Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements

Fredrik Orderud
Sem Sælands vei 7-9, NO-7491 Trondheim

Abstract

The Extended Kalman Filter (EKF) has long been the de-facto standard for nonlinear state space estimation [11], primarily due to its simplicity, robustness and suitability for realtime implementations. However, an alternative approach has emerged over the last few years, namely the unscented Kalman filter (UKF). This filter claims both higher accuracy and robustness for nonlinear models. Several papers have investigated the accuracy of UKF for nonlinear process models, but none has addresses the accuracy for nonlinear measurement models in particular. This paper claims to bridge this gap by comparing the performance of EKF to UKF for two tracking models having nonlinear measurements.

1 Introduction

The problem of *state estimation* concerns the task of estimating the state of a process while only having access to noisy and/or inaccurate measurements from that process. It is a very ubiquitous problem setting, encountered in almost every discipline within science and engineering.

The most commonly used type of state estimator is the Kalman filter. It is an optimal estimator for linear systems, but unfortunately very few systems in real world are linear. A common approach to overcome this problem is to linearize the system first before using the Kalman filter, resulting in an extended Kalman filter. This linearization does however pose some problems, e.g. it can result in nonstable estimates [8]. The development of better estimator algorithms for nonlinear systems has therefore attracted a great deal of interest in the scientific community, because improvements here will undoubtedly have great impact in a wide range of engineering fields.

Notation

Algebraic letters are distinguished based on their appearance: Normal (a) denotes scalars, bold ($\mathbf{a}$) denotes vectors and uppercase (A) denotes matrices. Subscripts (x_a) denote discrete time. Conditional subscripts ($x_{a|b}$) denote state x in time a, given measurements up to time b. A 'hat' superscript ($\hat{a}$) denotes an estimated value, known only with a certain belief.

2 Background

A state space model is a mathematical model of a process, where the process' *state* $\mathbf{x}$ is represented by a numerical vector. State-space models actually consists of two separate models: the *process model*, which describes how the state propagates in time based on external influences, such as input and noise; and the *measurement model*, which describe how measurements $\mathbf{z}$ are taken from the process, typically simulating noisy and/or inaccurate measurements.

2.1 General State Space Model

The most general form of state-space models is the nonlinear model.
This model does typically consist of two functions, f and h:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$$
$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k)$$

which govern state propagation and measurements, respectively. $\mathbf{u}$ is process input, and $\mathbf{w}$ and $\mathbf{v}$ are state and measurement noise vectors, respectively while k is the discrete time.
State-space models are remarkably usable for modelling almost all sorts of processes. f and h are usually based upon a set of discretized differential equations, governing the dynamics of and observations from the process.

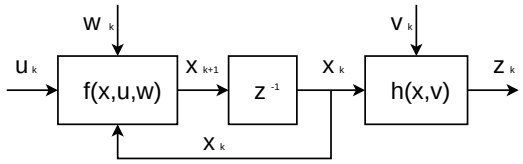

Figure 1: *A general state-space model. z^{-1} is the unit delay function known from the Z-transform in digital signal processing.*

2.2 Linear State Space Model

A linear state-space model is a model where the functions f and h are linear in both state and input. The functions can then be expressed by using the matrices F, B and H, reducing state propagation calculations to linear algebra. Overall this results in the following state-space model:

$$\mathbf{x}_{k+1} = F_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathbf{w}_k$$
$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k$$

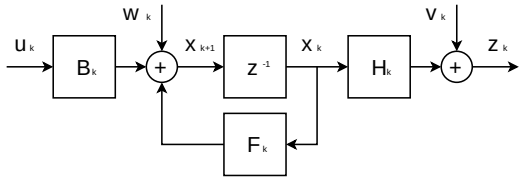

Figure 2: *A linear state-space model*

This linear model is easier both to calculate and analyze. Enabling modellers to investigate properties such as controllability, observability and frequency response [3].
Linear state models are either based on inherently linear processes, or simply a linearized versions of a nonlinear process by means of a first order Taylor approximation.

3 State Estimation

State estimation concerns the problem of estimating the probability density function (*pdf*) for the state of a process which is not directly observable. This involves both predicting the next state based on the current, and updating/correcting this prediction based on noisy measurements taken.

