AERO 626 Challenge Problem

Tim Woodbury

December 5, 2015

1 Filter algorithms

Five estimation algorithms are considered for the nonlinear sequential estimation problem: the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), the Sequential Importance Resampling (SIR) particle filter (PF), the Ensemble Kalman Filter (ENKF), and the Gaussian Mixture Model (GMM). For completeness, the algorithm governing each filter is described in the following sections.

1.1 Extended Kalman filter

The Extended Kalman filter is the most popular sequential estimation algorithm in the field of guidance, navigation, and control. Simply put, the EKF replaces the linear process and measurement influence matrices in the linear Kalman filter algorithm with the Jacobian of nonlinear governing functions evaluated at an aposteriori state estimate. The algorithm description is taken from Ref. [**?**]. The assumed prior estimate $\hat{x}(0)$, covariance $P(0)$, and dynamic system are of the following form:

$$\hat{x}(0) = E[x_0] \tag{1}$$

$$P(0) = E[(\hat{x}(0) - x_0)(\hat{x}(0) - x_0)^T] \tag{2}$$

$$x_k = f(x_{k-1}, t_{k-1}) + G_k w_k, w_k \sim N(0, Q_k) \tag{3}$$

$$y_k = h(x_k, t_k) + v_k, v_k \sim N(0, R_k) \tag{4}$$

The discrete propagation equations for the state and covariance follow:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, t_{k-1}) \tag{5}$$

$$P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \tag{6}$$

$$F_{k-1} \equiv \frac{\partial f}{\partial x}(\hat{x}_{k-1}, t_{k-1}) \tag{7}$$

The update equations for the state and covariance, given a measurement $\tilde{y}_k$, are:

$$\hat{x}_k = \hat{x}_k^- + K_k(\tilde{y}_k - h(\hat{x}_k^-)) \tag{8}$$

$$P_k = (I - K_k H_k)P_k^- \tag{9}$$

$$K_k \equiv P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{10}$$

$$H_k \equiv \frac{\partial h}{\partial x}(\hat{x}_k^-, t_k) \tag{11}$$

The EKF has been shown to work well for a variety of systems, given that the estimate PDF remains well approximated by a Gaussian. Many studies have also shown that the UKF generally performs better for nonlinear systems, at the cost of slightly greater computational complexity.

1.2 Unscented Kalman filter

The UKF is based on the same idea as the EKF; extend the linear Kalman update to nonlinear systems in an approximate fashion. In the UKF, however, function Jacobians are no longer used; instead, a discrete set of "sigma points" are defined and passed through the propagation and update equations. The output points are used to evaluate the system covariances from their definitions.

The algorithm here is repeated from Wan[7]; for a complete algorithmic description including the value of the weights and other algorithm parameters, the reader is encouraged to review this resource, or one of numerous others in the literature. Define the augmented state vector $x^a \in \mathcal{R}^L$ as follows:

$$x_k^a = \begin{bmatrix} x_k \\ w_k \\ v_k \end{bmatrix} \equiv \begin{bmatrix} x_k^x \\ x_k^w \\ x_k^v \end{bmatrix} \tag{12}$$

In Eq. 12, x_k^x, x_k^w, and x_k^v are used as shorthand for the augmented state members associated with the state, process noise, and measurement noise. The $L \times 2L{+}1$ array of sigma points is defined as follows. Note that λ is an algorithm parameter.

$$\chi_{k-1}^a = [\hat{x}_{k-1}^a \; \hat{x}_{k-1}^a \pm \sqrt{(L+\lambda)P_{k-1}^a}] \tag{13}$$

Each sigma point is then propagated through the governing nonlinear equation as follows:

$$\chi_k^-(:,j) = f(\chi_{k-1}^x(:,j), t_{k-1}) + G_k \chi_{k-1}^w(:,j), \; \forall \; j \tag{14}$$

The state and covariance apriori estimates are given as weighted averages of the sigma points:

$$\hat{x}_k^- = \sum_{i=1}^{2L+1} W_i^m \chi_k^{x-}(:,i) \tag{15}$$

$$P_k^- = \sum_{i=1}^{2L+1} W_i^c (\chi_k^{x-}(:,i) - \hat{x}_k^-)(\chi_k^{x-}(:,i) - \hat{x}_k^-)^T \tag{16}$$

In a similar fashion, the update is performed by first passing the propagated sigma points through the measurement equation:

$$Y_k(:,j) = h(\chi_k^{x-}(:,j), t_k) + \chi_{k-1}^v(:,j) \tag{17}$$

The measurement covariance and state-measurement cross-correlation are evaluated as follows:

$$\hat{y}_k \equiv \sum_{i=1}^{2L+1} W_i^m Y_k(:,i) \tag{18}$$

$$P_{yy} \equiv \sum_{i=1}^{2L+1} W_i^c (Y_k(:,i) - \hat{y}_k)(Y_k(:,i) - \hat{y}_k)^T \tag{19}$$

$$P_{xy} \equiv \sum_{i=1}^{2L+1} W_i^c (\chi_k^{x-}(:,i) - \hat{x}_k^-)(Y_k(:,i) - \hat{y}_k)^T \tag{20}$$

The Kalman update is then performed as follows:

$$K_k \equiv P_{xk} P_{yy}^{-1} \tag{21}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(\tilde{y}_k - \hat{y}_k) \tag{22}$$

$$P_k = P_k^- - K_k P_{yy} K_k^T \tag{23}$$

This completes the algorithm description.

