SIGGRAPH 2001

Course 8

An Introduction to the Kalman Filter

Greg Welch
Gary Bishop

University of North Carolina at Chapel Hill
Department of Computer Science
Chapel Hill, NC 27599-3175

http://www.cs.unc.edu/~{welch, gb}
{welch, gb}@cs.unc.edu

TABLE OF CONTENTS

Preface

In putting together this course pack we decided not to simply include copies of the slides for the course presentation, but to attempt to put together a small booklet of information that could stand by itself. The course slides and other useful information, including a new Java-based *Kalman Filter Learning Tool* are available at

`http://www.cs.unc.edu/~tracker/ref/s2001/kalman/`

In addition, we maintain a popular web site dedicated to the Kalman filter. This site contains links to related work, papers, books, and even some software.

`http://www.cs.unc.edu/~welch/kalman/`

We expect that you (the reader) have a basic mathematical background, sufficient to understand explanations involving basic linear algebra, statistics, and random signals.

Course Syllabus

Time	Speaker	Topic	**Time**
10:00 AM	Bishop	Welcome, Introduction, Intuition	0:30
10:30 AM	Welch	Concrete examples	0:30
11:00 AM	Bishop	Non-linear estimation	0:15
11:15 AM	Welch	System identification and multi-modal filters	0:30
11:45 AM	Welch	Conclusions (summary, resources, etc.)	0:15
12:00 PM			
		Total time	2:00

1. Introduction

The Kalman filter is a mathematical power tool that is playing an increasingly important role in computer graphics as we include sensing of the real world in our systems. The good news is you don't have to be a mathematical genius to understand and effectively use Kalman filters. This tutorial is designed to provide developers of graphical systems with a basic understanding of this important mathematical tool.

1.1 Course Description

While the Kalman filter has been around for about 30 years, it (and related optimal estimators) have recently started popping up in a wide variety of computer graphics applications. These applications span from simulating musical instruments in VR, to head tracking, to extracting lip motion from video sequences of speakers, to fitting spline surfaces over collections of points.

The Kalman filter is the best possible (optimal) estimator for a large class of problems and a very effective and useful estimator for an even larger class. With a few conceptual tools, the Kalman filter is actually very easy to use. We will present an intuitive approach to this topic that will enable developers to approach the extensive literature with confidence.

1.2 Speaker/Author Biographies

Greg Welch is a Research Assistant Professor in the Department of Computer Science at the University of North Carolina at Chapel Hill. His research interests include hardware and software for man-machine interaction, 3D interactive computer graphics, virtual environments, tracking technologies, tele-immersion, and projector-based graphics. Welch graduated with *highest distinction* from Purdue University with a degree in Electrical Engineering Technology in 1986 and received a Ph.D. in computer science from UNC-Chapel Hill in 1996. Before coming to UNC he worked at NASA's Jet Propulsion Laboratory and Northrop-Grumman's Defense Systems Division. He is a member of the IEEE Computer Society and the Association of Computing Machinery.

Gary Bishop is an Associate Professor in the Department of Computer Science at the University of North Carolina at Chapel Hill. His research interests include hardware and software for man-machine interaction, 3D interactive computer graphics, virtual environments, tracking technologies, and image-based rendering. Bishop graduated with highest honors from the Southern Technical Institute in Marietta, Georgia, with a degree in Electrical Engineering Technology in 1976. He completed his Ph.D. in computer science at UNC-Chapel Hill in 1984. Afterwards he worked for Bell Laboratories and Sun Microsystems before returning to UNC in 1991.

2. Probability and Random Variables

What follows is a very basic introduction to probability and random variables. For more extensive coverage see for example (Maybeck 1979; Brown and Hwang 1996; Kailath, Sayed et al. 2000).

2.1 Probability

Most of us have some notion of what is meant by a "random" occurrence, or the probability that some event in a *sample space* will occur. Formally, the probability that the outcome of a discrete event (e.g., a coin flip) will favor a particular event is defined as

$$p(A) = \frac{\text{Possible outcomes favoring event A}}{\text{Total number of possible outcomes}}.$$

The probability of an outcome favoring either A *or* B is given by

$$p(A \cup B) = p(A) + p(B). \tag{2.1}$$

If the probability of two outcomes is *independent* (one does not affect the other) then the probability of *both* occurring is the product of their individual probabilities:

$$p(A \cap B) = p(A)p(B). \tag{2.2}$$

For example, if the probability of seeing a "heads" on a coin flip is 1/2, then the probability of seeing "heads" on both of two coins flipped at the same time is 1/4. (Clearly the outcome of one coin flip does not affect the other.)

Finally, the probability of outcome A given an occurrence of outcome B is called the *conditional probability* of A given B, and is defined as

$$p(A|B) = \frac{p(A \cap B)}{p(B)}. \tag{2.3}$$

2.2 Random Variables

As opposed to discrete events, in the case of tracking and motion capture, we are more typically interested with the randomness associated with a *continuous* electrical voltage or perhaps a user's motion. In each case we can think of the item of interest as a *continuous*

random variable. A random variable is essentially a function that maps all points in the sample space to real numbers. For example, the continuous random variable $X(t)$ might map time to position. At any point in time, $X(t)$ would tell us the expected position.

In the case of continuos random variables, the probability of any *single* discrete event A is in fact 0. That is, $p(A) = 0$. Instead we can only evaluate the probability of events within some interval. A common function representing the probability of random variables is defined as the *cumulative distribution function*:

$$F_X(x) = p(-\infty, x]. \tag{2.4}$$

This function represents the cumulative probability of the continuous random variable X for all (uncountable) events up to and including x. Important properties of the cumulative density function are

1. $F_X(x) \to 0$ as $x \to -\infty$
2. $F_X(x) \to 1$ as $x \to +\infty$
3. $F_X(x)$ is a non-decreasing function of x.

Even more commonly used than equation (2.4) is its derivative, known as the *probability density function*:

$$f_X(x) = \frac{d}{dx} F_X(x). \tag{2.5}$$

Following on the above given properties of the cumulative probability function, the density function also has the following properties:

1. $f_X(x)$ is a non-negative function
2. $\int_{-\infty}^{\infty} f_X(x)dx = 1$.

Finally note that the probability over any interval $[a, b]$ is defined as

$$p_X[a, b] = \int_a^b f_X(x)dx.$$

So rather than summing the probabilities of discrete events as in equation (2.1), for continuous random variables one integrates the probability density function over the interval of interest.