3.1 Recursive Bayesian Estimation

The most general form for state estimation is known as recursive Bayesian estimation [1]. This is the optimal way of predicting a state pdf for any process, given a system and a measurement model. In this section we will discuss this estimator, which recursively calculates a new estimate for each time-step, based on the estimate for the previous timestep and new measurements.
Recursive Bayesian estimation works by simulating the process, while at the same time adjusting it to account for new measurements $\mathbf{z}$, taken from the real process. The calculations are performed recursively in a two step procedure. First, the next state is predicted, by extrapolating the current state onto next time step using state propagation belief $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ obtained from function f. Secondly, this prediction is corrected using measurement likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ obtained from function h, taking new measurements into account.
The Chapman-Kolmogorov equation is first used to calculate a prior pdf for state $\mathbf{x}_k$, based on measurements up to time $k-1$:

$$p(\mathbf{x}_k|\mathbf{z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})\,d\mathbf{x}_{k-1}$$

Bayes rule is then used to calculate the updated pdf for state $\mathbf{x}_k$, after taking measurements up to time k into account:

$$p(\mathbf{x}_k|z_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{k-1})\,d\mathbf{x}_k}$$

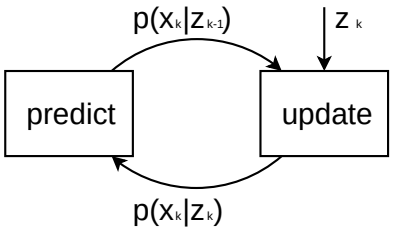

Figure 3: *Recursive Bayesian estimator loop*

Unfortunately, this method does not scale very well in practice, mainly due to the large state space for multidimensional state vectors. Calculating the prior probability of each point in this state space involves a multidimensional integral, which quickly becomes intractable as the state space grows. Computers are also limited to calculation of the pdf in discrete point in state space, requiring a discretization of the state space. This technique is therefore mainly considered as a theoretic foundation for state estimation in general. Bayesian estimation by means of computers is only possible if either the state space can be discretized, or if certain limitations apply for the model.

3.2 Kalman Filter

The problem of state estimation can be made tractable if we put certain constrains on the process model, by requiring both f and h to be linear functions, and the noise terms **w** and **v** to be uncorrelated, Gaussian and white with zero mean. Put in mathematical notation, we then have the following constraints:

$$\begin{aligned} f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) &= F_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathbf{w}_k \\ h(\mathbf{x}_k, \mathbf{v}_k) &= H\mathbf{x}_k + \mathbf{v}_k \\ \mathbf{w}_k \sim N(0, Q_k) \quad & \mathbf{v}_k \sim N(0, R_k) \\ E(\mathbf{w}_i\mathbf{w}_j^T) = Q_i\delta(i-j) \quad & E(\mathbf{v}_i\mathbf{v}_j^T) = R_i\delta(i-j) \\ E(\mathbf{w}_k\mathbf{v}_k^T) &= 0 \end{aligned}$$

where Q and R are covariance matrices, describing the second-order properties of the state- and measurement noise. The constraints described above reduces the state model to:

$$\begin{aligned} \mathbf{x}_{k+1} &= F_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k &= H_k\mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

where F, B and H are matrices, possible time dependent.

As the model is linear and input is Gaussian, we know that the state and output will also be Gaussian [12]. The state and output pdf will therefore always be normally distributed, where mean and covariance are sufficient statistics. This implies that it is not necessary to calculate a full state pdf any more, a mean vector $\hat{\mathbf{x}}$ and covariance matrix P for the state will suffice.