1.3 Sequential importance resampling particle filter

The essential function of a particle filter is to propagate a set of candidate states forward in time, evaluate the likelihood of a given measurement for each point, and weight the points so that they form a discrete-valued approximation to the true system density function. Repeated applications of the likelihood function tend to reduce the weight associated with most particles to essentially zero, a phenomenon referred to as particle degeneracy. This phenomenon can be overcome in part by judicious application of resampling; sampling new particles from the aposteriori density function.

The SIR algorithm employed in this work is found in Ref. [2]. The density function is approximated by a set of weights ω_k^i and points x_k^i as follows:

$$p_k(x) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(x - x_k^i) \tag{24}$$

For the particular dynamic system under consideration, the SIR algorithm may be written as follows:

1. Draw process noise values as $w_{k-1}^i \sim N(0, Q_k)$, then propagate each particle according to:

$$x_k^i = f(x_{k-1}^{i+}, t_{k-1}) + G_k w_{k-1}^i \tag{25}$$

2. Update the weights according to:

$$v_k^i \equiv \tilde{y}_k - h(x_k^i, t_k) \tag{26}$$

$$\omega_k^i = p(v_k^i) \tag{27}$$

3. Normalize the weights to sum to one

4. Resample all points

The resampling algorithm is expanded below:

```
• Initialize the CDF as: $c_1 = 0$
• For $i = 2 : N_s$
    – Construct the CDF according to: $c_i = c_{i-1} + \omega_k^i$
• $i = 1$
• Draw a starting point: $u_1 \sim \mathcal{U}[0, N_s^{-1}]$
• For $j = 1 : N_s$
    – Move along the CDF: $u_j = u_1 + N_s^{-1}(j - 1)$
    – While $u_j > c_i$
        * $i = i + 1$
    – Assign sample: $x_k^{j+} = x_k^i$
    – Assign weight: $\omega_k^{j+} = N_s^{-1}$
```

The new particles x_k^{j+} are used in the propagation step of Eq. 25. The expectation and covariance at any time may be computed from the discrete PDF as follows:

$$E[x_k] \equiv \mu_k \approx \sum_{i=1}^{N_s} \omega_k^i x_k^i \tag{28}$$

$$E[(x_k - \mu_k)(x_k - \mu_k)^T] \approx \sum_{i=1}^{N_s} \omega_k^i (x_k^i - \mu_k)(x_k^i - \mu_k)^T \tag{29}$$

1.4 Ensemble Kalman Filter

The Ensemble Kalman Filter attempts to address two limitations associated with the EKF: (1) the need to store and propagate a covariance matrix in time; (2) the approximation of the nonlinear system by the linearized system [5]. The ENKF is a particle method; it relies on the propagation of a selection of points with random sampling to approximate the effects of process and measurement noise. Instead of using a weighted approximation to the PDF, the ENKF performs a Kalman update on each point in the ensemble, and the resulting set of points are treated in a Monte Carlo sense as approximating the true system PDF. The ENKF uses the same basic algorithm as the UKF, except the set of sigma points is replaced by an ensemble of points with associated random process and measurement noise.

Let the ensemble of points at time t_k be in the array $\gamma_k \in \mathcal{R}^{n \times N_s}$, where n is the dimension of the state and N_s is the ensemble size. An augmented ensemble with the process and measurement noise may be assembled as follows:

$$\gamma_k^a = \begin{bmatrix} \gamma_k & & & \\ w_k^{(1)} & w_k^{(2)} & \dots & w_k^{(N_s)} \\ v_k^{(1)} & v_k^{(2)} & \dots & v_k^{(N_s)} \end{bmatrix} \tag{30}$$

In Eq. 30, the process and measurement noise members are drawn from the underlying distributions (which are assumed zero-mean Gaussian here, but need not be in general). The UKF propagation and update equations of Eq. 14, 17-23 may then be used with one additional change. The mean weights W_i^m and covariance weights W_i^c are replaced by the constants used in the definitions of sample mean and covariance; $W_i^m \leftarrow \frac{1}{N_s}$, $W_i^c \leftarrow \frac{1}{N_s - 1}$.

