1.0 INTRODUCTION

1.1 Background

Simply stated, the Kalman filter provides a means of combining measurements and knowledge of a process or system to determine the value of some desired quantity. To illustrate this consider the following example. An experimenter selects a resistor that is labeled as being 100 ohms with a tolerance of 1%. To assure that the resistance is 100 ohms the experimenter measures the resistance with an ohmmeter that has an accuracy of 10%. The resistance the experimenter measures is 105 ohms. What value of resistance should the experimenter assume for the resistor? The resistor (which is termed the process or system) is known to be accurate within 1% and the ohmmeter within 10%. Therefore, it seems logical to lean toward the value of 100 ohms. However, the ohmmeter did read 105 ohms. Therefore, the experimenter may wish to account for this and specify a resistance of 100.5 or 101 ohms. Thus, in determining the resistance the experimenter has considered the known information about the resistor and the measurement. The experimenter gave more weight to the specified resistance than to the ohmmeter measurement because of the relative accuracies involved. Had the tolerance on the resistor been 20%, the experimenter may have been more inclined to assign a value closer to 105 ohms to the resistor.

In essence, the above decision processes are performed by the Kalman filter. By accounting for process, or system, uncertainties and measurement uncertainties, the Kalman filter estimates quantities based on measurements and expected behavior of the system.

Mathematically speaking, the Kalman filter is a linear, minimum mean squared estimator. This means that the filter estimate is a linear combination of measurements where the coefficients of the linear combination are chosen so as to minimize the expected value, or mean, of the square of the error between the estimate and the actual value. The expected value is used because of an assumption that the quantity being estimated is a random process and that the measurements are corrupted by noise. The fact that the filter design is linear, and based on a mean squared error criterion, is driven by mathematics. Specifically, these constraints result in a design technique that is mathematically tractable for a wide range of applications. Texts on estimation theory (see the bibliography for a list of a few) offer several other types of estimator designs. However, they usually apply to a limited set of situations.

It is widely accepted that Wiener is credited with the first organized treatment of the linear, minimum mean squared estimation problem, including the postulation of the filtering and prediction problem in the form of the Wiener-Hopft equation. Over the years since Wiener's early work, many authors have posed various solutions to the so-called Wiener problem. Most of these were difficult to implement and had considerable practical limitations. For example, Wiener's approach, which was a frequency domain approach, required computation of the noise spectrum and was only applicable to
stationary processes. Another approach for discrete time systems was a brute-force approach where a new set of coefficients was computed for each new measurement. The problem with this approach is that the order of the filter grows with time and the number of equations that must be solved for the coefficients grows prohibitively large.

Kalman’s contribution to the linear, minimum mean squared estimation problem was to use a model of the process, or system, to formulate a recursive estimator. With this formulation only one new coefficient (actually one matrix of coefficients) is computed for each new measurement. In his first paper (Kalman, 1960), Kalman addressed the discrete-time problem and in a second paper (Kalman and Bucy, 1961), Kalman and Bucy addressed the continuous-time problem. Although the Kalman filter is named after Kalman, according to WIKIPEDIA, Thorvald Nicolai Thiele and Peter Swerling developed similar algorithms prior to Kalman. Also, the current form of the Kalman filter is more closely related to the one in the Kalman-Bucy paper than in Kalman’s original paper. This is most likely why many people refer to the Kalman filter as the Kalman-Bucy filter.

One of the early motivations behind the Kalman filter appears to be its use in the optimal control problem. One of the requirements on optimal controller designs in the 1950’s and 1960’s was that estimates of the system states were needed to implement the control law. At that time the availability of state estimators was limited and the ones that were available were difficult to implement. The Kalman filter held the promise of solving this problem because it was easy to implement. Unfortunately, the Kalman filter is based on random process concepts whereas most optimal control problems are not. Thus, the optimal control designer had to cast the control problem in the stochastic domain, which created its own set of problems. As a result of these complexities, the Kalman filter did not find wide use in control theory and the optimal control theorists found other ways of designing effective controllers.

While Kalman filters didn’t find wide use in optimal control theory they did find wide use in tracking problems and other estimation problems. In fact, it wasn’t long after Kalman’s original paper that the extended Kalman filter was formulated and applied to the radar tracking problem. Since then, the Kalman filter has become a very widely accepted estimation technique for radar tracking. It is also used extensively in other applications such as transfer alignment, guidance and GPS systems. A recent IEEE Control Systems Magazine (December 2009) had articles on Kalman filtering for positioning and heading control of ships and offshore rigs, focal plane calibration of the Spitzer space telescope, and the use of Kalman filtering in economics.

