

ECEN 662 Group Report: Overview of the Kalman Filter and its Extensions

Kiran Vishnu, Daniel Whitten, Tim Woodbury

Abstract

The Kalman filter has a storied history in the estimation of correlated-in-time processes. In Kalman's original work, the filter estimate was shown to be optimal in a minimum mean square error sense for Gaussian systems. It was further shown that the optimal estimate for a sequence of measurements could be realized by operating the filter sequentially, treating only one measurement at a time. In the present work, the linear Kalman filter is reviewed. A simple numerical example is presented to demonstrate the implementation of the filter. Subsequently, a discussion of two extensions of the Kalman filter to nonlinear systems is given. These methods are the Extended Kalman Filter, in which the nonlinear system is replaced by a two-term Taylor series expansion, and the Unscented Kalman Filter, in which the estimate distribution is approximated by deterministically defined particles. The equations governing the nonlinear approximations are given, and the two filters are discussed at a high level.

I. INTRODUCTION

The Kalman filter's origins trace back to 1960 and Kalman's distinguished work [3]. The contribution of the paper is difficult to overstate; few methods in engineering have received so much attention, or been applied to so many diverse problems. Kalman showed that the optimal filter for a linear system with Gaussian forcing could be expressed as a sequential linear function of a prior and measurements. As a consequence, the optimal estimate at time k given all the measurements at $t = 0, 1, \dots, k - 1$ is equivalent to the optimal estimate at time k given the measurement at $k - 1$ and the optimal estimate at time $k - 1$. This has profound implications in terms of data reduction and, in an era of relatively limited computer memory, enabled many online estimation problems to be solved.

Many practical systems of interest are nonlinear, state-dependent processes. It has long been desired to extend the utility and simplicity of the Kalman filter to such systems. Unfortunately,

the optimal estimate for nonlinear systems in general is intractable[4]. Two popular approaches to approximate the Kalman filter for nonlinear systems are the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). The EKF is a classic approach in which the Kalman Filter is linearized about a prior state at each cycle of prediction and update. The EKF has been widely used and has many attractive qualities, but it is known to perform poorly for certain types of problems[5]. The UKF replaces the prior state and covariance by a particle approximation. However, unlike in other particle filters, the UKF points are deterministic functions of the prior, and the number of points is of the same order as the size of the state vector. The result is a nonlinear estimate that better approximates the true system statistics than the EKF, at a typically higher computational cost.

In this document, a short derivation of the Kalman Filter from a dynamic systems perspective is given. A brief simulation example is shown, demonstrating the simplicity of the filter. Subsequently, a discussion of the EKF and UKF is presented.

II. THE LINEAR DISCRETE KALMAN FILTER

The original Kalman filter considers a linear discrete dynamic model and a linear discrete measurement model.

A. System Models

The discrete dynamic model relates the state at some time k to the state at the current time $k + 1$. This is also known as the process model.

$$\mathbf{x}(k + 1) = \Phi(k)\mathbf{x}(k) + \Gamma(k)\mathbf{u}(k) + \Upsilon(k)\mathbf{w}(k) \quad (1)$$

$\mathbf{x}(k)$ is some state of interest at time k . $\mathbf{u}(k)$ is a deterministic input vector. $\mathbf{w}(k)$ is zero-mean Gaussian white-noise process with

$$E[\mathbf{w}(k)\mathbf{w}(k)] = \mathbf{Q}(k) \quad (2)$$

$$E[\mathbf{w}(k)\mathbf{w}(j)] = \mathbf{0} \quad j \neq k \quad (3)$$

It is useful to find the first and second moments of these models.

$$E[\mathbf{x}(k + 1)] = \hat{\mathbf{x}}^-(k + 1) = \Phi(k)\hat{\mathbf{x}}(k) + \Gamma(k)\mathbf{u}(k) \quad (4)$$

$$Cov[\mathbf{x}(k+1), \mathbf{x}(k+1)] = P_{xx}^-(k+1) = \Phi(k)P_{xx}(k)\Phi(k)^T + \Upsilon(k)Q(k)\Upsilon(k)^T \quad (5)$$

$\hat{\mathbf{x}}^-(k+1)$ and $P_{xx}^-(k+1)$ can be used to predict the state at time $k+1$ given the state at k .

A measurement vector $\tilde{\mathbf{y}}(k)$ is available. $\tilde{\mathbf{y}}(k)$ relates to the state according to

$$\tilde{\mathbf{y}}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \quad (6)$$

$$E[\mathbf{v}(k)\mathbf{v}(k)] = \mathbf{R}(k) \quad (7)$$

A measurement estimate $\hat{\mathbf{y}}(k)$ is found by taking the expectation of this model

$$E[\tilde{\mathbf{y}}(k)] = \hat{\mathbf{y}}(k) = \mathbf{H}(k)\hat{\mathbf{x}}^-(k) \quad (8)$$

It is also useful to define and determine the second moments of this model

$$Cov[\tilde{\mathbf{y}}(k), \tilde{\mathbf{y}}(k)] = P_{yy}(k) = \mathbf{H}(k)P_{xx}^-(k)\mathbf{H}(k)^T + \mathbf{R}(k) \quad (9)$$

$$Cov[\mathbf{x}(k), \tilde{\mathbf{y}}(k)] = P_{xy}(k) = P_{xx}^-(k)\mathbf{H}(k) \quad (10)$$

B. Minimum Mean Square

When a measurement $\tilde{\mathbf{y}}(k)$ becomes available, the process model can be used to predict, or propagate, the state estimate from the previous time to the current time. This is known as the prior estimate $\hat{\mathbf{x}}^-(k)$. It is desirable to update the state estimate $\hat{\mathbf{x}}^-(k)$ using the information about the state provided by the measurement. This updated measurement, $\hat{\mathbf{x}}^+(k)$, is known as the posterior estimate.

