediction of the impact point. This paper compares the performance of the three filters (EKF, UKF, and PF) for impact point prediction. The traditional Extended Kalman Filter equations are based on a first-order Taylor series approximation of the nonlinear transformations (expanded about the latest state estimate). Both the Unscented Kalman Filter and the Particle Filter allow nonlinear systems to be modeled without prior linearizion. The primary focus of the research presented in this paper is comparing the performance and accuracy of the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), and the Particle Filter (PF) for impact point prediction. The three filtering techniques are compared to the theoretical Cramer-Rao lower bounds (CRLB) of estimation error.

Keywords: Estimation, non-linear dynamic system, Cramer-Rao lower bound (CRLB), Kalman filter

1. INTRODUCTION

The objective is to accurately predict the impact point of a ballistic target. It is assumed that the ballistic target is tracked for a limited time. Therefore, at some point, estimation of the ballistic target's trajectory is stopped, and based on this track history; a predicted impact point is generated. In the case of nonlinear filtering, the optimal recursive Bayesian estimator requires the entire posterior density of the target state to be computed sequentially as new measurements are received. However, the recursive propagation of the posterior density, in most cases, cannot be determined analytically. This would require an infinite dimensional vector and therefore infinite storage. Consequently, suboptimal techniques, such as the EKF, UKF, and PF, are prevalent and are evaluated in this paper. These suboptimal techniques are formulated in Sections 3, 4, and 5. The evaluation tool used is the CRLB. The CRLB provides a lower bound for second-order error, that is, the theoretical lower bound of the variance of estimation error for the nonlinear dynamic system. The CRLB is formulated in Section 6. Section 2 details the system dynamic and measurement models. The dynamic model of the ballistic target is characterized by a nonlinear stochastic difference equation. This equation describes how the state vector of the object evolves over time. The measurement model defines how the noisy radar measurements relate to the state vector. In general, it is characterized by a nonlinear stochastic difference equation, as well. Every filter is compared with the CRLB. Section 7 presents IPP performance results, including error analysis, comparison with the CRLB, and covariance consistency. Finally, conclusions are drawn in Section 8.

2. MODELS

Before addressing the analysis, it is first helpful to review the underlying models and assumptions. In the interest of simplicity, the most basic models were selected while maintaining enough complexity to make the analysis interesting. Because of the geometries considered in this paper, a flat Earth model was chosen to simplify the transformations from the different frames internal to the simulation and tracker. As a result, the coordinates for the tracker was chosen to be East North Up (ENU) with the origin arbitrarily selected as the sensor's position on the ground.

2.1. Sensor

The tracker is fed with measurements collected by an electronically scanned phased array. This sensor model takes as input the target's truth position at a given time in ENU and maps it first into array face coordinates (AFC) and then into sine-space (RUV). The transformation from ENU into AFC is carried out via a translation which takes into account the sensor's location with respect to the reference and a rotation which takes into account the sensor's azimuth and tilt angles.

$$\mathbf{p}_{\text{target}_{AFC}} = \mathbf{T}_{\text{local to sensor}}(\mathbf{p}_{\text{target}_{ENU}} - \mathbf{p}_{\text{sensor}_{ENU}})$$
(1)

Given a point in AFC, the transformation into sine space is implemented with the following equations.

$$r = \sqrt{x_{AFC}^2 + y_{AFC}^2 + z_{AFC}^2}$$

$$U = \frac{x_{AFC}}{\sqrt{x_{AFC}^2 + y_{AFC}^2 + z_{AFC}^2}}$$

$$V = \frac{y_{AFC}}{\sqrt{x_{AFC}^2 + y_{AFC}^2 + z_{AFC}^2}}$$
(2)

These transformations may be combined into a single function, $\mathbf{p}_{\text{RUV}} = h(\mathbf{p}_{\text{ENU}})$.