Since the original development of the Kalman filter there have been many extensions to it that are aimed at solving various problems encountered when implementing the classical Kalman filter. Examples of these include the aforementioned extended Kalman filter that was developed to accommodate nonlinear system and measurement models, the square-root filter that was developed to solve problems associated with ill conditioning of some of the covariance matrices and the mixed, continuous-discrete Kalman filter that is used to solve problems that can arise from formulating a discrete-time model of
a nonlinear, continuous-time, system. Other extensions include the unscented Kalman filter and the particle filter.

1.2 Outline of the Notes and Course

Many Kalman filter derivations have been presented over the years since Kalman's first paper. We will present two in these notes. One is a straightforward technique that is more intuitive than rigorous. We present it because of its brevity and the fact that it is a direct application of the orthogonality principle, the basis of all Kalman filter derivations. The second technique is based on two papers by Kailath (Kailath, 1968) and is very rigorous and elegant. However, it has a detracting feature in that it appears to be a circuitous method of deriving the Kalman filter because of the background derivations that precede the actual Kalman filter derivation.

We will begin our development of the Kalman filter by briefly reviewing linear system theory and state variables in Chapter 2, and probability, random variables and stochastic processes in Chapter 3. A background in these topics is necessary because the Kalman filter is used to estimate the states of a system that is excited by random processes. Furthermore, the estimates are based on measurements that are corrupted by noise.

In the Chapter 4 we discuss the concept of covariance propagation. Specifically, we derive equations for the covariance (a generalization of variance) and autocovariance of the states of a system. Covariance propagation is a time domain method that is capable of accommodating time varying systems and non-stationary random process inputs.

In Chapter 5 we present a derivation of the discrete-time, scalar Kalman filter using the straightforward technique discussed above. We also present an in-depth discussion of the structure of the Kalman filter and how it works. This chapter also contains some examples of the Kalman filter that illustrate some of its properties. It also contains an illustration of some of the problems that can arise if the Kalman filter is not designed and used correctly.

Although the scalar Kalman filter is useful for understanding the operation of a Kalman filter it is not very practical in that it allows the estimation of only one state. In general we will want to estimate several states (i.e. a state vector). To quickly get to the point where we can build more practical Kalman filters we will present a heuristic development of the vector Kalman filter. This allows us to construct Kalman filters before we present the formal derivation of the Kalman filter.

Chapter 6 contains a derivation of the extended Kalman filter. With the extended Kalman filter we derive a Kalman filter formulation for the case where the system is non-linear and/or the measurement vector is a non-linear function of the states. We include an example of the radar tracking problem to illustrate the design and use of the extended Kalman filter.

Chapter 7 contains a formal derivation of the vector Kalman filter and begins with a formal statement of the linear, minimum mean squared estimation problem, and a derivation of the orthogonality condition. We also show that the linear, minimum mean squared estimator is the optimum
minimum mean squared estimator when the system and measurement noises are Gaussian. Finally, Chapter 7 contains a rigorous derivation of the discrete-time, vector, Kalman filter. The formal derivation of the Kalman filter is delayed to the end of the course because it is fairly tedious and time consuming. By delaying it to the end, we can be building filters for homework while performing the formal derivation in class.

1.3 Notation

We will use the following fundamental notation throughout this text.

- Non random, scalar quantities – lower- and upper-case, italic symbols (e.g., \( a, x, B, \beta, \Phi \), etc.).
- Non random, column vectors – lower-case, bold, italic symbols (e.g., \( \textbf{a}, \textbf{x}, \textbf{b}, \beta, \phi \), etc.).
- Non random, matrices or row vectors – upper-case, bold, italic symbols (e.g., \( \textbf{A}, \textbf{X}, \textbf{B}, \textbf{B}, \Phi \), etc.).
- Random, scalar quantities – lower- and upper-case, non-italic symbols (e.g., \( a, x, B, \beta, \Phi \), etc.).
- Random, column vectors – lower-case, bold, non-italic symbols (e.g., \( \textbf{a}, \textbf{x}, \textbf{b}, \beta, \phi \), etc.).
- Random, matrices or row vectors – upper-case, bold, non-italic symbols (e.g., \( \textbf{A}, \textbf{X}, \textbf{B}, \textbf{B}, \Phi \), etc.).