Comparison of Extended and Unscented Kalman, Particle, and Smooth Variable Structure Filters on a Bearing-Only Target Tracking Problem

S. A. Gadsden[a], D. Dunne[b], S. R. Habibi[a], and T. Kirubarajan[b]
[a]MECH Department, McMaster University, {gadsdesa, habibi}@mcmaster.ca
[b]ECE Department, McMaster University, {ddunne, kiruba}@mcmaster.ca

ABSTRACT

In this paper, we study a nonlinear bearing-only target tracking problem using four different estimation strategies and compare their performances. This study is based on a classical ground surveillance problem, where a moving airborne platform with a sensor is used to track a moving target. The tracking scenario is set in two dimensions, with the measurement providing angle observations. Four nonlinear estimation strategies are used to track the target: the popular extended and unscented Kalman filters (EKF/UKF), the particle filter (PF), and the relatively new smooth variable structure filter (SVSF). The SVSF is a predictor-corrector method used for state and parameter estimation. It is a sliding mode estimator, where gain switching is used to ensure that the estimates converge to true state values. An internal model of the system, either linear or nonlinear, is used to predict an a priori state estimate. A corrective term is then applied to calculate the a posteriori state estimate, and the estimation process is repeated iteratively. The performances of these methods applied on a bearing-only target tracking problem are compared in terms of estimation accuracy and filter robustness.

Keywords: bearing-only tracking, state estimation, extended Kalman filter, unscented Kalman filter, particle filter, smooth variable structure filter

1. INTRODUCTION

In target tracking applications, one may be concerned with surveillance, guidance, obstacle avoidance or tracking a target given some measurements [1]. In a typical scenario, sensors provide a signal that is processed and output as a measurement. These measurements are related to the target state, and are typically noise-corrupted observations [1]. The target state usually consists of kinematic information such as position, velocity, and acceleration. The measurements are processed in order to form and maintain tracks, which are a sequence of target state estimates that vary with time [1]. A tracking filter is used in a recursive manner to carry out the target state estimation.

A good target tracking benchmark is the nonlinear bearing-only measurement problem [1, 2]. These problems are of great interest in sonar applications, where two-dimensional bearing-only targets are prevalent [3]. Many techniques have been developed to solve these types of nonlinear problems. A very popular method is the EKF, which is an extension of the Kalman filter (KF) [4]. The KF provides an elegant and statistically optimal solution for linear dynamic systems in the presence of Gaussian white noise. The EKF is conceptually similar to the iterative KF process, except that the nonlinear system and measurement matrices are linearized according to its corresponding Jacobian, which is a first-order partial derivative. In this simulation scenario, only the measurement equation requires linearization. Furthermore, a moving observer measures noisy bearings to a target on the ground [2]. Based on these measurements, filters are used to obtain estimates of the position and velocity of the target.

The following section describes the simulation scenario and tracking problem in more detail. Section 3 provides a general overview and implementation strategy for the EKF. Following that, the UKF is described. The PF is introduced in Section 5, followed by the SVSF is in Section 6. The results of the simulation are then provided followed by a brief comparison of the two methods. A list of the nomenclature may be found in Appendix A.

2. Bearing-Only Tracking Scenario

The benchmark problem that is studied here is shown in Fig. 1. This problem is described in [2, 4], and will be presented as such. An elevated platform with a sensor travels according to the following equations:

$$x_{p_k} = \bar{x}_{p_k} + \Delta x_{p_k} \tag{1}$$

$$y_{p_k} = \bar{y}_{p_k} + \Delta y_{p_k} \tag{2}$$

Where x_p and y_p are the horizontal and vertical position coordinates, respectively. The first term on the right-hand side of the above two equations refers to the average platform position coordinates. The last term represents perturbations (i.e. random wind disturbances), and are assumed to be zero-mean Gaussian and independent with variances of $R_x = 1\ \mathrm{m}^2$ and $R_y = 1\ \mathrm{m}^2$, respectively. Note that k represents the discrete time sequence (from 0 to 20 seconds).