2.3 Mean and Variance

Most of us are familiar with the notion of the *average* of a sequence of numbers. For some N samples of a discrete random variable X, the average or *sample mean* is given by

$$\bar{X} = \frac{X_1 + X_2 + \ldots + X_N}{N}.$$

Because in tracking we are dealing with continuous signals (with an uncountable sample space) it is useful to think in terms of an infinite number of trials, and correspondingly the outcome we would *expect* to see if we sampled the random variable infinitely, each time seeing one of n possible outcomes $x_1 \ldots x_n$. In this case, the *expected value* of the discrete random variable could be approximated by averaging probability-weighted events:

$$\bar{X} \approx \frac{(p_1 N)x_1 + (p_2 N)x_2 + \ldots + (p_n N)x_N}{N}.$$

In effect, out of N trials, we would expect to see $(p_1 N)$ occurrences of event x_1, etc. This notion of infinite trials (samples) leads to the conventional definition of *expected value* for *discrete* random variables

$$\text{Expected value of } X = E(X) = \sum_{i=1}^{n} p_i x_i \tag{2.6}$$

for n possible outcomes $x_1 \ldots x_n$ and corresponding probabilities $p_1 \ldots p_n$. Similarly for the continuous random variable the expected value is defined as

$$\text{Expected value of } X = E(X) = \int_{-\infty}^{\infty} x f_X(x) dx. \tag{2.7}$$

Finally, we note that equation (2.6) and equation (2.7) can be applied to functions of the random variable X as follows:

$$E(g(X)) = \sum_{i=1}^{n} p_i g(x_i) \tag{2.8}$$

and

$$E(g(X)) = \int_{-\infty}^{\infty} g(x) f_X(x) dx. \tag{2.9}$$

The expected value of a random variable is also known as the *first statistical moment*. We can apply the notion of equation (2.8) or (2.9), letting $g(X) = X^k$, to obtain the k^{th} statistical moment. The k^{th} statistical moment of a continuous random variable X is given by

$$E(X^k) = \int_{-\infty}^{\infty} x^k f_X(x)dx . \tag{2.10}$$

Of particular interest in general, and to us in particular, is the *second moment* of the random variable. The second moment is given by

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x)dx . \tag{2.11}$$

When we let $g(X) = X - E(X)$ and apply equation (2.11), we get the *variance* of the signal about the mean. In other words,

$$\begin{aligned} \text{Variance } X &= E[(X - E(X))^2] \\ &= E(X^2) - E(X)^2 . \end{aligned}$$

Variance is a very useful statistical property for random signals, because if we knew the variance of a signal that was otherwise supposed to be "constant" around some value—the mean, the magnitude of the variance would give us a sense how much jitter or "noise" is in the signal.

The square root of the variance, known as the *standard deviation*, is also a useful statistical unit of measure because while being always positive, it has (as opposed to the variance) the same units as the original signal. The standard deviation is given by

$$\text{Standard deviation of } X = \sigma_X = \sqrt{\text{Variance of } X} .$$

2.4 Normal or Gaussian Distribution

A special probability distribution known as the *Normal* or *Gaussian* distribution has historically been popular in modeling random systems for a variety of reasons. As it turns out, many random processes occurring in nature actually appear to be normally distributed, or very close. In fact, under some moderate conditions, it can be proved that a sum of random variables with *any* distribution tends toward a normal distribution. The theorem that formally states this property is called the *central limit theorem* (Maybeck 1979; Brown and Hwang 1996). Finally, the normal distribution has some nice properties that make it mathematically tractable and even attractive.

Given a random process $X \sim N(\mu, \sigma^2)$, i.e. a continuous random process X that is normally distributed with mean μ and variance σ^2 (standard deviation σ), the probability density function for X is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

for $-\infty < x < \infty$. Any linear function of a normally distributed random process (variable) is also a normally distributed random process. In particular if $X \sim N(\mu, \sigma^2)$ and $Y = aX + b$, then

$$Y \sim N(a\mu + b, a^2\sigma^2). \tag{2.12}$$

The probability density function for Y is then given by

$$f_Y(y) = \frac{1}{\sqrt{2\pi a^2\sigma^2}} e^{-\frac{1}{2}\frac{(y-(a\mu+b))^2}{a^2\sigma^2}}. \tag{2.13}$$

Finally, if X_1 and X_2 are independent (see Section 2.5 below), $X_1 \sim N(\mu_1, \sigma_1^2)$, and $X_2 \sim N(\mu_2, \sigma_2^2)$, then

$$X_1 + X_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2), \tag{2.14}$$

and the density function becomes

$$f_X(x_1 + x_2) = \frac{1}{\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)}} e^{-\frac{1}{2}\frac{(x-(\mu_1+\mu_2))^2}{(\sigma_1^2+\sigma_2^2)}}. \tag{2.15}$$

See (Kelly 1994) pp. 351-358 for further explanation and proofs of the above. Graphically, the normal distribution is what is likely to be familiar as the “bell-shaped” curve shown below in Figure 2.1.

2.5 Continuous Independence and Cond. Probability

Finally we note that as with the discrete case and equations (2.2) and (2.3), independence and conditional probability are defined for continuous random variables. Two continuous random variables X and Y are said to be *statistically independent* if their *joint* probability $f_{XY}(x, y)$ is equal to the product of their individual probabilities. In other words, they are considered independent if

$$f_{XY}(x, y) = f_X(x) f_Y(y).$$

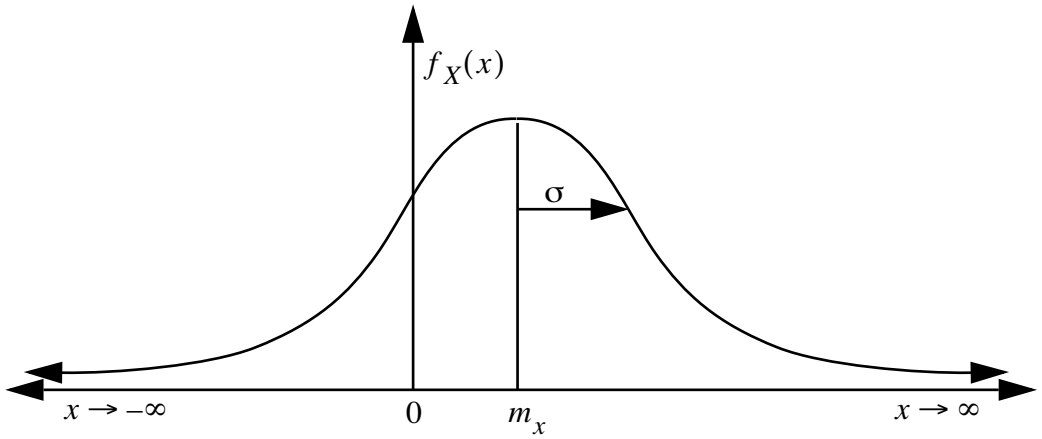

Figure 2.1: The Normal or Gaussian probability distribution function.

Bayes' Rule

In addition, Bayes' rule follows from (2.3), offering a way to specify the probability density of the random variable X given (in the presence of) random variable Y. Bayes' rule is given as

$$f_{X|Y}(x) = \frac{f_{Y|X}(y) f_X(x)}{f_Y(y)}.$$

Continuous-Discrete

Given a *discrete* process X and a *continuous* process Y, the discrete probability mass function for X conditioned on $Y = y$ is given by

$$p_X(x \mid Y = y) = \frac{f_Y(y \mid X = x) p_X(x)}{\sum_z f_Y(y \mid X = z) p_X(z)}. \tag{2.16}$$

Note that this formula provides a discrete probability based on the conditioning density, *without any integration*. See (Kelly 1994) p. 546 for further explanation and proofs of the above.

2.6 Spatial vs. Spectral Signal Characteristics

In the previous sections we looked only at the *spatial* characteristics of random signals. As stated earlier, the magnitude of the variance of a signal can give us a sense of how much jitter or "noise" is in the signal. However a signal's variance says nothing about the

spacing or the rate of the jitter over time. Here we briefly discuss the temporal and hence *spectral* characteristics of a random signal. Such discussion can be focused in the time or the frequency domain. We will look briefly at both.