To simplify the analysis, it is assumed that waveform management is performed in a way that guarantees constant SNR regardless of RCS fluctuations and range dependency. It is assumed that some type of angle-of-arrival estimation takes place to produce U and V estimates. The exact algorithm is not specified, but the covariance of the measurements are. Likewise, some type of pulse compression is used to produce range estimates with a constant variance. Given that it is assumed that the SNR is constant throughout the measurements, it is not unreasonable that the variance associated with the measurements is fixed also. Consequently, the measurements, $\mathbf{z}_{\text{meas}_k}$, are generated based on truth, $\mathbf{z}_{\text{truth}_k}$, with added noise based on these statistics, \mathbf{P}_{meas} .

$$\mathbf{z}_{\text{meas}_{k}} = \mathbf{z}_{\text{truth}_{k}} + \sqrt{\mathbf{P}_{\text{meas}}}\mathbf{r}_{k} \tag{3}$$

where $\sqrt{\mathbf{P}_{\text{meas}}} \sqrt{\mathbf{P}_{\text{meas}}}^T = \sqrt{\mathbf{P}_{\text{meas}}}$ and $\mathbf{r}_k \in \mathbb{R}^3$ and the elements are *i.i.d* ~ $\mathcal{N}(0, 1)$. In this way the white noise process \mathbf{r}_k is mapped into a colored noise process which is added to the truth to produce measurements. Such a simple model offers the advantage that the parameters needed to describe radar performance such as waveform parameters, aperture, beamwidth, bandwidth, losses, *etc.* may be ignored.

To further simplify the analysis the need for an association algorithm is eliminated by making the following assumptions.

- only a single target is present
- there are no missed detections
- there are no false alarms

2.2. Dynamics

The ballistic equations considered here are obtained by applying Newtons laws to a point mass ballistic model with drag, and have the general form

$$\dot{\mathbf{x}}_{\mathbf{p}} = \mathbf{x}_{\mathbf{v}}$$
 (4)

$$\dot{\mathbf{x}}_{\mathbf{v}} = \mathbf{g} - \rho \, \mathbf{x}_{\gamma} | \mathbf{x}_{\mathbf{v}} | \mathbf{x}_{\mathbf{v}} \tag{5}$$

$$\dot{\mathbf{x}}_{\gamma} = 0, \tag{6}$$

with initial conditions

$$\mathbf{x}_{\mathbf{p}}(0) = \mathbf{p}_0, \quad \mathbf{x}_{\mathbf{v}}(0) = \mathbf{v}_0, \quad \mathbf{x}_{\gamma} = const.$$

Here the state space is assumed to be seven dimensional cartesian space based on a flat, non-rotating Earth coordinates, $S = \mathbb{R}^7$. The state vector $\mathbf{x} \in S$ is the stacked vector consisting of the position $\mathbf{x}_{\mathbf{p}} \in \mathbb{R}^3$, the velocity $\mathbf{x}_{\mathbf{v}} \in \mathbb{R}^3$,

and the drag coefficient in the direction of the velocity $\mathbf{x}_{\gamma} \in \mathbb{R}$. The gravitational acceleration vector, \mathbf{g} , points towards the surface of the earth and is assumed here to have magnitude 9.8 m/s^2 . This model does not account for important dynamical effects that would be seen by a tracking or fire control radar, such as lift, coning, or the Magnus effect, to name a few.

Dynamic range issues arrise from the above formulation of the ballistic equations when propagating covariances and in performaning measurement updates in the filter. This is due to the fact that $\mathbf{x}_{\gamma} \approx 10^{-3} (kg/m^2)$ while $\mathbf{x}_{\mathbf{v}} \approx 10^2 (m/s)$ and $\mathbf{x}_{\mathbf{p}} \approx 10^4 (m)$. For this reason, the ballistic coefficient $\beta = \gamma^{-1}$ is typically used in the seventh state variable. This change of variable introduces a term with poor linearization properties, and we choose to work with the $\delta\beta$ formulation of the dynamical equations, introduced by N. Speakman ref[], using the change of variable $\beta^{-1} = \beta_0^{-1} - \beta_0^{-2} \delta\beta$. Here β_0 is a reference ballistic coefficient and $\mathbf{x}_{\delta\beta}$ is the new state variable. With this substitution, the dynamic equations are expressed in the following form:

