PERFORMANCE COMPARISON OF UNSCENTED AND EXTENDED KALMAN FILTER IMPLEMENTATION ON A CONTINIOUS STIRRED TANK REACTOR

by

Saman Mostafavi

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2014

Approved by:

____
Dr. Yogendra Kakad

____
Dr. Mehdi Miri

____
Dr. Mohammad Kazemi

ABSTRACT

SAMAN MOSTAFAVI. Performance comparison of Unscented and Extended Kalman Filter implementation on a continuous stirred tank reactor. (Under the direction of Dr.YOGENDRA KAKAD)

In this thesis work, it is intended to investigate not only the Extended Kalman Filter (EKF) but to further study the more recent nonlinear Kalman Filters for their application to the nonlinear problem of a continuous stirred tank reactor. The various filters studied in this thesis are:

- Extended Kalman Filter (EKF)
- Monte Carlo Kalman Filter (MCKF)
- Unscented Kalman Filter (UKF) and its various forms and alternate editions

The study goes on to provide a comparison of the computational and accuracy costs involved in the cases of EKF and UKF.

Until recently, the EKF has proven to be a better filter in terms of accuracy for the majority of non-linear cases such as the various cases of a continuous stirred tank reactor. The research work in this thesis also applies these filtering methods to the system dynamics of an exothermic continuous stirred tank reactor. Results obtained in this research corroborate the notion that an Unscented Kalman Filter tends to have a better accuracy than the previously trusted family of Kalman Filters, however after investigating the application of the UKF to the problem studied, it is still not possible to make a strong case for the UKF in general and the conclusion is that the application of this filter remains to be case sensitive.

ACKNOWLEDGMENTS

I am using this opportunity to express my gratitude to m advisor, Dr. Yogendra Kakad who supported me throughout the course of this thesis. I am thankful for his aspiring guidance, invaluably constructive criticism and friendly advice during the writing of this thesis. I am sincerely grateful to him for sharing his truthful and illuminating views on a number of issues related to this thesis and I will forever remember all the funny stories he told me about many famous scientists mentioned in this thesis, specially Dr. Kalman!

Dedications

I dedicate this thesis to my mom and my dad to whom I owe a great deal for everything and anything that I have and will achieve in my life. I could not have done this without their emotional and financial support. I would also like to thank my little brother, Sina, and my dear uncle, Taghi, for always being there when I needed reassurance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
CSTR	Continuous Stirred Tank Reactor
RV	Random Variable
GRV	Gaussian Random Variable
MCKF	Monte Carlo Kalman Filter
PDF	Probability Distribution Function

CHAPTER 1: A BRIEF HISTORY ON KALMAN FILTERING

In 1960, Rudolf Kalman proposed a new approach to linear filtering and prediction problems [1] that were later named Kalman filter in his honor. At that juncture, linear filtering that we know today was completely revolutionized.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared errors. The filter is very powerful in several aspects: it supports estimations of past, present and even future states and it can achieve this in spite of the fact that the precise nature of the modeled system is unknown [2].

Before we begin to explain the Kalman Filtering method in detail, it is worth mentioning as to why we would like to stick to this particular approach in filtering:

Most of the Control problems that are currently being faced are non-linear, but it can be argued that the extent of their non-linearity is not high. They are effectively only going to be non-linear within certain boundaries. These types of problems can be efficiently tackled by the derivatives of the Kalman Filter. These derivatives tend to either linearize around estimated points approaching the Taylor series expansion scheme or use nonlinear transforms. We can assert that although the prime Gaussian Assumption is not always correct, these filters do render their equations to be represented in state space matrix form, which reduces the math to being elegant. (Meaning readable to those with less mathematical gift!) One can now rely on linear algebra to simplify the equations and also make them computationally efficient. In addition, most of the problems can be accurately modeled via a Gaussian distribution with little to negligible error in performance. Therefore, the Kalman Filter (and its derivatives) is a good choice not only in theory but also in practice as it effectively performs the tasks asked of it, being only a little below the optimal solution and it is computationally proves to be highly efficient.

1.1 The Discrete Kalman Filter

The Kalman filter addresses the general problem of estimating the state $x \in R^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} , \tag{1.1.1}$$

with a measurement $z \in R^m$ that is

$$z_k = Hx_k + v_k \quad . \tag{1.1.2}$$

The random variables $w_k$and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions

$$p(w) \sim N(0, Q), \tag{1.2.2}$$

$$p(v) \sim N(0, R). \tag{1.1.4}$$

In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement, however in case of a linear Kalman filter we assume that they are constant.

Method of estimation in Kalman filter can be said to be very similar to a feedback control systems: the filter estimates the process state at some time and then obtains feedback in the form of measurements. (Considered to be noisy most of the time). So in essence there are a group of equations called the time update equations that are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step and there is another group of equations called the measurement update equations that are responsible for the feedback—i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

So in essence the time update equations constitute the predicting phase while the measurement update equations represents the corrector phase. Indeed the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems. In Figure 1.1 we can see a complete picture as to how the filter actually operates:

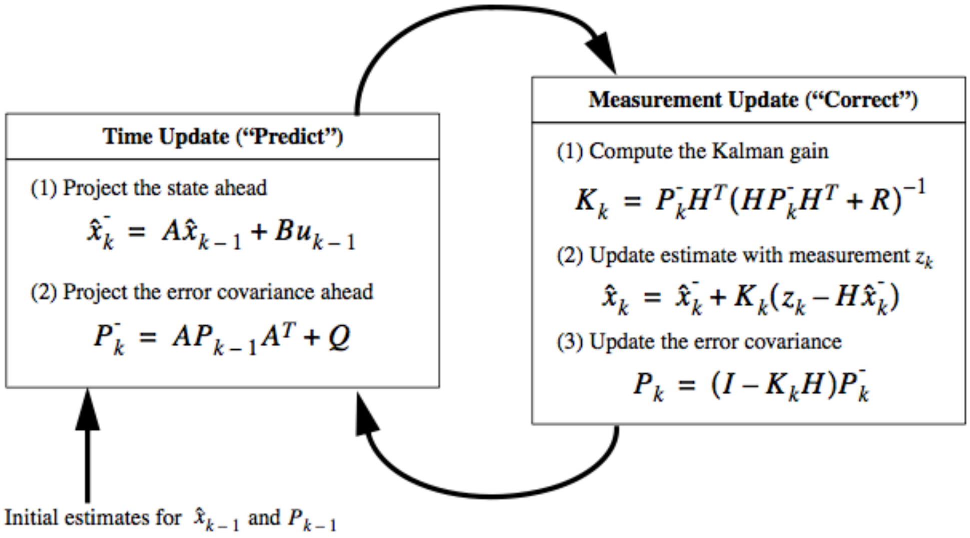
Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$
(2) Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Figure 1.1: A complete picture of the operation of the Kalman filter[2]

Note that in the equations represented in the figure 1.1:

e_k and $e^{-}{}_{k}$ are the a posteriori and a priori estimate error respectively in a sense that:

$$e_k \equiv x - \hat{x}_k \qquad (1.1.5)$$

$$e^{-}{}_k \equiv x^{-}{}_k - \hat{x}^{-}{}_k$$

P_k and $P^{-}{}_k$ are the a posteriori and a priori estimate error covariance respectively in a sense that

$$P_k \ = E[e_k {e_k}^T] \qquad (1.1.6)$$

$$P^{-}{}_k \ = E[e^{-}{}_k {e^{-}{}_k}^T]$$

and K_k is a matrix chosen to be the gain or blending factor that minimizes the a posteriori error covariance:

$$K_k = P^{-}_k H^T (H P_k H^T + R)^{-1} \qquad (1.1.7)$$

To explain a little further, the time update equations project the state and covariance estimates forward from time step k – 1 to step k and the measurement update computes the Kalman gain, K_k. The next step is to actually measure the process to obtain z_k, and then to generate an a posteriori state estimate by incorporating the measurement. The final step is to obtain an a posteriori error covariance estimate.

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman filter—it makes practical implementations much more feasible than (for example) an implementation of a Wiener filter [4] which is designed to operate on all of the data directly for each estimate. The Kalman filter instead recursively conditions the current estimate on all of the past measurements.

1.2 Limitations

In the actual implementation of the filter, the measurement noise covariance R is usually measured prior to operation of the filter. Measuring the measurement error covariance R is generally practical (possible) because we need to be able to measure the process anyway (while operating the filter) so we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement noise.

The determination of the process noise covariance Q is generally more difficult as we typically do not have the ability to directly observe the process as we are estimating. Sometimes a relatively simple (poor) process model can produce acceptable results if one "injects" enough uncertainty into the process via the selection of Q. Certainly in this case one would hope that the process measurements are reliable.

In either case, whether or not we have a rational basis for choosing the parameters, often times superior filter performance (statistically speaking) can be obtained by tuning the filter parameters Q and R. The tuning is usually performed off-line, frequently with the help of another (distinct) Kalman filter in a process generally referred to as *system identification*.

It is frequently the case however, that the measurement error (in particular) does not remain constant. For example, when sighting beacons in our optoelectronic tracker ceiling panels, the noise in measurements of nearby beacons will be smaller than that in far-away beacons. Also, the process noise Q is sometimes changed dynamically during filter operation,becoming Q_k, in order to adjust to different dynamics. For example, in the case of tracking the head of a user of a 3D virtual environment we might reduce the magnitude of Q_k if the user seems to be moving slowly, and increase the magnitude if the dynamics start changing rapidly. In such cases Q_k might be chosen to account for both uncertainty about the user's intentions and uncertainty in the model [4].

With all of that in mind, The Linear Kalman Filter has two main problems:

- Linear assumption of the system
- Gaussian assumption of Random Variable Probability Distribution

The first limitation is because of the following line of reasoning: the non-linear function of the expectation of a Random Variable is generally not the same as the expectation of the non-linear function of the Random Variable. The equation below delineates it in a mathematical form.

$$E\,\{f(x)\}\ \neq\ f(E\,\{x\}) \tag{1.1.8}$$

, where f(·) is a non-linear function.

This non-equality is not present for linear functions as the following is true for a linear function.

$$E\,\{g(x)\} =\ g(E\,\{x\}) \tag{1.1.9}$$

, where g(·) is a linear function.

The second limitation is more inherent in the nature of the Kalman Filter. The reason for this is that it assumes all of its distributions i.e., the measurement noise, the process disturbance and the main state variable being estimated to be Gaussian. This has its benefits since it allows the Kalman Filter to be broken down to linear algebraic steps; thus making it mathematically elegant and computationally efficient. However, it places an important limitation on its uses in the practical world of problems.

CHAPTER 2: THE EXTENDED KALMAN FILTER

As described in the previous chapter, the Kalman filter addresses the general problem of estimating the state $x \in R^n$ of a discrete-time controlled process that is governed by a linear stochastic difference equation. But what if the process to be estimated and (or) the measurement relationship to the process is non-linear? Some of the most interesting and successful applications of Kalman filtering have been for such situations. A Kalman filter that linearizes about the current mean and covariance is referred to as an Extended Kalman Filter or EKF.

2.1. New approaches to improve EKF

Using a similar method as it is often obtained using a Taylor series expansion, we can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even if we face a non-linear process. Let us assume that our process again has a state vector $x \in R^n$, but that the process is now in the form of a non-linear stochastic difference equation

$$x_k = f(x_{k-1},\ u_{k-1}, w_{k-1}), \tag{2.1.1}$$

with a measurement $z \in R^m$ that is

$$z_k = h(x_k, v_k), \tag{2.1.2}$$

where the random variables w_k and v_k again represent the process and measurement noise. In this case the non-linear function f in the difference equation

(2.1.1) relates the state at the previous time step k – 1 to the state at the current time step k. It includes as parameters any driving function u_{k-1} and the zero-mean process noise w_k. The non-linear function h in the measurement equation (2.1.2) relates the state x_k to the measurement z_k.

In practice of course one does not know the individual values of the noise w_k and v_k at each time step. However, one can approximate the state and measurement vector without them as

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{2.1.3}$$

and

$$\tilde{z}_k = h(\tilde{x}_k, 0), \tag{2.1.4}$$

where $\hat{x}_k$ is some a posteriori estimate of the state (from a previous time step k). (Note that process disturbance and the measurement noise are assumed to be zero-mean Gaussian process)

It is important to note that a fundamental flaw of the EKF is that the distributions (or probability densities in the continuous case) of the various random variables are no longer normal after undergoing their respective nonlinear transformations. The EKF is simply an ad hoc state estimator that only approximates the optimality of Bayes' rule by linearization [2].