A useful time-related characteristic of a random signal is its *autocorrelation*—its correlation with itself over time. Formally the autocorrelation of a random signal $X(t)$ is defined as

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] \tag{2.17}$$

for sample times t_1 and t_2. If the process is *stationary* (the density is invariant with time) then equation (2.17) depends only on the difference $\tau = t_1 - t_2$. In this common case the autocorrelation can be re-written as

$$R_X(\tau) = E[X(t)X(t+\tau)]. \tag{2.18}$$

Two hypothetical autocorrelation functions are shown below in Figure 2.1. Notice how compared to random signal X_2, random signal X_1 is relatively short and wide. As $|\tau|$ increases (as you move away from $\tau = 0$ at the center of the curve) the autocorrelation signal for X_2 drops off relatively quickly. This indicates that X_2 is less correlated with itself than X_1.

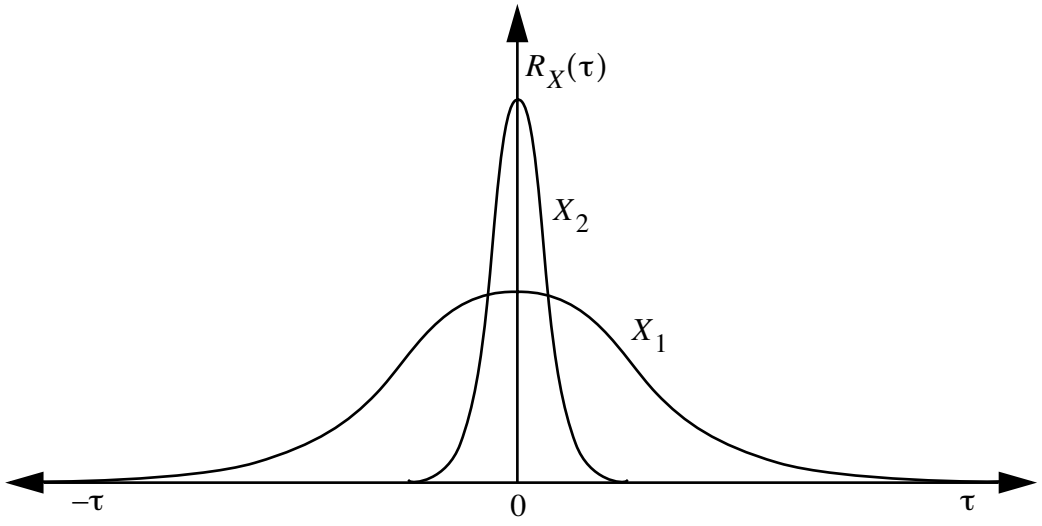

Figure 2.2: Two example (hypothetical) autocorrelation functions X_1 and X_2.

Clearly the autocorrelation is a function of time, which means that it has a spectral interpretation in the frequency domain also. Again for a stationary process, there is an important temporal-spectral relationship known as the *Wiener-Khinchine relation*:

$$S_X(j\omega) = \Im[R_X(\tau)] = \int_{-\infty}^{\infty} R_X(\tau)e^{-j\omega\tau}d\tau$$

where $\Im[\bullet]$ indicates the Fourier transform, and ω indicates the number of (2π) cycles per second. The function $S_X(j\omega)$ is called the *power spectral density* of the random signal. As you can see, this important relationship ties together the time and frequency spectrum representations of the same signal.

White Noise

An important case of a random signal is the case where the autocorrelation function is a *dirac delta* function $\delta(\tau)$ which has zero value everywhere except when $\tau = 0$. In other words, the case where

$$R_X(\tau) = \begin{cases} \text{if } \tau = 0 \text{ then } A \\ \text{else } 0 \end{cases}$$

for some constant magnitude A. In this special case where the autocorrelation is a "spike" the Fourier transform results in a *constant* frequency spectrum. as shown in Figure 2.3. This is in fact a description of *white noise*, which be thought of both as having power at all

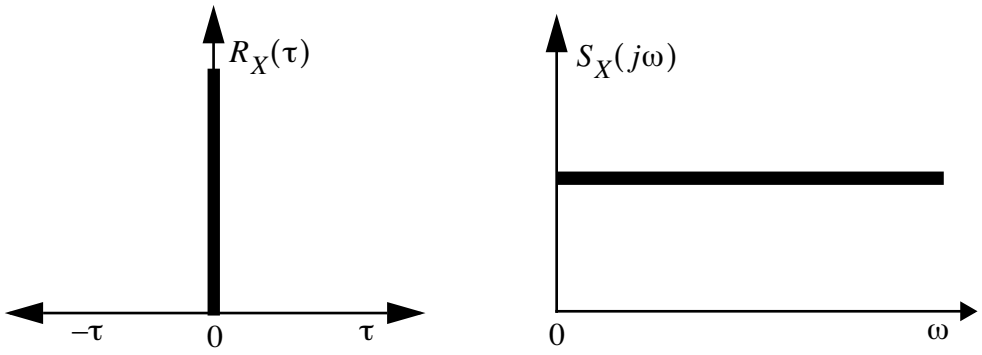

Figure 2.3: White noise shown in both the time (left) and frequency domain (right).

frequencies in the spectrum, and being completely uncorrelated with itself at any time except the present $(\tau = 0)$. This latter interpretation is what leads white noise signals to be called *independent*. Any sample of the signal at one time is completely independent (uncorrelated) from a sample at any other time.

While impossible to achieve or see in practice (no system can exhibit infinite energy throughout an infinite spectrum), white noise is an important building block for design and analysis. Often random signals can be modeled as filtered or *shaped* white noise. Literally this means that one could filter the output of a (hypothetical) white noise source to achieve a non-white or *colored* noise source that is both band-limited in the frequency domain, and more correlated in the time domain.

3. Stochastic Estimation

While there are many application-specific approaches to "computing" (estimating) an unknown state from a set of process measurements, many of these methods do not inherently take into consideration the typically *noisy* nature of the measurements. For example, consider our work in tracking for interactive computer graphics. While the requirements for the tracking information varies with application, the fundamental source of information is the same: pose estimates are derived from noisy electrical measurements of mechanical, inertial, optical, acoustic, or magnetic sensors. This noise is typically statistical in nature (or can be effectively modeled as such), which leads us to *stochastic* methods for addressing the problems. Here we provide a very basic introduction to the subject, primarily aimed at preparing the reader for Chapter 4. For a more extensive discussion of stochastic estimation see for example (Lewis 1986; Kailath, Sayed et al. 2000).

3.1 State-Space Models

State-space models are essentially a notational convenience for estimation and control problems, developed to make tractable what would otherwise be a notationally-intractable analysis. Consider a dynamic process described by an n-th order difference equation (similarly a differential equation) of the form

$$y_{i+1} = a_{0,i} y_i + \ldots + a_{n-1,i} y_{i-n+1} + u_i, \; i \geq 0,$$

where $\{u_i\}$ is a *zero-mean* (statistically) *white* (spectrally) random "noise" process with autocorrelation

$$E(u_i, u_j) = R_u = Q_i \delta_{ij},$$

and initial values $\{y_0, y_{-1}, \ldots, y_{-n+1}\}$ are zero-mean random variables with a known $n \times n$ *covariance matrix*

$$P_0 = E(y_{-j}, y_{-k}), \; j, k \in \{0, n-1\}.$$

Also assume that

$$E(u_i, y_i) = 0 \text{ for } -n+1 \leq j \leq 0 \text{ and } i \geq 0,$$

which ensures (Kailath, Sayed et al. 2000) that

$$E(u_i, y_i) = 0\,,\; i \geq j \geq 0\,.$$

In other words, that the noise is statistically independent from the process to be estimated. Under some other basic conditions (Kailath, Sayed et al. 2000) this difference equation can be re-written as

