Using Scrum in Global Software Development: A Systematic Literature Review

N. Mownika,J.VijaySree,G.Srinivasa Rao
Assistant Professor[1,2], Associate Professor
Dept. of CSE,
mail-id: mounika.nekkanti@gmail.com, jvijaya.pavani@gmail.com,
gsr.anurag@gmail.com
Anurag Engineering College,Anatagiri(V&M),Suryapet(Dt),Telangana-508206

ABSTRACT
This paper presents a tutorial on Kalman filtering that is designed for instruction to undergraduate students. The idea behind this work is that undergraduate students do not have much of the statistical and theoretical background necessary to fully understand the existing research papers and textbooks on this topic. Instead, this work offers an introductory experience for students which takes a more practical usage perspective on the topic, rather than the statistical derivation. Students reading this paper should be able to understand how to apply Kalman filtering tools to mathematical problems without requiring a deep theoretical understanding of statistical theory.

KEYWORDS
Data Processing, Kalman Filtering, Tutorial

INTRODUCTION

Kalman filtering is a useful tool for a variety of different applications. However, this technique is not easily accessible to undergraduate students due to the high level details in existing publications on this topic. While it may not be practical to expect undergraduates to obtain a deep and thorough understanding of the stochastic theory behind Kalman filtering techniques, it is reasonable to expect undergraduates to be capable of making use of this computational tool for different applications.

While there are some excellent references detailing the derivation and theory behind the Kalman filter [1,2,3], this article aims to take a more teaching-based approach to presenting the Kalman filter from a practical usage perspective. The goal of this work is to have undergraduate students be able to use this guide in order to learn about and implement their own Kalman filter. One of the major differences between this work and the current state of the art Kalman filtering tutorial
[3] is that the statistical theory is minimized, and focus is given to developing skills in implementing Kalman filters, rather than to understand the inner workings.

WHAT IS KALMAN FILTERING

So what is a Kalman filter? Let us start by breaking it down. The “Kalman” part comes from the primary developer of the filter, Rudolf Kalman [4]. So this is just a name that is given to filters of a certain type. Kalman filtering is also sometimes called “linear quadratic estimation.” Now let us think about the “filter” part. All filters share a common goal: to let something pass through while something else does not. An example that many people can relate to is a coffee filter. This coffee filter will allow the liquid to pass through, while leaving the solid coffee grounds behind. You can also think about a low-pass filter, which lets low frequencies pass through while attenuating high frequencies. A Kalman filter also acts as a filter, but its operation is a bit more complex and harder to understand. A Kalman filter takes in information which is known to have some error, uncertainty, or noise. The goal of the filter is to take in this imperfect information, sort out the useful parts of interest, and to reduce the uncertainty or noise. Diagrams of these three filtering examples are offered in Figure 1.

The state vector, **x**, are the values that will be estimated by the filter. Using the filter analogy, thecomponents of this vector are the things that you want to pass through the filter. Sometimes you may include more items in the state vector than you really care about if they are necessary in order to determine what you really want. For example if you want to determine the position of anobject using information about the acceleration, you will likely need to determine the velocity as well. This is an important distinction between the state and output vectors of the system. While in other situations the "output" is what you are trying to get, for state estimation problems using Kalman filtering, the "state" is actually the desired result.

The output vector, **y**, is not what you are trying to get out of the filter, but rather what you are ableto measure. You need to be able to express your measurements in terms of the states so that you can compare them with the measurements, i.e. you need to get apples to apples to know how much (or little) to correct. Not all measurements need to appear in the output vector for a particular formulation. Sometimes certain measurements are necessary for use in the state dynamics (1). The output vector should consist of values which can be both determined mathematically from the states as well as through some independent measurement system, i.e. the measurements are not used elsewhere in the filter.

The input vector, **u**, is probably the trickiest part of the Kalman filter definitions. This vector contains information that is necessary coming into the filter in order to define the system dynamics. These values can be sensor measurements, however in this case the uncertainty in these input values would need to be considered. In general, when defining your system equations
(1) and (2), after determine the necessary states, any other terms which appear in the filter that donot need to be estimated as states can be considered as inputs. For example, in order to find the dynamics of a velocity state, you might include an acceleration measurement as an input.