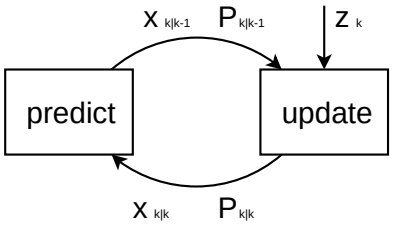

Figure 4: *Kalman filter loop*

The recursive Bayesian estimation technique is then reduced to the *Kalman filter*, where f and h is replaced by the matrices F, B and H. The Kalman filter is, just as the Bayesian estimator, decomposed into two steps: predict and update. The actual calculations required are:

Predict next state, before measurements are taken:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= F_k\hat{\mathbf{x}}_{k-1|k-1} + B_k\mathbf{u}_k \\ P_{k|k-1} &= F_kP_{k-1|k-1}F_k^T + Q_k \end{aligned}$$

Update state, after measurements are taken:

$$\begin{aligned} K_k &= P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - H_k\hat{\mathbf{x}}_{k|k-1}) \\ P_{k|k} &= (I - K_kH_k)P_{k|k-1} \end{aligned}$$

where K is the Kalman gain matrix, used in the update observer, and P is the covariance matrix for the state estimate, containing information about the accuracy of the estimate. More details and background for this filter can be found in [2].

The Kalman filter is quite easy to calculate, due to the fact that it is mostly linear, except for a matrix inversion. It can also be proved that the Kalman filter is an optimal estimator of process state, given a quadratic error metric [2].

3.3 Extended Kalman Filter

Most processes in real life are unfortunately not linear, and therefore needs to be linearized before they can be estimated by means of a Kalman filter. The extended Kalman filter (EKF) solves this problem by calculating the Jacobian[1] of f and h around the estimated state, which in turn yields a trajectory of the model function centered around this state.

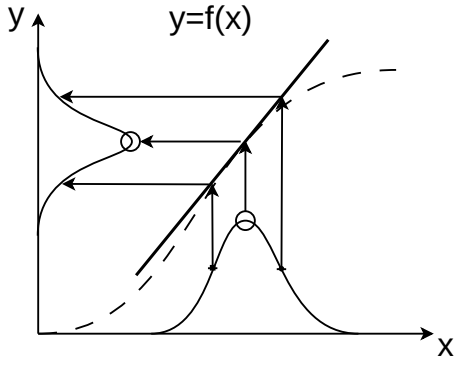

Figure 5: *Illustration of how the Extended Kalman filter linearizes a nonlinear function around the mean of a Gaussian distribution, and thereafter propagates the mean and covariance through this linearized model*

$$\begin{aligned} F_k &= \left.\frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, 0} \\ H_k &= \left.\frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}, 0} \end{aligned}$$

[1] The Jacobian is the matrix of all partial derivatives of a vector

The extended Kalman filter works almost like a regular Kalman filter, except for F and H, which vary in time based on the estimated state $\hat{\mathbf{x}}$. The actual calculations required are:
Predict next state, before measurements are taken:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k, 0)$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Update state, after measurements are taken:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}, 0))$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

where K is the Kalman gain matrix, used in the update observer, and P is the covariance matrix for the state estimate, containing information about the accuracy of the estimate.

3.4 Unscented Kalman Filter

3.4.1 Introduction

The problem of propagating Gaussian random variables through a nonlinear function can also be approached using another technique, namely the *unscented transform*. Instead of linearizing the functions, this transform uses a set of points, and propagates them through the actual nonlinear function, eliminating linearization altogether. The points are chosed such that their mean, covariance, and possibly also higher order moments, match the Gaussian random variable. Mean and covariance can be recalculated from the propagated points, yielding more accurate results compared to ordinary function linearizaton. The underlying idea is also to approximate the probability distribution instead of the function. This strategy typically does both decrease the computational complexity, while at the same time increasing estimate accuracy, yielding faster, more accurate results.

3.4.2 Background

The underlying method of unscented transform was first proposed by Uhlmann et al. in [11] and [10], where they laid out the framework for representing a Gaussian random variable in N dimensions using $2N+1$ samples, called *sigma points*. They utilized the matrix square root and covariance definitions to select these points in such a way that they had the same covariance as the Gaussian they approximated. Skewness was avoided by selecting the points in a symmetric way, such that any approximation error would only originate from the fourth and higher moments.
Usage of the unscented transform in Kalman filtering was then presented by Julier in [8], where he introduced the *Unscented Kalman filter* (UKF), which approximates the state estimate using sigma points. Later, it was analyzed more in depth in [16].
A limitation associated with the unscented Kalman filter is that it has a lower bound on the *safe spread* of the sigma points, meaning the distance between the points in state space. Sigma point spreads below this bond are not guaranteed to yield positive semidefinite correlation matrices. This distance also increases with the dimension of the state space, a limitation that may cause problems in highly nonlinear models, since high sigma point spread may result in sampling of non-local features.
The technique presented here is therefore based on the *scaled unscented transform* [6], which provides an additional tuning parmeter, α, compared to the original unscented transform. This parameter is used to arbitrary control the spread of the sigma-points, while at the same time guaranteeing positive semidefinite covariance matrices. Even models of high dimensonality can then keep a tight sigma point spread to avoid nonlocal effects.