1.5 Gaussian mixture models

The implementation of the Gaussian mixture model (GMM) is based on Ref. [3]. The algorithm is repeated here for completeness. The prior density function is approximated as a sum of multivariate Gaussians with normalizing weights $\alpha_{ki}^{'}$ as follows:

$$p(x_k|y_k) = \sum_{i=1}^{\chi_k} \alpha_{ki} \eta(x_k - a_{ki}, P_{ki}) \tag{31}$$

$$\eta(a, P) \equiv \frac{1}{\sqrt{(2\pi)^n \det(P)}} e^{-\frac{1}{2} a^T P^{-1} a} \tag{32}$$

$$\sum_{i=1}^{\chi_k} \alpha_{ki} = 1 \tag{33}$$

For a discrete-time dynamic system with state influence function $x_{k+1} = f(x_k) + G_k w_k, w_k \sim N(0, Q_k)$, the individual Gaussian filters are propagated using the Extended Kalman Filter update for the means a_{ki} and the covariances $P^{'}_{ki}$:

$$p(x_k|y_{k-1}) = \sum_{i=1}^{\chi^{'}_k} \alpha^{'}_{ki}\eta(x_k - a^{'}_{ki}, P^{'}_{ki}) \tag{34}$$

$$\chi^{'}_k = \chi_{k-1} \tag{35}$$

$$\alpha^{'}_{ki} = \alpha_{(k-1)i} \tag{36}$$

$$a^{'}_{ki} = f_k(a_{(k-1)i}) \tag{37}$$

$$P^{'}_{ki} = F_{ki}P_{(k-1)i}F^T_{ki} + G_kQ_kG^T_k \tag{38}$$

$$F_{ki} \equiv \frac{\partial f_k}{\partial x}(a_{(k-1)i}) \tag{39}$$

The update equations assume a nonlinear measurement of the form $y_k = h(x_k) + n_k, n_k \sim N(0, R_k)$. The aposteriori density function is approximated as follows:

$$p(x_k|y_k) = \sum_{i=1}^{\chi_k} \alpha_{ki}\eta(x_k - (a^{'}_{ki} + K_{ki}(y_k - h_k(a^{'}_{ki})), P_{ki}) \tag{40}$$

$$\chi_k \equiv \chi^{'}_k \tag{41}$$

$$P_{ki} \equiv P^{'}_{ki} - K_{ki}H_{ki}P^{'}_{ki} \tag{42}$$

$$K_{ki} \equiv P^{'}_{ki}H^T_{ki}(H_{ki}P^{'}_{ki}H^T_{ki} + R_k)^{-1} \tag{43}$$

$$\alpha_{ki} \equiv \frac{\alpha^{'}_{ki}\beta_{ki}}{\sum_{j=1}^{\chi_k} \alpha^{'}_{kj}\beta_{kj}} \tag{44}$$

$$\beta_{kj} \equiv \eta(y_k - h_k(a_{kj}), H_{kj}P^{'}_{kj}H^T_{kj} + R_k) \tag{45}$$

This completes the description of the basic algorithm. Initialization of the prior is not addressed in Ref. [3]. Two schemes are employed in the current work:

1. Initialization of the means a_{ki} to the sigma points used by the Unscented Kalman Filter

2. Monte Carlo selection of the initial means

In each case, the initial covariances P_{0i} are selected to satisfy a prior Gaussian proposal with mean $\mu(0)$ and covariance $P(0)$. The mean and covariance of the initial distribution are computed as follows:

$$\mu \equiv E(x_k) = \sum_{i=1}^{\chi_k} \alpha_{ki} a_{ki} \tag{46}$$

$$cov(x_k) = \sum_{i=1}^{\chi_k} \alpha_{ki}(P_{ki} + (a_{ki} - \mu)(a_{ki} - \mu)^T) \tag{47}$$

Eq. 47 represents an underconstrained system. By specifying that the initial distributions have equal covariance, the expression can be made tractable and solved for P_0:

$$P_0 = P(0) - \sum_{i=1}^{\chi_k} (\alpha_{ki}(a_{ki} - \mu)(a_{ki} - \mu)^T) \tag{48}$$

One practical note: astute readers will observe that the sum in Eq. 48 with equal weights matches the statistical definition of the covariance (to within a small bias in the weights). Depending on the number of initial points and their distribution, the value of P_0 may not satisfy the requirement of positive definiteness for the covariance matrix. To ensure positive definiteness, P_0 is processed additionally by taking the absolute value of all members, and setting the diagonal elements to zero, to ensure the determinant remains positive.

The initial weights are initially set equal to one another, $\alpha_{ki} = \chi_k^{-1}$. It should be observed that for a unimodal near-Gaussian system, the algorithm will collapse to the Extended Kalman filter after a few iterations, with the means all concentrated near a single point. (This assumes that the initial means are all within the region of convergence for the EKF algorithm.) In the dynamics considered in the following section, the GMM is intended to address the case of bimodality in the propagated density function. The GMM is of most interest as an alternative to the PF and ENKF for handling multi-modality of the density.