$$\vec{x}_{i+1} \equiv \begin{bmatrix} y_{i+1} \\ y_i \\ y_{i-1} \\ \vdots \\ y_{i-n+2} \end{bmatrix} = \underbrace{\begin{bmatrix} a_0 & a_1 & \dots & a_{n-2} & a_{n-1} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y_i \\ y_{i-1} \\ y_{i-2} \\ \vdots \\ y_{i-n+1} \end{bmatrix}}_{\vec{x}_i} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}}_{G} u_i$$

which leads to the *state-space model*

$$\vec{x}_{i+1} = A\vec{x}_i + Gu_i$$

$$\vec{y}_i = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \vec{x}_i$$

or the more general form

$$\vec{x}_{i+1} = A\vec{x}_i + Gu_i \tag{3.1}$$

$$\vec{y}_i = H_i\vec{x}_i\,. \tag{3.2}$$

Equation (3.1) represents the way a new state $\vec{x}_{i+1}$ is modeled as a linear combination of both the previous state $\vec{x}_i$ and some *process noise* u_i. Equation (3.2) describes the way the process measurements or *observations* $\vec{y}_i$ are derived from the internal state $\vec{x}_i$. These two equations are often referred to respectively as the *process model* and the *measurement model*, and they serve as the basis for virtually all linear estimation methods, such as the *Kalman filter* described below.

3.2 The Observer Design Problem

There is a related general problem in the area of linear systems theory generally called the *observer design problem*. The basic problem is to determine (estimate) the internal *states* of a linear system, given access only to the system's *outputs*. (Access to the system's control inputs is also presumed, but we omit that aspect here. See for example (Kailath, Sayed et al. 2000) for more information.) This is akin to what people often think of as the "black box" problem where you have access to some signals coming from the box (the outputs) but you cannot directly observe what's inside.

The many approaches to this basic problem are typically based on the state-space model presented in the previous section. There is typically a *process model* that models the transformation of the process state. This can usually be represented as a linear stochastic difference equation similar to equation (3.1):

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}. \tag{3.3}$$

In addition there is some form of *measurement model* that describes the relationship between the process state and the measurements. This can usually be represented with a linear expression similar to equation (3.2):

$$z_k = Hx_k + v_k. \tag{3.4}$$

The terms w_k and v_k are random variables representing the process and measurement noise respectively. Note that in equation (3.4) we changed the dependent variable to z_k instead of y_k as in equation (3.2). The rationale is to reinforce the notion that the measurements to not have to be of elements of the state specifically, but can be any linear combination of the state elements.

Measurement and Process Noise

We consider here the common case of noisy sensor measurements. There are many sources of noise in such measurements. For example, each type of sensor has fundamental limitations related to the associated physical medium, and when pushing the envelope of these limitations the signals are typically degraded. In addition, some amount of random electrical noise is added to the signal via the sensor and the electrical circuits. The time-varying ratio of "pure" signal to the electrical noise continuously affects the *quantity* and *quality* of the information. The result is that information obtained from any one sensor must be qualified as it is interpreted as part of an overall sequence of estimates, and analytical measurement models typically incorporate some notion of random measurement noise or uncertainty as shown above.

There is the additional problem that the actual state transform model is completely unknown. While we can make predictions over relatively short intervals using models based on recent state transforms, such predictions assume that the transforms are predictable, which is not always the case. The result is that like sensor information, ongoing estimates of the state must be qualified as they are combined with measurements in an overall sequence of estimates. In addition, process models typically incorporate some notion of random motion or uncertainty as shown above.

4. The Kalman Filter

Within the significant toolbox of mathematical tools that can be used for stochastic estimation from noisy sensor measurements, one of the most well-known and often-used tools is what is known as the *Kalman filter*. The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem (Kalman 1960). A very "friendly" introduction to the general idea of the Kalman filter is offered in Chapter 1 of (Maybeck 1979)—which is available from the above Kalman filter web site, *and* we have included it (with permission) in this course pack. A more complete introductory discussion can be found in (Sorenson 1970), which also contains some interesting historical narrative. More extensive references include (Gelb 1974; Maybeck 1979; Lewis 1986; Jacobs 1993; Brown and Hwang 1996; Grewal and Andrews 2001). In addition, for many years we have maintained a web site dedicated to the Kalman filter. This site contains links to related work, papers, books, and even some software including a new Java-based *Kalman Filter Learning Tool*.

`http://www.cs.unc.edu/~welch/kalman/`

The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is *optimal* in the sense that it minimizes the estimated *error* covariance—when some presumed conditions are met. Since the time of its introduction, the *Kalman filter* has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This is likely due in large part to advances in digital computing that made the use of the filter practical, but also to the relative simplicity and robust nature of the filter itself. Rarely do the conditions necessary for optimality actually exist, and yet the filter apparently works well for many applications in spite of this situation.

Of particular note here, the Kalman filter has been used extensively for tracking in interactive computer graphics. We use a *single-constraint-at-a-time* Kalman filter (see Section 5.4 on page 41) in our HiBall Tracking System (Welch, Bishop et al. 1999; Welch, Bishop et al. 2001) which is commercially available from 3rdTech (3rdTech 2000). It has also been used for motion prediction (Azuma and Bishop 1994; Azuma 1995), and it is used for multi-sensor (inertial-acoustic) fusion in the commercial Constellation™ wide-area tracking system by Intersense (Foxlin, Harrington et al. 1998; Intersense 2000). See also (Fuchs (Foxlin) 1993; Van Pabst and Krekel 1993; Azarbayejani and Pentland 1994; Emura and Tachi 1994; Emura and Tachi 1994; Mazuryk and Gervautz 1995).

4.1 The Discrete Kalman Filter

This section describes the filter in its original formulation (Kalman 1960) where the measurements occur and the state is estimated at discrete points in time.

4.1.1 The Process to be Estimated

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

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}, \tag{4.1}$$

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

$$z_k = Hx_k + v_k. \tag{4.2}$$

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

$$p(w) \sim N(0, Q), \tag{4.3}$$

$$p(v) \sim N(0, R). \tag{4.4}$$

In practice, the *process noise covariance* Q and *measurement noise covariance* R matrices might change with each time step or measurement, however here we assume they are constant.

The $n \times n$ matrix A in the difference equation equation (4.1) relates the state at the previous time step $k-1$ to the state at the current step k, in the absence of either a driving function or process noise. Note that in practice A might change with each time step, but here we assume it is constant. The $n \times l$ matrix B relates the optional control input $u \in \Re^l$ to the state x. The $m \times n$ matrix H in the measurement equation equation (4.2) relates the state to the measurement z_k. In practice H might change with each time step or measurement, but here we assume it is constant.