3.4.3 Augmented state

The unscented transform approach also has another advantage, namely that noise can be treated in a nonlinear fashion to account for non-Gaussian or non-additive noises. The strategy for doing so involves propagation of noise through the functions by first augmenting the state vector to also include noise sources, a technique first introduced by Julier in [7], and later refined more in depth by Merwe in [16]. Sigma point samples are then selected from the augmented state, $\mathbf{x}^a$, which also includes noise values. The net result is that any nonlinear effects of process and measurement noise are captured with the same accuracy as the rest of the state, which in turn increases accuracy for non-additive noise sources.

3.4.4 Filter Formulation

The filter starts by augmenting the state vector to L dimensions, where L is the sum of dimensions in the original state-vector, model noise and measurement noise. The covariance matrix is similarly augmented

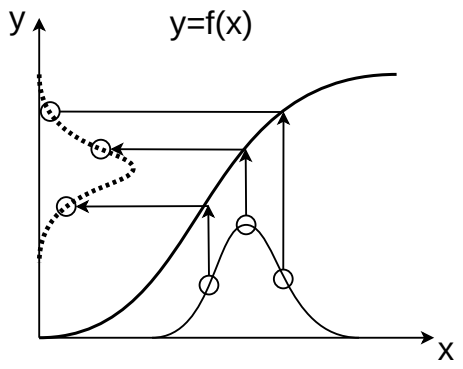

Figure 6: *Illustration of how the unscented Kalman filter propagates sigma-points from a Gaussian distribution through a nonlinear function, and recreates a Gaussian distribution, by calculating the mean and covariance of the results*

to a L^2 matrix. Together this forms the augmented state estimate vector $\mathbf{x}^a$ and covariance matrix P^a:

$$\mathbf{x}^a_{k-1} = \begin{bmatrix} \mathbf{x}_{k-1} \\ 0_w \\ 0_v \end{bmatrix}$$

$$P^a_{k-1} = E\{(\mathbf{x}^a_{k-1} - \hat{\mathbf{x}}^a_{k-1})(\mathbf{x}^a_{k-1} - \hat{\mathbf{x}}^a_{k-1})^T\} = \begin{bmatrix} P_{k-1} & 0 & 0 \\ 0 & Q_{k-1} & 0 \\ 0 & 0 & R_{k-1} \end{bmatrix}$$

The next step consists of creating $2L+1$ sigma-points in such a way that they together captures the full mean and covariance of the augmented state. The χ^a matrix is chosen to contain these points, and its columns are calculated as follows:

$$\begin{aligned} \chi^a_{0,k-1} &= \mathbf{x}^a_{k-1} && i = 0 \\ \chi^a_{i,k-1} &= \mathbf{x}^a_{k-1} + (\alpha\sqrt{LP^a_{k-1}})_i, && i = 1...,L \\ \chi^a_{i,k-1} &= \mathbf{x}^a_{k-1} - (\alpha\sqrt{LP^a_{k-1}})_{i-L}, && i = L+1...,2L \end{aligned}$$

where subscript i means the i-th column of the square root of the covariance matrix[2]. The α parameter, in the interval $0 < \alpha \leq 1$, determines sigma-point spread. This parameter is typically quite low, often around 0.001, to avoid non-local effects.

The resulting χ^a_{k-1} matrix can now be decomposed vertically into the χ^x_{k-1} rows, which contains the state; the χ^w_{k-1} rows, which contains sampled process noise and the χ^v_{k-1} rows, which contains sampled measurement noise.