2 Governing equation and bifurcation time

The system dynamics are governed by Eq. 49, given below. The system consists of a cubic nonlinearity with white-noise and cosine forcing terms. For the purposes of estimation, the cosine forcing term is considered "unknown"; that is, the magnitude may be known but the exact form cannot be used in the system propagation equation. Therefore, in practice both the white-noise term and the cosine forcing terms are lumped into a single process noise term that must encapsulate both sources.

$$\ddot{x} = x - \epsilon x^3 + a_0 \cos \omega_t t + w(t) \tag{49}$$

$$w(t) \sim N(0, q) \tag{50}$$

2.1 System parameters and measurement model

The system measurement model is given as follows:

$$y_k = x_k + n_k, n_k \sim N(0, r) \tag{51}$$

System measurements occur at discrete times $t_k = kT$. The total set of parameters that govern system behavior are defined below:

- $\epsilon = 0.01$
- $a_0 = 0.5$
- $\omega_t = 1.25$
- $q = 1.0$
- $r = 1.0$
- $T = 0.1$

In addition, for numerical convenience in simulating the system dynamics, the process noise $w(t)$ is discretized at a fixed rate and held piecewise constant between samples. This effectively transforms as follows:

$$w(t) \to w_k \sim N(0, qT_s), t_k \leq t \leq t_{k+1} \tag{52}$$

The dynamic uncertainty discretization time is taken as $T_s = 0.001$, which is an order of magnitude smaller than any of the measurement sample times used later.

2.2 Bifurcation

One feature of interest for the system is its equilibria: the unforced system has roots at $x = 0, \pm\sqrt{\frac{1}{\epsilon}}$. The roots at $\pm\sqrt{\frac{1}{\epsilon}}$ are attractive; the root at the origin is repulsive. The net effect is that, when propagated forward for a long enough time, a probability density initially centered at the origin will bifurcate into two distributions, roughly centered around the attractive equilibria. It is of interest to determine the bifurcation time, in examining the effectiveness of various filters.

Bifurcation time is analyzed with no forcing ($a_0 = 0$) by performing simulating a number of points from a normal distribution about the origin, then propagating forward in time. Results are plotted, and the bifurcation time estimated qualitatively. Fig. 1 shows the development of the system bifurcation from its initial condition. The unforced motion is also shown by plotting the system after 9.5 seconds of development. From these plots, it is clear that bifurcation appears qualitatively on the order of a second or so. Therefore, the sample time of 10.0 seconds is used to consider the bifurcated system in later study.

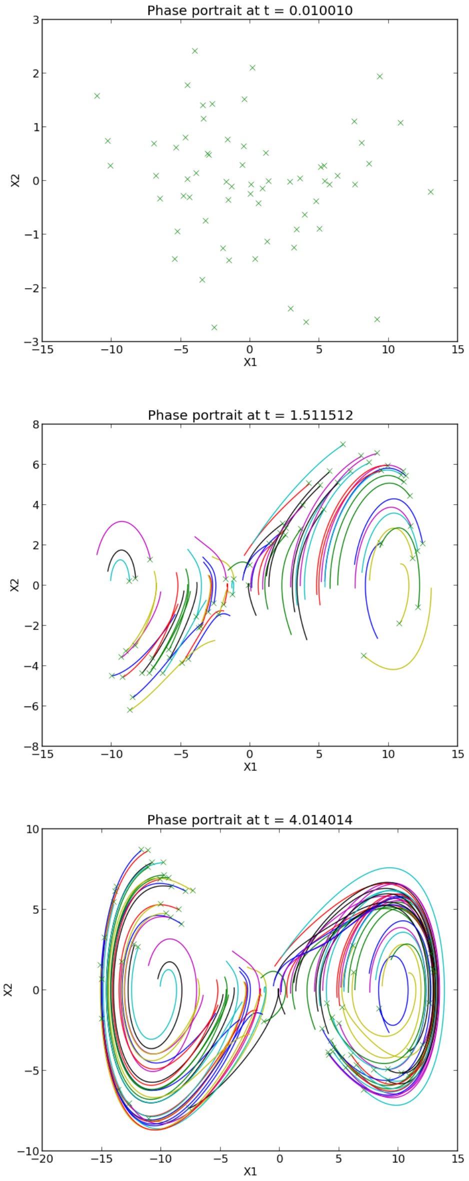

Figure 1: Figures showing the development of the phase portrait of the distribution. 'x' is used to indicate the current condition of the state with the contours tracing the path history of the phase portrait. Axis labels: $X1$ indicates the position, $X2$ the speed. Top: Initially Gaussian points with position variance 2.0 and speed variance 1.0. Middle: Condition of the sustem after 1.5 seconds; clear separation between points in the phase portrait is visible. Bottom: Development of the system after several seconds, showing separation of points.

It is clear that so long as position is measured, at least one state of the bimodal system can be estimated in a stable fashion. It is of greater interest to consider the case where measurements provide limited or ambiguous information about the states under consideration. Therefore, in the bimodality study, measurements of the form $y = x_1^2$ are considered. To exaggerate the effect of the bimodality on the system, the process and measurement noise are each reduced by a factor of 100; this ensures that the system estimates tend to cluster about two candidate solution states, rather than spreading over a domain unimodally.

2.3 Test for multimodality

The test for multimodality is Silverman's[6]; for implementation, reference is also made to [1]. The KDE is a method of nonparametric estimation of a density function. The KDE for a particular kernel $K(.)$ is defined for samples $\chi = \{X_1, X_2, ...X_n\}$ as follows:

$$\hat{f}(x; h) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{53}$$

In Eq. 53, h is a smoothing parameter. The kernel function in general may take different forms, but the Silverman test uses a univariate normal distribution. For a given data set χ, the number of modes in the data (determined by the number of local extrema) is monotonically decreasing with the value of the smoothing parameter, according to Silverman[6]. So, a simple binary search can be used to determine the critical smoothing parameter h_{crit} at which the KDE has a particular number of modes. The objective of the test is to if the distribution of the position and velocity states at a fixed time is unimodal; therefore, the smoothing parameter for which the KDE has only one extrema is sought.