The terms **w** and **v** which correspond to the process and measurement noise vectors for the systemare interesting in that they do not typically appear directly in the equations of interest. Instead, these terms are used to model the uncertainty (or noise) in the equations themselves. This can manifest itself in a number of ways. The first is modelling error, which is uncertainty in the equation itself. How much can we trust this equation? Some equations, when derived from physics principles for instance, have negligible modelling error. However, some situations contain heuristically defined equations which may not be used with full confidence in their correctness. In this case, a certain amount of error can be considered to appear in the equation in the form of **w** or **v**. Another possible source of error is error in sensor measurements used in the equations. The **w** and **v** terms can be used to include the errors due to sensor measurements. In fact, this is the most common use of v in the output equations to account for the error in the measurement of the output. Note that this error is not in the equation itself, but accounts for how good the equation should be relative to the measurement. Note that **w** and **v** are not actually implemented in the calculations of (1) and (2) since they are assumed to be random errors with zero mean, but rather are just used to determine information about the process and measurement noise covariance matrices **Q** and **R**. If you are unfamiliar with the definition of a covariance matrix, please see Appendix A for more information.

What about the system matrices **F**, **G**, and **H**? These matrices will depend on the considered problem, and are used in order to represent the equations as a linear system of states and inputs. **F** will contain the coefficients of the state terms in the state dynamics (1), while **H** serves a similar function in the output equations (2). The **G** matrix contains coefficients of the input terms in the state dynamics (1). These matrices can in general vary with time, but cannot change with respect to the states or inputs. For many problems, these matrices are constant. Kalman Filtering Algorithm

The Kalman filter uses a prediction followed by a correction in order to determine the states of the filter. This is sometimes called predictor-corrector, or prediction-update. The main idea is that using information about the dynamics of the state, the filter will project forward and predict what the next state will be. A simple example of this would be if I know where I was before (previous state), and how fast I was moving (state dynamics), I can guess where I am at now (current state). This can be thought of as a numerical integration technique such as Euler's method or Runge-Kutta [5]. The correction or update part then involves comparing a measurement with what we predict that measurement should be based on our predicted states.

The Kalman filtering technique is now discussed in equation format. Starting from some initial state estimate, $\hat{\mathbf{x}}_0$, and initial state error covariance matrix, $\mathbf{P}_0$, the predictor-corrector format is applied recursively at each time step, e.g. using a loop. First, the state vector is predicted from the state dynamic equation using

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \tag{3}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state vector, $\hat{\mathbf{x}}_{k-1}$ is the previous estimated state vector, **u** is the input vector, and **F** and **G** are matrices defining the system dynamics. Note that the subscript $k|k$-1 is read as "k given k-1" and is a shorthand notation for the state at discrete time k given its previous state at discrete time k-1, i.e. this is the prediction of the state using the system model projected forward one step in time. Next, the state error covariance matrix must also be predicted using

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}^T_{k-1} + \mathbf{Q}_{k-1} \tag{4}$$

where $\mathbf{P}_{k|k-1}$ represents the predicted state error covariance matrix, $\mathbf{P}_{k-1}$ is the previous estimated state error covariance matrix, and **Q** is the process noise covariance matrix. Again, $k|k$-1 is indicating that this is the expected covariance matrix at k based on the system model and the covariance at k-1. Once the predicted values are obtained, the Kalman gain matrix, $\mathbf{K}_k$, is calculated by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5)$$

where **H** is a matrix necessary to define the output equation and **R** is the measurement noisecovariance. The state vector is then updated by scaling the "innovation," which is the difference between the measurement of the output, $\mathbf{z}_k$, and the predicted output, $\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}$ (sometimes called $\hat{\mathbf{y}}_{k|k-1}$), by the calculated Kalman gain matrix in order to correct the prediction by the appropriate amount, as in

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \quad (6)$$

Similarly, the state error covariance is updated by

A LINEAR KALMAN FILTERING EXAMPLE

In order to illustrate the use of the Linear Kalman Filter, a simple example problem is offered. This example considers a simple object in freefall assuming there is no air resistance. The goal of the filter is to determine the position of the object based on uncertain information about the initialposition of the object as well as measurements of the position provided by a laser rangefinder.

Using particle kinematics, we expect that the acceleration of the object will be equal to the acceleration due to gravity. Defining the height of the object in meters, h, we have:

$$\ddot{h}(t) = -g \quad (18)$$

where g is the acceleration due to gravity (g = 9.80665 m/s^2). Integrating this relationship over asmall time interval, Δt, gives

$$\ddot{h}(t) = \frac{\dot{h}(t) - \dot{h}(t-\Delta t)}{\Delta t} = -g \quad (19)$$

which is a backward difference equation, which is useful for Kalman filtering applications due tothe recursive structure of the filter, i.e. each time step in the Kalman filter always references the previous time step. Simplifying this expression gives

$$\dot{h}(t) = \dot{h}(t-\Delta t) - g\Delta t \quad (20)$$

Integrating again yields a commonly used kinematic equation relating successive positions of aparticle with respect to time for constant acceleration

$$h(t) = h(t-\Delta t) + \dot{h}(t-\Delta t) - \frac{1}{2} g(\Delta t)^2 \tag{21}$$