$$\dot{\mathbf{x}}_{\mathbf{p}} = \mathbf{x}_{\mathbf{v}}$$
 (7)

$$\dot{\mathbf{x}}_{\mathbf{v}} = \mathbf{g} - \frac{\rho}{2\beta_0} \left(1 - \frac{\mathbf{x}_{\delta\beta}}{\beta_0} \right) |\mathbf{x}_{\mathbf{v}}| \mathbf{x}_{\mathbf{v}}$$
(8)

$$\dot{\mathbf{x}}_{\delta\beta} = 0, \tag{9}$$

with initial conditions

 $\mathbf{x}_{\mathbf{p}}(0) = \mathbf{p}_0, \quad \mathbf{x}_{\mathbf{v}}(0) = \mathbf{v}_0, \quad \mathbf{x}_{\delta\beta} = const.$

and $\rho = 1.2255 e^{0.000129 \mathbf{p}(3)}$, representing the atmospheric density.

These equations apply to the propagation of the state in the EKF, UKF, and PF and are solved by fourth order Runga-Kutta integration in all cases. The EKF requires the solution of the matrix Ricatti equation for the covariance:

$$\dot{P}_t = \frac{\partial f}{\partial \mathbf{x}} P_t + P_t \frac{\partial f}{\partial \mathbf{x}}$$

where P_0 is given, and

$$f(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_{\mathbf{v}} \\ -\frac{\rho}{2\beta_0} \left(1 - \frac{\mathbf{x}_{\delta\beta}}{\beta_0}\right) |\mathbf{x}_{\mathbf{v}}| \mathbf{x}_{\mathbf{v}} \\ 0 \end{bmatrix}$$
(10)

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times1} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times1} \end{bmatrix}$$
(11)

$$M_{2,1} = \frac{1}{2\beta_0} \left(1 - \frac{\delta\beta}{\beta_0} \right) |\mathbf{x}_{\mathbf{v}}| \ \mathbf{x}_{\mathbf{v}} \frac{\partial\rho}{\partial\mathbf{x}_{\mathbf{p}}}^T$$
(12)

$$\frac{\partial \rho}{\partial \mathbf{x}_{\mathbf{p}}}^{T} = [0, 0, -c_{1} c_{2} \exp\left(-c_{2}[0, 0, 1]^{T} \mathbf{x}_{\mathbf{p}}\right)]$$
(13)

$$M_{2,2} = -\frac{\rho}{2\beta_0} \left(1 - \frac{\delta\beta}{\beta_0}\right) \left(\frac{1}{|\mathbf{x}_{\mathbf{v}}|} \mathbf{x}_{\mathbf{v}} \mathbf{x}_{\mathbf{v}}^T + |\mathbf{x}_{\mathbf{v}}| I_{3\times 3}\right)$$
(14)

$$M_{2,3} = \frac{\rho}{2\beta_0} |\mathbf{x}_{\mathbf{v}}| \mathbf{x}_{\mathbf{v}}$$
(15)

3. EXTENDED KALMAN FILTER (EKF)

The EKF implemented was the method commonly described in the accepted literature [3].bibitem The two components of the EKF are the prediction step and the update step. In prediction the most recent state including its covariance are propagated to the time of the current measurement. This was implemented with a 4^{th} order Runga Kutta integrator with the same underlying model that generated the truth data.

$$\mathbf{x}_{\mathbf{k}+\mathbf{1}} = f_{\text{propagate}}(\mathbf{x}_{\mathbf{k}}) \tag{16}$$

$$\mathbf{P}_{\mathbf{k}+\mathbf{1}} = f_{\text{propagate}}(\mathbf{P}_{\mathbf{k}}) \tag{17}$$

Equation 16 is implemented via Equations ?? and ??, while Equation 17 is implemented via an extension of these to include covariance.