To estimate a process with non-linear difference and measurement relationships, we linearize the non-linear function about its first order term of the Taylor series expansion. It can be summarized in the following explanation.

Assuming that the nonlinear function f in equation (2.1.1) is differentiable, then we can simply linearize it around the estimated point $\hat{x}_k$ with the following equation:

$$f(x(t),t) =$$

$$f(\hat{x}(t),t) + \left(\frac{\partial f(x(t),t)}{\partial x(t)}\bigg| x(t) = \hat{x}(t)\right)\left(x(t) - \hat{x}(t)\right) +$$

$$\frac{1}{2}\left(\frac{\partial^2 f(x(t),t)}{\partial x(t)^2}\bigg| x(t) = \hat{x}(t)\right)\left(x(t) - \hat{x}(t)\right)^2 + \cdots \tag{2.1.5}$$

The key fact that enables the working of the Extended Kalman Filter is that, since we deal with non-linear yet almost linear dynamics (around a certain boundary) the quadratic and higher order terms can be neglected. This would result in the following about the estimated trajectory $\delta\hat{x}$:

$$\delta\dot{x} = f(\hat{x}(t),t) + \left(\frac{\partial f(x(t),t)}{\partial x(t)}\bigg| x(t) = \hat{x}(t)\right)\delta \mathrm{x(t)} \tag{2.1.6}$$

using this method, writing the new governing equations that linearize an estimate about (2.2.3) and (2.2.4) would result into :

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1}\ , \tag{2.1.7}$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k\ . \tag{2.1.8}$$

where,

A is the Jacobian matrix of partial derivatives of f with respect to x,

W is the Jacobian matrix of partial derivatives of f with respect to w,

H is the Jacobian matrix of partial derivatives of h with respect to x,

V is the Jacobian matrix of partial derivatives of h with respect to v,

So for example the observer matrix H can be given as:

$$H_j = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}\Bigg\| \ x(t) = \hat{x}(t) \tag{2.1.9}$$

and the rest just follows the same pattern.

Like the previous case of Kalman Filter, Here is a diagram of how the filter actually works:

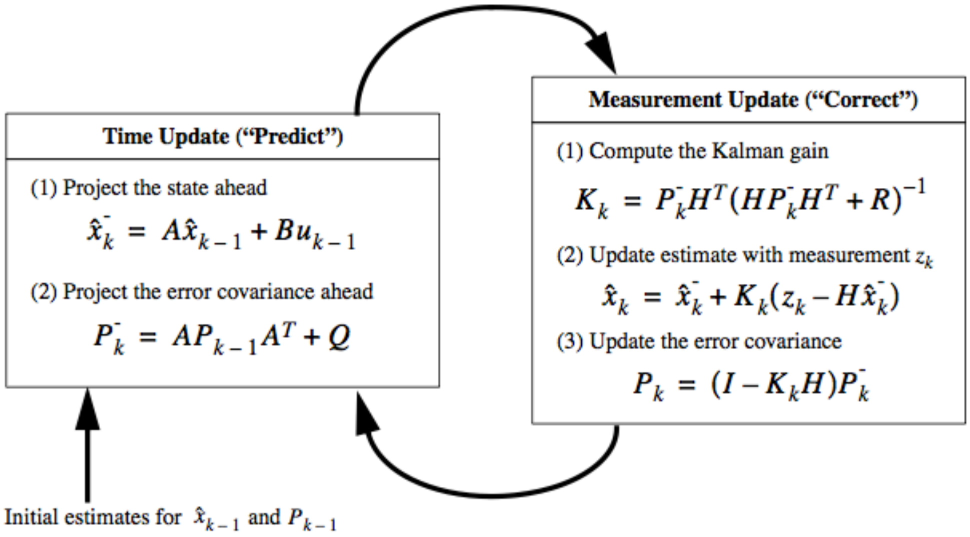
Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$
(2) Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Figure 2.1: A complete picture of the operation of the Extended Kalman Filter [2]

An important feature of the EKF is that the matrix H in the equation for the Kalman gain K serves to correctly propagate or "magnify" only the relevant component of the measurement information. Of course if, over all in measurements there is not a one-to-one mapping between the measurement and the state, then as you might expect the filter will quickly diverge. In this case the process is *unobservable*.

2.2 New approaches to improve EKF

Before we get to the methods that actually improved upon EKF, it is necessary to have a look at an approach once considered to be the ultimate solution, the Particle Filter. It is actually way older than the Kalman Filter. Dating back to 1933, particle filters have become a very popular class of numerical methods for the solution of optimal estimation problems in non-linear non-Gaussian scenarios, both of which are considered to be the weaknesses of Kalman Filtering. In comparison with standard approximation methods, such as the Extended Kalman Filter, the principal advantage of particle methods is that they do not rely on any local linearization technique or any crude functional approximation. The price that must be paid for this flexibility is computational: these methods are computationally expensive [5].

The accuracy of the particle filter relies on the number of samples (particles) it uses. It requires a very large number of particles to produce a decent estimate and an even larger set to actually yield better results than an Extended Kalman Filter. While it does eventually overcome the EKF by a good amount, the simultaneous propagation of all these particles in real-time puts a heavy load on the computation part of the scheme. Therefore, it levies a time constraint in a practical scenario where extraordinary computational power is not available for filtering and tracking. Therefore, in a case where one wants to have the luxury of real time estimation process, it renders itself to be very slow and in effective. And the problem only scales up with higher dimensionality of the system dynamics. So unless one has a very fast processing machine to spare for this, the Particle Filter becomes somewhat a lesser-desired alternative in many cases.

2.2.1 The Iterated Extended Kalman Filter

The EKF linearized the state and measurement equations about the predicted state as an operating point. This prediction is often inaccurate in practice. Re-evaluating the filter around the new estimated state operating point can refine the estimate. This refinement procedure can be iterated until little extra improvement is obtained and hence it is called the Iterated Extended Kalman Filter.

2.2.2 Monte Carlo Kalman Filter

Before we proceed further, it should be noted that this is not actually a real filter used in practice and it is rather presented here as a precursor to the discussion of the *Unscented Kalman Filter* that comes in the next chapter.

To explain how a Monte Carlo Kalman filter works, we first have to introduce the statistical linearization technique. So far we managed to discuss the expansion of Taylor series for the design of an EKF. Statistical approximation is an alternate approach and one that is generally considered to be more accurate. The basic principle of this technique is conveniently illustrated for a scalar function, $s(x)$, of a random variable x [6].

A random variable x is a variable that can take on a set of possible different values (similarly to other mathematical variables), each with an associated probability.

Consider that $s(x)$ is to be approximated by a series expansion of the form

$$s(x) \cong n_0 + n_1 x + n_2 x^2 + \cdots + n_m x^m \tag{2.2.1}$$

The problem of determining appropriate coefficients n_k is similar to the estimation problem where an estimate of a random variable is sought from given measurement data. Analogous to the concept of estimation error, we define a function representation error, e, of the form

$$e = s(x) - n_0 - n_1 x - n_2 x^2 - \cdots - n_m x^m \tag{2.2.2}$$

It is desirable that n_k be chosen so that e is small in some ”average” sense; any procedure that is used to accomplish this goal, which is based upon the statistical properties of x, so it can be thought of as a statistical approximation technique. The most frequently used method for choosing the coefficients in (2.2.1) is to minimize the mean square error value of e. This is achieved by forming

$$E[(s(x) - n_0 - n_1 x - n_2 x^2 - \cdots - n_m x^m)^2] \tag{2.2.3}$$

and setting the partial derivatives of this quantity with respect to each n_k equal to zero. The result is a set of algebraic equations, linear in the n_k, that can be solved in terms of the moments and cross-moments of x and $s(x)$. We can now abstractly determine that statistical approximation has a distinct advantage over the Taylor Series expansion; it simply does not require the existence of derivatives of $s(\cdot)$ function.

With all that being said, there is a major drawback that we face in statistical linearization techniques, and that lies in the nature of the expectation operation. This would require the knowledge of the probability density function of x in order to compute the coefficients n_k. This requirement does not exist for the Taylor Series expansion employed in the EKF. One may argue that approximations can often be made for the probability distribution function used to calculate the coefficients n_k, such that the resulting expansion for $s(x)$ is considerably more accurate than the Taylor series. While, this is generally true, we choose not to dwell on this topic because the nature of approximate probability distribution functions is complex and beyond the scope of this thesis.

By approximating the Jacobian matrix of the system in a broader region centered at the state of the system, we have a statistically linearized Kalman Filter that tries to avoid the problems of an EKF. This type of approach also offers the benefit that it does not require continuity or differentiability of the motion and measurement models. Since it is not necessary to compute Jacobian matrices, these methods can offer benefits in terms of computational efficiency as well.

Mante-Carlo Kalman Filter (MCKF) samples the points around the mean and tries to find the best fit (a line, a plane, etc.) over the points and use this approximation to run the Kalman Filter. [7] Consider the prior distribution of X_k. MCKF draws sample from this distribuition and passes them through the non-linear state propagation function $f(X)$. Now, with these transformed samples, we can draw the a priori mean and covariance as follows:

$$\hat{X}^{-}{}_{k+1} = \frac{1}{N}\sum_{i=1}^{N} f(X_k^i) \qquad (2.2.4)$$

This gives us the mean of the distribution that is propagated forward through the non-linear function f. Here X_k^i denotes the samples taken from the prior distribution of X. These points are taken about the principal axis of the pdf. Now, for the covariance,

$$P_{k+1}^{-} = Q + \frac{1}{N}\sum_{i=1}^{N}[(f(X_k^i) - \hat{X}^{-}{}_{k+1})\left(f\left(X_k^i\right) - \hat{X}^{-}{}_{k+1}\right)^T] \qquad (2.2.5)$$

As we can see, here we first draw samples from the prior distribution, pass it through the non-linear function and then approximate their mean by averaging methods. We could also have chosen to propagate the mean of the prior distribution of X i.e. $\hat{X}_k$ through the function, in case f is linear and the samples are perfectly Gaussian. The primary difficulty encountered in the later approach for non-linear functions was explained in section 1.2.

Now we pass the samples through the observer function h and obtain the mean and covariance of $\hat{Y}_{k+1}$ as well as the cross covariance of $\hat{X}$ and $\hat{Y}$:

$$Y_{k+1}^i = h(f\left(X_k^i\right))$$

$$\hat{Y}_{k+1} = \frac{1}{N}\sum_{i=1}^{N} h(f\left(X_k^i\right)) \qquad (2.2.6)$$

The error covariance in the estimated observed output $\hat{Y}_{k+1}$is given by the equation below and the measurement noise is incorporated in this step:

$$Y_{cov} = R + \frac{1}{N}\sum_{i=1}^{N}[(h(f(X_k^i)) - \hat{Y}_{k+1})(h(f(X_k^i)) - \hat{Y}_{k+1})^T] \qquad (2.2.7)$$

The cross-covariance between X and Y is given by,

$$XY_{cov} = \frac{1}{N}\sum_{i=1}^{N}[f(X_k^i) - \hat{X}^-{}_{k+1}))(h(f(X_k^i)) - \hat{Y}_{k+1})^T] \qquad (2.2.8)$$

The Kalman gain K_{k+1} would be simply the cross covariance matrix divided by the error covariance:

$$K_{k+1} = \frac{XY_{cov}}{Y_{cov}} \qquad (2.2.9)$$

and after we obtain the actual measurement for the $(k + 1)th$ step, post-measurement a posteriori mean is given by:

$$\hat{X}_{k+1} = \hat{x}_{k+1}^- + K_{k+1}[Y_{k+1} - \hat{Y}_{k+1}] \qquad (2.2.10)$$

and finally, the a posteriori error covariance is going to be:

$$P_{k+1} = P_{k+1}^- - K_{k+1}Y_{cov}{K_{k+1}}^T \qquad (2.2.11)$$

The Monte-Carlo Kalman Filter is an intuitive example to grasp the concept of the Statistical Linearized Kalman Filter, as well as to understand our transition to Sigma Point Kalman Filters (which is explained in the following chapter). However, it is not a very practical filter and is rarely (if at all) used in real world applications. This is due to the fact that it requires knowledge of the prior pdf of X, the number of samples it demands for proper functioning is very high and the fact that if one uses these many samples one may as well go with the Particle Filter which is multi-modal compared to the unimodal of the MCFK [7].