The update equations will first be developed by assuming an estimator in the form of a sequential linear estimator with a gain $\mathbf{K}(k)$ and then finding $\mathbf{K}(k)$ to minimize the mean square error (MSE). It will then be shown that if the initial state is Gaussian then this estimator is also the minimum mean square error (MMSE) estimator.

The updated state $\hat{\mathbf{x}}^+(k)$ estimator is assumed to be in the form of a linear sequential estimator.

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)(\tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k)) \quad (11)$$

It is necessary to find $\mathbf{K}(k)$ so that $\hat{\mathbf{x}}^+(k)$ minimizes some performance criteria. The mean square error is a natural choice.

$$MSE = P_{xx}^+(k) = E[(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))^T] \quad (12)$$

By using the previously defined value of $\hat{\mathbf{x}}^-(k)$, the error $(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))$ can be written as

$$\mathbf{x}(k) - \hat{\mathbf{x}}^+(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) + \mathbf{K}(k)\mathbf{v}(k) \quad (13)$$

After expanding out the terms of the square error and finding its expected value

$$\mathbf{P}_{xx}^+(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}_{xx}^-(k)(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))^T + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}(k) \quad (14)$$

To minimize MSE, it is equivalent to minimize the trace of \mathbf{P}_{xx}^+ .

$$J = \text{trace}(\mathbf{P}_{xx}^+) \quad (15)$$

Finding the partial derivative of J with respect to $\mathbf{K}(k)$

$$\frac{\partial J}{\partial \mathbf{K}(k)} = 0 = -2(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T + 2\mathbf{K}(k)\mathbf{R}(k) \quad (16)$$

Solving for $\mathbf{K}(k)$

$$\mathbf{K}(k) = \mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T (\mathbf{H}(k)\mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T + \mathbf{R}(k))^{-1} \quad (17)$$

This choice of $\mathbf{K}(k)$ minimizes the MSE for a linear estimator. That is, the Kalman filter provides the optimal *linear* MMSE estimate.

If the initial state is sampled from a normal distribution, the state remains linear because a Gaussian random variable that undergoes linear transformations remains linear. In this case, $\mathbf{x}(k)$ and $\tilde{\mathbf{y}}(k)$ are jointly Gaussian

The MMSE estimator is known to be

$$\hat{\mathbf{x}}_{MMSE}(k) = E[\mathbf{x}(k)|\tilde{\mathbf{y}}(k)] \quad (18)$$

It is well known that for jointly Gaussian $\mathbf{x}(k)$ and $\tilde{\mathbf{y}}(k)$

$$E[\mathbf{x}(k)|\tilde{\mathbf{y}}(k)] = E[\mathbf{x}(k)] + \mathbf{P}_{xy}(k)\mathbf{P}_{yy}^{-1}(k)(\tilde{\mathbf{y}}(k) - E[\tilde{\mathbf{y}}(k)]) \quad (19)$$

Having previously defined each term of this equation in terms of the process and measurement model, it can be re-written as

$$E[\mathbf{x}(k)|\tilde{\mathbf{y}}(k)] = \hat{\mathbf{x}}^-(k) + \left(\mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T (\mathbf{H}(k)\mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T + \mathbf{R}(k))^{-1} \right) (\tilde{\mathbf{y}}(k) - \mathbf{H}(k)\hat{\mathbf{x}}^-(k)) \quad (20)$$

$$E[\mathbf{x}(k)|\tilde{\mathbf{y}}(k)] = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)(\tilde{\mathbf{y}}(k) - \mathbf{H}(k)\hat{\mathbf{x}}^-(k)) \quad (21)$$

Thus, for a Gaussian state prior, the Kalman filter is the MMSE estimator.

C. Algorithm Summary

The linear discrete Kalman filter can be described and implemented in three distinct stages.