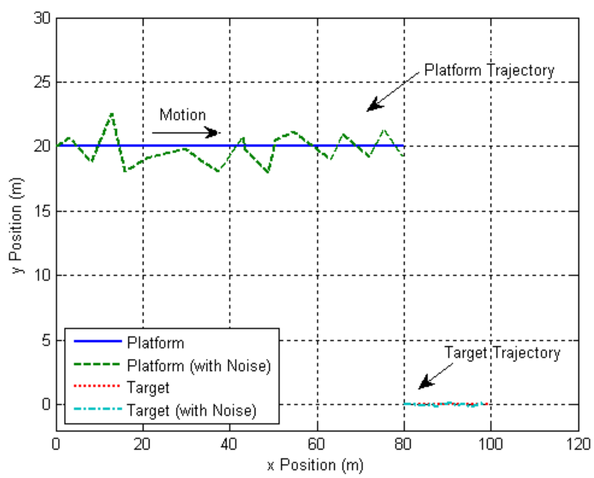

Figure 1. Platform and Target Trajectory

The average platform motion is assumed to be horizontal with constant velocity, and may be described by the following two equations [2]:

$$\bar{x}_{p_k} = 4k \tag{3}$$

$$\bar{y}_{p_k} = 20 \tag{4}$$

The system equation (for the target) is defined according to the following:

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} w_k \tag{5}$$

The state vector is defined by the position (m) and velocity (m/s) of the target. The sampling period used in this simulation was 1 seconds, and is defined by T. The system noise described by w_k is zero-mean Gaussian with a variance of $Q = 10^{-2}$ m^2/s^4. The initial position of the target was set to 80 m, and the initial velocity was set to 1 m/s.

The nonlinear measurement (sensor) equation is defined by:

$$z_{k+1} = \tan^{-1} \frac{y_{p_{k+1}}}{x_{1_{k+1}} - x_{p_{k+1}}} + v_{k+1} \tag{6}$$

The first term on the right-hand side of equation 6 is the measured bearing between the horizontal and the line-of-sight from the sensor to the target [2]. The measurement noise v_{k+1} is defined as zero-mean Gaussian with a variance of $R_s = (3°)^2$.

3. Extended Kalman Filter

3.1 Introduction

As previously mentioned, the KF provides an elegant and statistically optimal solution for linear dynamic systems in the presence of Gaussian white noise. It is a method that utilizes measurements linearly related to the states, and error covariance matrices, to generate a gain referred to as the Kalman gain. This gain is applied to the a priori state estimate, thus creating an a posteriori estimate. The estimation process continues in a predictor-corrector fashion while maintaining a statistically minimal state error covariance matrix for linear systems.

The following two equations describe the system dynamic model and the measurement model used in general for (linear) state estimation.

$$x_{k+1} = F_k x_k + w_k \tag{7}$$

$$z_{k+1} = H_{k+1} x_{k+1} + v_{k+1} \tag{8}$$

The next five equations form the KF algorithm, and are used in an iterative fashion, in conjunction with Eqs. (7) and (8). Equation 9 extrapolates the a priori state estimate, and Eq. (10) is the corresponding state error covariance. The Kalman gain may be calculated by Eq. (11), and is used to update the state estimate and error covariance, described by Eqs. (12) and (13), respectively.

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k} \tag{9}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \tag{10}$$

$$K_k = P_{k+1|k} H_k^T [H_k P_{k+1|k} H_k^T + R_k]^{-1} \tag{11}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_k [z_k - H_k \hat{x}_{k+1|k}] \tag{12}$$

$$P_{k+1|k+1} = [I - K_k H_k] P_{k+1|k} \tag{13}$$

The effects due to mismodeling can be negative, as both the Kalman gain and covariance matrix calculations are dependent on the system and measurement matrices. Furthermore, the performance and stability of the KF may also be dependent on the definition of the process and measurement noise, made through covariance matrices [5, 6]. Overlooked nonlinearities in the system may also cause the KF to become unstable. The EKF may be used for nonlinear systems. It is conceptually similar to the iterative KF process. The nonlinear system (*F*) and measurement (*H*) matrices are linearized according to its corresponding Jacobian, which is a first-order partial derivative. This linearization can sometimes cause instabilities when implementing the EKF [5].