CHAPTER 3: THE UNCENTED KALMAN FILTER

Although the EKF maintains the elegant and computationally efficient recursive update form of the KF, it suffers from a number of serious limitations:

Linearized transformations are only reliable if the error propagation can be well approximated by a linear function. If this condition does not hold, the linearized approximation can be extremely poor. At best, this undermines the performance of the filter. At worst, it causes its estimates to diverge altogether. However, determining the validity of this assumption is extremely difficult because it depends on the transformation, the current state estimate, and the magnitude of the covariance. This problem is well documented in many applications and has been mentioned in an example in [9] as well.

Linearization can be applied only if the Jacobian matrix exists. However, this is not always the case. Some systems contain discontinuities (for example, the process model might be jump-linear, in which case the parameters can change abruptly, or the sensor might return highly quantized sensor measurements), where as others have singularities (for example, perspective projection equations and in others the states themselves are inherently discrete [9].

Calculating Jacobian matrices can be a very difficult and error-prone process. The Jacobian equations frequently produce many pages of dense algebra that must be converted to code. As far as programming is concerned, this is actually going to open up windows of opportunities for human errors and it is often said that these errors are difficult to debug and identify.

The Unscented Kalman Filter introduced by Julier, Simon J. and Uhlmann, Jeffrey K. [18] led the way for a family of Sigma Point Kalman Filters (which now include the UKF, scaled UKF, iterated UKF, Central-Difference Kalman Filter(CDKF) and some more).

So before we actually develop the UKF, it is better to start of by explaining the sigma point approach:

1. A set of weighted sigma-points is deterministically calculated using the mean and square-root decomposition of the covariance matrix of the prior random variable. As a minimal requirement the sigma-point set must completely capture the first and second order moments of the prior random variable. Higher order moments can be captured, if so desired, at the cost of using more sigma-points.

2. The sigma-points are then propagated through the true nonlinear function using functional evaluations alone, i.e., no analytical derivatives are used, in order to generate a posterior sigma-point set.

3. The posterior statistics are calculated (approximated) using tractable functions of the propagated sigma-points and weights. (These steps are going to be further explained under the unscented transform)

3.1. The Unscented Transform

Uhlmann’s and Julier’s first paper on the UKF [18] outlines a scheme for Unscented Transform that utilized in this research work as well.

Let us consider an example: Let our main state vector be described by a random variable x having a distinct probability distribution function. Assume a set of sigma points chosen from this distribution of x, whose mean is denoted by $\bar{x}$ and the covariance by P_{xx}. Applying a nonlinear transformation to these sigma points would result in a new distribution that will not retain the same properties as that of x. Let the nonlinear function transform our random variable x into another random variable y. (y being a nonlinear function f of x)

The prime objective here is to obtain the statistics of the random variable y, its mean $\bar{y}$ and covariance P_{yy}. We do this by transforming each of our sigma points drawn deterministically from x, through the nonlinear function f. This results in a set of transformed sigma points whose mean and covariance are now $\bar{y}$ and P_{yy}.

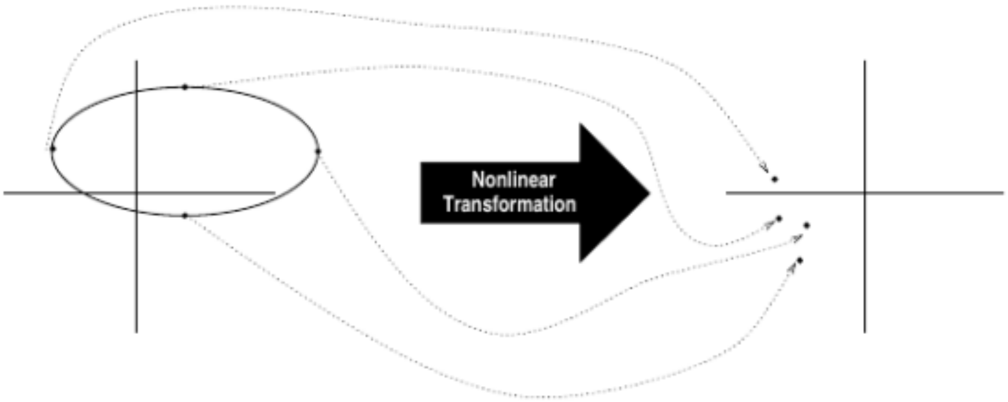

Figure 3.1: The principle of the unscented transform [18]

Although this method bares a resemblance to Monte Carlo Kalman Filter there is an extremely important and fundamental difference. The samples are not drawn at random but rather according to a specific, deterministic algorithm with weights attached to each sigma points. The different versions of the sigma point filters differ on how these weights are selected. All versions choose weights so that the method behaves perfectly for a Gaussian model (linear dynamics) and then optimize the weights for different criteria. Since the problems of statistical convergence are not an issue, high order information about the distribution can be captured using only a very small number of points.

The n-dimensional random variable x with mean $\bar{x}$ and covariance P_{xx} is approximated by 2n + 1 weighted points given by:

$$\begin{aligned} X_{s\,0} &= \bar{x} & W_0 &= \frac{\kappa}{n+\kappa} \\ X_{s\,i} &= \bar{x} + (\sqrt{(n+\kappa)P_{xx}}\,)_i & W_i &= \frac{1}{2(n+\kappa)} \\ X_{s\,i+n} &= \bar{x} - (\sqrt{(n+\kappa)P_{xx}}\,)_i & W_{i+n} &= \frac{1}{2(n+\kappa)} \end{aligned} \tag{3.1.1}$$

where $\kappa \in \Re$, $(\sqrt{(n+\kappa)P_{xx}}\,)_i$ is the ith row or column of the matrix square root of $(n+\kappa)P_{xx}$ and W_i is the weight which is associated with the ith point. It is worth mentioning that essentially κ is the tuning factor which gives us extra room to fine tune the UKF to be application specific.

This procedure consists of these steps:

1. Generate sigma points from the mean and covariance of the given random variable using the weighted selection scheme mentioned in (3.1.1).

2. Pass each of these sigma points through the nonlinear function, to yield the set of transformed sigma points:

$$\mathrm{Y}_{s\,i} = f(X_{s\,i}) \tag{3.1.2}$$

3. The mean of the transformed sigma points is given by the weighted averages of the transformed points:

$$\bar{Y} = \sum_{i=0}^{2n} W_i \, \Upsilon_{s\,i} \qquad (3.1.3)$$

4. The covariance is the weighted outer product of the transformed points:

$$P_{yy} = \sum_{i=0}^{2n} W_i \, [\Upsilon_{s\,i} - \bar{Y}][\Upsilon_{s\,i} - \bar{Y}]^T \qquad (3.1.4)$$

3.2. The Scaled Unscented Transform

Julier suggested a modified selection scheme for the selection of the sigma points in his papers on The Scaled Unscented Transformation [8]. He states that his modification to the unscented transformation, would help increase the robustness of the sampling method against higher order nonlinearities beyond the second order.

This method uses adjustable scaling parameters (α, β, κ) to allow for some fine tuning of the transform for specific applications. Note that λ and γ are auxiliary parameters used for notational ease. The structure of the scalar weights used in the unscented transform is given by:

$$\begin{array}{ll}
X_{s\,0} = \bar{x} & W_0^m = \frac{\lambda}{n+\lambda} \\
X_{s\,i} = \bar{x} + (\gamma\sqrt{P_{xx}}\,)_i & W_0^m = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \qquad (3.2.1) \\
X_{s\,i+n} = \bar{x} - (\gamma\sqrt{P_{xx}}\,)_i & W_i^m = W_i^c = \frac{1}{2(n+\lambda)} \; for \; i \neq 0
\end{array}$$

(note: There are a total of 2n + 1 weights one for each sigma point)

where $\lambda = \alpha^2(n + \kappa) - n$ and $\gamma = \sqrt{n + \lambda}$ and n is the dimension of random variable x.

In the above equations the W_i^m are the weights used to calculate the mean of the transformed distribution and the W_i^c are the weights used to calculate the covariance of the transformed distribution. The procedure for sigma points is updated to reflect scaling as follows:

1. Generate sigma points from the mean and covariance of the given RV using the weighted selection scheme mention in (3.1.1).

2. Pass each of these sigma points through the nonlinear function, to yield the set of transformed sigma points:

$$\Upsilon_{s\,i} = f(X_{s\,i}) \tag{3.2.2}$$

3. The mean of the transformed sigma points is given by the weighted averages of the transformed points:

$$\bar{Y} = \sum_{i=0}^{2n} W_i^m \, \Upsilon_{s\,i} \tag{3.2.3}$$

4. The covariance is the weighted outer product of the transformed points:

$$P_{yy} = \sum_{i=0}^{2n} W_i^c \, [\Upsilon_{s\,i} - \bar{Y}][\Upsilon_{s\,i} - \bar{Y}]^T \tag{3.2.4}$$

It should be noted that κ is a secondary scaling parameter, which is usually set to 0. Putting a negative value in κ might affect the positive semi-definiteness of the transformed covariance. α is the main scaling factor. An acceptable range is usually somewhere between 10^{-4} and 1 [12]. In summary as α approaches one the scaled unscented transform is the same as the normal unscented transform and as it goes towards 0 it further moves sigma points towards the mean. β is used to incorporate prior knowledge of the distribution of (for Gaussian distributions, $\beta = 2$ is optimal [10]).

3.3. The Filter Algorithm

We have to slightly restructure the state vector and the dynamic process and the observation models. The state vector is augmented with the process and noise terms to give an n + q dimensional augmented vector x^a, where q is dimension of the process disturbance variance Q. Therefore

$$\hat{x}^a = \begin{bmatrix} \hat{x}_k \\ 0_{q\times 1} \end{bmatrix} \quad (3.3.1)$$

One additional step that we need to do is to augment the main error covariance matrix P_{xx}:

$$P^a{}_{xx_k} = \begin{bmatrix} P_{xx_k} & 0_{n\times q} \\ 0_{q\times n} & Q \end{bmatrix} \quad (3.3.2)$$

Because of these augmentation this algorithm is commonly refered to as the augmented UKF.

In his paper, Julier describes the algorithms steps as being:

1. The $2n^a$+1 sigma point is created by applying (3.1.1) to the above augmented system. Here n^a is the new dimension of our state vector. If the original state vector is of dimension n and the process disturbance is of dimension q then n^a is equal to n+q.
2. The transformed sigma points is given by instantiating each point through the process model (nonlinear function):

$$X^{a^-}_{k+1_i} = f^a(X^a_{k_i}, u_k) \quad (3.3.3)$$

3. The predicted mean should be calculated as:

$$\hat{x}^{a^-}_{k+1_i} = \sum_{i=0}^{2n^a} W_i \, \hat{X}^{a^-}_{k+1_i} \quad (3.3.4)$$

   It should be noted that this is essentially a predicted priori state.