Each sigma-point is also assigned a weight. These weight are derived by comparing the moments of the sigma-points with a Taylor series expansion of the models while assuming a Gaussian distribution, as derived in [9]. The resulting weights for mean (m) and covariance (c) estimates then becomes:

$$\begin{aligned} w^{(m)}_0 &= 1 - \frac{1}{\alpha^2} && i = 0 \\ w^{(c)}_0 &= 4 - \frac{1}{\alpha^2} - \alpha^2 && i = 0 \\ w^{(m)}_i &= w^{(c)}_i = \frac{1}{2\alpha^2 L} && i = 1...,2L \end{aligned}$$

The filter then predicts next state by propagating the sigma-points through the state and measurement models, and then calculating weighted averages and covariance matrices of the results:

$$\begin{aligned} \chi^x_{k|k-1} &= f(\chi^x_{k-1}, \mathbf{u}_k, \chi^w_{k-1}) \\ \hat{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{2L} w^{(m)}_i \chi^x_{k|k-1} \\ P_{k|k-1} &= \sum_{i=0}^{2L} w^{(c)}_i [\chi^x_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}][\chi^x_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}]^T \\ Z_{k|k-1} &= h(\chi^x_{k|k-1}, \chi^v_{k-1}) \\ \hat{\mathbf{z}}_{k|k-1} &= \sum_{i=0}^{2L} w^{(m)}_i Z_{i,k|k-1} \end{aligned}$$

The predictions are then updated with new measurements by first calculating the measurement covariance and state-measurement cross correlation matrices, which are then used to determine the Kalman gain:

$$\begin{aligned} P_{zz} &= \sum_{i=0}^{2L} w^{(c)}_i [Z_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}][Z_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}]^T \\ P_{xz} &= \sum_{i=0}^{2L} w^{(c)}_i [\chi^x_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}][Z_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}]^T \\ K_k &= P_{xz} P^{-1}_{zz} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \\ P_{k|k} &= P_{k|k-1} - K_k P_{yy} K^T_k \end{aligned}$$

Experimental results indicate [16] that Unscented Kalman filters yield results comparable to a third order Taylor series expansion of the state-model, while Extended Kalman filters of course only are accurate to a first order linearization. Consult [14] for a comparison of accuracy between the two kinds of filters.

[2]The square root of a symmetric matrix is typically calculated by means of a lower triangular Cholesky decomposition. The square root A of matrix P is then on the form $P = AA^T$.

The most computationally demanding part of the Unscented Kalman filter is the matrix square-root used to calculate sigma points. Matrix diagonalization or Cholesky factorization of the covariance matrix can be used to solve this problem, but a more direct square root approach, propagating only the square-roots of the covariance matrices, offers higher computationally efficiency. Merwe et al. proposes a approach for doing this in [15].

3.5 Other Approaches

Central difference Kalman filter (CDKF) proposed in [4] offers an alternative Kalman formulation for propagating Gaussian *pdf*s through nonlinear functions. This formulation, although it is different, remains very much like UKF and is reported to perform indistinguishable from UKF in all tests performed [13]. The CDKF is therefore not addressed by this paper.
Particle filters [1]: While the various types of Kalman filters often offers superb estimation accuracy, there are situations where they are not suited for the task. This problem relates to the fact that all Kalman filters are constrained to only model Gaussian probabilities, and are therefore incapable of handling skewed or multimodal distributions. A more general approach is therefore needed when trying to estimate non-Gaussian distributions. Particle filters, which are based on sequential Monte-Carlo simulations using importance sampling, can then often be a good alternative. Particle filters are not addressed here, since this paper focuses on estimating only the most probable state of a process, and not the entire process pdf, thus rendering particle filters superfluous. Particle filters are also much more computationally demanding than Kalman filters, often making them intractable for usage in real-time settings.

4 Experiments

The experiments described here aims at determining whether there are any difference between EKF and UKF for practical tracking applications, having linear process models and nonlinear measurement models.