4.1.2 The Computational Origins of the Filter

We define $\hat{x}_k^- \in \Re^n$ (note the "super minus") to be our *a priori* state estimate at step k given knowledge of the process prior to step k, and $\hat{x}_k \in \Re^n$ to be our *a posteriori* state estimate at step k given measurement z_k. We can then define *a priori* and *a posteriori* estimate errors as

$$e_k^- \equiv x_k - \hat{x}_k^-, \text{ and}$$

$$e_k \equiv x_k - \hat{x}_k.$$

The *a priori* estimate error covariance is then

$$P_k^- = E[e_k^- e_k^{-T}], \tag{4.5}$$

and the *a posteriori* estimate error covariance is

$$P_k = E[e_k e_k^T]. \tag{4.6}$$

In deriving the equations for the Kalman filter, we begin with the goal of finding an equation that computes an *a posteriori* state estimate $\hat{x}_k$ as a linear combination of an *a priori* estimate $\hat{x}_k^-$ and a weighted difference between an actual measurement z_k and a measurement prediction $H\hat{x}_k^-$ as shown below in equation (4.7). Some justification for equation (4.7) is given in "The Probabilistic Origins of the Filter" found below.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \tag{4.7}$$

The difference $(z_k - H\hat{x}_k^-)$ in equation (4.7) is called the measurement *innovation*, or the *residual*. The residual reflects the discrepancy between the predicted measurement $H\hat{x}_k^-$ and the actual measurement z_k. A residual of zero means that the two are in complete agreement.

The $n \times m$ matrix K in equation (4.7) is chosen to be the *gain* or *blending factor* that minimizes the *a posteriori* error covariance equation (4.6). This minimization can be accomplished by first substituting equation (4.7) into the above definition for e_k, substituting that into equation (4.6), performing the indicated expectations, taking the derivative of the trace of the result with respect to K, setting that result equal to zero, and then solving for K. For more details see (Maybeck 1979; Jacobs 1993; Brown and Hwang 1996). One form of the resulting K that minimizes equation (4.6) is given by[1]

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ &= \frac{P_k^- H^T}{H P_k^- H^T + R} \end{aligned} . \tag{4.8}$$

Looking at equation (4.8) we see that as the measurement error covariance R approaches zero, the gain K weights the residual more heavily. Specifically,

$$\lim_{R_k \to 0} K_k = H^{-1}.$$

1. All of the Kalman filter equations can be algebraically manipulated into to several forms. Equation equation (4.8) represents the Kalman gain in one popular form.

On the other hand, as the *a priori* estimate error covariance P_k^- approaches zero, the gain K weights the residual less heavily. Specifically,

$$\lim_{P_k^- \to 0} K_k = 0 .$$

Another way of thinking about the weighting by K is that as the measurement error covariance R approaches zero, the actual measurement z_k is "trusted" more and more, while the predicted measurement $H\hat{x}_k^-$ is trusted less and less. On the other hand, as the *a priori* estimate error covariance P_k^- approaches zero the actual measurement z_k is trusted less and less, while the predicted measurement $H\hat{x}_k^-$ is trusted more and more.

4.1.3 The Probabilistic Origins of the Filter

The justification for equation (4.7) is rooted in the probability of the *a priori* estimate $\hat{x}_k^-$ conditioned on all prior measurements z_k (Bayes' rule). For now let it suffice to point out that the Kalman filter maintains the first two moments of the state distribution,

$$E[x_k] = \hat{x}_k$$

$$E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k .$$

The *a posteriori* state estimate equation (4.7) reflects the mean (the first moment) of the state distribution— it is normally distributed if the conditions of equation (4.3) and equation (4.4) are met. The *a posteriori* estimate error covariance equation (4.6) reflects the variance of the state distribution (the second non-central moment). In other words,

$$\begin{aligned} p(x_k | z_k) &\sim N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) \\ &= N(\hat{x}_k, P_k) . \end{aligned}$$

For more details on the probabilistic origins of the Kalman filter, see (Brown and Hwang 1996).

4.1.4 The Discrete Kalman Filter Algorithm

We will begin this section with a broad overview, covering the "high-level" operation of one form of the discrete Kalman filter (see the previous footnote). After presenting this high-level view, we will narrow the focus to the specific equations and their use in this version of the filter.

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: *time update* equations and *measurement update* equations. The time update equations 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. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

The time update equations can also be thought of as *predictor* equations, while the measurement update equations can be thought of as *corrector* equations. Indeed the final estimation algorithm resembles that of a *predictor-corrector* algorithm for solving numerical problems as shown below in Figure 4.1.

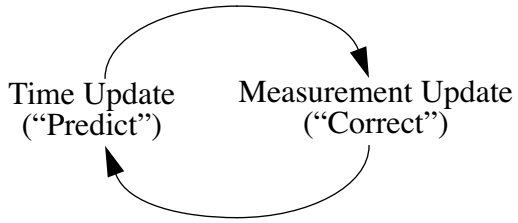

Figure 4.1: The ongoing discrete Kalman filter cycle. The *time update* projects the current state estimate ahead in time. The *measurement update* adjusts the projected estimate by an actual measurement at that time.

The specific equations for the time and measurement updates are presented below in table 4.1 and table 4.2.

Table 4.1: Discrete Kalman filter time update equations.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{4.9}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.10}$$

Again notice how the time update equations in table 4.1 project the state and covariance estimates forward from time step $k-1$ to step k. A and B are from equation (4.1), while Q is from equation (4.3). Initial conditions for the filter are discussed in the earlier references.

Table 4.2: Discrete Kalman filter measurement update equations.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{4.11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4.12}$$

$$P_k = (I - K_k H)P_k^- \tag{4.13}$$

The first task during the measurement update is to compute the Kalman gain, K_k. Notice that the equation given here as equation (4.11) is the same as equation (4.8). 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 as in equation (4.12). Again equation (4.12) is simply equation (4.7) repeated here for completeness. The final step is to obtain an *a posteriori* error covariance estimate via equation (4.13).

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 (Brown and Hwang 1996) 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. Figure 4.2 below offers a complete picture of the operation of the filter, combining the high-level diagram of Figure 4.1 with the equations from table 4.1 and table 4.2.

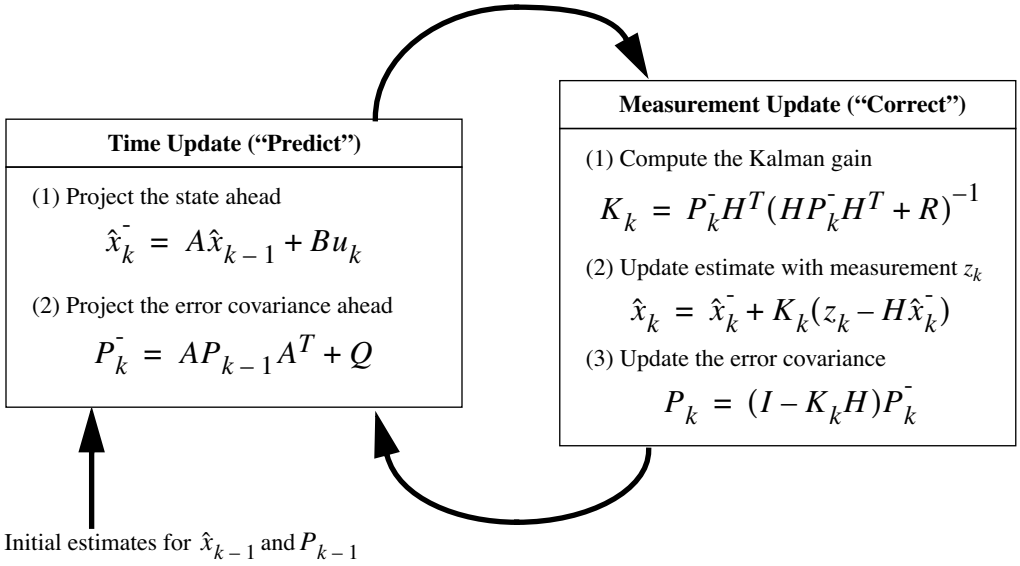

Figure 4.2: A complete picture of the operation of the Kalman filter, combining the high-level diagram of Figure 4.1 with the equations from table 4.1 and table 4.2.