4. The prior covariance should be computed too:

$$P^{a^-}{}_{xx_{k+1}} = \sum_{i=0}^{2n^a} W_i \, [X^{a^-}_{k+1_i} - \hat{x}^{a^-}_{k+1}][X^{a^-}_{k+1_i} - \hat{x}^{a^-}_{k+1}]^T \quad (3.3.5)$$

5. Passing the points through the observer function, we instantiate each of the prediction points:

$$Y_{k+1_i} = h(X^{a^-}_{k+1_i}, u_k) \tag{3.3.6}$$

6. The predicted observation should be calculated as:

$$\widehat{y}_{k+1} = \sum_{i=0}^{2n^a} W_i\, Y^a_{k+1_i} \tag{3.3.7}$$

7. Since the observation noise is additive and independent, we can just add the measurement noise variance R in calculating the covariance:

$$P_{yy_{k+1}} = R + \sum_{i=0}^{2n^a} W_i\, [Y_{k+1_i} - \hat{y}_{k+1}][Y_{k+1_i} - \hat{y}_{k+1}]^T \tag{3.3.8}$$

8. Finally we can find the cross correlation matrix as:

$$P_{xy} = R + \sum_{i=0}^{2n^a} W_i\, [X^{a^-}_{k+1_i} - \hat{x}^{a^-}_{k+1}][Y_{k+1_i} - \hat{y}_{k+1}]^T \tag{3.3.9}$$

9. The measurement update (correction) in the filter is going to be as follows:

$$Kalman\ Gain{:}\ K_{k+1} = \frac{P_{xy}}{P_{yy_{k+1}}} \tag{3.3.10}$$

$$Estimate\ Update\ with\ Measurment{:}\ \hat{x}_{k+1} = \hat{x}^{a^-}_{k+1} + K_{k+1}[Y_{k+1} - \hat{y}_{k+1}]$$

$$A\ Posteriori\ Error\ Covariance{:}\ P\ \ _{xx_{k+1}} = P^-{}_{xx_{k+1}} - K_{k+1} P_{yy_{k+1}} K^T{}_{k+1}$$

By carefully noting that in order to incorporate the effects of the process disturbance on the mean and covariance of the sigma points, we augment the state vectors. One side effect of this is that it will require us to include more sigma points in the propagation. However, Julier has gone on to note that this augmented scheme implies that correlated noise sources can be implemented easily and the process disturbance if not additive can be incorporated in the mean and the covariance [18]. Overall in most cases the augmented implementation is favored when the process disturbance is introduced into the system in a nonlinear and/or non-additive way.

If we know that the process disturbance is additive then it is not necessary to augment the state vector. This can lead to a reduction in the number of the sigma points to be propagated and can reduce the computational cost.

This algorithm utilizes the same non-augmented sigma points as in (3.1.1), so n, $\hat{x}_k$ and P_{xx_k} are going to remain the same. The main difference here is the fact that when calculating a priori covariance matrix here, we should add the process disturbance variance matrix Q to it. Thus the step 4 is as follows:

$$P^{-}{}_{xx_{k+1}} = Q + \sum_{i=0}^{2n} W_i \, [X^{-}{}_{k+1_i} - \hat{x}_{k+1}][Y^{-}{}_{k+1_i} - \hat{x}_{k+1}]^T \tag{3.3.11}$$

Now that we have included the effect of the process disturbance into our covariance matrix, the sigma points in (3.1.1) have to be calculated again and the rest of the algorithm is going to remain the same.

Some have argued that the non-augmented UKF is equivalent to the augmented counterpart only if $n + \kappa = const$ is satisfied and that the non-augmented UKF usage can lead to noticeable losses in accuracy [12]. The basic difference between the augmented and non-augmented UKFs is that the former draws sigma points only once in the recursive process while the latter has to redraw a new set of sigma points to incorporate the effect of additive process disturbance. This difference generally favors the augmented UKF in that the odd-order moment information is captured by the propagated sigma points and well propagated within one recursion.

3.4. A Refined Version of the UKF

Ever since the introduction of the Unscented Kalman Filter, there have been arguments about its practicality. There have been mentions of its computational complexity and comparatively slow speed of operation (as to that of the EKF), and therefore in spite of its argued superior accuracy over the EKF, it is still not used in applications that are constrained by low computational capabilities. There have been many refinements suggested to address these issues. One of the most popular ones is the Square-Root Unscented Kalman Filter.

Rudolph Van der Merwe introduced the Square-Root Unscented Kalman Filter in his paper published in 2001 [11]. The most computationally expensive operation in the UKF corresponds to calculating the new set of sigma points at each time update. This requires taking a matrix square-root of the state covariance matrix ($SS^T = P_{xx}$). While $\sqrt{P_{xx}}$ is an integral part of the UKF, it is still the full covariance P_{xx} which is recursively updated. In the SR-UKF implementation, S will be propagated directly, avoiding the need to refectories at each time step.

There are three linear algebraic techniques that we use in SR-UKF namely *QR decomposition, Cholesky factor up- dating and efficient least squares*.

QR Decomposition: The QR Decomposition or factorization of a matrix $A\epsilon\, \Re^{L\times N}$ is given by $A^T = QR$, where $Q\epsilon\, \Re^{N\times N}$ is an orthogonal matrix and $R\epsilon\, \Re^{N\times L}$ is upper triangular and $N \geq L$. The upper right triangular part of R, $\tilde{R}$, is the transpose of the Cholesky factor of $P = AA^T$. This culminates into $\tilde{R} = S^T$. We use the short hand notation $qr\{.\}$ to donate a QR Decomposition of a matrix where only $\tilde{R}$ is returned.

Cholesky Factor Updating: If S is the original Cholesky factor of $P = AA^T$, then the Cholesky factor of the rank-1 update (or downdate) $P \pm \sqrt{v}uu^t$ is denoted as S= choleupdate $\{S, u, \pm v\}$. If u is a matrix and not a vector, then the result is M

consecutive updates of the Cholesky factor using the M columns of u. (Available in Matlab as *cholupdate*)

Efficient Least Squares: The solution to the equation $(AA^T)x = A^T b$ also corresponds to the solution of the over determined least squares problem $Ax = b$. This can be solved efficiently using a QR decomposition with pivoting (implemented in Matlab's '/' operator).

It is important to note that the positive semi-definite matrix P can have many square roots in the form of $P = SS^T$. The reason for using the Cholesky Decomposition of the covariance matrix as an alternative to its usual matrix square root is that it adds numerical efficiency and stability as the Cholesky factors of P are generally much better conditioned for matrix operations than P itself. The idea of using a Cholesky factor for covariance matrix was first implemented by James E. Potter in his efforts to improve the numerical stability of the measurement update equations. His implementation came to be called as square-root filtering [11].

The updated UKF algorithm would be as follows:

1. Initialize with:

$$\hat{X}_0 = E[X]$$

$$P_0 = E[(X_0 - \hat{X}_0)(X_0 - \hat{X}_0)^T] \tag{3.4.1}$$

$$S_0 = chol\{P_0\}$$

   Also the scaled unscented transform mentioned in (3.2) is used to assign the weights.

2. Generate the sigma points using the Cholesky factor of the error covariance:

$$X_{k_0} = \hat{x}_k^+$$

$$X_{k_i} = \hat{x}_k^+ + \gamma S_k \qquad for\ i = 1,2,\dots,n \tag{3.4.2}$$

$$X_{k_{i+n}} = \hat{x}_k^+ + \gamma S_k \qquad for\ i = n+1,\dots,n$$

3. The transformed sigma points are given by instantiating each point through the process model (nonlinear function):

$$X_{k+1_i}^- = f\left(X_{k_i}, u_k\right) \qquad for\ i = 0,1,\dots,2n \tag{3.4.3}$$

4. The predicted mean should be calculated as:

$$\hat{x}_{k+1_i}^- = \textstyle\sum_{i=0}^{2n} W_i{}^m\, \hat{X}_{k+1_i}^- \tag{3.4.4}$$

5. Cholesky update is calculated as:

$$S_{k+1}^- = qr\left\{\left[\sqrt{W_i^c}(X_{k+1}^-|_{i=1:2n} - \hat{x}_{k+1}) \qquad \sqrt{Q}\right]\right\} \tag{3.4.5}$$

Since the weight of W_0^c might be negative, we use this specific update for that:

$$S_{k+1}^- = cholupdate\left\{S_{k+1}^-, X_{k+1_0}^- - \hat{x}_{k+1}, W_0^c\right\} \tag{3.4.6}$$

6. Now we go through step 2 again with the updated S_k essentially reproduce the sigma points. Recall that this is as a result of not using the augmented UKF
7. Passing the points through the observer function, we instantiate each of the prediction points:

$$Y_{k+1_i} = h(X_{k+1_i}^-, u_k) \tag{3.4.7}$$

8. Now we go through the measurement update phase in which we have to calculate the a posteriori values:

$$\hat{y}_{k+1} = \sum_{i=0}^{2n} W_i{}^m\, Y_{k+1_i}^-$$

$$S_{y_{k+1}}^- = qr\left\{\left[\sqrt{W_i^c}(Y_{k+1}^-|_{i=1:2n} - \hat{y}_{k+1}) \qquad \sqrt{R}\right]\right\}$$

$$S_{y_{k+1}}^- = cholupdate\{S_{y_{k+1}}^-, Y_{k+1_0}^- - \hat{y}_{k+1}, W_0^c\} \tag{3.4.8}$$

9. Finally we solve the cross correlation matrix as follows:

$$P_{xy} = \sum_{i=0}^{2n} W_i{}^c \, [X_{k+1_i}^- - \hat{x}_{k+1}^-][Y_{k+1_i} - \hat{y}_{k+1}]^T \qquad (3.4.9)$$

10. And we conclude the algorithm by finding the Kalman gain and using that gain to minimize the a posteriori error covariance:

$$Kalman\ Gain\colon\ K_{k+1} = (P_{xy}/S_{y_{k+1}}^T)/S_{y_{k+1}}$$

$$Estimate\ Update\ with\ Measurment\colon\ \hat{x}_{k+1}^+ = \hat{x}_{k+1}^{a^-} + K_{k+1}[Y_{k+1} - \hat{y}_{k+1}]$$

$$A\ Posteriori\ Cholesky\ factor\ of\ Error\ Covariance\colon$$

$$S_{k+1} = cholupdate\{S_{k+1}^-, K_{k+1}S_{y_{k+1}}, -1\} \qquad (3.4.10)$$

Note that since $S_{y_{k+1}}$ is a square and triangular matrix, efficient ”back-substitutions” can be used to solve for the Kalman gain K_{k+1} directly without the need for a matrix inversion.

It has been argued that among all the UKF methods in use, this one has better numerical properties and guarantees positive semi-definiteness of the underlying state covariance [11].

3.5 Comparing UKF and EKF

A central and vital operation performed in the Kalman Filter is the propagation of a Gaussian random variable through the system dynamics. In the EKF, the state distribution is approximated by a Gaussian random variable, which is then propagated analytically through the first-order linearization of the nonlinear system. This can introduce large errors in the true posterior mean and covariance of the transformed GRV, which may lead to sub-optimal performance and sometimes divergence of the filter. The UKF addresses this problem by using a deterministic sampling approach. The state distribution is again approximated by a GRV, but is now represented using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the GRV, and when propagated through the true non-linear system, captures the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any nonlinearity. The EKF, in contrast, only achieves first-order accuracy. Remarkably, the computational complexity of the UKF is of the same order as that of the EKF.

Below is an illustration of an example of mean a covariance propagation through a nonlinear function and how our unscented transform achieves to stay true to the propagated values in comparison with a linearization method based on the first order term of Taylor series expansion.

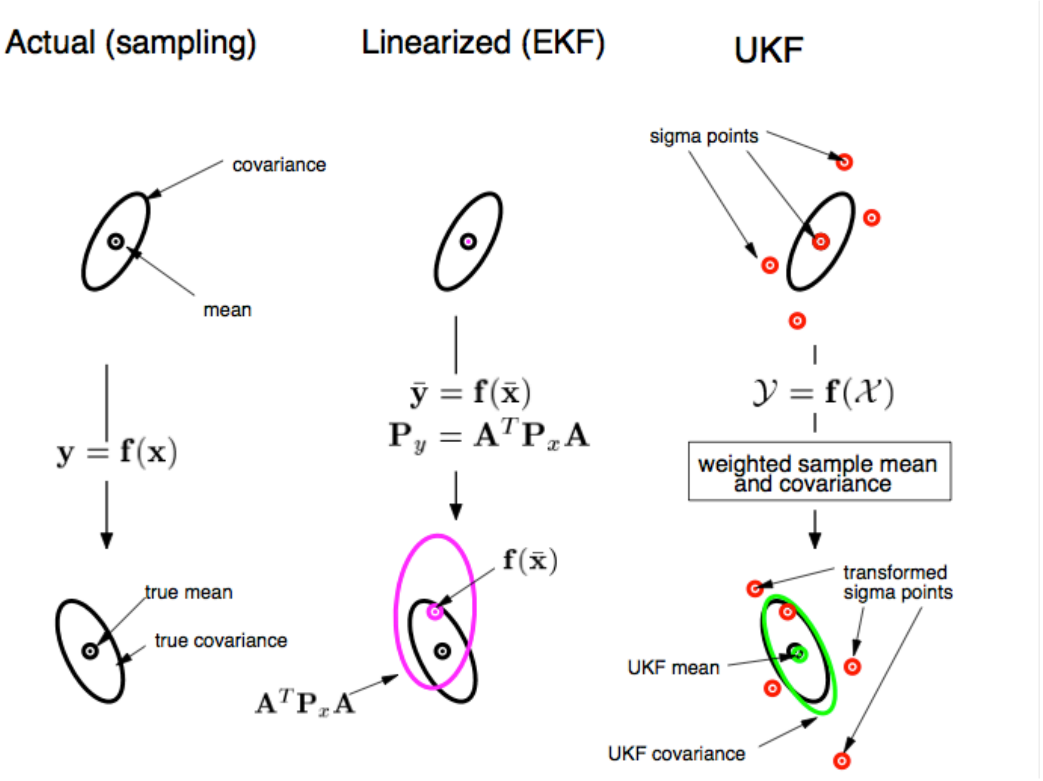
Actual (sampling)
Linearized (EKF)
UKF
covariance
mean
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
true mean
true covariance
$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{P}_y = \mathbf{A}^T\mathbf{P}_x\mathbf{A}$
$\mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{A}^T\mathbf{P}_x\mathbf{A}$
sigma points
$\mathcal{Y} = \mathbf{f}(\mathcal{X})$
weighted sample mean and covariance
transformed sigma points
UKF mean
UKF covariance

Figure 3.2: Example of mean and covariance propagation. a) actual, b) first-order linearization (EKF), c) new "sampling" approach (UKF) [11].