3.2 Implementation of the EKF

The EKF was implemented in what is referred to as mixed coordinates. The measurement was left in polar coordinates (bearing only), while the states of the target were in Cartesian coordinates. The system matrix in this case is already linear; however the measurement matrix is nonlinear. Taking the partial derivative of the nonlinear component in Eq. (6) with respect to the first state yields:

$$\frac{\partial h}{\partial x_1} = -\frac{y_p}{(x_1 - x_p)^2 + y_p^2} \tag{14}$$

Furthermore, note that only the knowledge of the average platform positions are used such that the linearized form of the nonlinear measurement matrix becomes:

$$H = \begin{bmatrix} -\dfrac{\bar{y}_p}{(\hat{x}_1 - \bar{x}_p)^2 + \bar{y}_p^2} & 0 \end{bmatrix} \tag{15}$$

Two cases were studied: normal conditions, and poor initial conditions. Under the first scenario, the initial position used by the EKF was set to the true value (80 m) and the initial velocity was set to 1 m/s. During the second scenario, the initial position estimate was set to 40 m. The initial covariance matrix used by the EKF is defined as follows:

$$P_{0|0} = \begin{bmatrix} 30 & 0 \\ 0 & 1 \end{bmatrix} \tag{16}$$

4. UNSCENTED KALMAN FILTER

4.1 Introduction

The UKF is a popular extension of the KF, and is typically more accurate than the EKF since it is able to capture a higher order of the nonlinearities. The UKF approximates the posterior distribution of the states by a Gaussian density, using a set of deterministically chosen sample points which, after a transformation, captures the true mean and covariance up to the second order of nonlinearity [1]. The UKF is sometimes referred to as a type of linear regression Kalman filter since it is based on statistical linearization, rather than analytical linearization like the EKF [1]. An advantage of the UKF method is the fact that no calculation of the first-order Jacobian is necessary, which can be a challenging task. An unscented transform is a method for calculating the random variable statistics after it goes through a nonlinear transformation [1]. A number ($2n$ + 1) of weighted sample points are deterministically chosen in an attempt to accurately approximate the true mean and covariance of the state distribution [1]. Given a nonlinear system, and based on the assumption that the posterior density of the state is Gaussian, one may first attempt to represent this density using the following two equations:

$$\hat{x}_{k+1|k} = \sum_{i=0}^{N-1} W_k^i f_k(X_k^i) \qquad (17)$$

$$P_{k+1|k} = Q_{k+1} + \sum_{i=0}^{N-1} W_k^i [f_k(X_k^i) - \hat{x}_{k+1|k}][f_k(X_k^i) - \hat{x}_{k+1|k}]^T \qquad (18)$$

The predicted density (represented by N sample points) and measurement may be calculated as follows:

$$X_{k+1|k}^i = f_k(X_k^i) \qquad (19)$$

$$z_{k+1|k} = \sum_{i=0}^{N-1} W_k^i h_k(X_{k+1|k}^i) \qquad (20)$$

The update step may be described by:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - \hat{z}_{k+1|k}) \qquad (21)$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} S_{k+1} K_{k+1}^T \qquad (22)$$

Please refer to Appendix B for further equations supporting the UKF derivation. Note that the UKF assumes a Gaussian distribution in its derivation, when in fact many nonlinear systems are non-Gaussian. This difference can lead to discrepancies and errors in the estimated values.

4.2 Implementation of the UKF

The UKF was implemented using the above equations.

5. Particle Filter

5.1 Introduction

The particle filter (PF) has many names: Monte Carlo filters, interacting particle approximations [7], bootstrap filters [8], condensation algorithm [9], and survival of the fittest [10], to name a few. Compared to the KF, it is a newer development, being introduced in 1993. Since then, the PF has become a very popular method for solving nonlinear estimation problems, ranging from predicting chemical processes to target tracking. The PF takes the Bayesian approach to dynamic state estimation, in which one attempts to accurately represent the probability distribution function (PDF) of the values of interest [1]. The PDF contains all of the pertinent statistical information, and may be considered as holding the solution to the estimation problem [1]. Essentially, the distribution holds a probability of values for the state being observed. The stronger or tighter the prediction PDF, the more accurate the state estimate.

The PF obtains its name from the use of weighted particles or 'point masses' that are distributed throughout the PDF to form an approximation. These particles are used in a recursive manner to obtain new particles and importance weights, with the goal of creating a more accurate approximation of the PDF. In general, as the number of implemented particles becomes very large, the PDF becomes more accurate [1]. An important step in the PF is that of resampling, which eliminates particles with low weights and multiplies those with high weights [1]. This helps to avoid the degeneracy problem with the PF, which refers to only one particle having a significant importance weight after a large number of recursions. Furthermore, it also increases the accuracy of the PDF approximation by replicating particles with high weights. The sequential importance resampling (SIR) algorithm is a very popular form of the PF, and may be summarized by Eqs. (23) to (26). The first equation draws samples or particles from the proposal distribution. Equation (24) updates the importance weights up to a normalizing constant. Next, the normalized weights are calculated for each particle. Finally, a constant known as the effective number of particles is calculated as shown in Eq. (26). Resampling is performed if the effective number is lower than some design threshold.