4.2 The Extended Kalman Filter (EKF)

4.2.1 The Process to be Estimated

As described above in Section 4.1.1, the Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by a *linear* stochastic difference equation. But what happens 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 such situations. A Kalman filter that linearizes about the current mean and covariance is referred to as an *extended Kalman filter* or EKF.

In something akin to a Taylor series, we can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships. To do so, we must begin by modifying some of the material presented in Section 4.1. Let us assume that our process again has a state vector $x \in \Re^n$, but that the process is now governed by the *non-linear* stochastic difference equation

$$x_k = f(x_{k-1}, u_k, w_{k-1}), \tag{4.14}$$

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

$$z_k = h(x_k, v_k), \tag{4.15}$$

where the random variables w_k and v_k again represent the process and measurement noise as in equation (4.3) and equation (4.4). In this case the *non-linear* function f in the difference equation equation (4.14) 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 and the zero-mean process noise w_k. The *non-linear* function h in the measurement equation equation (4.15) 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

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0) \tag{4.16}$$

and

$$\tilde{z}_k = h(\tilde{x}_k, 0), \tag{4.17}$$

where $\hat{x}_k$ is some *a posteriori* estimate of the state (from a previous time step k).

It is important to note that a fundamental flaw of the EKF is that the distributions (or 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. Some interesting work has been done by Julier et al. in developing a variation to the EKF, using methods that preserve the normal distributions throughout the non-linear transformations (Julier and Uhlmann 1996).

4.2.2 The Computational Origins of the Filter

To estimate a process with non-linear difference and measurement relationships, we begin by writing new governing equations that linearize an estimate about equation (4.16) and equation (4.17),

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1}\,, \tag{4.18}$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k\,. \tag{4.19}$$

where

- x_k and z_k are the actual state and measurement vectors,
- $\tilde{x}_k$ and $\tilde{z}_k$ are the approximate state and measurement vectors from equation (4.16) and equation (4.17),
- $\hat{x}_k$ is an *a posteriori* estimate of the state at step k,
- the random variables w_k and v_k represent the process and measurement noise as in equation (4.3) and equation (4.4).
- A is the Jacobian matrix of partial derivatives of f with respect to x, that is

$$A_{[i,\,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_k, 0)\,,$$

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

$$W_{[i,\,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_k, 0)\,,$$

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

$$H_{[i,\,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0)\,,$$

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

$$V_{[i,\,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0)\,.$$

Note that for simplicity in the notation we do not use the time step subscript k with the Jacobians A, W, H, and V, even though they are in fact different at each time step.

Now we define a new notation for the prediction error,

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k \,, \tag{4.20}$$

and the measurement residual,

$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k \,. \tag{4.21}$$

Remember that in practice one does not have access to x_k in equation (4.20), it is the *actual* state vector, i.e. the quantity one is trying to estimate. On the other hand, one *does* have access to z_k in equation (4.21), it is the actual measurement that one is using to estimate x_k. Using equation (4.20) and equation (4.21) we can write governing equations for an *error process* as

$$\tilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k \,, \tag{4.22}$$

$$\tilde{e}_{z_k} \approx H\tilde{e}_{x_k} + \eta_k \,, \tag{4.23}$$

where ε_k and η_k represent new independent random variables having zero mean and covariance matrices WQW^T and VRV^T, with Q and R as in (4.3) and (4.4) respectively.

Notice that the equations equation (4.22) and equation (4.23) are linear, and that they closely resemble the difference and measurement equations equation (4.1) and equation (4.2) from the discrete Kalman filter. This motivates us to use the actual measurement residual $\tilde{e}_{z_k}$ in equation (4.21) and a second (hypothetical) Kalman filter to estimate the prediction error $\tilde{e}_{x_k}$ given by equation (4.22). This estimate, call it $\hat{e}_k$, could then be used along with equation (4.20) to obtain the *a posteriori* state estimates for the original non-linear process as

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \,. \tag{4.24}$$

The random variables of equation (4.22) and equation (4.23) have approximately the following probability distributions (see the previous footnote):

$$p(\tilde{e}_{x_k}) \sim N(0, E[\tilde{e}_{x_k}\tilde{e}_{x_k}^T])$$

$$p(\varepsilon_k) \sim N(0, WQ_kW^T)$$

$$p(\eta_k) \sim N(0, VR_kV^T)$$

Given these approximations and letting the predicted value of $\hat{e}_k$ be zero, the Kalman filter equation used to estimate $\hat{e}_k$ is

$$\hat{e}_k = K_k\tilde{e}_{z_k} \,. \tag{4.25}$$

By substituting equation (4.25) back into equation (4.24) and making use of equation (4.21) we see that we do not actually need the second (hypothetical) Kalman filter:

$$\begin{aligned} \hat{x}_k &= \tilde{x}_k + K_k \tilde{e}_{z_k} \\ &= \tilde{x}_k + K_k (z_k - \tilde{z}_k) \end{aligned} \tag{4.26}$$

Equation equation (4.26) can now be used for the measurement update in the extended Kalman filter, with $\tilde{x}_k$ and $\tilde{z}_k$ coming from equation (4.16) and equation (4.17), and the Kalman gain K_k coming from equation (4.11) with the appropriate substitution for the measurement error covariance.

The complete set of EKF equations is shown below in table 4.3 and table 4.4. Note that we have substituted $\hat{x}_k^-$ for $\tilde{x}_k$ to remain consistent with the earlier "super minus" a priori notation, and that we now attach the subscript k to the Jacobians A, W, H, and V, to reinforce the notion that they are different at (and therefore must be recomputed at) each time step.

Table 4.3: EKF time update equations.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0) \tag{4.27}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{4.28}$$

As with the basic discrete Kalman filter, the time update equations in table 4.3 project the state and covariance estimates from the previous time step $k-1$ to the current time step k. Again f in equation (4.27) comes from equation (4.16), A_k and W_k are the process Jacobians at step k, and Q_k is the process noise covariance equation (4.3) at step k.

Table 4.4: EKF measurement update equations.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{4.29}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \tag{4.30}$$

$$P_k = (I - K_k H_k) P_k^- \tag{4.31}$$

As with the basic discrete Kalman filter, the measurement update equations in table 4.4 correct the state and covariance estimates with the measurement z_k. Again h in equation (4.30) comes from equation (4.17), H_k and V are the measurement Jacobians at step k, and R_k is the measurement noise covariance equation (4.4) at step k. (Note we now subscript R allowing it to change with each measurement.)

The basic operation of the EKF is the same as the linear discrete Kalman filter as shown in Figure 4.1. Figure 4.3 below offers a complete picture of the operation of the EKF, combining the high-level diagram of Figure 4.1 with the equations from table 4.3 and table 4.4.