CHAPTER 4: MATHEMATICAL MODELING AND FILTERING OF A CONTINUOUS STIRRED TANK REACTOR

Continuous Stirred Tank Reactor (CSTR) is a typical chemical reactor system with complex nonlinear dynamic characteristics. The reason that this specific model was selected for this research work is because the variables that characterize the quality of the final product in Continuous Stirred Tank Reactor are often difficult to measure in real-time and cannot be directly controlled using feedback configuration. Hence, there has been a great deal of research done and papers published recently, implementing a variety of KF methods for this specific problem.

A CSTR exhibits highly nonlinear behavior and usually has wide operating ranges. Chemical reactions in a reactor are either exothermic or endothermic and require that energy can either be removed or added to the reactor to maintain a constant temperature. The CSTR is normally run at steady state and is usually operated so as to be quite well mixed. As a result of this quality, the CSTR is generally modeled as having no special variations in concentration, temperature or reaction rate throughout the vessel. Since the temperature and concentration are identical everywhere within the reaction vessel, they are the same at the exit point as they are elsewhere in the tank. Thus the temperature and concentration at the exit are modeled as being the same as those inside the reactor [13]. In systems where mixing is not ideal, the well-mixed model is inadequate. The following presents the mathematical modeling of an ideal CSTR.

4.1. Process Description

The first principles model of the continuous stirred tank reactor and the operating data (given in Table 4.1) as specified in the Pottman and Seborg [14] has been used in filter design and the simulation studies. Highly nonlinear CSTR is common in chemical and petrochemical plants. In the process considered for the simulation study as shown in Figure 4.1, an irreversible, exothermic chemical reaction $A \rightarrow B$ occurs in constant volume reactor that is cooled by a single coolant stream. A feed material of composition C_{A_0}enters the reactor at temperature T_0, at a constant volumetric flow rate q. Product is withdrawn from the reactor at the same volumetric flow rate q. The mixing is assumed to be efficient enough to guarantee homogeneity of the liquid content within the reactor.

In a jacketed CSTR the heat is added or removed by virtue of the difference between the jacket fluid and the reactor fluid. Often, the heat transfer fluid is pumped through the agitation nozzles that circulate the fluid through the jacket at a high velocity. The coolant flows at a flow rate of q_c and at a feed temperature T_{c_0}. The exit temperature of the coolant fluid is T_c.

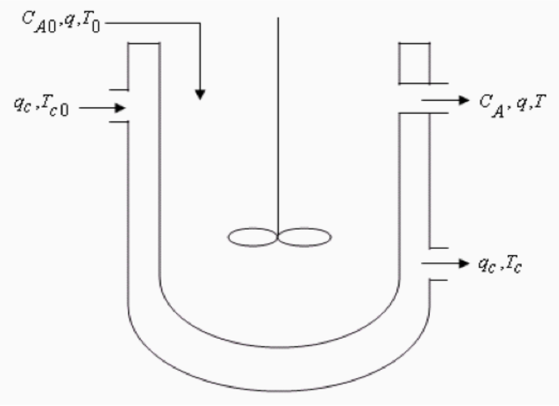

Figure 4.1: Schematic diagram of a Continuous Stirred Tank Reactor

4.2. Modeling

The following assumptions are made to obtain the simplified modeling equations of an ideal CSTR:

- Perfect mixing in the reactor and jacket
- Constant volume reactor and jacket

Carrying out mass and energy balances, and introducing appropriate constitutive equations we formulate the mathematical model for this process:

Reactor mass balance:

$$V\frac{dC_A}{dt} = q(C_{A_0} - C_A) - Vr_A \tag{4.2.1}$$

where, C_A is the product (effluent) concentration of component A in the reactor and $r_A$is the rate of reaction per unit volume. The Arrhenius expression is normally used for the rate of reaction. A first order reaction results in the following expression:

$$r_A = k_0 \exp\left(\frac{-E}{RT}\right) C_A \tag{4.2.2}$$

where, k_0 is the reaction rate constant, E is the activation energy, R is the ideal gas constant and T is the reactor temperature on an absolute scale (R, Rankine or K, Kelvin).

Reactor energy balance:

$$V_\rho C_p \frac{dT}{dt} = q\rho C_p\left(T_0 - T\right) - (-\Delta H) - Vr_A + \rho_C C_{pc}\, q_c \left[1 - \exp\left(\frac{-hA}{\rho_C C_{pc}\, q_c}\right)\right]\left(T_{C_0} - T\right) \tag{4.2.3}$$

where, $(-\Delta H)$ is the heat of reaction, hA is the heat transfer coefficient, T_0 is the feed temperature and T_{C_0} is the inlet coolant temperature. The mass balance and energy balance equations of the CSTR are obtained as follows:

$$\frac{dC_A}{dt} = \frac{q}{V}(C_{A_0} - C_A) - k_0 C_A \exp\left(\frac{-E}{RT}\right) \tag{4.2.4}$$

$$\frac{dT}{dt} = \frac{q}{V}\left(T_0 - T\right) - \left(-\frac{\Delta H}{\rho C_p}\right) k_0 C_A \exp\left(\frac{-E}{RT}\right) + \frac{\rho_C C_{pc}}{\rho C_p V}\, q_c \left[1 - \exp\left(\frac{-hA}{\rho_C C_{pc}\, q_c}\right)\right] (T_{C_0} - T) \tag{4.2.5}$$

The modeling equations of the CSTR contain nonlinear functions of T and C_A. They are coupled and it is not possible to solve one equation independently of the other.

Table 4.1: Steady state operating data [13]

Parameters	*Symbols*	*Values*
Product concentration	C_A	0.0882 mol/l
Reactor temperature	T	441.2 K
Coolant flow rate	q_c	100 l/min
Feed flow rate	q	100 l/min
Feed concentration	C_{A0}	1 mol/l
Feed temperature	T_0	350 K
Inlet coolant temperature	T_{c0}	350 K
CSTR volume	V	100 l
Heat transfer term	hA	7x10^{5} cal/(min K)
Reaction rate constant	k_0	7.2x10^{10} min^{-1}
Activation energy term	$\frac{E}{A}$	1x10^{4} K
Heat of reaction	$-\Delta H$	-2x10^{5} cal/mol
Liquid densities	ρ, ρ_C	1x10^{3} g/l
Specific heat	C_p, C_{pc}	1 cal/(g K)

4.3. Filtering

Schaffner and Zeitz have suggested the following model of an exothermic continuous stirred tank reactor (CSTR)[13] to be considered for nonlinear estimation problems. The model is a standard CSTR model whose parameters were tuned in such a way that it is prone to autonomous periodic oscillations, which usually are undesired and often pose a safety risk. We attempt to design a monitoring system for such a reactor that may serve as a basis for a stabilizing feedback control scheme. It still uses the same parameters as discussed before. The state-space model is going to be as follows:

$$\frac{dx_1}{dt} = -a_1x_1 + b_1r(x_1,x_2)$$

$$\frac{dx_2}{dt} = -a_2x_2 + b_2r(x_1,x_2) + gu$$

$$y(t_k) = x_2(t_k) + v_k \qquad t_k = 0.1k\,, k = 1,2,\ldots \tag{4.3.1}$$

In the equations, x_1 denotes the conversion of the chemical reaction ($0 \le x_1 \le 1$), x_2 is a scaled reactor temperature, u is a scaled cooling temperature. For the reaction rate $r(x_1,x_2)$, the rate expression

$$r(x_1,x_2) = (1 - x_1)\exp\,(-\tfrac{E}{1+x_2}) \tag{4.3.2}$$

is assumed. The variable v_k denotes measurement noise with mean value 0 and variance R.

4.3.1. Extended Kalman Filter

The predictor step of the Extended Kalman Filter for the reactor model is going to be as follows:

Prediction of system states:

$$\frac{d\hat{x}_1}{dt} = -a_1\hat{x}_1 + b_1 r(\hat{x}_1, \hat{x}_2)\,, \quad t_{k-1} \leq t \leq t_k$$

$$\frac{d\hat{x}_2}{dt} = -a_2\hat{x}_2 + b_2 r(\hat{x}_1, \hat{x}_2) + gu$$

Initial condition:

$$\hat{x}_{k-1} = \begin{pmatrix} \hat{x}_1(t_{k-1}) \\ \hat{x}_2(t_{k-1}) \end{pmatrix}$$

Final predicted value at t_k:

$$\hat{x}^{-}{}_{k} = \begin{pmatrix} \hat{x}_1(t_k) \\ \hat{x}_2(t_k) \end{pmatrix}$$

Error Propagation:

$$\frac{dP}{dt} = JP + PJ^T\,, t_{k-1} \leq t \leq t_k$$

Where:

$$J = \begin{bmatrix} -a_1 - b_1 \exp\left(-\frac{E}{1+\hat{x}_2}\right) & -(1-\hat{x}_1) b_1 \exp\left(-\frac{E}{1+\hat{x}_2}\right)\left(\frac{E}{1+\hat{x}_2}\right) \\ -b_2 \exp\left(-\frac{E}{1+\hat{x}_2}\right) & a_2 - b_2 \exp\left(-\frac{E}{1+\hat{x}_2}\right) \end{bmatrix}$$

is a positive semi-definite Jacobean matrix.

The initial conditions read: $P(t_{k-1}) = P_{k-1}$

The covariance matrix of the predicted states becomes: $P_k^- = P(t_k)$

The corrector step of the Extended Kalman Filter for the reactor model is formulated as follows:

$$P_k^- = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

Covariance and cross covariance matrices for calculating the filter gain:

$$P_{xy} = P_k^- (\frac{\partial h}{\partial x}) = P_k^- \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_{12} \\ P_{22} \end{bmatrix}$$

$$P_{yy} = \frac{\partial h}{\partial x} P_k^- \left(\frac{\partial h}{\partial x}\right) + R = P_{22} + R$$

Kalman gain is:

$$K_k = \frac{P_{xy}}{P_{yy}} = \begin{bmatrix} \dfrac{P_{12}}{P_{22} + R} \\ \dfrac{P_{22}}{P_{22} + R} \end{bmatrix}$$

Updated estimates are:

$$x_k = x_k^- + K_k(y - \hat{x}_2(t_k))$$

The updated error covariance is:

$$P_k = \begin{bmatrix} P_{11} - \dfrac{{P_{12}}^2}{P_{22} + R} & P_{12} - \dfrac{P_{12}P_{22}}{P_{22} + R} \\ P_{12} - \dfrac{P_{12}P_{22}}{P_{22} + R} & P_{22} - \dfrac{{P_{22}}^2}{P_{22} + R} \end{bmatrix}$$

It can easily be shown that given P_k^- is positive semi-definite, P_k is also going to be positive semi definite.

4.3.1. Unscented Kalman Filter

For UKF, we have to first calculate the positions of the sigma points for the reactor model.