$$x_k^{(n)} \sim \pi(x_k \mid x_{k-1}^{(n)}, y_k) \tag{23}$$

$$\hat{\omega}_k^{(n)} = \omega_{k-1}^{(n)} \frac{p(y_k \mid x_k^{(n)}) p(x_k^{(n)} \mid x_{k-1}^{(n)})}{\pi(x_k \mid x_{k-1}^{(n)}, y_k)} \tag{24}$$

$$\omega_k^{(n)} = \frac{\hat{\omega}_k^{(n)}}{\sum_{i=1}^{n} \hat{\omega}_k^{(i)}} \tag{25}$$

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{n} (\omega_k^{(i)})^2} \tag{26}$$

5.2 Implementation of the PF

The PF was implemented using N = 5,000 particles with an effective number of particles set to 0.8. The code was initialized by sampling from the distribution used to initialize the EKF.

6. Smooth Variable Structure Filter

6.1 Introduction

In 2002, the variable structure filter (VSF) was introduced as a new predictor-corrector method used for state and parameter estimation [6, 11]. It is a type of sliding mode estimator, where gain switching is used to ensure that the estimates converge to true state values. An internal model of the system, either linear or nonlinear, is used to predict an a priori state estimate. A corrective term is then applied to calculate the a posteriori state estimate, and the estimation process is repeated iteratively. The SVSF was later derived from the VSF, and uses a simpler and less complex gain calculation [12]. In its present form, the SVSF is stable and robust to modeling uncertainties and noise, given an upper bound [12]. The basic concept of the SVSF is shown in Fig. 2. Assume that the solid line in Fig. 2 is a trajectory of some state (amplitude versus time). An initial value is selected for the state estimate. The estimated state is pushed towards the true value. Once the value enters the existence subspace, the estimated state is forced into switching along the system state trajectory [12].

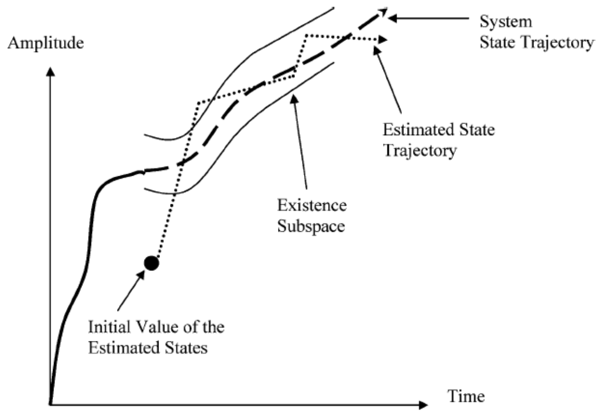

Figure 2. SVSF Estimation Concept [12]

The SVSF method is model based and applies to smooth nonlinear dynamic equations. The estimation process may be summarized by Eqs. (27) to (30), and is repeated iteratively. An a priori state estimate is calculated using an estimated model of the system. In this case, there is no input into the system. This a priori value is then used to calculate an a priori estimate of the measurement, defined by Eq. (28). A corrective term, referred to as the SVSF gain, is calculated as a function of the error in the predicted output, as well as a gain matrix and the smoothing boundary layer width. The corrective term calculated in Eq. (29) is then used in Eq. (30) to find the a posteriori state estimate.

$$\hat{x}_{k+1|k} = \hat{f}(\hat{x}_{k|k}, u_k) \tag{27}$$

$$\hat{z}_{k+1|k} = \hat{h}(\hat{x}_{k+1|k}) \tag{28}$$

$$K_{k+1} = \hat{H}^{+} \left| \left(\left| e_{z_{k+1|k}} \right|_{ABS} + \gamma \left| e_{z_{k|k}} \right|_{ABS} \right) \right|_{ABS} \circ sat(e_{z_{k+1|k}}, \Psi) \tag{29}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \tag{30}$$

$$e_{z_{k|k}} = z_k - \hat{z}_{k|k} \tag{31}$$

$$e_{z_{k+1|k}} = z_{k+1} - \hat{z}_{k+1|k} \tag{32}$$

Two critical variables in this process are the a priori and a posteriori output error estimates, defined by Eqs. (31) and (32), respectively [12]. Note that Eq. (31) is the output error estimate from the previous time step, and is used only in the gain calculation.

