



Understanding Nonlinear Kalman Filters, Part I: Selection between EKF and UKF

Matthew Rhudy* and Yu Gu†

West Virginia University, Morgantown, WV 26506, USA

Abstract—Kalman filters provide an important technique for estimating the states of engineering systems. With several variations of nonlinear Kalman filters, there is a lack of guidelines for filter selection with respect to a specific research or engineering application. This creates a need for an in-depth discussion of the intricacies of different nonlinear Kalman filters. Particularly of interest for practical state estimation applications are the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). This tutorial is divided into three self-contained articles. Part I gives a general comparison of EKF and UKF, and offers a guide to the selection of a filter. Part II presents detailed information about the implementation of EKF and UKF, including equations, tips, and example codes. Part III examines different techniques for determining the assumed noise characteristics of the system as well as tuning procedures for these nonlinear Kalman filters.

Index Terms—Kalman Filters, Nonlinear Filters, Extended Kalman Filter, Unscented Kalman Filter

I. INTRODUCTION

The Kalman filter (Kalman, 1960) is a popular and effective state estimator for linear systems. It provides a way to estimate the state vector using an optimal observer gain to minimize the expected Mean Square Error (MSE) of the estimate. There are many different applications of the Kalman filter, and its statistical properties have been comprehensively studied (Anderson and Moore, 1979). Nonlinear systems, however, require approximations in order to handle the nonlinearity in the state dynamic equations and/or the output equations of the system. While there are many variations on the Kalman filter for nonlinear systems, two of the most common ones are the Extended Kalman Filter (EKF) (Kalman and Bucy, 1961) and the Unscented Kalman Filter (UKF) (Julier and Uhlmann, 1997). These two filters both operate within the existing Kalman filter framework, but use different approaches to handle the nonlinearity. The EKF uses an analytical linearization approach involving Jacobian matrices, while the UKF uses a statistical approach called the Unscented Transformation (UT) (Julier and Uhlmann, 1997).

II. SELECTION OF A NONLINEAR KALMAN FILTER

When starting to approach a nonlinear state estimation problem, the practical question becomes: which filter should I use, and why? There is no simple answer for this question. Several existing works claim superiority of the UKF over the EKF due to the more accurate linearization technique (Julier and Uhlmann, 1997; Orderud, 2005; Sadhu et al., 2006;

* Ph.D. Candidate, Department of Mechanical and Aerospace Engineering

† Assistant Professor, Department of Mechanical and Aerospace Engineering and Adjunct Assistant Professor, Lane Department of Computer Science and Electrical Engineering

Reviewers: Tanmay Mandal and Kyle Lassak

Citation: Rhudy, M., and Gu, Y., “Understanding Nonlinear Kalman Filters Part I: Selection of EKF or UKF,” *Interactive Robotics Letters*, West Virginia University, June 2013. Link: <http://www2.statler.wvu.edu/~irl/page13.html>

Copyright: © 2013 Matthew Rhudy and Yu Gu. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Kandepu et al., 2008). While this claim has been verified for certain example problems, the advantage is not always significant. Additionally there are other practical considerations to consider between the EKF and UKF other than just the accuracy of the filter. This article briefly explores both the technical and practical differences between the EKF and UKF, and offers some general guidelines regarding the selection of a nonlinear filtering technique.