Initialize:

$$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0 \quad (22)$$

$$\mathbf{P}_{xx}(t_0) = \text{Cov}[\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_0] \quad (23)$$

Predict:

$$\hat{\mathbf{x}}^-(k+1) = \Phi(k)\hat{\mathbf{x}}(k) + \Gamma(k)\mathbf{u}(k) \quad (24)$$

$$\mathbf{P}_{xx}^-(k+1) = \Phi(k)\mathbf{P}_{xx}(k)\Phi(k)^T + \Upsilon(k)\mathbf{Q}(k)\Upsilon(k)^T \quad (25)$$

Update:

$$\mathbf{K}(k) = \mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T (\mathbf{H}(k)\mathbf{P}_{xx}^-(k)\mathbf{H}(k)^T + \mathbf{R}(k))^{-1} \quad (26)$$

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)(\tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k)) \quad (27)$$

$$\mathbf{P}_{xx}^+(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}_{xx}^-(k) \quad (28)$$

Using the models:

$$\mathbf{x}(k+1) = \Phi(k)\mathbf{x}(k) + \Gamma(k)\mathbf{u}(k) + \Upsilon(k)\mathbf{w}(k) \quad \mathbf{w}(k) \sim N(\mathbf{0}, \mathbf{Q}(k)) \quad (29)$$

$$\tilde{\mathbf{y}}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \quad \mathbf{v}(k) \sim N(\mathbf{0}, \mathbf{R}(k)) \quad (30)$$

This section has introduced the linear Kalman Filter and justified its optimality in a MMSE sense. The next section presents a numerical implementation of the Kalman Filter.

III. KALMAN FILTER SIMULATION

Consider a simple linear system with known analytical solution: a projectile launched with an initial velocity, falling under the influence of a constant gravity, with no air resistance. The

gravitational acceleration is treated as a deterministic input. The y axis is taken as normal to the ground, leading to the following true dynamics for the position states x, y and velocities v_x, v_y :

$$\dot{x} = v_x \quad (31)$$

$$\dot{v}_x = 0 \quad (32)$$

$$\dot{y} = v_y \quad (33)$$

$$\dot{v}_y = -g \quad (34)$$

$$\mathbf{x}(k) \equiv \begin{bmatrix} x(k) & v_x(k) & y(k) & v_y(k) \end{bmatrix}^T \quad (35)$$

The initial states are chosen such that $v_x = v_y \approx 70.7$ m/s, and the gravitational acceleration is taken as a constant 9.81 m/s². The system is discretized with a simple Euler integration model at $\Delta T = 0.01$ seconds, leading to the following discrete-time process model:

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\Delta T \end{bmatrix} g \quad (36)$$

Note that no explicit process noise is included in the dynamics. However, the simulated states are evaluated using the analytical solution, which leads to a discretization error due to the use of Euler integration. This discretization error is modeled as a zero-mean process with the following covariance:

$$\mathbf{Q}(k) = \text{diag} \begin{bmatrix} \Delta T^{0.25} & \Delta T & \Delta T & \Delta T \end{bmatrix} \quad (37)$$

This covariance was tuned experimentally to produce more consistent filter outputs. The measurement matrix is taken simply to be identity, with linear measurements of the full state. The measurement covariance matrix is diagonal with each measurement having variance 10. Python simulation results are now presented.

For brevity, we show only the filter tracking errors (Fig. 1) and consistency histograms (Fig. 2). The tracking errors are simply the difference between the filter states and true states at each moment in time. Three times the square root of the covariance diagonal elements is plotted as a metric of filter accuracy and consistency. The filter appears to be somewhat conservative, since