6.2 Implementation of the SVSF

The initial conditions used by the SVSF were the same as those used by the EKF. The nonlinear measurement matrix is linearized as per Eq. (15) and correspondingly used in Eq. (29). There are two main SVSF design parameters. The first parameter (γ) controls the speed of convergence, where as the second (Ψ) refers to the boundary layer width which is used to smooth out the switching action. These parameters were tuned by trial-and-error, based on minimizing the estimation error. The first parameter was set to the following:

$$\gamma = 0.2 \tag{33}$$

To increase the quality of the estimate (in terms of convergence speed and estimation accuracy), the boundary layer was made to change with time. Essentially, the variable boundary layer allows the state estimate to approach the true value as quickly as possible by using a large boundary layer width. Once the state estimate is within an acceptable range of the true value, the layer width is decreased, and the estimate is smoothed out. A simple two-stage approach was used, as shown in Tab. 1, where the values were determined by trial-and-error. For example, if the absolute position error (between the estimate and the true value) was greater than 2.0, a value of 10 was set for the boundary layer width. Once the absolute position error was less than 2.0, the boundary layer was reduced to a smaller value (5x10^{-3}).

Table 1. Values for the SVSF Variable Boundary Layer

State	$1 / \Psi_1$	$1 / \Psi_2$	Error Range
Position	0.1	200	> |2.0|
Velocity	80	50,000	> |0.4|

Furthermore, since there is no velocity measurement available, the SVSF algorithm described in the previous section needs to be slightly modified. The measurement function described by Eq. (6) needs to be augmented such that a position estimate is formed based on the horizontal and vertical platform positions, and the measurement, as follows:

$$\hat{\sigma}_{1k+1} = \frac{\bar{y}_{p\,k+1}}{\tan(z_{k+1})} + \bar{x}_{p\,k+1} \tag{34}$$

Furthermore, an estimate of the velocity based on Eq. (34) may be defined by:

$$\hat{\sigma}_{2k+1} = \frac{\hat{\sigma}_{1k+1} - \hat{\sigma}_{1k}}{T} \tag{35}$$

The output errors used in Eqs. (31) and (32) are then determined by calculating the following, where the second term is found by Eqs. (27) and (35):

$$e_{\sigma\,k+1|k} = \hat{\sigma}_{2k+1|k} - \hat{x}_{2k+1|k} \tag{36}$$

7. POSTERIOR CRAMÉR-RAO LOWER BOUND

The Cramér-Rao lower bound (CRLB) is defined as the inverse of the Fisher information matrix (FIM), which quantifies the available information found in the observations about a state [4]. The CRLB provides a lower bound on the achievable variance in the estimation of a parameter. A derivation that can be used for discrete-time nonlinear filtering is the posterior form (PCRLB) [13-15]. This allows meaningful evaluations of estimation techniques, such that the root mean square error (RMSE) for each filter can be determined and compared with the PCRLB. Ideally, one would want the RMSE to reach the PCRLB, or be as close as possible. The CRLB of the error covariance matrix is defined as the inverse of the FIM [4]:

$$C = E\{[\hat{x} - x][\hat{x} - x]^T\} \ge J^{-1} \tag{37}$$

The inverse of the PCRLB may be calculated recursively as follows [13]:

$$J_{k+1} = (Q_k + F_k J_k^{-1} F_k^T)^{-1} + H(x_k)^T R_k^{-1} H(x_k) \tag{38}$$

8. Simulation Results

An example of the position and velocity target estimates (single run) is shown in Fig. 3. Under normal conditions, when compared with the EKF and UKF, the position was estimated more accurately using the PF and SVSF. Since there was no measurement directly associated with the velocity, the performance of the four filters was found to be significantly worse when attempting to estimate the velocity. However, the filters remained relatively stable, with the EKF providing a slightly worse estimate for the velocity. Under poor initial conditions, as shown in Fig. 4, the SVSF converged towards the true position value faster than the other three filters. Notice how the EKF overshot the estimate whereas the SVSF did not. The SVSF appears to be more robust, mainly due to the inherent switching function shown in Eq. (29) that allows the estimate to stay within a close proximity of the true value. Once within an accurate range of the true value trajectory, the EKF, UKF, and SVSF yielded relatively the same performance as in the normal case. However, the initial estimation errors were too great for the PF to catch up and provide an accurate estimate. Note that the EKF and UKF were found to be more sensitive than the SVSF to the influence of the poor initial condition, as shown in the estimate of the target velocity.