A. Comparison of Estimation Accuracy

First, the accuracy of the two filters is considered. Some simulation studies have shown that the UKF consistently and significantly outperforms the EKF for certain problems such as bearing-only tracking (Orderud, 2005; Sadhu et al., 2006) and the Van der Pol oscillator (Kandepu et al., 2008). However, other researchers have found that the differences between the performance of the EKF and UKF are small (LaViola Jr., 2003; Saulson and Chang, 2003; Gross et al., 2012; Rhudy et al., 2013a). A thorough sensitivity analysis of both filters for the problem of GPS/INS attitude estimation revealed that two filters give a similar result for that particular problem, even as a function of various design parameters such as noise covariance tuning, sampling rate, and initialization error (Rhudy et al., 2013a). Other works state that the only noticeable performance difference between EKF and UKF occur under large initialization errors, which are handled better by the UKF (Crassidis, 2005; Fiorenzani et al., 2008; Wendel, et al., 2006). Basically, the differences between the EKF and UKF become more significant as the nonlinearity in the system increases. Simulation studies of highly nonlinear problems reveal the advantages of the UT over analytical linearization. Also, some problems such as the attitude estimation problem studied in (Gross et al., 2012; Rhudy et al., 2013a), while nonlinear in nature, do not contain a high enough level of nonlinearity in order to notice the advantages of the UKF. Additionally, for certain problems, a large initialization error will cause the linearization errors to become much more significant, thus explaining the differences in EKF and UKF initial error convergence. In general, depending on the application of interest, the UKF may provide improved performance for highly nonlinear problems; however for many applications that do not contain such strong nonlinearity, the EKF provides nearly identical estimation performance. In addition to the case studies discussed earlier, a thorough analytical comparison of the linearization techniques of these two filters for different nonlinear functions can be found in (Rhudy et al., 2013b).

B. Examples Comparing Linearization Errors of EKF and UKF

To help illustrate the differences in the EKF and UKF linearization techniques, a few examples are provided. Each example considers the nonlinear transformation of a 2D normally distributed random variable, \mathbf{x} , with covariance matrix, \mathbf{P} . Both the analytical (as in EKF) and statistical (as in UKF) linearization techniques are applied to a nonlinear function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ in order to recover the mean and covariance after the transformation. A Monte Carlo approach using 10^6 points is used to provide a ‘truth’ reference for the post transformation statistics, i.e. 10^6 points are generated from the distribution of \mathbf{x} , propagated through $\mathbf{f}(\mathbf{x})$, and then the mean and covariance are calculated based on the transformed points.

First, the following vector-valued nonlinear function is considered

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{f}(\mathbf{x}) = \mathbf{f} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \begin{bmatrix} x_1^2 + x_2^2 \\ x_1 x_2 \end{bmatrix} \quad (1)$$

Three different cases of prior distribution are considered, as summarized in Table 1.

Table 1. Different Cases of Prior Distribution

Case	Mean, \mathbf{x}	Covariance, \mathbf{P}
1	$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$	$\mathbf{P} = 10^{-3} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$
2	$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$	$\mathbf{P} = 10^{-2} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$
3	$\mathbf{x} = \begin{bmatrix} 0.1 \\ -1 \end{bmatrix}$	$\mathbf{P} = 10^{-2} \begin{bmatrix} 8 & 2 \\ 2 & 3 \end{bmatrix}$

For each of the cases described in Table 1, the prior distribution is shown on the left, and the post transformation statistics are shown on the right of Figure 1, Figure 2, and Figure 3. In these figures, the ellipses are used as a visual representation of the covariance, i.e. the ellipse is a $1-\sigma$ constant probability curve. Note that while the same nonlinear function is considered,

different effects of nonlinearity are observed by varying the prior statistics, i.e. starting from a different point along the nonlinear curve with different levels of uncertainty changes how much nonlinearity is experienced during the transformation. Case 1 shown in Figure 1 demonstrates nearly identical performance of EKF and UKF, both of which converge to the truth determined from Monte Carlo. Example MATLAB code for Case 1 is provided in the Appendix. Case 2 shown in Figure 2 demonstrates a slightly worse performance of the EKF relative to the UKF, which is able to closely match the Monte Carlo truth. Finally, Case 3 shown in Figure 3 offers a situation where the EKF is significantly less accurate in mean than the UKF, while the covariance estimates are similarly correct.