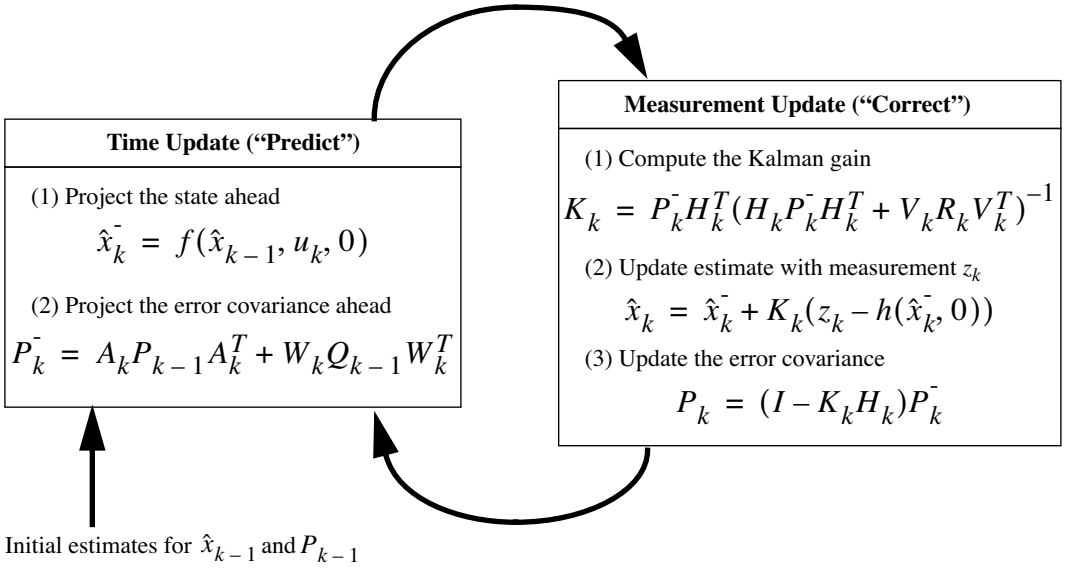

Figure 4.3: A complete picture of the operation of the *extended* Kalman filter, combining the high-level diagram of Figure 4.1 with the equations from table 4.3 and table 4.4.

An important feature of the EKF is that the Jacobian H_k in the equation for the Kalman gain K_k serves to correctly propagate or "magnify" only the relevant component of the measurement information. For example, if there is not a one-to-one mapping between the measurement z_k and the state via h, the Jacobian H_k affects the Kalman gain so that it only magnifies the portion of the residual $z_k - h(\hat{x}_k, 0)$ that does affect the state. Of course if over *all* measurements there is *not* a one-to-one mapping between the measurement z_k and the state via h, then as you might expect the filter will quickly diverge. In this case the process is *unobservable*.

4.3 An Example: Estimating a Random Constant

In the previous two sections we presented the basic form for the discrete Kalman filter, and the extended Kalman filter. To help in developing a better feel for the operation and capability of the filter, we present a very simple example here.

4.3.1 The Process Model

In this simple example let us attempt to estimate a scalar random constant, a voltage for example. Let's assume that we have the ability to take measurements of the constant, but that the measurements are corrupted by a 0.1 volt RMS *white* measurement noise (e.g. our analog to digital converter is not very accurate). In this example, our process is governed by the linear difference equation

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_k + w_k \\ &= x_{k-1} + w_k \end{aligned},$$

with a measurement $z \in \Re^1$ that is

$$\begin{aligned} z_k &= Hx_k + v_k \\ &= x_k + v_k \end{aligned}.$$

The state does not change from step to step so $A = 1$. There is no control input so $u = 0$. Our noisy measurement is of the state directly so $H = 1$. (Notice that we dropped the subscript k in several places because the respective parameters remain constant in our simple model.)

4.3.2 The Filter Equations and Parameters

Our time update equations are

$$\hat{x}_k^- = \hat{x}_{k-1},$$

$$P_k^- = P_{k-1} + Q,$$

and our measurement update equations are

$$\begin{aligned} K_k &= P_k^-(P_k^- + R)^{-1} \\ &= \frac{P_k^-}{P_k^- + R} \end{aligned}, \tag{4.32}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-),$$

$$P_k = (1 - K_k)P_k^-.$$

Presuming a very small process variance, we let $Q = 1e-5$. (We could certainly let $Q = 0$ but assuming a small but non-zero value gives us more flexibility in "tuning" the filter as we will demonstrate below.) Let's assume that from experience we know that the

true value of the random constant has a standard normal probability distribution, so we will "seed" our filter with the guess that the constant is 0. In other words, before starting we let $\hat{x}_{k-1} = 0$.

Similarly we need to choose an initial value for P_{k-1}, call it P_0. If we were absolutely certain that our initial state estimate $\hat{x}_0 = 0$ was correct, we would let $P_0 = 0$. However given the uncertainty in our initial estimate $\hat{x}_0$, choosing $P_0 = 0$ would cause the filter to initially and always believe $\hat{x}_k = 0$. As it turns out, the alternative choice is not critical. We could choose almost any $P_0 \neq 0$ and the filter would *eventually* converge. We'll start our filter with $P_0 = 1$.

4.3.3 The Simulations

To begin with, we randomly chose a scalar constant $z = -0.37727$ (there is no "hat" on the z because it represents the "truth"). We then simulated 50 distinct measurements z_k that had error normally distributed around zero with a standard deviation of 0.1 (remember we presumed that the measurements are corrupted by a 0.1 volt RMS *white* measurement noise). We could have generated the individual measurements within the filter loop, but pre-generating the set of 50 measurements allowed me to run several simulations with the same exact measurements (i.e. same measurement noise) so that comparisons between simulations with different parameters would be more meaningful.

In the first simulation we fixed the measurement variance at $R = (0.1)^2 = 0.01$. Because this is the "true" measurement error variance, we would expect the "best" performance in terms of balancing responsiveness and estimate variance. This will become more evident in the second and third simulation. Figure 4.4 depicts the results of this first simulation. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve.

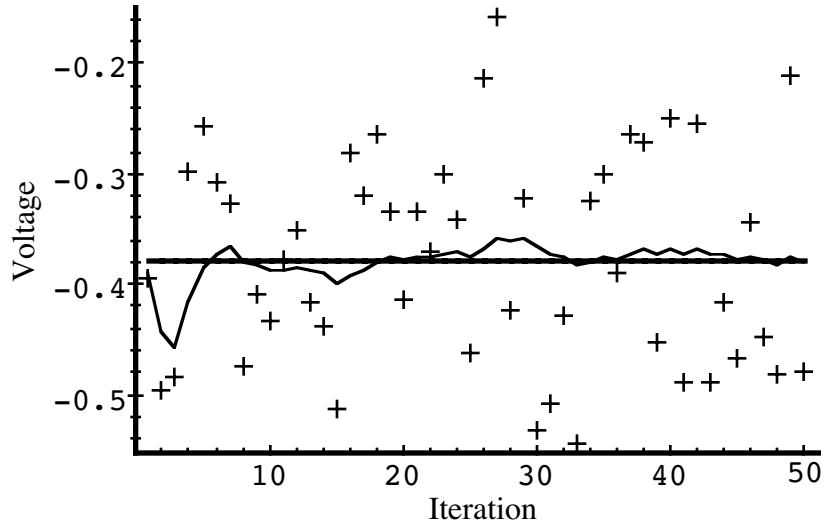

Figure 4.4: The first simulation: $R = (0.1)^2 = 0.01$. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve.

When considering the choice for P_0 above, we mentioned that the choice was not critical as long as $P_0 \neq 0$ because the filter would eventually converge. Below in Figure 4.5 we have plotted the value of P_k versus the iteration. By the 50^{th} iteration, it has settled from the initial (rough) choice of 1 to approximately 0.0002 (Volts^2).