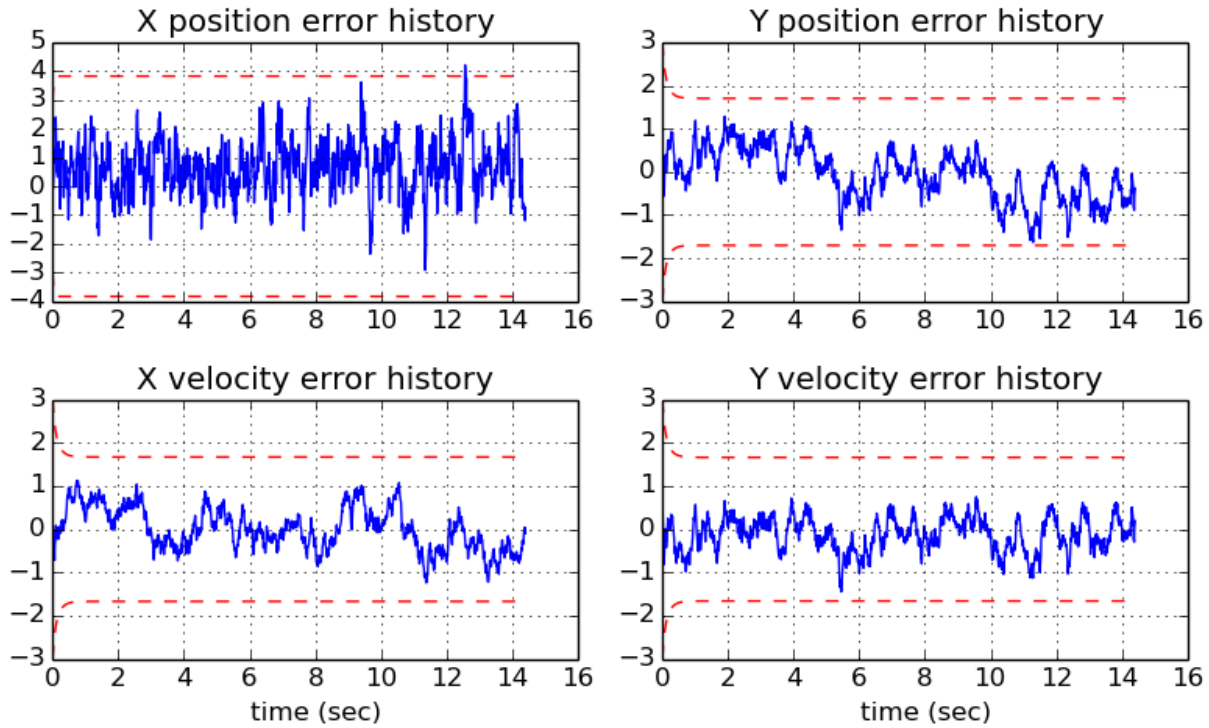


Fig. 1: Kalman filter errors relative to truth. $3\times$ the square root of covariance diagonals plotted in red for reference.

the errors are typically less than the 3σ approximations. However, the x position history shows a bias. No doubt this bias is largely attributable to the use of first-order discretized integration.

For Fig. 2, the filter errors are regularized to states \mathbf{z} having expected mean and covariance 0, 1: $\mathbf{z} \equiv \mathbf{P}^{1/2}(\mathbf{x} - \hat{\mathbf{x}})$. If the filter is consistent, the regularized states \mathbf{z} should be zero-mean and have unit variance. In Fig. 2, histograms of the regularized states are plotted. Note that states z_1, z_2, z_3, z_4 correspond approximately to x position, x velocity, y position, and y velocity. It is clear that the states are approximately Gaussian but with some error. In particular, the z_1 and z_4 states exhibit a noticeable bias and the filter is conservative, because the actual variance is less than 1. The z_2 and z_3 states are more consistent.

Overall, the filter implementation is imperfect, but adequate for the purposes of demonstrating the basics of the Kalman Filter. The following section explores advanced topics in nonlinear sequential estimation.

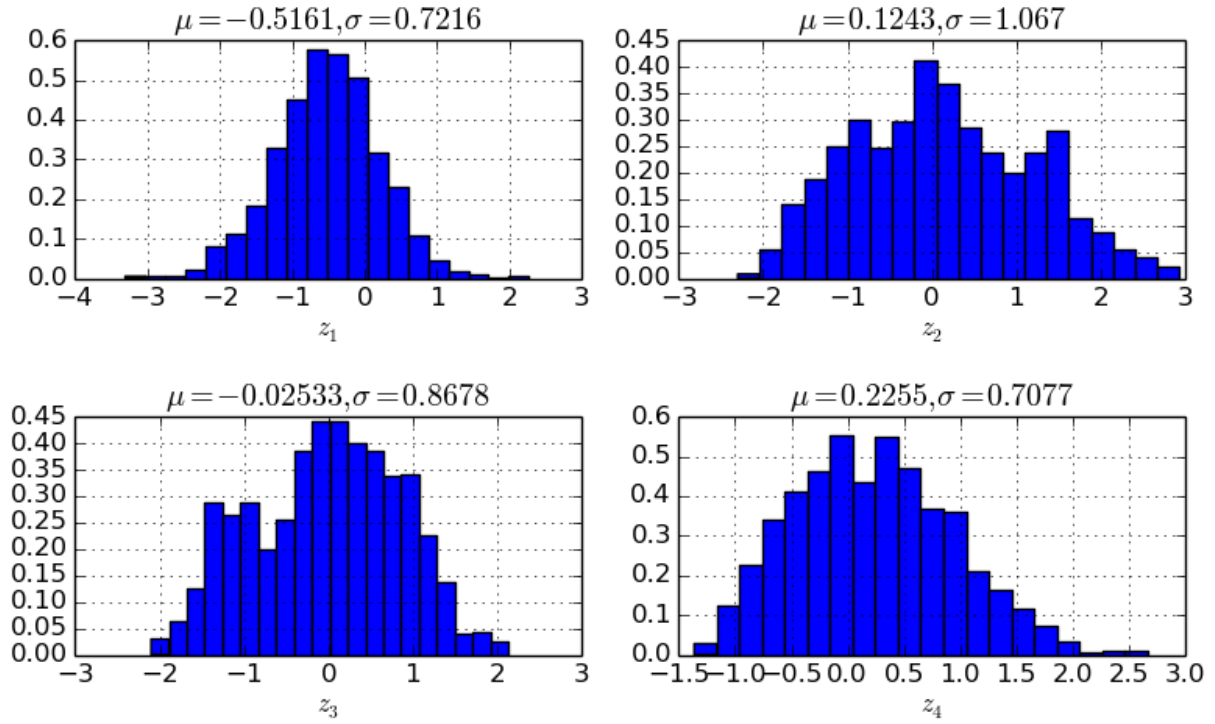


Fig. 2: Histograms of normalized filter states.