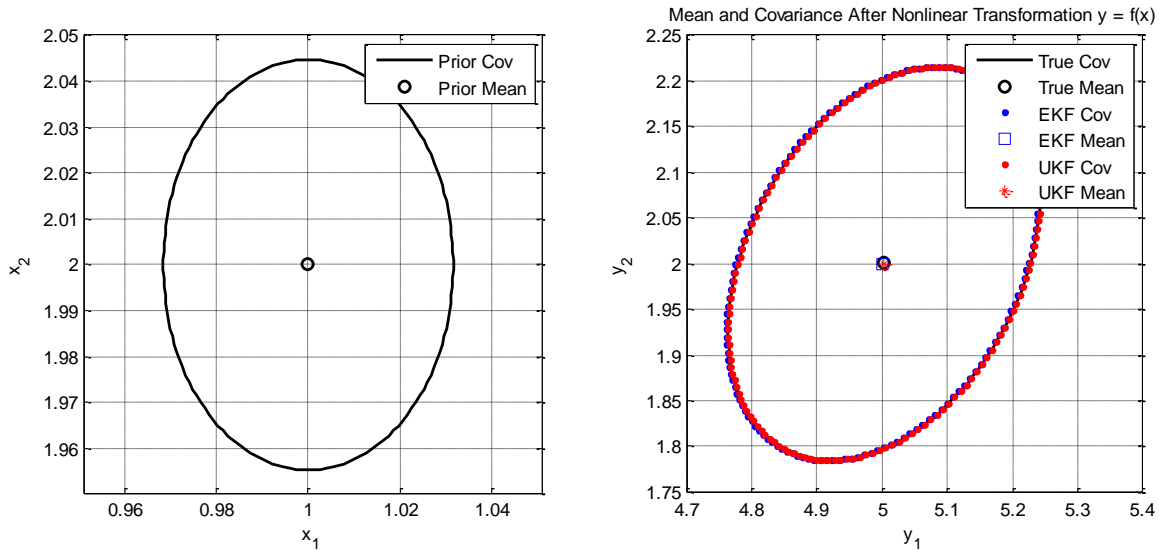


Figure 1. Example Linearization Comparison: Case 1

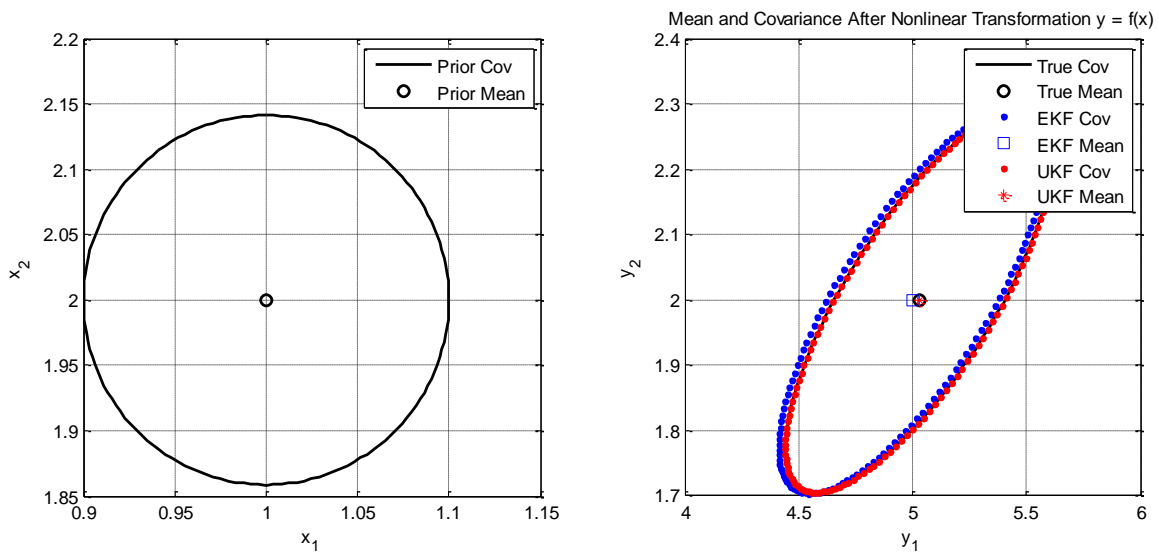


Figure 2. Example Linearization Comparison: Case 2

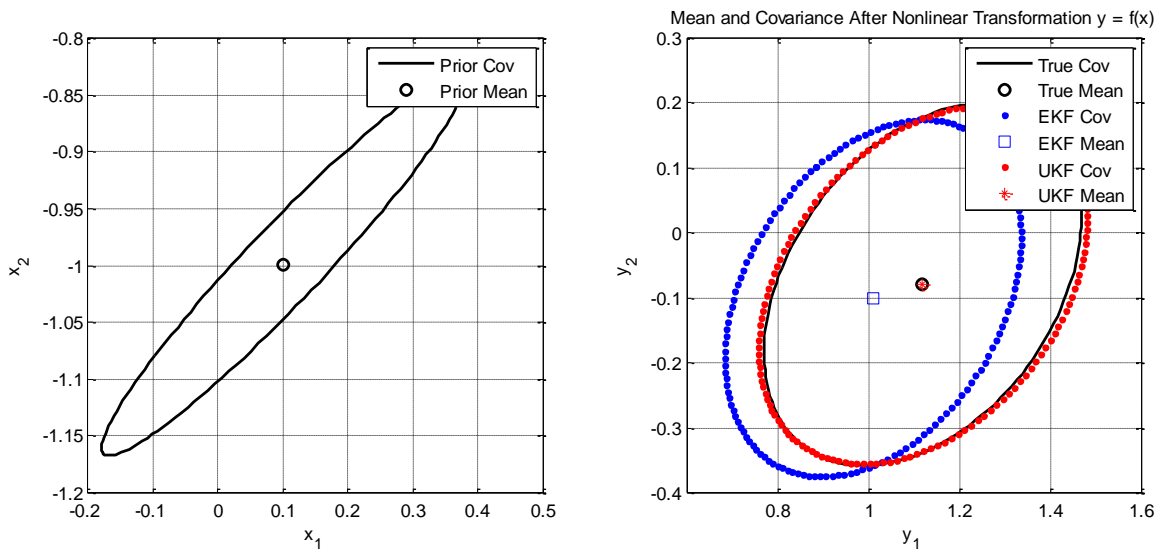


Figure 3. Example Linearization Comparison: Case 3

These three cases show how even for a particular problem, the linearization methods can vary in their results just by changing the prior conditions.

One additional example is also considered, which contains a highly nonlinear function

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{f}(\mathbf{x}) = \mathbf{f}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^4 x_2 \\ x_1 + x_2^5 \end{bmatrix} \quad (2)$$

For this example, the prior conditions are considered as

$$\mathbf{x} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \quad \mathbf{P} = 10^{-2} \begin{bmatrix} 8 & 2 \\ 2 & 3 \end{bmatrix} \quad (3)$$

Using these conditions, the prior and post transformation statistics are given in Figure 4. This example clearly demonstrates the superior performance of the UKF linearization technique over the EKF under strong nonlinearities. There are still some errors in the UKF covariance estimate, but the mean estimate is much more accurate than the EKF.