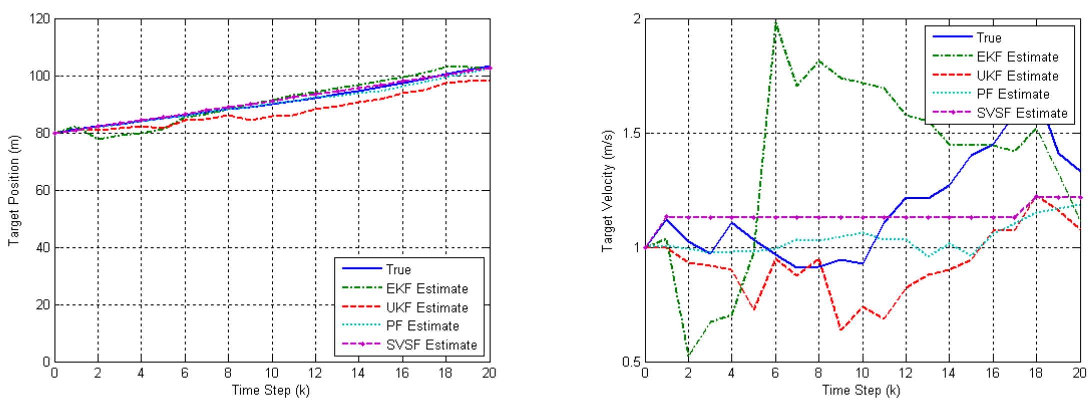

Figure 3. Position and Velocity Target Estimates, Respectively
(Under Normal Conditions)

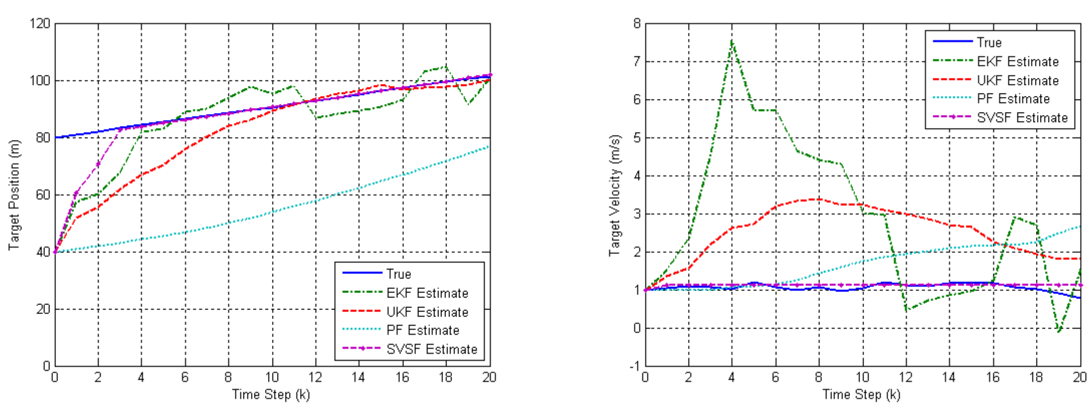

Figure 4. Position and Velocity Target Estimates, Respectively
(Under Poor Initial Conditions)

The PCRLB and root mean square (RMS) position and velocity errors for 10,000 Monte Carlo runs are shown in Figs. 5 and 6. For the normal conditions case, it was found that the PF and SVSF performed significantly better than the EKF and UKF, in terms of estimation error. For the RMS position error, the PF and SVSF yielded relatively the same results. However, when estimating the velocity, the SVSF initially had difficulty due to the lack of a velocity measurement. The velocity estimate had to be extracted from the estimated position, as described earlier by Eqs. (34) to (36). This estimation process was sensitive to error due to the relatively large sampling time. If a smaller sampling time was used, it is expected that the SVSF would yield a more accurate velocity estimate. The SVSF was able to overcome the lack of information after a few time steps, and provide a relatively stable estimate.