After prediction, the update was implemented with the following equations:

$$\mathbf{H} = \frac{\partial h(\mathbf{x}_{\mathbf{k}})}{\partial \mathbf{x}_{\mathbf{k}}} \tag{18}$$

$$\mathbf{dz}_{\mathbf{k}} = \mathbf{z}_{\text{meas}_{\mathbf{k}}} - h(\mathbf{x}_{\mathbf{k}}) \tag{19}$$

$$\mathbf{S}_{\mathbf{k}} = \mathbf{H} \mathbf{P}_{\mathbf{k}} \mathbf{H}^{T} + \mathbf{P}_{\text{meas}}$$
(20)

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\mathbf{k}} \mathbf{H}^T \mathbf{S}_{\mathbf{k}}^{-1}$$
(21)

$$\mathbf{x}_{\mathbf{k}+\mathbf{1}} = \mathbf{x}_{\mathbf{k}} + \mathbf{K}_{\mathbf{k}} \mathbf{d} \mathbf{z}_{\mathbf{k}}$$
(22)

$$\mathbf{P}_{\mathbf{k}+\mathbf{1}} = \mathbf{P}_{\mathbf{k}} - \mathbf{K}_{\mathbf{k}} \mathbf{S}_{\mathbf{k}} \mathbf{K}_{\mathbf{k}}^{T}$$
(23)

4. UNSCENTED KALMAN FILTER (UKF)

The Unscented Kalman Filter, first introduced by Simon J. Julier and Jeffrey K. Uhlmann, circumvents the problem of approximating a nonlinear system with a linear system. Its basis is that it is easier to approximate a probability distribution than to approximate a nonlinear function [1]. Unlike the EKF, the UKF does not approximate the nonlinear dynamic and measurement models. Instead, it approximates the posterior by a Gaussian density, which is represented by a set of deterministically chosen sample points. These sample points completely capture the true mean and covariance of the Gaussian density. When propagated through the true nonlinear system, they capture the posterior mean and covariance accurately to the 2^{nd} order for any nonlinearity [2].

4.1. The Unscented Transform

The Unscented Transform (UT) is a method for calculating the statistics of a random variable that undergoes a nonlinear transformation. Consider propagating a random variable, \mathbf{x} , with mean, $\bar{\mathbf{x}}$, and covariance, $\mathbf{P}_{\mathbf{x}}$, through an arbitrary nonlinear function, $g: \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ to produce a random variable, \mathbf{y} .

$$\mathbf{y} = g(\mathbf{x}) \tag{24}$$

The first two moments of \mathbf{y} are computed using the UT as follows. First, $2n_x + 1$ weighted sample points, (\mathbf{x}_i, w_i) , are deterministically chosen so that they completely capture the true mean of \mathbf{x} denotes respectively as $\bar{\mathbf{x}}$ and $\mathbf{P}_{\mathbf{x}}$. A scheme that satisfies this requirement is:

$$\begin{aligned} \mathbf{x}_0 &= \bar{\mathbf{x}} & w_0 &= \frac{\kappa}{n_x + \kappa} & i = 0\\ \mathbf{x}_i &= \bar{\mathbf{x}} + \sqrt{(n_x + \kappa)} \mathbf{P}_{\mathbf{x}} & w_i &= \frac{\kappa}{2(n_x + \kappa)} & i \in \{1, \cdots, n_x\}\\ \mathbf{x}_i &= \bar{\mathbf{x}} - \sqrt{(n_x + \kappa)} \mathbf{P}_{\mathbf{x}} & w_i &= \frac{\kappa}{2(n_x + \kappa)} & i \in \{n_x + 1, \cdots, 2n_x\} \end{aligned}$$