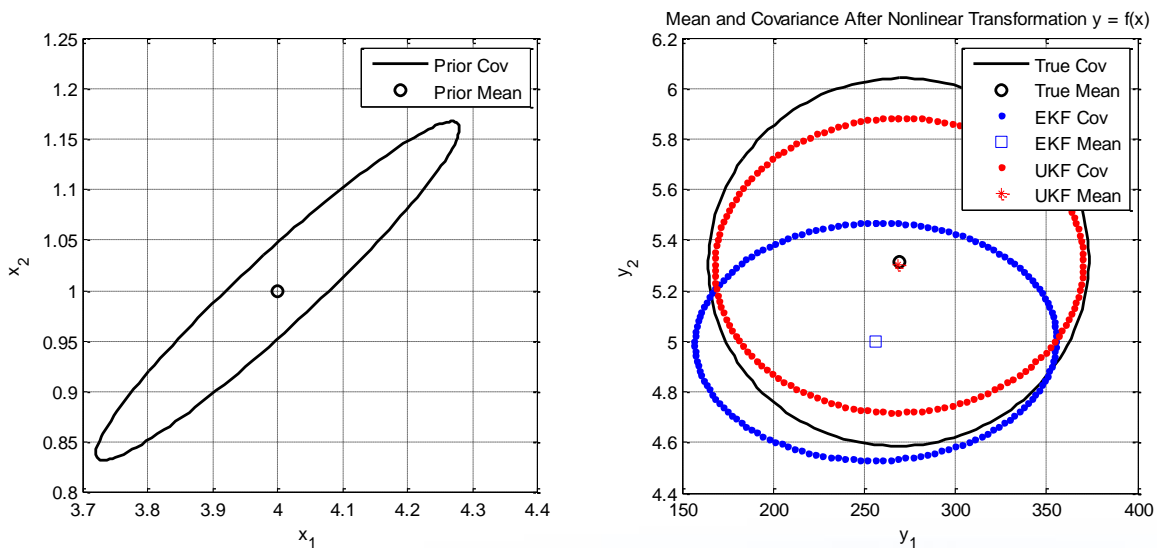


Figure 4. Example Linearization Comparison: Case 4

C. Comparison of Computational Requirements

Although estimation performance is an important consideration for nonlinear filters, there are other practical issues related to the filter selection. Perhaps one of the most important practical considerations is the computational requirements of the filter, especially for on-line applications. Based on both theoretical and empirical analyses of a particular application, the EKF was found to be significantly (approximately 10 times) more computationally efficient than the UKF (Gross et al., 2012; Rhudy et al., 2013a). The computational is application dependent, however, due to the dependence on the number of states and the nonlinear functions (and their derivatives for EKF) for each particular problem. The relative complexity of the nonlinear functions and their derivatives is also important, as the UKF requires the calculation of the nonlinear functions multiple times (proportional to the number of states), while the EKF requires this calculation only once, but in addition requires the calculation of derivatives to construct the Jacobian matrices. In general, the EKF seems to be more computationally efficient for most applications, as long as the EKF is using analytically determined Jacobian matrices instead of numerical derivatives.

D. Other Practical Issues

While the calculations of Jacobian matrices for the EKF may seem straightforward, in practice these calculations are prone to errors. Many mistakes in EKF implementations are due to mistakes in either the calculation or coding of the Jacobian matrices, which can be long and complex. The UKF does not require this calculation, and therefore is typically easier and faster to implement than an EKF. Another advantage of the UKF in this regard is that changes to the problem formulation can be made much more easily within the UKF, since Jacobian matrices do not need to be calculated. On the other hand, the EKF tends to be easier to understand conceptually than the UKF, which tends to make the UKF harder to implement initially, especially for students who are new to the state estimation process. These implementation issues are discussed in detail in Part II of this tutorial.

III. CONCLUSIONS

In conclusion, the EKF and UKF each serve a practical purpose. In fact, there are many situations where either filter could be used effectively, and the differences between the two might be negligible. One situation that merits the use of the UKF is highly nonlinear problems. Another benefit of the UKF is that it tends to be easier to implement and modify, which makes it a great tool for prototyping for a filtering application. The EKF, however, tends to be more computationally efficient, which increases its practicality for on-line and real-time applications. The EKF is also a bit easier to understand conceptually, so it can provide a better starter filter for students to learn about nonlinear state estimation.