The main difference between the filters becomes apparent in the case of the poor initial conditions. The PF was unable to overcome the large initial estimate error; however the estimates were approaching the correct values and most likely would have reached them given enough time. The PF yields better results if one were to increase the number of particles from 5,000 to 20,000. However, this increases the computational time required for the estimation process. Since the PF was already running the slowest, it was not desirable to increase the number of particles. The EKF had difficulty with estimating the velocity but was able to yield a relatively stable estimate after about 10 to 12 time steps. The overall RMSE was significantly lower for the SVSF, thus suggesting that the SVSF is more robust to handling initial errors in the estimation process. As already mentioned, this is most likely due to the inherent switching found within the SVSF gain.

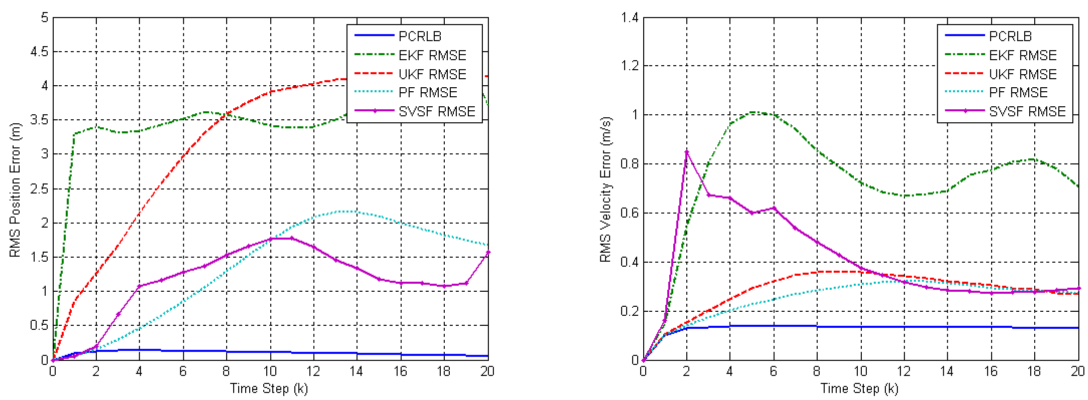

Figure 5. RMS Position and Velocity Errors, Respectively
(Under Normal Conditions)

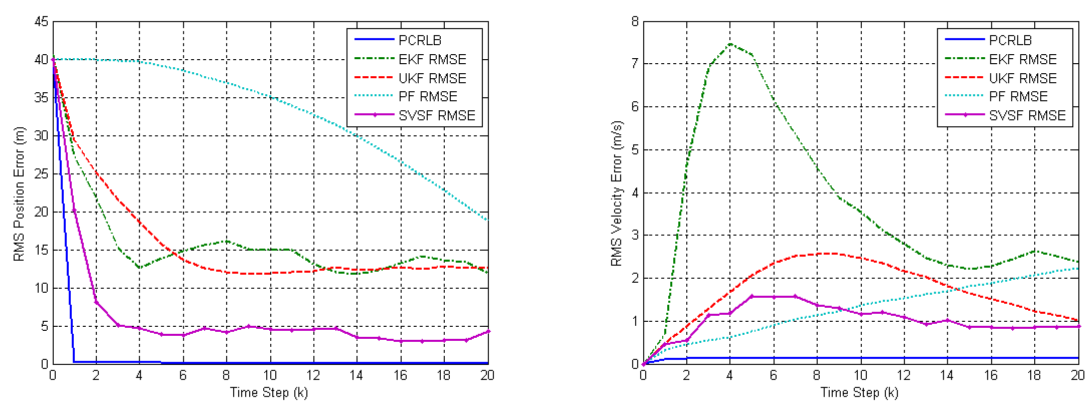

Figure 6. RMS Position and Velocity Errors, Respectively
(Under Poor Initial Conditions)

9. CONCLUSIONS

In this paper, we studied a nonlinear bearing-only target tracking problem using four different estimation strategies and compared their performances. This study was based on a classical ground surveillance problem, which appears to be deceptively simple. The nonlinear position measurement and lack of a velocity measurement create an interesting estimation problem. The performances of the popular EKF and UKF, the PF, and relatively new SVSF were compared. For the normal conditions case, the simulation shows that both the PF and SVSF performed relatively the same, beating out the EKF and UKF in terms of accuracy. In the poor initial conditions case, the SVSF yielded the most accurate and stable results suggesting that the SVSF is more robust to handling initial errors in the estimation process. In its current form, the SVSF offers two very important advantages: robustness to modeling errors and uncertainties, as well as estimation stability. The only main disadvantage of the SVSF at this point is the tuning process in determining the SVSF parameters. However, future forms of the SVSF will include methods such that tuning will not be necessary, which will further make this a powerful and useful filter for handling nonlinear estimation target tracking problems.