Where κ is a scaling parameter (such that $\kappa + n_x \neq 0$) and $(\sqrt{(n_x + \kappa)}\mathbf{P}_{\mathbf{x}})_i$ is the i^{th} row of the matrix square root \mathbf{L} of $(n_x + \kappa)\mathbf{P}_{\mathbf{x}}$, such that $(n_x + \kappa)\mathbf{P}_{\mathbf{x}} = \mathbf{L}^T \mathbf{L}$, w_i is the weight associated with the i^{th} point such that $\sum_{i=0}^{2n_x} w_i = 1$. Each sigma point is now propagated through the nonlinear function g:

$$\mathbf{y}_i = g(\mathbf{x}_i), \quad \text{for } i \in \{0, 1, \cdots, 2n_x\}$$

$$\tag{25}$$

and the first two moments are computed as follows:

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n_x} w_i \mathbf{y}_i \tag{26}$$

$$\mathbf{P}_{\mathbf{y}} = \sum_{i=0}^{2n_x} w_i (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}})^T$$
(27)

4.2. Filtering Equations

The first step is in the calculation of the $2n_x + 1$ weighted sample points $(\mathbf{x}_{k-1}^i, w_{k-1}^i), i \in \{0, \dots, 2n_x\}$, as shown in section 4.1 (where n_x is the dimension of the state). The prediction is then performed as follows:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n_x} w_i f_{k-1}(\mathbf{x}_{k-1}^i)$$
(28)

$$\mathbf{P}_{\mathbf{x}k|k-1} = \mathbf{Q}_{k} - 1 + \sum_{i=0}^{2n_{x}} w_{i} (f_{k-1}(\mathbf{x}_{k-1}^{i}) - \hat{\mathbf{x}}_{k|k-1}) (f_{k-1}(\mathbf{x}_{k-1}^{i}) - \hat{\mathbf{x}}_{k|k-1})^{T}$$
(29)

The predicted density, $p(\mathbf{x}_k|z_{k-1}) \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{\mathbf{x}_k|k-1})$ is represented by a set of $2n_x + 1$ sample points: $\mathbf{x}_{k|k-1}^i = f_{k-1}(\mathbf{x}_{k-1}^i)$. The predicted measurement is then: $\hat{z}_{k|k-1} = \sum_{i=0}^{2n_x} w_{k-1}^i h(\mathbf{x}_{\mathbf{k}_k|k-1}^i)$. The update step is as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k} + \mathbf{K}_{\mathbf{k}}(z_k - \hat{z}_{k|k-1})$$
(30)

$$\mathbf{P}_{\mathbf{x}k|k} = \mathbf{P}_{\mathbf{x}k|k-1} - \mathbf{K}_{\mathbf{k}}\mathbf{S}_{\mathbf{k}}\mathbf{K}_{\mathbf{k}}^{T}$$
(31)

where

$$\mathbf{S}_{\mathbf{k}} = \mathbf{P}_{\mathbf{z}\mathbf{z}} + \mathbf{P}_{\text{meas}} \tag{32}$$

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\mathbf{x}\mathbf{z}}\mathbf{S}_{\mathbf{k}}^{-1} \tag{33}$$

$$\mathbf{P_{zz}} = \sum_{i=0}^{2n_x} w_{k-1}^i (h(\mathbf{x_k}_{k|k-1}^i) - \hat{z}_{k|k-1}) (h(\mathbf{x_k}_{k|k-1}^i) - \hat{z}_{k|k-1})^T$$
(34)

$$\mathbf{P_{xz}} = \sum_{i=0}^{2n_x} w_{k-1}^i (\mathbf{x_k}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}) (h(\mathbf{x_k}_{k|k-1}^i) - \hat{z}_{k|k-1})^T$$
(35)

5. PARTICLE FILTER (PF)

Particle filters perform sequential Monte Carlo (SMC) estimation based on point mass representation of probability densities. Particle filters address system nonlinearities with little or no approximation at the cost of more computational complexity. Particle filters impose no structure on the underlying probability density function of the unknown quantity (recall the Kalman filter imposes the Gaussian structure). State estimation problems with nonlinear dynamic or measurement models and non-Gaussian noise processes pose fundamental difficulties when applying standard Kalman filtering techniques. Alternatively, particle filtering techniques are well-suited to problems in which nonlinearities and non-Gaussian noise processes exist. A particle filter approximates the probability density function of an unknown random variable (or random vector) by a set of samples, in which each sample represents a possible value of the unknown random quantity. Thus, the filter's accuracy is a function of the number of particles propagated by the filter. While the particle filter method can require significant computational resources due to the fact that many particles need to be used at each time step, the introduction of new computational methods and the ever-increasing speed of computers have made particle filters useful for many real-world applications. The filter consists of two steps: the particles are propagated through the system dynamic equations in a prediction step and the weights associated with each particle are updated based on the new observations. A re-sampling step is used to eliminate particles with low weights and to repopulate the sample set to maintain accuracy.