IV. ADVANCED TOPICS

In many processes of interest in practical problems, the strict assumptions of a linear system with Gaussian process and measurement noise are not satisfied. For instance, consider the radar tracking problem: radar will usually measure the range and bearing and elevation angles to a target. The measurement noise is not likely to be perfectly Gaussian. One can either write the target dynamic equations in terms of range and angles (in which case the dynamics are nonlinear); or, one may compute 3-dimensional coordinates x, y, z from the raw measurements, in which case the measurement noise undergoes a nonlinear transformation. There is no perfect general transformation that allows the strict assumptions of linear dynamics and linear measurements to be guaranteed. This motivates the need for nonlinear estimators.

Perhaps the most intuitive approach to the nonlinear estimation problem is to linearize the nonlinear system about the prior state, and use the Jacobians of the system in the Kalman covariance equations. This process is the EKF, and is widely used for nonlinear estimation in

many fields including aerospace and process engineering. The linearized approximation, however, may fail in some cases, particularly if an accurate prior for the estimate is not known. This has motivated the development of other techniques. The UKF, sometimes called the Sigma Point Filter, uses a discrete approximation to the prior density. The literature indicates that this approximation is more accurate than the linearization of the EKF [5]. The remainder of this section presents an overview of the EKF and UKF. We begin with a presentation of the EKF.

A. Extended Kalman Filter

This discussion is motivated by Ref. [1]. Consider a generic discrete nonlinear system with discrete nonlinear measurements and discrete controls $\mathbf{u}(k)$, and a Gaussian prior for the estimated state. Let $\mathbf{w}(k)$ and $\mathbf{v}(k)$ indicate zero-mean process noise and measurement noise:

$$\mathbf{x}(k+1) = \phi(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{G}(k)\mathbf{w}(k) \quad (38)$$

$$\mathbf{y}(k) = \mathbf{v}(\mathbf{x}(k)) + \mathbf{v}(k) \quad (39)$$

$$E[\mathbf{x}(0)] = \hat{\mathbf{x}}(0) \quad (40)$$

$$\text{var}[\mathbf{x}(0)] = \mathbf{P}_{\text{xx}}(0) \quad (41)$$

$$E[\mathbf{w}(k)] = 0 \quad (42)$$

$$\text{var}[\mathbf{w}(k)] = \mathbf{Q}(k) \quad (43)$$

$$E[\mathbf{v}(k)] = 0 \quad (44)$$

$$\text{var}[\mathbf{v}(k)] = \mathbf{R}(k) \quad (45)$$

In general, the MMSE for the nonlinear system is intractable [4]. An approximate solution can be obtained by linearizing the nonlinear governing equations, sacrificing the optimality properties of the linear KF. A brief summary of the derivation of the EKF equations is presented. Let $\hat{\mathbf{x}}(k)$ denote the (prior) estimate of the state at time step k and $\mathbf{x}(k)$ denote the unknown true state. The nonlinear functions can be expanded in two terms of a Taylor series approximation as follows:

$$\phi(\mathbf{x}(k), \mathbf{u}(k)) \approx \phi(\hat{\mathbf{x}}(k), \mathbf{u}(k)) + \left. \frac{d\phi(\mathbf{x}, \mathbf{u}(k))}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} (\mathbf{x}(k) - \hat{\mathbf{x}}(k)) \quad (46)$$

$$\mathbf{v}(\mathbf{x}(k)) \approx \mathbf{v}(\hat{\mathbf{x}}(k)) + \left. \frac{d\mathbf{v}(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} (\mathbf{x}(k) - \hat{\mathbf{x}}(k)) \quad (47)$$

$$\Phi(k) \equiv \left. \frac{d\phi(\mathbf{x}, \mathbf{u}(k))}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \quad (48)$$

$$\Upsilon(k) \equiv \left. \frac{d\mathbf{v}(x)}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \quad (49)$$

It is straightforward to show that the EKF equations can be found by modifying the linear Kalman filter equations as follows: (1) Replace the state prediction and measurement expectation terms by the nonlinear equations ϕ and \mathbf{v} ; (2) Replace the state and measurement influence matrices by the Jacobians of the nonlinear equations evaluated at the estimate ($\Phi(k)$ and $\Upsilon(k)$). For convenience to the reader, the derivation of the EKF equations is skipped; merely note that the mean and variance of the Kalman prediction and update steps are taken, and lead to the following results. The EKF prediction equations are as follows:

$$\boxed{\hat{\mathbf{x}}^-(k+1) = \phi(\hat{\mathbf{x}}(k), \mathbf{u}(k))} \quad (50)$$

$$\boxed{\mathbf{P}_{xx}^-(k+1) = \Phi(k)\mathbf{P}_{xx}(k)\Phi(k)^T + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}(k)^T} \quad (51)$$

The posterior estimate and variance equations are below, introducing the Kalman gain again for convenience:

$$\boxed{\mathbf{K}(k) \equiv (\mathbf{P}_{xx}^-(k)\Upsilon(k)^T)(\Upsilon(k)\mathbf{P}_{xx}^-(k)\Upsilon(k)^T + \mathbf{R}(k))^{-1}} \quad (52)$$

$$\boxed{\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)(\tilde{\mathbf{y}}(k) - \mathbf{v}(\hat{\mathbf{x}}^-(k)))} \quad (53)$$

$$\boxed{\mathbf{P}_{xx}(k) = (\mathbf{I} - \mathbf{K}(k)\Upsilon(k))\mathbf{P}_{xx}^-(k)} \quad (54)$$