One additional procedure is used in the test; smoothed bootstrap sampling. The test works by evaluating the significance of the null hypothesis that the underlying distribution is unimodal. If the significance is too small, the null hypothesis is rejected, and the distribution is assumed to be multimodal. Smoothed bootstrap sampling is used to evaluate the significance level. Bootstrap samples X_i^* are drawn from the original data set $\chi = \{X_1, X_2, ...X_n\}$ as follows:

$$X_i^* = \frac{X_{j(i)} + h\epsilon_i}{\sqrt{1 + h^2/\sigma^2}}, i \in [1, \dots, n] \tag{54}$$

In Eq. 54, the sample $X_{j(i)}$ is selected uniformly with replacement. σ is the sample standard deviation of the data. ϵ_i is a normally distributed random variable with uniform variance. The smoothed bootstrap samples are used to evaluate the null hypothesis that the distribution is unimodal with a critical KDE parameter h_{crit}; N_b sets are drawn using smoothed bootstrap sampling, and the number of maxima of the resulting KDE with h_{crit} are tabulated. The fraction of sets with only one maximum is taken to be the significance level of the null hypothesis.

The test procedure is summarized as follows:

1. Simulate N_p Monte Carlo simulations for a fixed time
2. For each time t^* in the simulation outputs:
    (a) Collect the N_p state values at $t = t^*$
    (b) Compute the critical smoothing parameter h_{crit} for which the KDE is unimodal
    (c) Evaluate the probability that the distribution has one mode
        i. Perform smoothed bootstrap sampling from the critical KDE N_b times
        ii. For each of the N_b samples:
            A. Evaluate the KDE with the critical smoothing parameter for the current sample. Record the number of modes.
        iii. The fraction of unimodal KDEs is taken as the P-value for the test.

3 Simulation

The estimation algorithms are evaluated in Monte Carlo simulation. Two categories of simulation are conducted: (1) a typical sequential filtering problem, referred to as the **standard case**, with relatively frequent measurements and approximately unimodal state distributions; (2) a case where significant bifurcation occurs due to the system dynamics, with infrequent, ambiguous measurements. The latter is referred to as the **bifurcation case**. All five algorithms are tested in the first scenario; in the second scenario, only the ENKF, SIR PF, and GMM are considered (that is, those models that have the ability to model non-unimodal distributions).

In the standard case, forcing functions of white noise only, cosine forcing only, and cosine forcing with white noise are considered. In addition, sample rates varying from 100 Hz to 1 Hz are considered. This broad sweep of parameters helps highlight the performance differences between the various filters.

In the bifurcation case, the process and measurement noise levels are reduced, and only the case of Gaussian white noise is considered. The sample period is increased to five seconds, allowing the system pdf more than enough time to bifurcate. The modified system parameters are listed below for completeness:

- $a_0 = 0.0$
- $q = 0.01$
- $r = 0.01$
- $T = 5.0$

4 Results

4.1 Standard case

For the standard case, results are broken into three sections, each corresponding to a different sample rate. The sample rates used are 0.1, 0.01, and 1.0 seconds, corresponding to medium, fast, and slow sample rates. For each sample rate, 100 Monte Carlo simulations of 30 seconds each are performed for three different cases of unmodelled forcing: (1) Gaussian white noise, (2) cosine forcing, (3) Gaussian white noise and cosine forcing.

For the SIR PF, 200 particles are used in each filter. For the ENKF, ensembles of size 100 are used in the non-adaptive case. Filter process noise terms were tuned for each filter and forcing case individually. It should be noted that the distinction is made between the **filter process noise** and the **simulation process noise**, the former being a tuning parameter for filter performance, and the latter being a global setting that changes the behavior of the underlying system.

Two performance metrics are used. The first is mean squared error (MSE), computed for the position states x_1 and speed states x_2 individually as follows:

$$\text{MSE}_i = \frac{1}{N}\sum_{j=1}^{N}(\hat{x}_j(i) - x_j(i))^2 \tag{55}$$

MSE is computed across all times and all Monte Carlo simulations as an average metric of accuracy. To evaluate the consistency of each filter, the normalized expectation of the error squared (NEES) is employed. NEES is computed at a given simulation time as follows:

$$\text{NEES}(t_k) = (\hat{x}_k - x_k)^T P_k^{-1}(\hat{x}_k - x_k) \tag{56}$$

NEES is employed in two ways: first, the average NEES across the Monte Carlo runs at each simulation time is plotted as a function of time; second, 95% confidence bounds for the NEES are computed as in Ref. [4]. The 95% confidence bounds are used to evaluate the fraction of times at which the mean NEES exceeded the 95% bounds (indicating overconfidence of the estimate) and fell below the 95% bounds (indicating conservatism of the estimate). This process yields two scalars that indicate the fraction of time steps at which a given filter was overconfident or overly conservative.