4.1 Process Model

The basis for the experiment is an aeroplane, modelled linearly as a dual Wiener process[3] for position and velocity respectively, driven by white noise acceleration. The model $\dot{p}_x = v_x$, $\dot{p}_y = v_y$, $\dot{v}_x = a_x$ and $\dot{v}_y = a_y$ yields the following discrete time process model when assuming zero order hold with timestep $T_s = 1$.

$$\begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix}_k + \begin{bmatrix} 0 \\ 0 \\ a_x \\ a_y \end{bmatrix}_k$$

where the accelerations a_x and a_y are modelled as uncorrelated white noise with a variance of 0.5.
The process is assumed to start in the following state:

$$\mathbf{x}_0 = [-200 \;\; 200 \;\; 4 \;\; 0]^T$$

with the squared error metric $(\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}})$ for estimation accuracy.

4.2 Tracking by Radar

Radar tracking can be modelled with a measurement model observing distance and angle to the target:

$$\begin{bmatrix} d \\ \Theta \end{bmatrix} = \begin{bmatrix} \sqrt{p_x^2 + p_y^2} \\ atan(p_y / p_x) \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

where the measurement model is clearly nonlinear. The radar is assumed to be positioned in the coordinates $(0,0)$ with measurement noises n_1 and n_1, having a variance of 200 and 0.003, respectively.

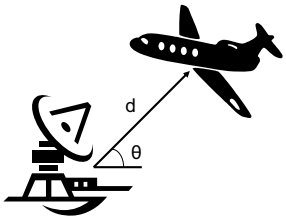

Figure 7: *Tracking of plane motion by means of a radar*

4.3 Tracking by Triangulation

Tracking by triangulation can similarly be modelled with a measurement model observing distances to the target from two observators:

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \sqrt{(p_x - p_{1x})^2 + (p_y - p_{1y})^2} \\ \sqrt{(p_x - p_{2x})^2 + (p_y - p_{2y})^2} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

where the measurement model is also clearly nonlinear. The observators are assumed to be positioned in

[3] Wiener processes are integrated white noise

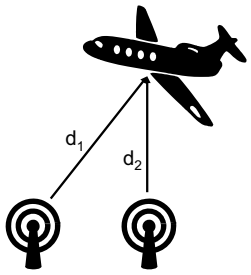

Figure 8: *Tracking of plane motion by means of triangulation*

the coordinates $(-300, 0)$ and $(300, 0)$, with measurement noises n_1 and n_2, both having a variance of 200 .

This configuration also poses an ambiguity problem, since measurements coming from a given position will be equal to the same position flipped about the x-axis. This will probably not be a big problem, since the process model prediction will help resolving this ambiguity.

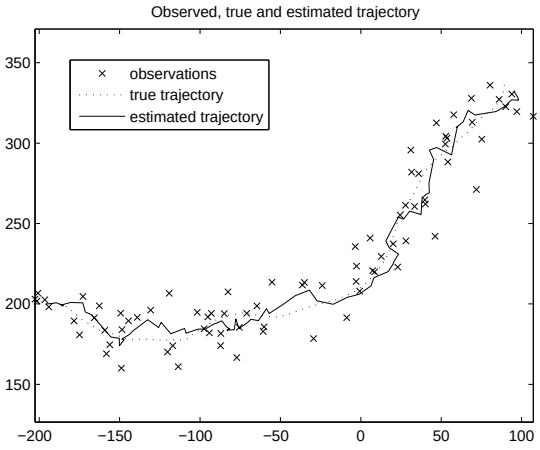

Figure 9: *Example illustration of observations, true trajectory and estimated trajectory for the experiments using the UKF*

4.4 Results

A dual simulation/estimation experiment was run 10000 times. Each time, a simulated plane trajectory were estimated over 80 time steps, by both EKF and UKF for both of the observation models. The estimation accuracy results are shown in the table below, with accuracy distribution plots in the accompanying figure. The MSE estimate variance is calculated from the empirical error distribution, using

$$VAR(MSE) = VAR(error)/N$$

Model	EKF MSE	UKF MSE
Radar	174.4 (5.00)	116.9 (0.363)
Triangulation	185.2 (3.15)	183.1 (2.81)

Table 1: *Mean squared error (MSE) for the estimations, saturating errors above 1000. Accuracy variance is given in parenthesis.*