ACKNOWLEDGEMENTS

This research was partially supported by NASA grant # NNX07AT53A and grant # NNX10AI14G. The authors would like to thank Dr. Jason Gross for his contribution to this work.

REFERENCES

- Anderson, B. D. O., and Moore, J. B., *Optimal Filtering*, Prentice-Hall, NJ, 1979.
- Crassidis J. L., "Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation," *AIAA Guidance, Navigation and Control Conference and Exhibit*, San Francisco, CA, 2005.
- Fiorenzani, T., et al., "Comparative Study of Unscented Kalman Filter and Extended Kalman Filter for Position/Attitude Estimation in Unmanned Aerial Vehicles," *IASI-CNR*, R. 08-08, 2008.
- Gross, J., Gu, Y., Rhudy, M., Gururajan, S., and Napolitano, M., "Flight Test Evaluation of GPS/INS Sensor Fusion Algorithms for Attitude Estimation," *IEEE Transactions on Aerospace Electronic Systems*, Vol. 48, No. 3, July 2012, pp. 2128-2139.
- Julier, S. and Uhlmann, J., "A New Extension of the Kalman Filter to Nonlinear Systems." *SPIE Proceedings Series*, 1997, Vol. 3068, pp. 182-193.
- Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Trans. of the ASME – Journal of Basic Engineering*, March 1960, pp. 35-45.
- Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory." *Journal of Basic Engineering (Transactions of ASME)*, Vol. 83, 1961, pp. 95-108.
- Kandepu, R., Foss, B., and Imsland, L., "Applying the unscented Kalman filter for nonlinear state estimation," *Journal of Process Control*, vol. 18, no. 7/8, Aug. 2008, pp. 753-768.
- LaViola Jr., J. J., "A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion," *Proc. of the American Control Conference*, Denver, CO, June, 2003, pp. 2435-2440.
- Orderud, F., "Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements," *Proc. of Scandinavian Conference on Simulation and Modeling*, 2005.

- Rhudy, M., Gross, J., Gu, Y., Gururajan, S., and Napolitano, M. R. (2013a), "Sensitivity and Robustness Analysis of EKF and UKF Design Parameters for GPS/INS Sensor Fusion," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 10, No. 3, March 2013, pp. 131-143.
- Rhudy, M., Gu, Y., and Napolitano, M. R. (2013b), "An Analytical Approach for Comparing Linearization Methods in EKF and UKF," *International Journal of Advanced Robotic Systems*, Vol. 10, No. 208, 2013.
- Sadhu, S., Mondal, S., Srinivasan, M., and Ghoshal, T. K., "Sigma point Kalman filter for bearing only tracking," *Signal Processing*, vol. 86, no. 12, April 2006, pp. 3769-3777.
- Saulson, B., and Chang, K. C., "Comparison of Nonlinear Estimation for Ballistic Missile Tracking," *Signal Processing, Sensor Fusion, and Target Recognition XII*, vol. 5096, pp. 13-24, 2003.
- Wendel, J., Metzger, J., Moenikes, R., Maier, A., and Trommer, G. F., "A Performance Comparison of Tightly Coupled GPS/INS Navigation Systems Based on Extended and Sigma Point Kalman Filters." *Journal of the Institute of Navigation*, Vol. 53, No. 1, 2006.