First thing to do here is to calculate the square-root of the covariance matrix:

$$P_{k-1} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

$$\sqrt{P_{k-1}} = \begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix}$$

$$s_{11} = \sqrt{P_{11}}$$

$$s_{12} = \frac{P_{12}}{\sqrt{P_{11}}}$$

$$s_{22} = \sqrt{P_{22} - \frac{{P_{12}}^2}{P_{11}}}$$

Now the positions of the sigma points are obtained as:

$$X_{k-1,0} = \hat{x}_{k-1}$$

$$X_{k-1,1} = \hat{x}_{k-1} + \sqrt{2+\kappa}\begin{bmatrix} s_{11} \\ s_{12} \end{bmatrix}$$

$$X_{k-1,2} = \hat{x}_{k-1} + \sqrt{2+\kappa}\begin{bmatrix} 0 \\ s_{22} \end{bmatrix}$$

$$X_{k-1,3} = \hat{x}_{k-1} - \sqrt{2+\kappa}\begin{bmatrix} s_{11} \\ s_{12} \end{bmatrix}$$

$$X_{k-1,4} = \hat{x}_{k-1} - \sqrt{2+\kappa}\begin{bmatrix} 0 \\ s_{22} \end{bmatrix}$$

Prediction of the states for the sigma points are obtained as:

$$\frac{d{\hat{x}_1}^{(i)}}{dt} = -a_1{\hat{x}_1}^{(i)} + b_1 r\left({\hat{x}_1}^{(i)}, {\hat{x}_2}^{(i)}\right), \quad t_{k-1} \le t \le t_k$$

$$\frac{d{\hat{x}_2}^{(i)}}{dt} = -a_2{\hat{x}_2}^{(i)} + b_2 r\left({\hat{x}_1}^{(i)}, {\hat{x}_2}^{(i)}\right) + gu$$

Where the initial condition is given as:

$$\hat{X}_{k-1,i} = \begin{pmatrix} \hat{x}_1{}^{(i)}(t_{k-1}) \\ \hat{x}_2{}^{(i)}(t_{k-1}) \end{pmatrix}$$

Final predicted value at t_k:

$$\hat{X}^-{}_{k,i} = \begin{pmatrix} \hat{x}_1{}^{(i)}(t_k) \\ \hat{x}_2{}^{(i)}(t_k) \end{pmatrix}$$

Predicted expectation of states is:

$$\hat{x}^-{}_k = \sum_{i=0}^{4} W_i \, \hat{X}^-{}_{k,i}$$

$$= \begin{bmatrix} \frac{\kappa}{2+\kappa}\hat{x}_1{}^{(0)}(t_k) + \frac{1}{2(2+\kappa)}(\hat{x}_1{}^{(1)}(t_k) + \hat{x}_1{}^{(2)}(t_k) + \hat{x}_1{}^{(3)}(t_k) + \hat{x}_1{}^{(4)}(t_k)) \\ \frac{\kappa}{2+\kappa}\hat{x}_2{}^{(0)}(t_k) + \frac{1}{2(2+\kappa)}(\hat{x}_2{}^{(1)}(t_k) + \hat{x}_2{}^{(2)}(t_k) + \hat{x}_2{}^{(3)}(t_k) + \hat{x}_2{}^{(4)}(t_k)) \end{bmatrix}$$

Predicted measurement value is:

$$\hat{y}^-{}_k = \sum_{i=0}^{4} W_i \, [\hat{X}^-{}_{k,i}]_2$$

$$= \frac{\kappa}{2+\kappa}\hat{x}_2{}^{(0)}(t_k) + \frac{1}{2(2+\kappa)}(\hat{x}_2{}^{(1)}(t_k) + \hat{x}_2{}^{(2)}(t_k) + \hat{x}_2{}^{(3)}(t_k) + \hat{x}_2{}^{(4)}(t_k))$$

Cross covariance matrix is:

$$P_{xy} = \sum_{i=0}^{4} W_i \, [X_{k,_i}^- - \hat{x}_k^-][[\hat{X}^-{}_{k,i}]_2 - \hat{y}_k]^T$$

Covariance matrix is:

$$P_{xy} = R + \sum_{i=0}^{4} W_i \, [[\hat{X}^-{}_{k,i}]_2 - \hat{y}_k][[\hat{X}^-{}_{k,i}]_2 - \hat{y}_k]^T$$

$$P^-{}_k = \sum_{i=0}^{4} W_i \, [X_{k,_i}^- - \hat{x}_k^-][X_{k,_i}^- - \hat{x}_k^-]^T$$

Kalman gain is:

$$K_k = \frac{P_{xy}}{P_{yy}}$$

Updated estimates are:

$$x_k = x_k^- + K_k(y - \hat{x}_2(t_k))$$

The updated error covariance is:

$$P_k = P_k^- - P_{yy} K_k K_k^T$$

CHAPTER 5: COMPARISON OF SIMULATION AND RESULTS USING EKF AND UKF FILTERING METHODS

Matlab is used for simulating the system and for the filter design in this research. There are a number of reasons for this selection:

- Controls and Electrical engineering community favors it.

- The major reason for using Matlab is its prototyping capability. So implementation is fast and effective and so more effort can be paid to the filter design.

- It provides an extensive library of Linear Algebra. The Kalman Filter relies heavily on Matrix math, thus, it makes Matlab a convenient environment.

- Matlab provides excellent plotting and graphing tools.

Matlab does have certain inefficiencies, which makes it very slow when compared to other programming languages like C and Java. However, all the filters are coded using Matlab, thus it provides a common platform for comparing different filter designs. One can switch to a Java implementation when dealing with a real-time implementation of the Filter for efficient and fast operation.

The Simulink models used for the simulations and other verification schemes can be used to recreate any of the results and plots documented in the sections below.

5.1. Simulation

The implementation of the model was carried out using Matlab's Simulink. Below are the demonstrations:

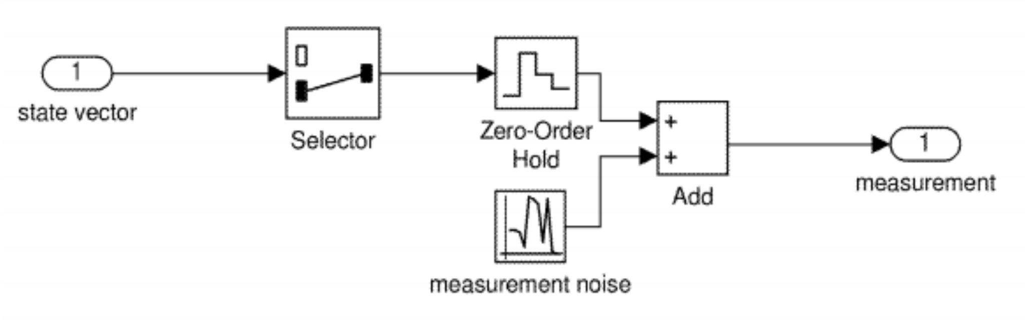

Figure 5.1: Model of the time discrete noisy sensor in Simulink

Our measurement noise is going to have a mean of 0 and a variance of $R = 10^{-4}$.

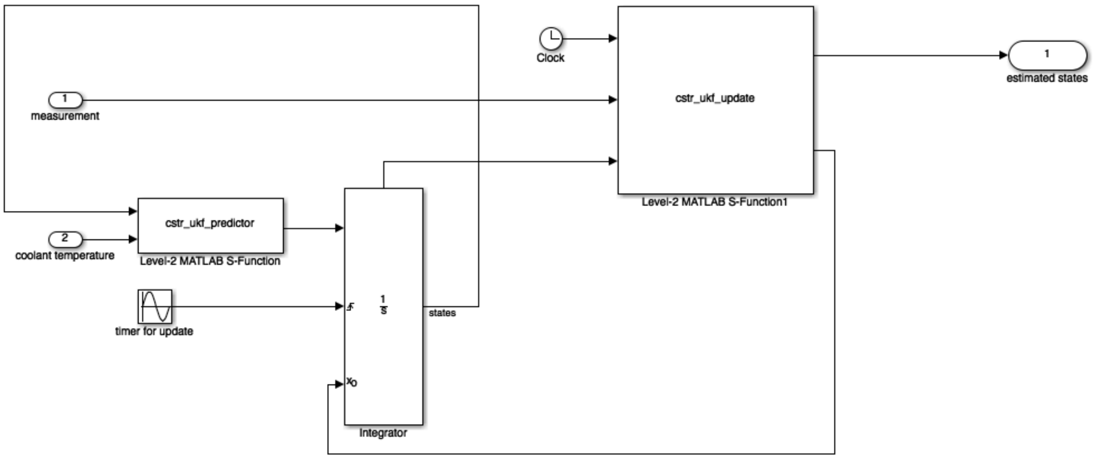

Figure 5.2: Model of the Unscented Kalman Filter

In the above model the cstr_ukf_predictor and cstr_ukf_update blocks are two separate m-files written in Matlab and then implemented in Simulink as s-functions. They perform prediction and measurement update respectively. The update block for small simulation times has to transmit the initial guesses for the states $\widehat{x_k}$ and for the error covariance matrix P_0 to the initial condition port of the integrator. For larger simulation time steps the update block has to transmit the updated states $\widehat{x_k}$ and P_k to the initial condition port of the integrator. In the Unscented Kalman Filter (UKF) case, we use the Matlab function *chol* to compute $\sqrt{P_k}$.

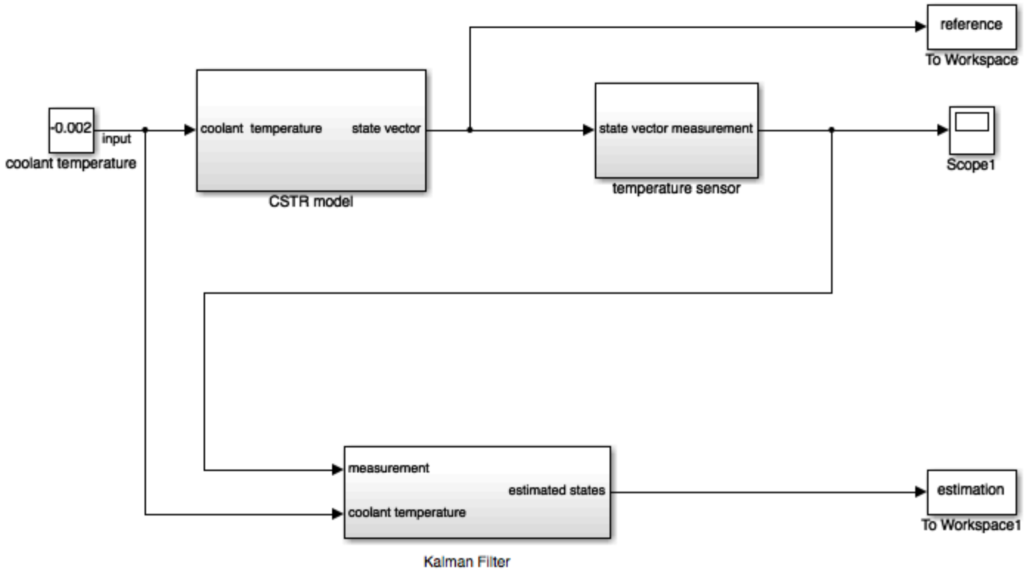

Figure 5.3: Complete Implementation of Kalman Filter in Simulink(KF)

The following values are used for the model parameters discussed in the previous chapter: $a_1 = 0.2674$, $a_2 = 1.815$, $B = 0.4686$, $k_1 = 0.139$, $g = 1.5476$, $E = 34.2583$, $u = -0.002$, the following parameters are explained in more depth:

Discretization Time Interval: δt is the time step we use for our numerical integration method in seconds. It is set as follows:

$$\delta t = 0.1 \text{ s}$$

Process Disturbance Variance Q: We have process disturbance only in our conversion rate and not for any other state variable. Thus it is a scalar and it takes on the following value:

$$Q = 0.0025$$

Measurement Noise R: is the variance of the measurement noise included only in our temperature state given as:

$$R = 10^{-4}$$

Initial Error Covariance P_0: is the value given by $E[(\hat{x}_0 - X_0)(\hat{x}_0 - X_0)^T]$. Here $\hat{x}_0$ is the initial best guess of the state vector before beginning with the filtering operation. X_0 is the true state vector (used for our simulations). Since we neither have knowledge of the probability distribution function (pdf) of the state vector X nor access to the true state vector X_0, we set this value arbitrarily. Trial and error methods are used to fine tune this value to increase the accuracy of the filter.