Rather than consider these equations in terms of continuous time, t, it is beneficial to rewrite the equations in terms of a discrete time index, k, which is defined by $t = k\Delta t$. Additionally, discrete time values are traditionally written as subscripts rather than as a functional dependence, i.e.

$$\begin{aligned} h(t) &= h(k\Delta t) = h_k \\ h(t-\Delta t) &= h(k\Delta t - \Delta t) = h(\Delta t(k-1)) = h_{k-1} \end{aligned} \tag{22}$$

Rewriting the kinematic relationships for the example problem in terms of the discrete timevariable, k, gives

$$\begin{aligned} \dot{h}_k &= \dot{h}_{k-1} - g\Delta t \\ h_k &= h_{k-1} + \dot{h}_{k-1}\Delta t - \frac{1}{2} g(\Delta t)^2 \end{aligned} \tag{23}$$

From these kinematic expressions, we see that we have two equations describing the motion of the object: one for velocity and one for position. Since we are interested in estimating the position, we know that the position must be included as a state. However, since we can see that the velocity also appears in the position equation, it is also necessary to obtain that information. One way to make sure that we have information about the velocity is to additionally include the velocity as a state. As a result, we now define the state vector for the Kalman filter as

$$\mathbf{x}_k = \begin{bmatrix} h_k \\ \dot{h}_k \end{bmatrix} \tag{24}$$

which results in the following expression

$$\mathbf{x}_k = \begin{bmatrix} h_{k-1} + \dot{h}_{k-1}\Delta t - \frac{1}{2} g(\Delta t)^2 \\ \dot{h}_{k-1} - g\Delta t \end{bmatrix} \tag{25}$$

With this definition, we can rewrite these equations in terms of the state vector, $\mathbf{x}$, as in

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} -\frac{1}{2}(\Delta t)^2 \\ -\Delta t \end{bmatrix} g \tag{26}$$

Now, we have the problem in necessary format for the Kalman filter

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \tag{27}$$

where the system matrices **F** and **G** are given by

$$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad \mathbf{G}_{k-1} = \begin{bmatrix} -\frac{1}{2}(\Delta t)^2 \\ -\Delta t \end{bmatrix} \tag{28}$$

and the input vector can be defined as

$$\mathbf{u}_{k-1} = g \tag{29}$$

The reason for selecting g as an input is because this information is necessary to define the state dynamics, but g is not defined as a state in the filter. This additional information is "input" into the equations at each time step. Note that for this particular problem the values of **F**, **G**, and **u** donot vary with respect to k. It is important to note in these equations that there is no process noiseuncertainty term, **w**. For this particular set of equations, we are assuming that there are no errorsin the equations themselves. For this problem, this is an assumption, as there could be disturbances from air resistance or other sources. However, assuming that these errors are small enough to ignore, we do not need to model the process noise for this problem. Because of this, the process noise covariance matrix, **Q**, can be set to zero.

Next, we need to consider the measurement part of the system. Let us consider a scenario wherethe position of the object can be measured using a laser rangefinder with 2 m standard deviation of error. Because the position is what can be measured, we need to define an output equation that gives the position as a function of the states of the filter. There is some uncertainty in the measurement, which is noted in the equations by the measurement noise vector, **v**:

$$\mathbf{y}_k = h_k + \mathbf{v}_k \tag{30}$$

Since the position can be written in terms of the state vector, this can be rewritten as

$$\mathbf{y}_k = [1 \quad 0]\mathbf{x}_k + \mathbf{v}_k$$

Now, we have the output equations of the system defined in the proper form as in (2), where thesystem matrix, **H**, is given by

The considered measurement system has a standard deviation of error of 2 m, which is a varianceof 4 m^2. Because of this, and the fact that there is only one term in the output vector, the resultingmeasurement noise covariance matrix reduces to a scalar value, **R** = 4 m^2.

In addition to the measurement noise, we also need to consider any uncertainty in the initial stateassumption. The initial position is approximately known to be 105 m before the ball is dropped, while the actual initial position is 100 m. The initial guess was roughly determined, and should therefore have a relatively high corresponding component in the assumed initial covariance. Forthis example, we consider an error of 10 m^2 for the initial position. For the initial velocity, we assume that the object starts from rest. This assumption is fairly reasonable in this case, so a smaller uncertainty value of 0.01 m^2/s^2 is assumed. To aid the reader in the implementation of this example, the necessary parameters and definitions for this filtering application are summarized in Table 4.