Figure 1. Laydown description with sensor array denoted by the box, the trajectory, and the region of ascent were measurements are collected.

6. CRAMER-RAO LOWER BOUND (CRLB) 7. EXAMPLES

The analysis is based on a single geometry test case (see Figure 1 and is defined with the following parameters describing the state at launch and impact along with the parameters denoting the sensor position and orientation.

$$\mathbf{x}(t=0)_{\text{launch}} = \begin{bmatrix} 1000\\ -250\\ 0\\ 100\\ 0\\ 100\\ 0 \end{bmatrix} \qquad \mathbf{x}(t=27+)_{\text{impact}} = \begin{bmatrix} 2393.4\\ -250.0\\ 0.0\\ 58.9\\ 0\\ -82.1\\ 0 \end{bmatrix}$$
$$\mathbf{p}_{\text{sensor}_{\text{ENU}}} = \begin{bmatrix} 0.0\\ 0.0\\ 10\\ 10 \end{bmatrix} \qquad \theta_{\text{sensor}_{\text{tilt}}} = 20^{\circ}\\ \theta_{\text{sensor}_{\text{azimuth}}} = 90^{\circ}$$

To better understand the relative performance of these three filters, several sets of Monte Carlo trials were conducted. All simulations were carried out over 100 Monte Carlo trials with each filter receiving the same measurements for a given trial. The trials varied parametrically the angle components of the measurement variance and the velocity components of the process noise. The process noise was defined as: $\mathbf{Q} = \text{diag}([0\ 0\ 0\ 1e3Q_{\text{scale}}\ 1e3Q_{\text{scale}}\ 1e3Q_{\text{scale}}\ 1e4])$.

Because each filter receives the same measurements the estimates of the three candidate methods are follow each other as can be seen in an example of a typical depicted in Figure 2. Because of their similarity, the EKF and UKF produce essentially the same plots with this data. The small deviations of the particle filter are due to the dissimilarity of structure with the other two.

Figures 2-11 denote individual trial results. For large Q_{scale} (see Figure 2) the σ curves of the track covariances at impact are closely matched.

As is expected the covariance will drop if the process noise is not inflated and the σ curves of the trackers tend to separate for a given trial. However, if the statics are collected as seen in Table 1, it is evident that the performance of all three trackers are closely matched over a wide range of the parametric parameters evaluated.