It should be noted that initial estimate of the state vector can be set to a completely random value and allow the filter to catch up to the true state at the expense of a certain number of iterations of the simulation. Some approaches even assume that they have exact knowledge of the initial conditions of the state, and set $\hat{x}_0 = X_0$ and thus making $P_0 = 0_{n\times n}$. The approach used in this thesis sets $\hat{x}_0$ to a value near the true state (by including some random noise over the value of the true state vector).

κ is set to 2 for the UKF.

5.2. Results

In testing out these filters, some scenarios have been proposed and in each case one or more parameters change. These includes the noise measurement variance and mean, initial conditions

5.2.1. Case One

This is carried out using an off guess for the initial condition. As it can be noticed in figure (5.4), EKF actually does have a convergence problem and never seems to be able to pick up from the initial conditions that weren't favorable to it whereas UKF carries out the task with impressive accuracy. The convergence problem of the EKF is shown in figure (5.6) on a longer run too.

5.2.2. Case Two

The initial condition has set to $\hat{x}_0 = X_0$ and everything else remains unchanged. As can be seen in figures 5.7 and 5.8, since EKF manages to converge faster than the UKF in this case, the overall MSE is actually less for EKF and time wise, there's a fine difference between these two, EKF outperforming UKF.

5.2.3. Case Three

In this scenario the initial condition is determined as described before (by including some random noise over the value of the true state vector). The reason behind this is that it actually gives us the benefit of averaging over time for this scenario (Discussed later). More important, variance level in measurement noise is increased enormously in this case. ($R = 10^{-2}$) Although both filters seem to have problems in terms of convergence, UKF does manage to show better result in terms of accuracy. Please do note that run time has been increased five times the original run time.