THE EXTENDED KALMAN FILTER

The Extended Kalman Filter (EKF) is just an extension of the Kalman filter to nonlinear systems. This means that the difference with the EKF is that the state and/or the output equations can contain nonlinear functions. So rather than considering systems of the form (1) and (2), the EKFcan consider a more general nonlinear set of equations:

Other Nonlinear Kalman Filtering Techniques

The Extended Kalman Filter (EKF) is not the only nonlinear Kalman filtering technique. Perhapsthe most common alternative to the EKF is the Unscented Kalman Filter (UKF) which is sometimes referred to as a Sigma-Point Kalman Filter (SPKF). Rather than use Jacobian matrices to handle the nonlinear in the system, the UKF uses a statistical linearization technique called the unscented transformation [9]. The UKF has demonstrated some advantages in more accurate linearization, particularly in highly nonlinear problems [9,10], however in some situations it has been shown to produce similar results to that of the EKF [11]. The EKF tends to be more computationally efficient than the UKF, thus making it more desirable for real-time applications[11,12]. Additionally, the EKF is a bit easier to understand conceptually, which is beneficial for student use. The UKF, however, can provide superior linearization in certain situations, and canalso be easier to implement since there is no need to calculate Jacobian matrices [10]. For detailson the implementation of the UKF, see [13] or [14].

1. CONCLUSION

This work provided a tutorial for Kalman filtering that is geared toward undergraduate students with little or no experience on this topic. Detailed descriptions and an example problem were offered in order to help aid in the basic understanding of this complex topic.

ACKNOWLEDGEMENTS

This work was partially supported by the Lafayette College EXCEL Scholar Program for undergraduate research and the Rodney A. Erickson Discovery Grant Program for undergraduateresearch.

REFERENCES

[1] Reference: [1] Simon, D., Optimal State Estimation, Wiley, New York, 2006.

[2] Optimal Filtering, by Brian D. O. Anderson and John B. Moore, Prentice Hall, New Jersey, 1979 [2].

[3] Technical Report TR 95-041, University of North Carolina, Department of Computer Science, 1995, Welch, G., and Bishop, G., "An introduction to the Kalman filter."

[4]

[5] Trans. of the ASME - Journal of Basic Engineering, March 1960, pp. 35-45, has an article by R. E. Kalman titled "A Novel Method to Linear Filtering and Prediction Issues."

[6] Reference: [5] Kreyszig, E., Advanced Engineering Mathematics, 9th Edition, Wiley, New York, 2006.

[7] This is according to Reif et al. (1999) in IEEE Transactions on Automatic Control, Volume 44, Issue 4 (April): "Stochastic Stability of the Discrete-Time Extended Kalman Filter" [6].

[8] A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press (New York), 1970[7].

[9] IEEE Colloquium on Kalman Filters: Introduction, Applications, and Future Advances, February 1989 [8] Hargrave, P., "A tutorial introduction to Kalman filtering."

[10] [9] S. Julier & J. Uhlmann, "A Novel Extension of the Kalman Filter to Nonlinear Systems." SPIE Proceedings, Volume 3068, Pages 181–193, 1997.

[11] Choosing between the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) is the first step in understanding nonlinear Kalman filters [10]. This article was first published in June 2013 in the journal Interactive Robotics Letters at West Virginia University. You may get more information at this location: http://www2.statler.wvu.edu/irl/page13.html.

[12] [11] "Sensitivity Study of Extended and Unscented Kalman Filters for Attitude Estimation," AIAA Journal of Aeronautical Information Systems, Vol. 10, No. 3,

March 2013, pp. 131-143; Rhudy, M.; Gu, Y.; Gross, J.; Gururajan, S.; Napolitano, M. doi: 10.2514/1.54899.

[13] According to [12] "Flight Test Assessment of GPS/INS Sensor Fusion Algorithms for Attitude Estimation," published in IEEE Transactions on Aerospace Electronic Systems, Volume 48, Issue 3, Pages 2128-2139, July 2012.

[14] According to [13] "Understanding Nonlinear Kalman Filters, Part II: An Implementation Guide," written by Matthew Rhudy and Ying Gu and published in the June 2013 issue of Interactive Robotics Letters at West Virginia University. You may get more information at this location: http://www2.statler.wvu.edu/irl/page13.html.

[15] Chapter 7 in Kalman Filtering and Neural Networks, Wiley, New York, March 2002, pages 221-282, by E. Wan and R. van der Merwe.