These equations are similar in form to those of the linear Kalman filter. In terms of computational complexity, the EKF is comparable to the linear Kalman filter, with the most intensive operation being a matrix inverse. The primary difference is that the Jacobians of the prediction and measurement steps must be computed, making the variance equations for the EKF functions of the state as well as functions of time. In some cases, this can lead to ill-conditioning of the covariance approximation $\mathbf{P}_{xx}(k)$, and it prevents offline precomputation of the variance histories

[4]. Furthermore, it is known in general that the EKF may diverge if a “good” initial value of the estimate is not known [5]. Nonetheless, the EKF remains popular for several reasons: (1) Its relatively low computational overhead make it suitable for systems with limited computing capability; (2) The EKF is an intuitive extension of the linear Kalman filter, and is easy to understand and implement; (3) For many types of systems, the EKF works well without the need to address the conditioning of the covariance estimate.

To address some of the limitations of the EKF, the Unscented Kalman Filter (UKF) has been developed recently. The next section presents a brief overview of the UKF.

B. Unscented Kalman Filter

The UKF is similar to the EKF in that it is an approximation to the MMSE estimator for a nonlinear system. However, the UKF is fundamentally different in that it employs statistical linearization rather than explicit Taylor series linearization. Instead of the function Jacobians, the UKF approximates the distribution of the estimated state by a finite set of “sigma” points. This approximation is conceptually similar to that of particle filters, but the UKF points are deterministically chosen. Furthermore, the number of points is of the same order as the state size, whereas particle filters typically require at least an order of magnitude more points. The UKF theoretically obtains means and variances that are accurate to second order, rather than the first order-accurate EKF approximation [2].

The idea underlying the UKF is now presented graphically. For brevity, we will restrict the discussion here to the high-level concept of the UKF. The equations governing the UKF algorithm are provided in the Appendix for reference. The filter begins with a state mean and variance prior, just as the EKF. The mean and variance are then replaced with a particle approximation. The particles are not sampled randomly; rather, they are determined from the variance in such a way that the particles lie along the principal axes of the covariance ellipse.

A graphical representation of the UKF algorithm is shown in Fig. 3. In the initial step (Fig. 3a), the 2D covariance ellipse is replaced by a particle approximation. Particles are placed at the mean, and along the positive and negative principal directions. For the 2D ellipse, this leads to five particles. Then (Fig. 3b), each of the particles is propagated until the next measurement, using the nonlinear dynamics. The prediction mean and covariance are determined using a weighted average and weighted variance of the propagated particles. In Fig. 3c, a measurement is