Q_{scale}	$[\mathbf{R}_z]_{2,2}$	$[\mathbf{R}_{z}]_{3,3}$	EKF			UKF				PF				
	/	/	Bia	s (m)	STD	(m)	Bia	s (m)	STD	(m)	Bia	s (m)	STD	(m)
			East	North	A	В	East	North	А	В	East	North	Α	В
1/1	1/4	1/4	0.9	-8.1	74.7	74.2	-0.1	-8.0	74.5	74.1	0.1	-7.7	75.6	75.2
1/2	1/4	1/4	-3.6	3.6	62.1	61.9	-4.3	3.6	62.0	61.7	-5.0	3.3	62.7	62.4
1/4	1/4	1/4	-12.9	-2.5	44.7	42.7	-13.4	-2.4	44.7	42.6	-12.7	-2.3	46.3	44.4
1/8	1/4	1/4	1.0	1.5	36.0	35.9	0.6	1.5	35.9	35.8	0.5	1.7	36.7	36.6
1/16	1/4	1/4	0.9	0.2	27.5	27.5	0.5	0.2	27.4	27.4	-0.5	0.1	29.3	29.3
1/32	1/4	1/4	0.6	-2.4	26.1	26.0	0.4	-2.4	26.0	25.8	-0.4	-2.4	27.6	27.5
1/64	1/4	1/4	-0.0	-0.2	22.3	22.3	-0.3	-0.2	22.2	22.2	-1.9	-0.2	24.4	24.4
1/128	1/4	1/4	-2.8	1.2	19.8	19.6	-3.1	1.1	19.7	19.4	-2.3	1.1	20.7	20.5
1/256	1/4	1/4	-2.8	1.0	18.8	18.6	-3.1	1.0	18.7	18.4	-2.4	1.0	19.9	19.7
1/512	1/4	1/4	-0.1	0.9	17.7	17.7	-0.4	0.9	17.5	17.5	-0.6	0.9	19.9	19.8
1/1024	1/4	1/4	-0.1	0.4	17.4	17.4	-0.4	0.4	17.1	17.1	-1.3	0.4	19.9	19.9
1/2048	1/4	1/4	1.6	-0.4	19.9	19.9	1.2	-0.4	19.6	19.6	0.4	-0.3	21.7	21.7
1/8	1/2	1/2	-0.8	0.1	38.8	38.8	-1.2	0.1	38.6	38.6	-2.3	-0.1	39.8	39.8
1/8	1	1	0.2	-5.4	47.1	46.7	-0.0	-5.4	46.9	46.5	-0.9	-5.7	48.0	47.7
1/8	2	2	1.5	5.7	50.9	50.6	1.5	5.5	50.6	50.3	1.2	5.8	52.0	51.7
1/8	4	4	11.5	-3.2	75.3	74.3	13.1	-2.9	74.2	73.0	13.8	-4.0	78.0	76.7
1/8	8	8	6.5	6.2	173.7	173.5	10.8	6.0	155.6	155.1	6.9	9.1	156.7	156.3
1/128	1/2	1/2	-0.8	0.1	38.8	38.8	-1.2	0.1	38.6	38.6	-2.3	-0.1	39.8	39.8
1/128	1	1	-0.8	0.1	38.8	38.8	-1.2	0.1	38.6	38.6	-2.3	-0.1	39.8	39.8
1/128	2	2	3.3	-0.2	17.5	17.2	3.0	-0.2	17.2	16.9	3.5	-0.1	20.0	19.7
1/128	4	4	11.5	-3.2	75.3	74.3	13.1	-2.9	74.2	73.0	13.8	-4.0	78.0	76.7
1/128	8	8	19.6	-28.3	187.6	184.4	14.9	-8.3	158.2	157.3	5.6	-16.1	180.4	179.6
1/2048	1/2	1/2	0.1	1.4	18.6	18.5	-0.5	1.4	18.4	18.4	-0.6	1.3	21.1	21.0
1/2048	1	1	1.5	-1.2	25.8	25.7	0.7	-1.2	25.7	25.7	-0.8	-1.7	28.7	28.6
1/2048	2	2	1.7	3.7	43.0	42.8	-0.0	3.8	42.9	42.8	0.6	4.7	46.6	46.4
1/2048	4	4	-1.3	2.1	78.1	78.0	-3.5	2.3	76.7	76.6	-4.3	1.8	79.5	79.4
1/2048	8	8	14.2	-12.7	160.5	159.4	13.7	-9.2	137.3	136.3	16.5	-4.7	144.4	143.4
1/8	1/2	1/4	-1.4	0.1	38.4	38.4	-1.7	0.1	38.3	38.3	-2.7	-0.1	39.5	39.4
1/8	1	1/4	-1.0	-5.5	45.3	44.9	-1.2	-5.5	45.2	44.8	-1.9	-5.9	46.2	45.8
1/8	2	1/4	1.4	3.1	47.4	47.3	1.2	3.0	47.3	47.2	0.1	2.4	49.4	49.3
1/8	4	1/4	-0.9	-5.0	71.1	70.9	-1.3	-5.2	70.5	70.3	-2.9	-7.8	73.2	72.7
1/8	8	1/4	4.9	-25.6	167.3	165.3	3.6	-12.2	146.0	145.4	0.9	-18.1	144.1	143.0
1/128	1/2	1/4	-1.4	0.1	38.4	38.4	-1.7	0.1	38.3	38.3	-2.7	-0.1	39.5	39.4
1/128	1	1/4	-1.4	0.1	38.4	38.4	-1.7	0.1	38.3	38.3	-2.7	-0.1	39.5	39.4
1/128	2	1/4	3.3	-0.2	17.5	17.2	3.0	-0.2	17.2	16.9	3.5	-0.1	20.0	19.7
1/128	4	1/4	-0.9	-5.0	71.1	70.9	-1.3	-5.2	70.5	70.3	-2.9	-7.8	73.2	72.7
1/128	8	1/4	7.2	-14.3	158.8	158.0	1.9	-7.0	128.7	128.5	1.5	-1.4	$13\overline{3.4}$	$13\overline{3.4}$
1/2048	1/2	1/4	0.0	1.4	18.6	18.5	-0.4	1.4	18.4	18.3	-0.5	1.3	20.7	20.6
1/2048	1	1/4	0.5	-1.2	24.6	24.6	-0.0	-1.2	24.5	24.5	-1.7	-1.7	26.7	26.6
1/2048	2	1/4	1.7	-2.4	36.1	35.9	0.6	-2.1	35.9	35.9	1.6	-2.2	39.1	39.0
1/2048	4	1/4	4.6	-4.9	70.8	70.5	2.4	-2.2	69.8	69.7	1.6	-3.8	71.7	71.6
1/2048	8	1/4	10.8	-33.4	164.0	160.2	6.8	-6.3	132.9	132.6	6.0	-8.6	136.3	135.8