4.1.1 Medium sample rate

Performance for the 0.1 second sample period case are collected in Tables 1-3 and Figs. 2-4. It is noted first that the MSE accuracy of the "Kalman" filters, the EKF, UKF, and ENKF, is very similar in each case, typically differing only in the second decimal place. For the case of purely Gaussian forcing, the UKF performs best, followed by the ENKF and EKF; however, the actual difference in errors is probably too small to be significant in application. For the case of cosine forcing, the EKF has lowest errors, followed by the ENKF and UKF;

again, the difference in the three is quite small. With combined forcing, the EKF performs best, followed by the UKF and ENKF.

In each case, it should be noted that the EKF estimates are extremely conservative compared to the UKF and ENKF. Extensive tuning of the filter process noise term allowed the EKF to achieve essentially identical performance at this T_s, but produces conservative error bounds that may or may not be acceptable in practice. The UKF is much less conservative in the white noise forcing case. For the cosine forcing and mixed forcing cases, the UKF is conservative at some times and optimistic at others. This is a product of using a white-noise forcing model to capture the effects of a cosine forcing term. To achieve good error performance, the filter's process noise term must be set so that it is conservative when the cosine term is small, and optimistic when the cosine term is near its maximum. The ENKF suffers from the same limitation in using a white-noise forcing model to capture a cosine forcing term; however, the extra flexibility of the ENKF in capturing nonlinear effects limits the extrema of conservatism and optimism.

Performance for the SIR PF in terms of MSE is typically worse than that of the other filters. However, it is noted that the fraction of conservative and optimistic points for the SIR is typically lower than for the other filters, especially in the cosine forcing cases. That indicates something about the PF produces less precise, yet more consistent estimates. Perhaps the process noise and number of particles has a "shotgun" effect on the propagation stage; that is, the filter propagation produces a broad set of candidate states, which are more likely to have some particles in the correct states to capture the effects of the nonlinear forcing. The filter covariance settings used are considered to be conservatively large. However, at smaller filter covariances, the particle degeneracy problem became much more significant and often led to zero weights after a single propagation step. Improved accuracy might be achieved with the SIR with the conservative covariance settings and larger numbers of particles.

It is noted that the GMM tracks the EKF performance in each case, but with slightly larger errors. This is to be expected; since Monte Carlo initialization of the GMM is used with the standard EKF update for each Gaussian in the mixture, the GMM essentially converges to the EKF within a few updates. The GMM has extra covariance built into the initialization step, which causes the slightly larger overall MSEs.

Figs. 2-4 reflect the underlying differences between the different forcing environments. Performance for the tuned filters in any single forcing case is qualitatively similar; for instance, in Fig. 2, each filter starts at a large initial mean NEES, due to the large initial covariance, and then decreases within about 10 seconds toward an apparent steady-state final value, with some oscillation. The UKF history is little "tighter" than the EKF, reflecting the improved consistency of the UKF. In Figs. 3 and 4, on the other hand, the time histories are highly oscillatory, with a period reflecting that of the driving cosine forcing. These figures should make clear the limitation of using a white-noise forcing model to capture a nonlinear forcing term; the best-tuned filters have oscillate between conservative and optimistic.

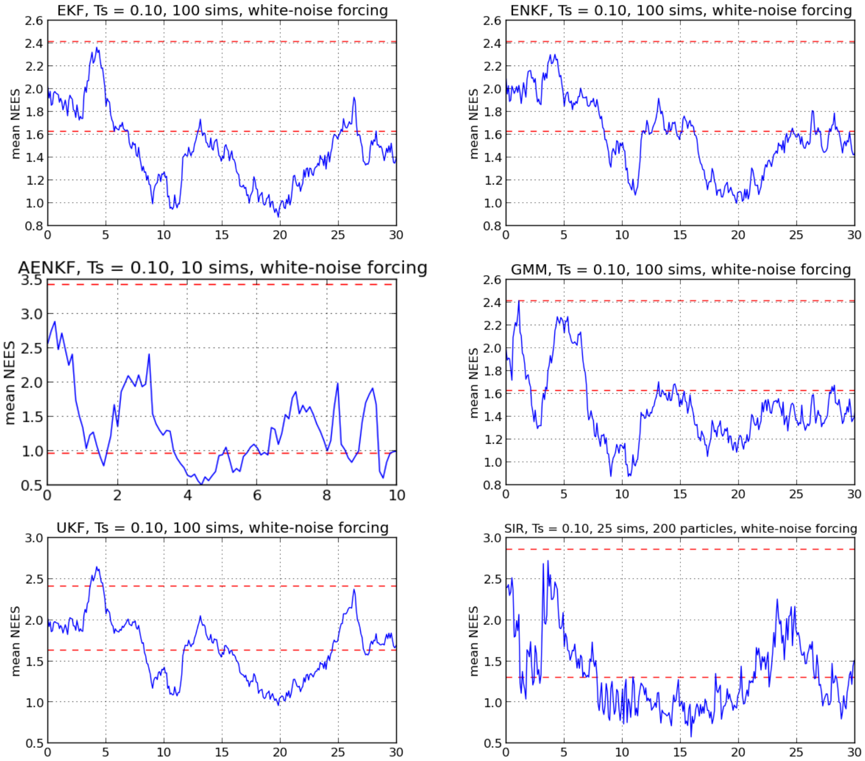

Figure 2: Mean NEES time histories for the pure white noise forcing cases.