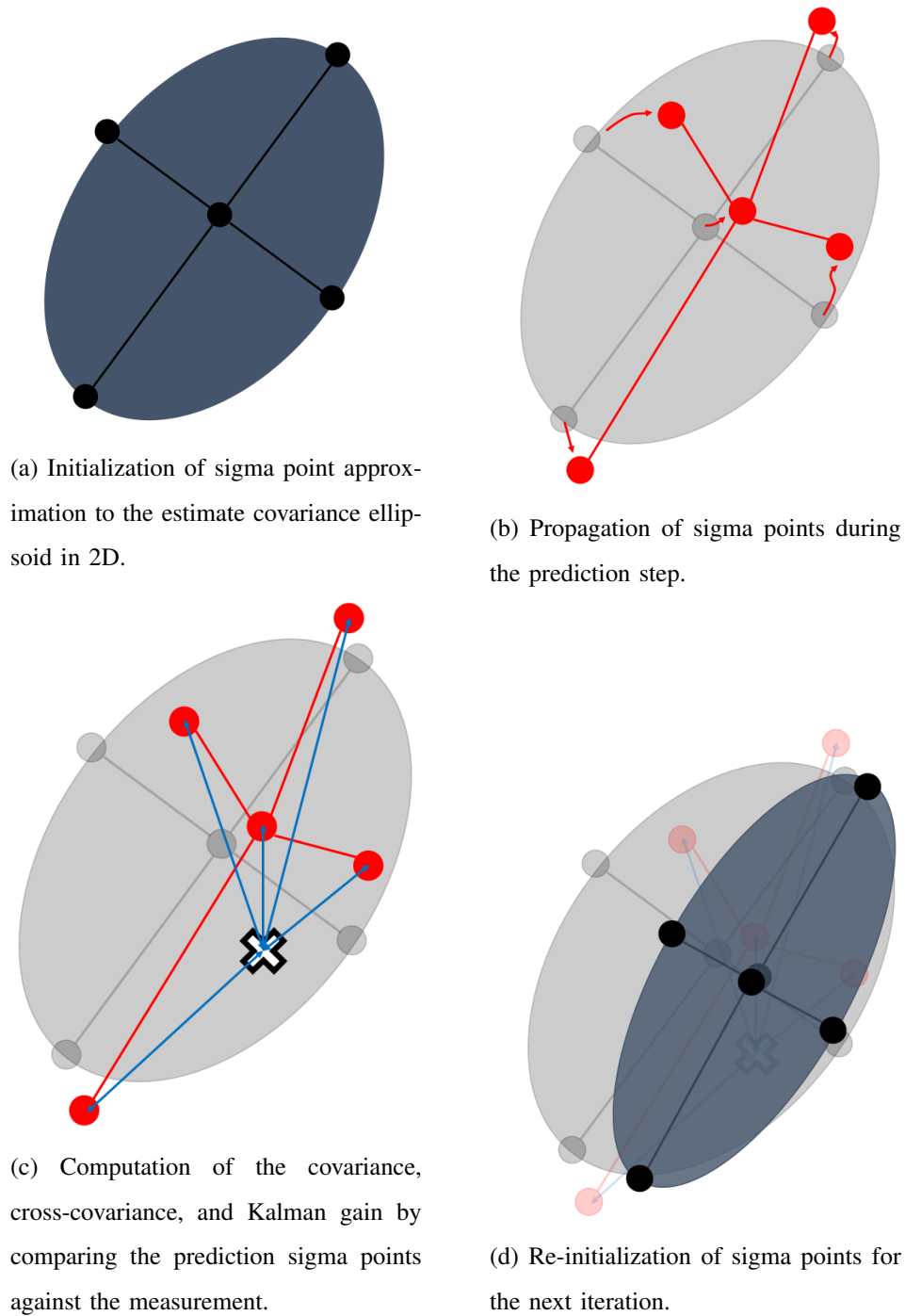


Fig. 3: Graphical summary of the UKF algorithm.

taken (represented by an "X" in the graphic). The predicted measurement associated with each prediction particle is computed using the nonlinear measurement function. Each expectation

of the measurement from each particle is compared against the particle values. The particle predictions are used to approximate a measurement expectation, measurement variance, and cross-covariance between the prediction state and measurement. The Kalman gain is computed from these approximations and used to update the prediction mean and variance to a posterior estimate and error variance. This completes the standard UKF prediction and update cycle. In Fig. 3d, five new particles are created from the posterior mean and variance, and the filter continues.

A high level-discussion of the UKF has been given; for the full mathematical description, the reader is encouraged to view the appendix. Now, we turn to a brief comparison of the EKF and UKF, from an application perspective.

Properly implemented, the UKF forms an estimate to a nonlinear process that is at least as good as an EKF estimate. Due to the higher order approximation implicit in the UKF, it may converge in some cases where an EKF fails. There is often a tradeoff in terms of computational cost; especially for long state vectors, the EKF algorithm can often exploit matrix sparsity and other manipulations to reduce the number of operations required. The UKF relies upon the nonlinear propagation and measurement expectation of sigma points and there are fewer opportunities to speed up the computation. However, it is true that the UKF and EKF have the same order of computational complexity[5], making either option much more tractable than methods like particle filters. The UKF is also somewhat less intuitive for those with a background in analytical estimation. Finally, an appropriate choice of the tuning parameters β , κ , and α may not always be obvious, and does not appear to be discussed frequently in the literature.

This section has presented the Extended and Unscented Kalman Filters for nonlinear state estimation. The following section briefly summarizes the overall report.

V. SUMMARY

This report has provided background and the underlying algorithm associated with the classical Kalman Filter. A simple numerical implementation on a linear system has been realized and results shown. Subsequently, a discussion of two extensions of the Kalman Filter to nonlinear systems have been presented: the Extended Kalman Filter and the Unscented Kalman Filter.

REFERENCES

- [1] John Crassidis and John L. Junkins. Optimal Estimation of Dynamic Systems. Chapman and Hall/CRC, 2nd edition, 2011.

- [2] Simon J. Julier and Jeffery K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. Proc. AeroSense: 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, pages 182–193, 1997.
- [3] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45, 1960.
- [4] Steven M. Kay. Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory. Prentice Hall PTR, 1st edition, 1993.
- [5] Eric Wan, Ronell Van Der Merwe, and et al. The unscented Kalman filter for nonlinear estimation. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, pages 153–158, 2000.
- [6] Tim Woodbury. Aero 626 challenge problem/final project, 2015. Final Class Project.

APPENDIX

The Unscented Kalman Filter algorithm is originally given by Ref. [5]. The notation of the following equations is motivated by Ref. [6] as well. Again, discrete nonlinear dynamics with discrete nonlinear measurements are considered. Define an augmented state vector as follows:

$$\mathbf{x}^a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{w}(k) \\ \mathbf{v}(k) \end{bmatrix} \equiv \begin{bmatrix} \mathbf{x}^x(k) \\ \mathbf{x}^w(k) \\ \mathbf{x}^v(k) \end{bmatrix} \quad (55)$$

Note the use of the superscripts x, w, v to denote the entries of the augmented state vector corresponding to the state, process noise, and measurement noise terms, respectively; this notation will be convenient later. Similarly, define the associated augmented state covariance matrix:

$$\mathbf{P}^a(k) \equiv \begin{bmatrix} \mathbf{P}_{xx}(k) & 0 & 0 \\ 0 & \mathbf{Q}(k) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{bmatrix} \quad (56)$$