APPENDIX: EXAMPLE MATLAB CODE (CASE 1)

```

% Example code for linearization comparison
clear all

% Prior conditions
x = [1; 2];
P = 1e-3*[1 0; 0 2];

% EKF Approximations
ye = [x(1)^2 + x(2)^2; x(1)*x(2)];
A = [2*x(1) 2*x(2); x(2) x(1)];
Pe = A*P*A';

% UKF Approximations
alpha = 0.5; % Define primary scaling parameter
beta = 2; % Define secondary scaling parameter (Gaussian)
L = 2; % Number of states
lambda = L*(alpha^2-1);
eta = sqrt(L + lambda);
wm = zeros(2*L+1,1);
wc = zeros(2*L+1,1);
wm(1) = lambda/(L + lambda);
wc(1) = lambda/(L + lambda) + (1 - alpha^2 + beta);
for i = 2:length(wm)
    wm(i) = 1/(2*(L + lambda));
    wc(i) = 1/(2*(L + lambda));
end
sP = sqrtm(P);
chi = [x x*[1 1]+eta*sP x*[1 1]-eta*sP];
ysig = zeros(2,2*L+1);
for i = 1:(2*L+1)
    ysig(:,i) = [chi(1,i)^2 + chi(2,i)^2; chi(1,i)*chi(2,i)];
end
yu = ysig*wm;
Pu = zeros(2,2);
for i = 1:(2*L+1)
    Pu = Pu + wc(i)*(ysig(:,i)-yu)*(ysig(:,i)-yu)';
end

% Monte Carlo Simulation
nsim = 1e6;
R = chol(P);
z = repmat(x',nsim,1) + randn(nsim,2)*R;
z = z';
ysv = zeros(2,nsim);
for i = 1:nsim
    ysv(:,i) = [z(1,i)^2 + z(2,i)^2; z(1,i)*z(2,i)];
end
ys = mean(ysv,2);
Ps = cov(ysv');

% Create Ellipses based on mean and covariance
[Xe Ye] = ellip2D(ye,Pe,151);
[Xu Yu] = ellip2D(yu,Pu,151);
[Xs Ys] = ellip2D(ys,Ps,151);
[X0 Y0] = ellip2D(x,P,151);

% Plot Results
figure; subplot(121); plot(X0,Y0,'k-',x(1),x(2),'ko');
xlabel('x_1'); ylabel('x_2'); legend('Prior Cov','Prior Mean');
subplot(122);
plot(Xe,Ye,'b.',ye(1),ye(2),'bs'); hold on;
plot(Xu,Yu,'r.',yu(1),yu(2),'r*'); hold on;
plot(Xs,Ys,'k-',ys(1),ys(2),'ko'); hold off;
xlabel('y_1'); ylabel('y_2');
title('Mean and Covariance After Nonlinear Transformation y = f(x)');
legend('EKF Cov','EKF Mean','UKF Cov','UKF Mean','True Cov','True Mean');

```

```

% Function to calculate covariance ellipses
function [X Y] = ellip2D(x,P,steps)

% Calculate "standard deviation" matrix
P = sqrtm(P);

% Determine rotation angle of ellipse
T = 0.5*atan(-2*P(1,2)/(P(1,1)^2-P(2,2)^2));
% Pre-calculate trigonometric functions for ellipse rotation
sinbeta = sin(T);
cosbeta = cos(T);

% Calculate equatorial radii of ellipse
a = sqrt(P(1,1)^2*sin(T)^2+2*P(1,2)*sin(T)*cos(T)+P(2,2)^2*cos(T)^2);
b = sqrt(P(1,1)^2*sin(T+pi/2)^2+2*P(1,2)*sin(T+pi/2)*cos(T+pi/2)+P(2,2)^2*cos(T+pi/2)^2);
a = abs(a);
b = abs(b);

% Generate points along the ellipse
alpha = linspace(0, 360, steps)' .* (pi / 180);
% Pre-calculate trigonometric functions for ellipse points
sinalpha = sin(alpha);
cosalpha = cos(alpha);

% Calculate Cartesian (x,y) coordinates of ellipse points
X = x(1) + (a * cosalpha * sinbeta + b * sinalpha * cosbeta);
Y = x(2) + (a * cosalpha * cosbeta - b * sinalpha * sinbeta);

```


DOCUMENT HISTORY

Initial Manuscript Submission: June 07, 2013

Initial Review Completed (V1.0 Upload): June 28, 2013