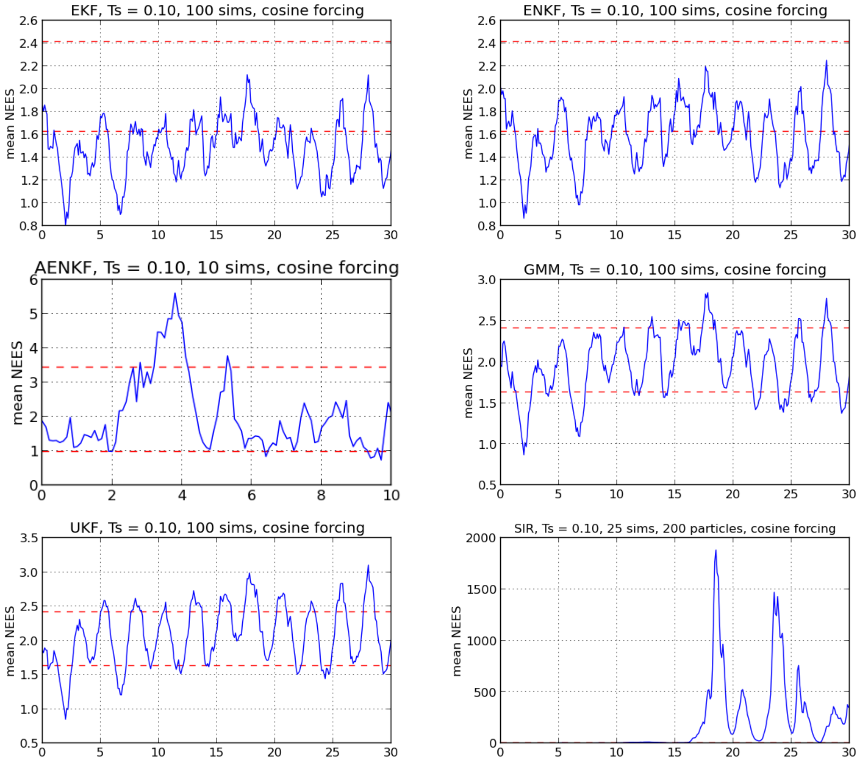

Figure 3: Mean NEES time histories for the pure cosine forcing cases.

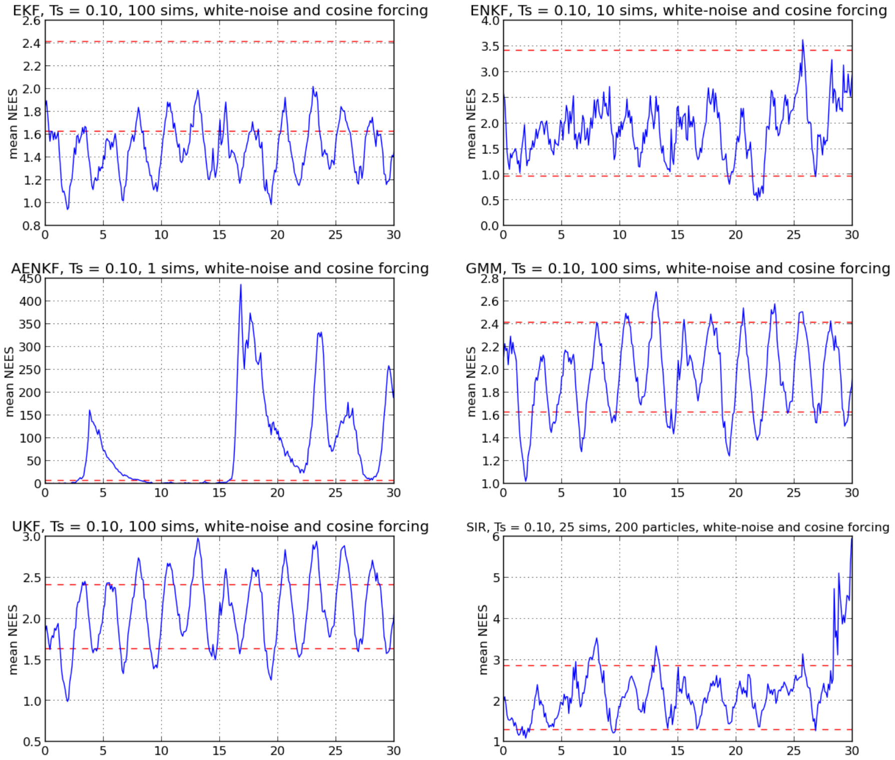

Figure 4: Mean NEES time histories for the pure cosine forcing cases.