Table 1. Bias and standard deviation IPP estimates averaged over 100 Monte Carlo Runs computed with the three candidate techniques.

		$[\mathbf{R}_z]_{2,2}$ (deg)									
		1/4	1/2	1	2	4	8				
	1/4	17.2	18.5	22.9	35.2	63.8	122.5				
	1/2		19.0								
$[\mathbf{R}_{z}]_{3,3}$	1			24.2							
(deg)	2				38.6						
	4					71.1					
	8						137.4				

Table 2. IPP CRLB (m) as a function of angle-of-arrival estimate variance in measurements.



Figure 2. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.



Figure 3. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.



IPP with $Q_{scale} = 1/2048$, $R_{z} = diag([30 (m) , (1/4)^{\circ}, (1/4)^{\circ}])^{2}$

Figure 4. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.



Figure 5. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

Figure 6. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

Figure 7. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

IPP with $Q_{scale} = 1/128$, $R_{z} = diag([30 (m), (8)^{\circ}, (1/4)^{\circ}])^{2}$

Figure 8. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

Figure 9. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

Figure 10. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

Figure 11. Comparison of Impact Point Prediction error of 100 Monte Carlos runs of the EKF, UKF, and PF.

8. CONCLUSIONS AND FURTHER RESEARCH

Because of the similarity of performance of the three filter methods over a wide range of parametric parameters, it was shown that for ballistic problems like the one evaluated here, there the particle filter offers no improvement over the EKF and UKF. In fact when operation counts are considered, the EKF must be considered the algorithm of choice.

REFERENCES

- 1. Julier, Simon J. and Uhlmann, Jeffrey K. "Unscented Filtering and Nonlinear Estimation" *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004.
- 2. Ristic, Branko, Arulampalam, Sanjeev, and Gordon, Neil, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, Boston, MA. 2004.
- 3. Y. bar-Shalom, X. R. Li, T. Kirubarajan *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY 2001.
- 4. Tichavsky, Petr, Muravchik, Carlos, Nehorai, Arye, Posterior Cremer-Rao Bounds for Discrete-Time Nonlinear Filtering, Vol 46, No. 5, May 1998.