APPENDIX A: LIST OF NOMENCLATURE

C	Cramér-Rao lower bound (CRLB)	t	Simulation time
e	State estimation error	T	Sampling time
f, F	Nonlinear, linear system matrix	**v**	Measurement noise
h, H	Nonlinear, linear output matrix	**w**	System noise
J	Fisher information matrix (FIM)	**W**	Sample weight (UKF)
k	Time step index	**x**	System states
K	Gain value (EKF, UKF, or SVSF)	**X**	Sample point (UKF)
N_{eff}	Effective number of particles	z	Measurement output
P	Error covariance matrix	γ	Constant diagonal gain matrix with elements having values between 0 and 1
Q	System noise covariance matrix	ω	Particle weight
R	Measurement noise covariance matrix	π	Probability distribution
Sat	Saturation function		

Furthermore, note that subscript $k+1|k$ refers to an a priori time step and the subscript $k+1|k+1$ refers an a posteriori time step. A superscript of T denotes a matrix transpose.

APPENDIX B: UKF SUPPORTING EQUATIONS

In the UKF process, Eqs. (17) to (22) also include the following terms:

$$K_{k+1} = P_{xz} S_{k+1}^{-1} \tag{39}$$

$$S_{k+1} = R_{k+1} + P_{zz} \tag{40}$$

$$P_{xz} = \sum_{i=0}^{N-1} W_k^i [X_{k+1|k}^i - \hat{x}_{k+1|k}][h_{k+1}(X_{k+1|k}^i) - \hat{z}_{k+1|k}]^T \tag{41}$$

$$P_{zz} = \sum_{i=0}^{N-1} W_k^i [h_{k+1}(X_{k+1|k}^i) - \hat{z}_{k+1|k}][h_{k+1}(X_{k+1|k}^i) - \hat{z}_{k+1|k}]^T \tag{42}$$

REFERENCES

1. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, 2004.
2. X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, *Comparison of EKF, Pseudomeasurement Filter and Particle Filter for a Bearings-only Target Tracking Problem*, Proceedings of the SPIE Conference of Signal and Data Processing of Small Targets, Orlando, Florida, April 2002.
3. V. J. Aidala, *Kalman Filter Behavior in Bearing-Only Tracking Applications*, IEEE Transactions on Aerospace Electronic Systems, Vol. 15, January 1979.
4. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation, Tracking and Navigation: Algorithms and Software for Information Extraction*, Wiley, New York, 2001.
5. M. S. Grewal, and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB, 2nd Edition*, John Wiley and Sons, Inc., New York, 1993.
6. S. Habibi, R. Burton, and Y. Chinniah, *Estimation Using a New Variable Structure Filter*, Proceedings of the American Control Conference, Anchorage, May 2002.
7. D. Moral, *Measure Valued Processes and Interacting Particle Systems: Application to Non-Linear Filtering Problems*, Annals of Applied Probability, Vol. 8, No. 2, pp. 438 to 495, 1998.
8. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, *Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation*, IEEE Proceedings – F, Vol. 140, No. 2, pp. 107 to 113, 1993.
9. J. MacCormick, and A. Blake, *A Probabilistic Exclusion Principle for Tracking Multiple Objects*, Proceedings of International Conference on Computer Vision, pp. 572 to 578, 1999.
10. K. Kanazawa, D. Koller, and S. J. Russell, *Stochastic Simulation Algorithms for Dynamic Probabilistic Networks*, Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, pp. 346 to 351, 1995.
11. S. Habibi, and R. Burton, *The Variable Structure Filter*, Journal of Dynamic Systems, Measurement, and Control, ASME, Vol. 125, pp. 287 to 293, 2003.
12. S. Habibi, *The Smooth Variable Structure Filter*, Proceedings of the IEEE, Vol. 95, No. 5, pp. 1026 to 1059, 2007.
13. M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, *Multisensor Resource Management in the Presence of Clutter Using Cramér-Rao Lower Bounds*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 40, No. 2, April 2004.
14. X. Zhang and P. Willett, *Cramér-Rao Bounds for Discrete-Time Linear Filtering with Measurement Origin Uncertainties*, Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom, California, May 2001.
15. P. Tichavsky, *Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering*, IEEE Transactions on Signal Processing, Vol. 46, No. 5, May 1998.