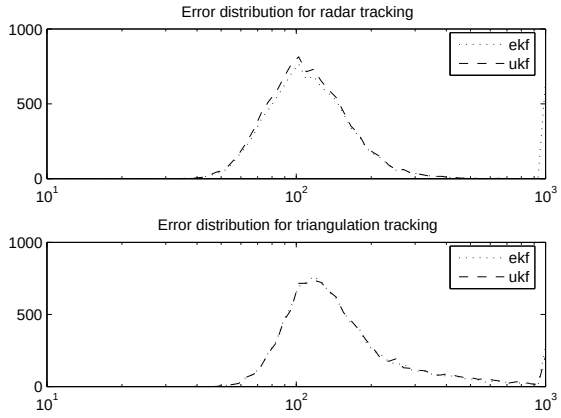

Figure 10: *Estimation accuracy distributions*

assuming validity of the central limit theorem.

The radar model, having measurements involving the highly nonlinear arcus-tangent, shows a wider difference in the estimation accuracy between EKF and UKF, compared to the triangulation model which only has Pythagoras' measurements, being significantly more linear. It can further be seen that UKF seems to show a higher degree of robustness, having fewer estimates with errors above 1000 for both of the models.

5 Conclusion

Several papers have investigated the accuracy of UKF for nonlinear process models [16], [5], but none has addresses the accuracy for nonlinear measurement models in particular.

This paper did therefore compare the relative estimation accuracy of UKF compared to EKF for linear state space models with nonlinear measurements. The empirical results shows a significant difference for the radar model, but not for the tracking model. This is believed to be caused by the difference in nonlinearity between the two models, having a highly nonlinear radar model and a relatively more linear tracking model. The relative advantage of using UKF does therefore seem to increase with the degree of nonlinearity in the measurement model. This finding is con-

sistent with the arguments for using the UKF presented in [16].

The estimation error distribution plots show that the two estimators yield quite similar results for both models, with the most significant exception being the amount of estimates having severely large errors. This leads us to the conclusion of UKF being a more robust estimator than EKF.

Statement of reproducibility: All program code required to reproduce the results shown in this paper are freely available upon request by contacting the author.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[2] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering, 3rd Edition*. Prentice Hall, 1996.

[3] Chi-Tsong Chen. *Linear System Theory and Design, third edition*. Oxford University Press, 1999.

[4] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *Kazufumi Ito and Kaiqi Xiong, Gaussian Filters for Nonlinear Filtering Problems, IEEE Transactions on Automatic Control, vol. 45, no. 5, pp. 910–927*, May 2000.

[5] Joseph J. and LaViola Jr. A comparison of unscented and extended kalman filtering for estimating quaternion motion. *Proceedings of the 2004 American Control Conference, IEEE Press, 2190-2195*, June 2004.

[6] Simon Julier. The scaled unscented transform. *Proceedings of the 2002 American Control Conference, IEEE Press*, May 2002.

[7] Simon Julier and Jeffrey Uhlmann. A general method for approximating nonlinear transformations of probability distributions. *University of Oxford*, November 1996.

[8] Simon Julier and Jeffrey Uhlmann. A new extension of the kalman filter to nonlinear systems. *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[9] Simon Julier and Jeffrey Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, March 2004.

[10] Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-Whyte. A new approach for filtering nonlinear systems. *Proceedings of the 1995 American Control Conference, IEEE Press*, June 1995.

[11] Ben Quine, Jeffrey Uhlmann, and Hugh Durrant-Whyte. Implicit jacobians for linearised state estimation in nonlinear systems. *Proceedings of the 1995 American Control Conference, IEEE Press*, June 1995.

[12] Charles W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice Hall, 1992.

[13] Rudolph van der Merwe. Sigma-point kalman filters for probabilistic inference in dynamic state-space models. *Workshop on Advances in Machine Learning, Montreal.*, June 2003.

[14] Rudolph van der Merwe and Eric Wan. Sigma-point kalman filters for integrated navigation.

[15] Rudolph van der Merwe and Eric Wan. The square-root unscented kalman filter for state and parameter-estimation. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, Utah*, May 2001.

[16] Eric Wan and Rudolph van der Merwe. The unscented kalman filter for nonlinear estimation. *Proc. of IEEE Symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada*, October 2000.