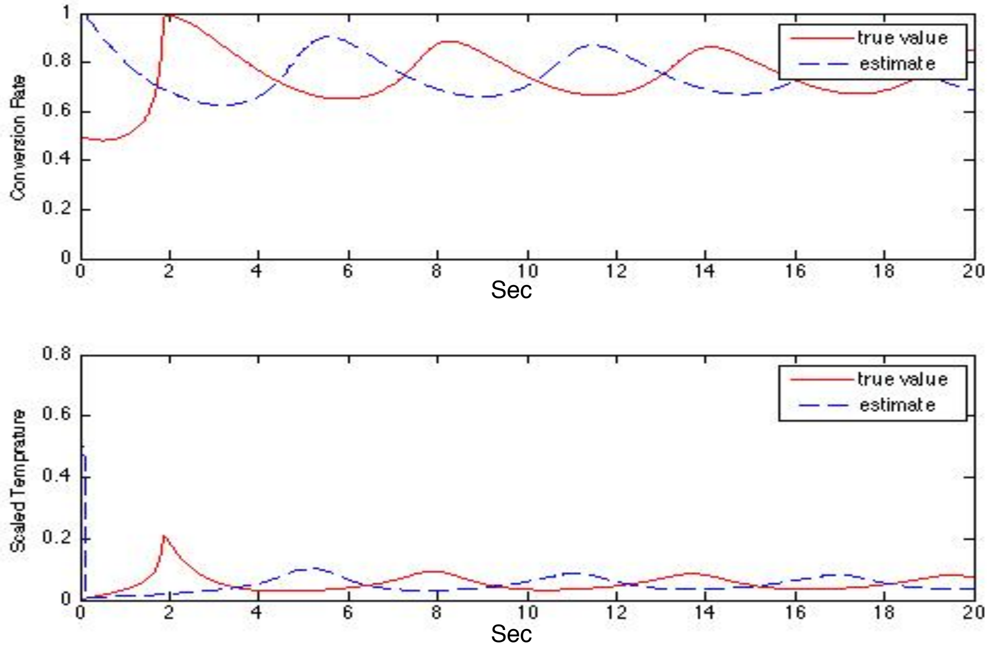

Figure 5.4: EKF plot for true v/s estimated of case one

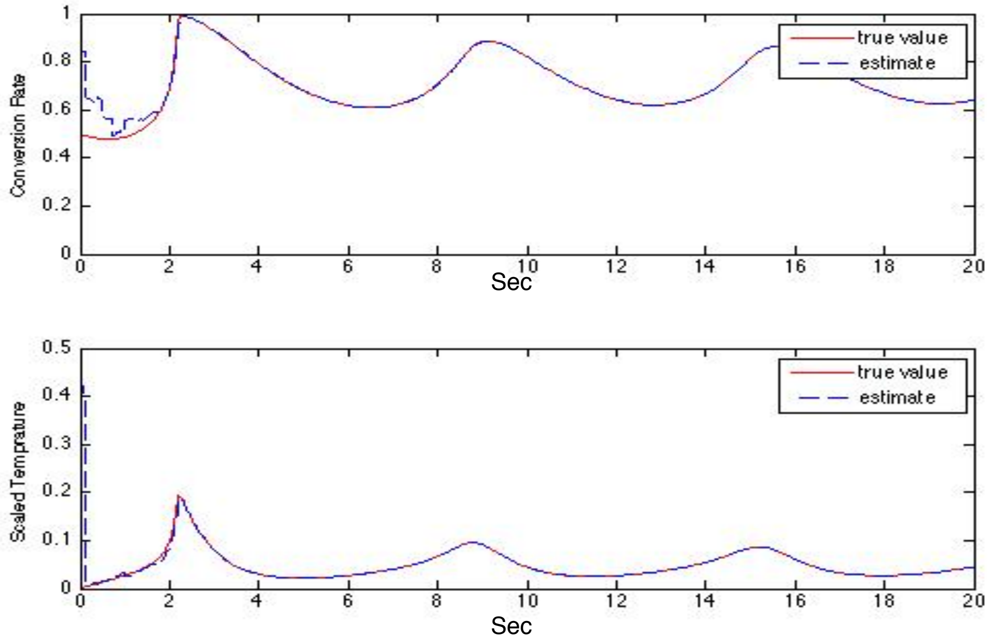

Figure 5.5: UKF plot for true v/s estimated of case one

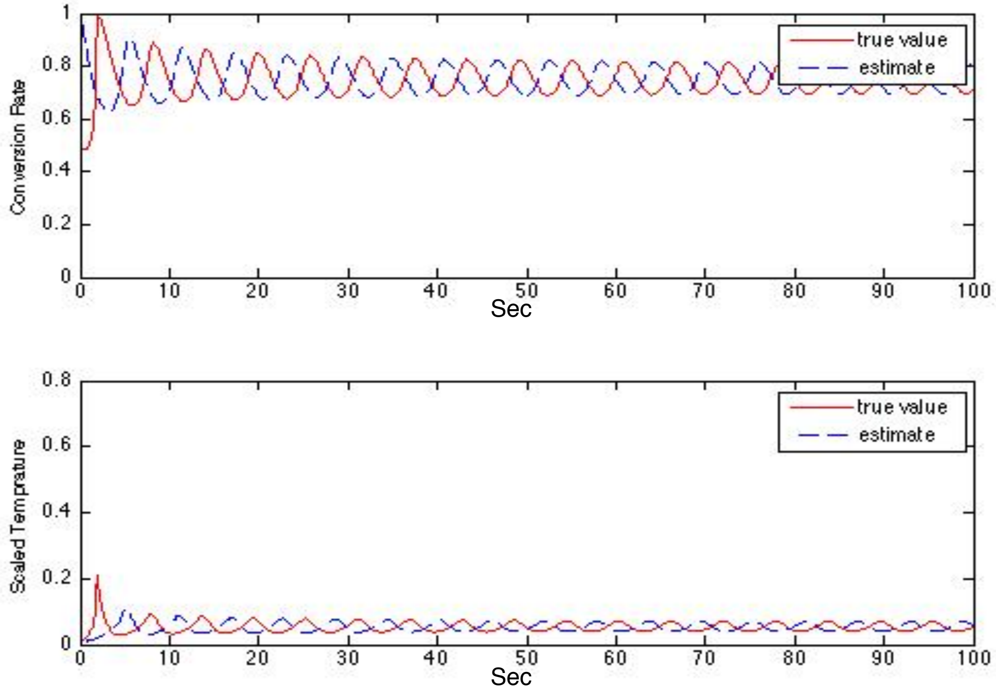

Figure 5.6: Convergence problem of EKF for case one

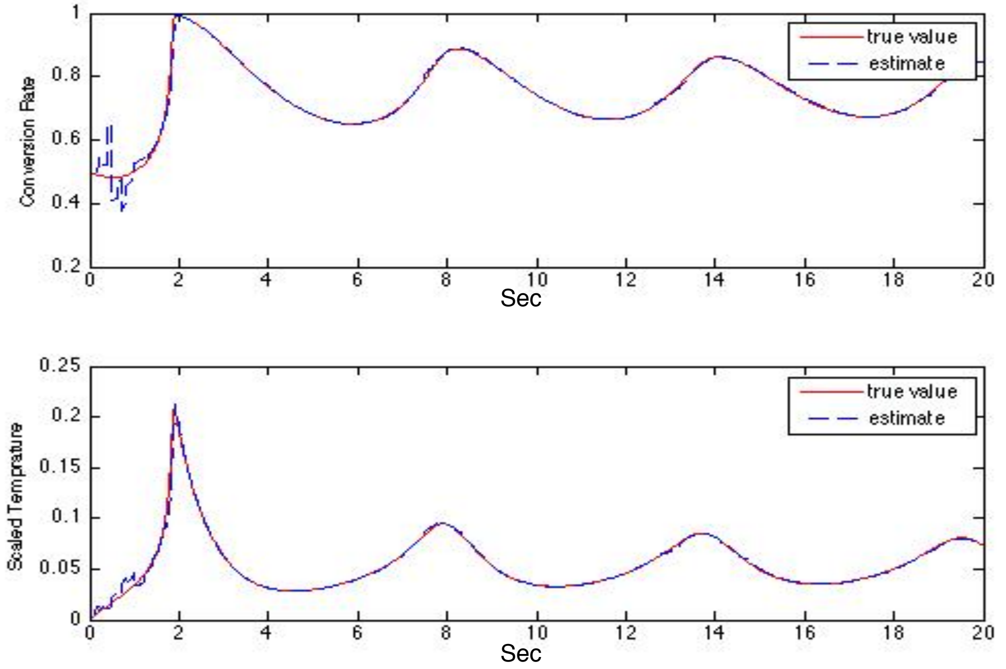

Figure 5.7: EKF plot for true v/s estimated of case two

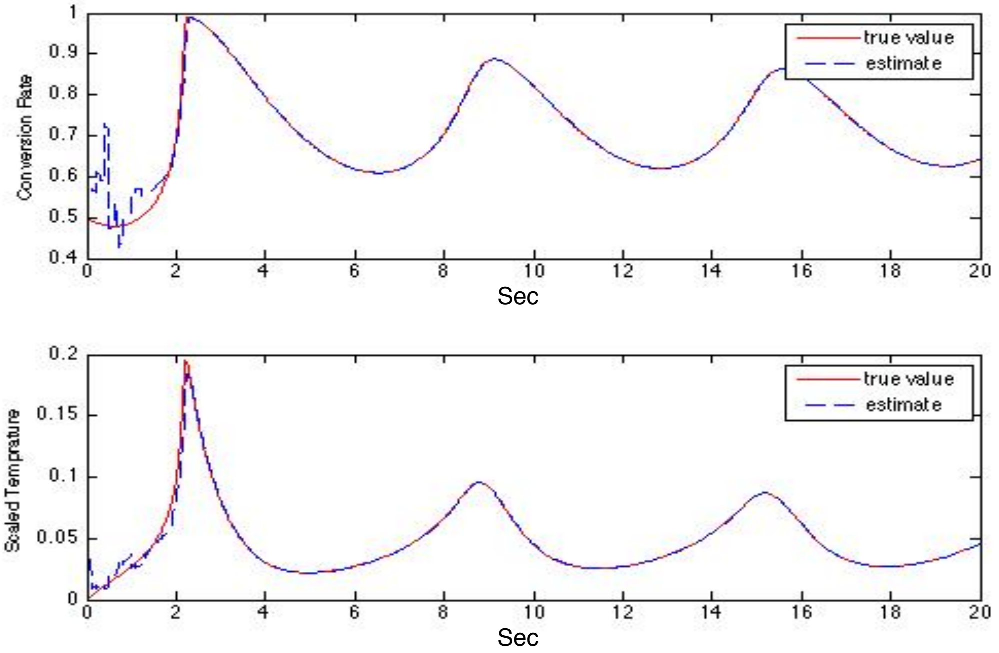

Figure 5.8: UKF plot for true v/s estimated of case two

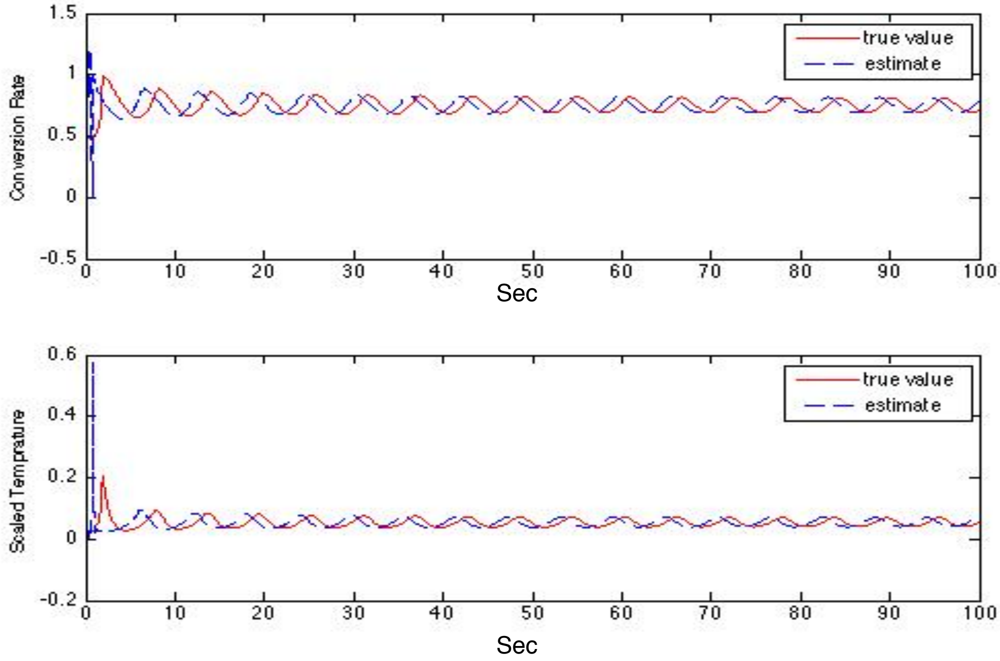

Figure 5.9: EKF plot for true v/s estimated of case three

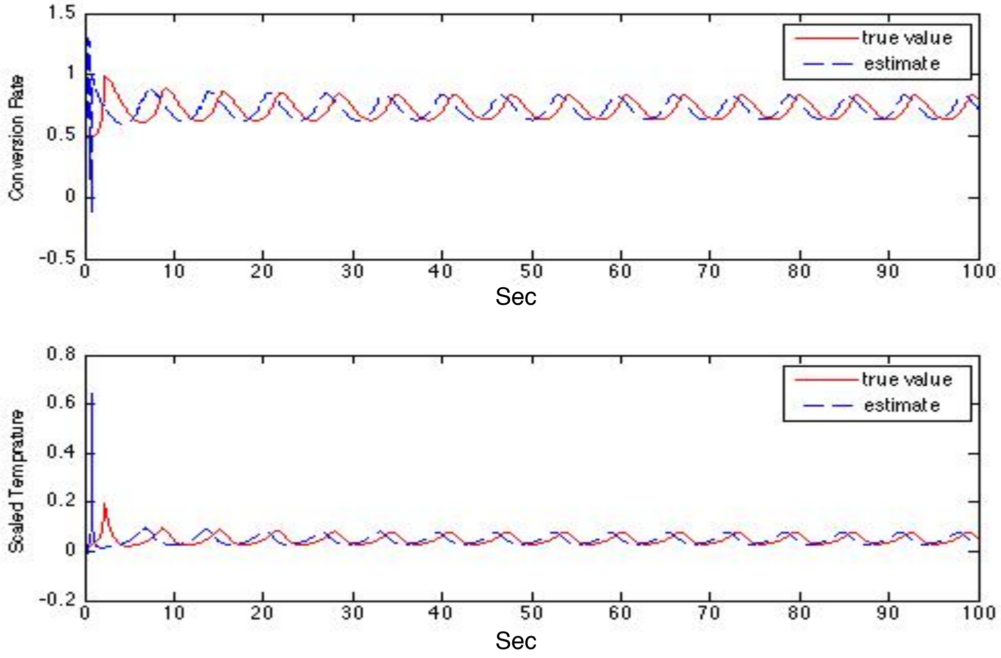

Figure 5.10: UKF plot for true v/s estimated of case three

5.3. Accuracy and Computational Cost

The accuracy of the filters is measured by calculating their Mean Squared Error (MSE) over the entire length of the simulation. Matlab's 'Run and Time' feature gives the necessary data to facilitate a time comparison between filters.

The tables below give us a comparison of the filters with respect to accuracy and speed. All times are in seconds.

Table 5.1: Comparison table for case one

Filters	MSE(x_1)	MSE(x_2)	Run Time
EKF	N/A	N/A	1.016
UKF	0.0190	0.0143	5.708

Table 5.2: Comparison table for scenario two

Filters	MSE(x_1)	MSE(x_2)	Run Time
EKF	3.3947e-04	1.5376e-05	0.908
UKF	8.7350e-04	5.0664e-05	8.901

Table 5.3: Comparison table for scenario three

Filters	MSE(x_1)	MSE(x_2)	Run Time
EKF	0.0373	0.0200	4.671
UKF	0.0190	0.0111	18.987

One possible explanation to the above results would be the random noise induced into the system via process disturbance and measurement noise. Since process disturbance and measurement noise vary from run to run, it is always considered a good practice to average any Means Square Error (MSE) over a good number of runs so that we eliminate the dependence of the results on randomness. For the case at hand, we will compare just the EKF and the UKF over a thousand runs as this would allow us a more in-depth analysis of what the true MSE of each of them in the long run might be. To play it safe, the MSEs of each of the filters were averaged over 1000 different runs (which means thousand different sets of random noise values). This is applied to case three:

Table 5.4: Comparison table for average performance

Filters	MSE(x_1)	MSE(x_2)	Run Time
EKF	0.0384	0.0194	4.513
UKF	0.0183	0.0109	17.897

5.4. Testing the Convergence of the Unscented Transform

The Unscented Transform should be convergent for our filter to maintain stability and not loose track of the state being estimated. There exists a very simple scheme to perform this verification. It is as follows:

1. With the initial Data given generate set of sigma points.
2. We then pass these points through an identity transform, which basically means we do not touch these points at all.
3. Next we generate the mean and covariance of the transformed sigma points respectively.
4. Since we subjected our sigma points through an identity transform, the state of the system should remain the same. Thus, the mean and covariance generated in the step above should match with the ones we started off with in the first place. If this does not hold then the Unscented Transform will be divergent and the entire procedure will fail.

A code snippet was written in Matlab for executing this verification scheme. Before the identity transform the mean and covariance were:

$$\hat{x} = \begin{bmatrix} 0.1761 \\ 0.1001 \end{bmatrix}, P = \begin{bmatrix} 0.0600 & 0 \\ 0 & 0.0600 \end{bmatrix}$$

And after the identity transform the calculated statistics were:

$$\hat{x}_{identity} = \begin{bmatrix} 0.1761 \\ 0.1001 \end{bmatrix}, P_{identity} = \begin{bmatrix} 0.0600 & -0.0000 \\ -0.0000 & 0.0600 \end{bmatrix}$$

As one can see the mean remains unaltered. As for the covariance, the misplaced – signs arise due to round off and other computational errors. But these too are after the fourth decimal place, thus it is safe to conclude that the transform is actually convergent and working properly.

CHAPTER 6: CONCLUSION AND FUTURE WORK

Recently there has been a great deal of interest in the specific problem of implementing the UKF for a tank reactor and nearly in all of those cases they come to the conclusion than EKF still shows better results and accuracy. Both [16] and [17] use the same model with slightly different parameter values and both come to the conclusion that indeed EKF is the preferred filter for this specific problem. This thesis actually shows better results for the implementation of UKF as the accuracy level is either better or near identical to the EKF. While again facing the dilemma of the underperforming UKF, several papres suggest that implementing the scaled UKF and fine tuning might be the key to solving this particular problem.

6.1. Conclusion

The current claim in the Controls Community is that the UKF outperforms the EKF for non-linear applications and does just as well for linear applications. While this is true for the various non-linear systems that the UKF has been applied to, the results specific to the case of the reactor tank, while impressive on its own, can not entirely confirm this for all cases. This might be a specific case of the peculiar system dynamics, which results into this outcome.

It should be noted that the most effective method for this case still remains to be the Extended Kalman Filter when the filter is converging in an effective manner. In cases, when the noise level is too high, the Unscented Kalman Filter should be employed if the computational costs are acceptable. The final conclusive finding of this research from comparing the results would be that for any given problem, there remains no definite answer and many factors should be considered and entertained before choosing a good filtering method.

6.2. Future Work

The following are some suggested extensions based on the work and analysis done in this research:

- Theoretical Investigation of why the UKF performs below expectation for this problem.
- A mathematical proof of concept as to why the UKF does not give higher accuracy than the EKF in most cases.
- Finding more cases with nonlinearities for which, EKF has a convergence problem.
- A possible change to UKF tailored for the current system dynamics filtering.

REFERENCES

[1] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Fluids Engineering, 82(1), 35-45.

[2] Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter.

[3] Sorenson, H. W. (1970). Least-squares estimation: from Gauss to Kalman. Spectrum, IEEE, 7(7), 63-68.

[4] Brown, R. G., & Hwang, P. Y. (1997). Introduction to random signals and applied Kalman filtering. NY John Wiley and Sons.

[5] Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of Nonlinear Filtering, 12, 656-704.

[6] Gelb, A. (Ed.). (1974). Applied optimal estimation. MIT press.

[7] Drew Bagnell and Keheng Zhang. Statistical techniques in robotics: Lecture on kalman filtering(part 2). CS lecture series in Carnegie Mellon University.

[8] Julier, S. J. (2002). The scaled unscented transformation. In American Control Conference, 2002. Proceedings of the 2002 (Vol. 6, pp. 4555-4559). IEEE.

[9] Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. Proceedings of the IEEE, 92(3), 401-422.

[10] Van Der Merwe, R. (2004). Sigma-point Kalman filters for probabilistic inference in dynamic state-space models (Doctoral dissertation, Oregon Health & Science University).

[11] Van Der Merwe, R., & Wan, E. A. (2001). The square-root unscented Kalman filter for state and parameter-estimation. In Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on (Vol. 6, pp. 3461-3464). IEEE.

[12] Wu, Y., Hu, D., Wu, M., & Hu, X. (2005, June). Unscented Kalman filtering for additive noise case: augmented vs. non-augmented. In American Control Conference, 2005. Proceedings of the 2005 (pp. 4051-4055). IEEE.

[13] Schaffner, J., & Zeitz, M. (1999). Variants of nonlinear normal form observer design. In New Directions in nonlinear observer design (pp. 161-180). Springer London.

[14] Willis, M. J. (2000). Continuous stirred tank reactor models. Dept. of Chemical and Process Engineering, University of Newcastle.

[15] Simon, D. (2006). Optimal state estimation: Kalman, H infinity, and nonlinear

approaches. John Wiley & Sons.

[16] Geetha, M., Kumar, P. A., & Jerome, J. (2014). Comparative Assessment of a Chemical Reactor Using Extended Kalman Filter and Unscented Kalman Filter. Procedia Technology, 14, 75-84.

[17] Gajare, V., Koturwar, R., & Kore, R. (2013). State Estimation of Non-linear Process Using Extended Kalman Filter. International Journal of Applied Engineering Research, 8(19).

[18] Julier, S. J., & Uhlmann, J. K. (1997, April). A new extension of the Kalman filter to nonlinear systems. In Int. symp. aerospace/defense sensing, simul. and controls (Vol. 3, No. 26, pp. 3-2).