Filter	MSE_1	MSE_2	Conservative fraction	Optimistic fraction
EKF	0.09895	0.1252	0.7043	0
ENKF	0.0946	0.1206	0.5548	0
AENKF	0.1488	0.2547	0.2871	0
GMM	0.1305	0.172	0.7841	0.003322
UKF	0.09088	0.1169	0.4385	0.03655
SIR	0.1634	0.2738	0.1584	0.1188

Table 1: Performance metrics comparison of all filters with Ts = 0.10 sec and Gaussian forcing term

Filter	MSE_1	MSE_2	Conservative fraction	Optimistic fraction
EKF	0.1444	0.2846	0.6977	0
ENKF	0.1455	0.2856	0.5249	0
AENKF	0.3373	1.936	0.0396	0.1188
GMM	0.1521	0.3213	0.206	0.113
UKF	0.1468	0.2898	0.1528	0.289
SIR	0.2694	1.171	0.0396	0

Table 2: Performance metrics comparison of all filters with Ts = 0.10 sec and cosine forcing term

Filter	MSE_1	MSE_2	Conservative fraction	Optimistic fraction
EKF	0.1381	0.277	0.7475	0
ENKF	0.1459	0.293	0.04319	0.003322
AENKF	4.459	13.75	0.01329	0.6545
GMM	0.1467	0.3121	0.2193	0.08638
UKF	0.1402	0.2813	0.1694	0.2691
SIR	0.1599	0.3126	0.04651	0.1196

Table 3: Performance metrics comparison of all filters with Ts = 0.10 sec and Gaussian and cosine forcing terms

4.1.2 Slow sample rate

With the sample period of $T_s = 1.0$ seconds, the filters begin to experience the effects of bifurcation and large covariance growth during the propagation step; however, the non-ambiguous measurements enable the filters to remain stable, and any bimodal effects are largely manageable.

For the case of purely Gaussian forcing, best performance in terms of MSE is obtained with the SIR, followed by the ENKF, EKF, GMM, and UKF. In terms of consistency, all the filters are quite conservative, with the GMM having probably the most consistency performance. This reflects the effect of the long propagation time and underlying system bimodality on these unimodal filters. Of note, the lowest MSE seems to correlate with the greatest degree of conservatism in the results.

For the case of pure cosine forcing, lowest MSE is found with the ENFK, followed by the EKF, GMM, UKF, then SIR. The ENKF is extremely conservative again; the EKF, UKF, and GMM also have notable conservative and optimistic fractions. As in the medium sample rate, the SIR has higher MSE but apparently better consistency compared to the other filters.

For the case of mixed Gaussian and cosine forcing, lowest MSE is achieved with the ENKF, followed by the EKF, GMM, SIR, and UKF. Of these, the EKF and GMM are most conservative. The ENKF is the most optimistic, despite its relatively low errors.

Filter	MSE_1	MSE_2	Conservative fraction	Optimistic fraction
EKF	0.6558	0.9599	0.3548	0.03226
ENKF	0.6215	0.822	0.4516	0
AENKF	0.6724	1.223	0.3636	0.1818
GMM	0.6822	1.042	0.1935	0.09677
UKF	0.7445	1.094	0.3871	0.06452
SIR	0.6023	0.7245	0.4516	0

Table 4: Performance metrics comparison of all filters with Ts = 1.00 sec and Gaussian forcing term

Filter	MSE_1	MSE_2	Conservative fraction	Optimistic fraction
EKF	0.5854	1.104	0.2903	0.03226
ENKF	0.5671	0.9629	0.7419	0
AENKF	0.7642	2.19	0	0.5455
GMM	0.6128	1.219	0.2258	0
UKF	0.6316	1.128	0.2903	0.1613
SIR	0.7376	2.383	0	0.09091

Table 5: Performance metrics comparison of all filters with Ts = 1.00 sec and cosine forcing term

Filter	MSE_1	MSE_2	Conservative fraction	Optimistic fraction
EKF	0.6076	1.077	0.2258	0.06452
ENKF	0.5489	0.8637	0.03226	0.3226
AENKF	0.6698	1.321	0.1613	0
GMM	0.6394	1.195	0.5484	0
UKF	0.661	1.116	0.1613	0.2258
SIR	0.6756	1.035	0.1613	0.03226

Table 6: Performance metrics comparison of all filters with Ts = 1.00 sec and Gaussian and cosine forcing terms

References

[1] M. Adereth. Silverman's mode estimation method explained. Online, October 2014. http://adereth.github.io/blog/2014/10/12/silvermans-mode-detection-method-explained/.

[2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[3] Daniel L. Aspach and Harold W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, AC-17(4), 1972.

[4] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3562–3568. IEEE, 2006.

[5] Geir Evensen. The ensemble kalman filter for combined state and parameter estimation. *Control Systems, IEEE*, 29(3):83–104, 2009.

[6] B.W. Silverman. Using kernel density estimates to investigate multimodality. *Journal of the royal statistical society*, 43(1):97–99, 1981. http://www.stat.washington.edu/wxs/Stat593-s03/Literature/silverman-81a.pdf.

[7] Eric Wan, Ronell Van Der Merwe, et al. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.