Denote the length of the augmented state vector as L . Now, define an $L \times (2L + 1)$ array of *sigma points*, $\chi(k)$. The i th column of $\chi(k)$ is defined as follows:

$$\chi_i(k) \begin{cases} \mathbf{x}^a(k) & i = 1 \\ \mathbf{x}^a(k) + \sqrt{(L + \lambda)}\sqrt{\mathbf{P}_i^a(k)} & i = 2, \dots, L + 1 \\ \mathbf{x}^a(k) - \sqrt{(L + \lambda)}\sqrt{\mathbf{P}_i^a(k)} & i = L + 2, \dots, 2L + 1 \end{cases} \quad (57)$$

Here, $\sqrt{P_i^a(k)}$ is used to denote the i th column of the matrix square root of the augmented covariance matrix. Observe that the sigma point matrix includes perturbed values of the state estimate *as well as of the process and measurement noise vectors*. Those process and measurement noise elements in the sigma points are passed into the governing dynamics. Note the presence of λ , a scalar tuning parameter, that is summarized in the next subsection. Each column of the sigma point array is a point approximation to the prior, and is passed through the prediction equation as follows to produce a *prediction sigma point array* $\chi^-(k+1)$, whose columns are determined as follows:

$$\chi_i^-(k+1) = \phi(\chi_i^x(k), \mathbf{u}(k)) + G(k)\chi_i^w(k), i = 1, \dots, 2L+1 \quad (58)$$

The state and covariance prediction approximations are weighted sums of the propagated sigma points. Denote the mean and covariance weight vectors by \mathbf{W}^m and \mathbf{W}^c ; they are defined separately in the sub-section “Tuning parameters and weights.”

$$\hat{\mathbf{x}}^-(k+1) = \sum_{i=1}^{2L+1} \mathbf{W}^m(i) \chi_i^{x-}(k+1) \quad (59)$$

$$P_{xx}^-(k+1) = \sum_{i=1}^{2L+1} \mathbf{W}^c(i) (\chi_i^{x-}(k+1) - \hat{\mathbf{x}}^-(k+1)) (\chi_i^{x-}(k+1) - \hat{\mathbf{x}}^-(k+1))^T \quad (60)$$

For the measurement update step, it is convenient to switch to the index k . Now, the prediction sigma points are each passed through the measurement model, leading to the *measurement sigma point array* Y :

$$Y_i(k) = \mathbf{v}(\chi_i^{x-}(k)) + \chi_i^{y-}(k) \quad (61)$$

The measurement expectation, measurement variance, and prediction-measurement covariance are then approximated by weighted sums:

$$\hat{\mathbf{y}}(k) \equiv \sum_{i=1}^{2L+1} \mathbf{W}^m(i) Y_i(k) \quad (62)$$

$$P_{yy}(k) \equiv \sum_{i=1}^{2L+1} \mathbf{W}^c(i) (Y_i(k) - \hat{\mathbf{y}}(k)) (Y_i(k) - \hat{\mathbf{y}}(k))^T \quad (63)$$

$$P_{xy}(k) \equiv \sum_{i=1}^{2L+1} \mathbf{W}^c(i) (\chi_i^{x-}(k) - \hat{\mathbf{x}}^-(k)) (Y_i(k) - \hat{\mathbf{y}}(k))^T \quad (64)$$

The Kalman gain is then computed from the variance terms, and the posterior state estimate and variance follow:

$$\mathbf{K}(k) \equiv \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1} \quad (65)$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)(\tilde{\mathbf{y}}(k) - \tilde{\mathbf{y}}^-(k)) \quad (66)$$

$$\mathbf{P}_{xx}(k) = \mathbf{P}_{xx}^-(k) - \mathbf{K}(k) \mathbf{P}_{yy}^{-1} \mathbf{K}^T(k) \quad (67)$$

The posterior estimate and variance are used to create a new augmented state vector and sigma point matrix, and the process repeats. The next subsection defines the weights and tuning parameters used in the UKF.

1) Tuning parameters and weights: The mean and covariance weights are selected so that the weighted sums of the prior sigma points is the prior mean and covariance. The weights are defined as follows:

$$\mathbf{W}^m(i) = \begin{cases} \frac{\lambda}{L+\lambda} & i = 1 \\ \frac{1}{2(L+\lambda)} & i = 2, \dots, 2L+1 \end{cases} \quad (68)$$

$$\mathbf{W}^c(i) = \begin{cases} \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) & i = 1 \\ \frac{1}{2(L+\lambda)} & i = 2, \dots, 2L+1 \end{cases} \quad (69)$$

β is used to incorporate prior knowledge of the “true” distribution; $\beta = 2$ is optimal for Gaussian priors.

λ is defined as follows. α determines the spread of the sigma points about the mean and is generally set to a small value, say 10^{-3} . κ is a “secondary scaling parameter,” and is commonly set to zero [2].

$$\lambda = \alpha^2(L + \kappa) - L \quad (70)$$