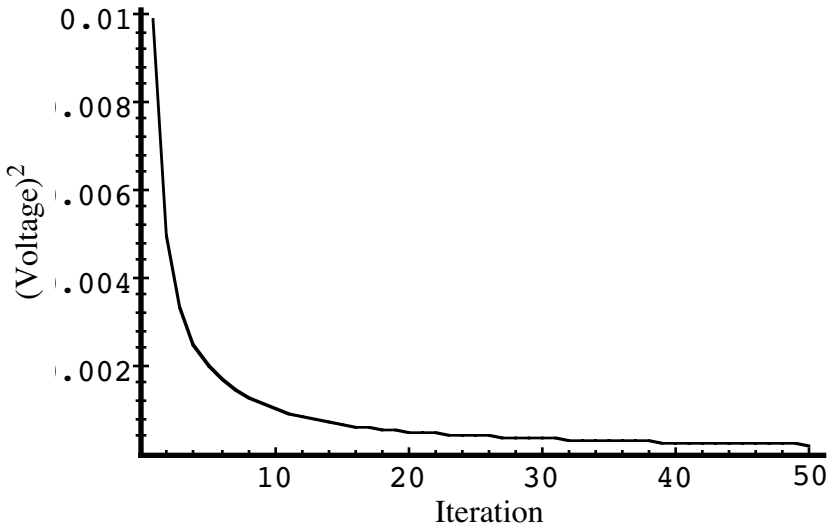

Figure 4.5: After 50 iterations, our initial (rough) error covariance P_k choice of 1 has settled to about 0.0002 (Volts^2).

In Section 5.1 under the topic “Parameter Estimation or Tuning” we briefly discussed changing or “tuning” the parameters Q and R to obtain different filter performance. In Figure 4.6 and Figure 4.7 below we can see what happens when R is increased or

decreased by a factor of 100 respectively. In Figure 4.6 the filter was told that the measurement variance was 100 times greater (i.e. $R = 1$) so it was "slower" to believe the measurements.

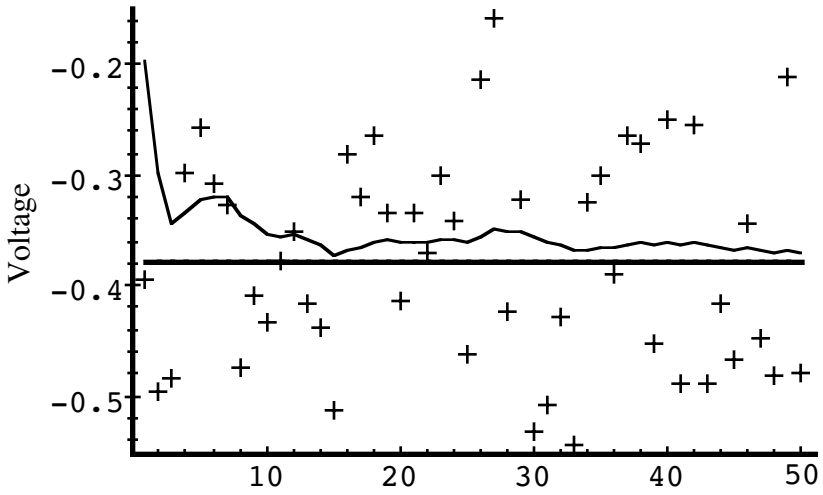

Figure 4.6: Second simulation: $R = 1$. The filter is slower to respond to the measurements, resulting in reduced estimate variance.

In Figure 4.7 the filter was told that the measurement variance was 100 times smaller (i.e. $R = 0.0001$) so it was very "quick" to believe the noisy measurements.

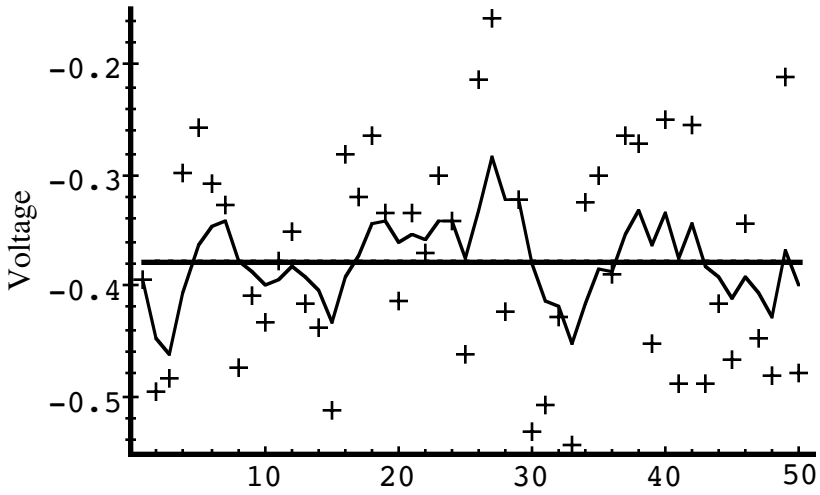

Figure 4.7: Third simulation: $R = 0.0001$. The filter responds to measurements quickly, increasing the estimate variance.

While the estimation of a constant is relatively straight-forward, it clearly demonstrates the workings of the Kalman filter. In Figure 4.6 in particular the Kalman "filtering" is evident as the estimate appears considerably smoother than the noisy measurements.

5. Other Topics

5.1 Parameter Estimation or Tuning

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 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*.

Under conditions where Q and R.are in fact constant, both the estimation error covariance P_k and the Kalman gain K_k will stabilize quickly and then remain constant. If this is the case, these parameters can be pre-computed by either running the filter off-line, or for example by determining the steady-state value of P_k as described in (Grewal and Andrews 2001).

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.

See (Welch 1996) for more information on this topic.

5.2 Multi-Modal (Multiple Model) Filters

5.2.1 Random Processes and the Kalman Filter

The Kalman filter is based on the assumption of a continuous system that can be modeled as a normally distributed random process X, with mean $\hat{\vec{x}}$ (the state) and variance P (the error covariance). In other words,

$$X \sim N(\hat{\vec{x}}, P) .$$

Similarly the Kalman filter is based on the assumption that the output of the system can be modeled as a random process Z which is a linear function of the state $\hat{\vec{x}}$ plus an independent, normally distributed, zero-mean white noise process V,[1]

$$\vec{z} = H\hat{\vec{x}} + \vec{v} \tag{5.1}$$

where $X \sim N(\hat{\vec{x}}, P)$, $V \sim N(0, R)$, and $E\{XV\} = 0$. From equations (2.12) and (2.14) we have

$$Z \sim N(H\hat{\vec{x}}, HPH^T + R) .$$

However, the expression $HPH^T + R$ reflects the filter's *estimate* of the measurement residuals (innovations), not the actual residuals. This becomes clear when one examines the update expressions for P in the Kalman filter: P does not depend on the measurement residual. The effect of this is that the expression $HPH^T + R$ may indicate some small residual variance, when in fact at particular points in time the variance is relatively large. This is indeed exactly the case when one is simultaneously considering multiple models for a process—one of the models, or some combination of them, is "right" and actually has small residuals, while others are "wrong" and will suffer from large residuals. Thus when one is computing the likelihood of a residual for the purpose of comparing model performance, one must consider the liklihood of the *actual* measurement $\vec{z}$ at each time step, given the *expected* performance of each model.

The Likelihood of the Measurements Given a Particular Model

Given equations (2.13) and (2.15), we can use the following conditional probability density function as an indicator of the *likelihood* of a measurement $\vec{z}$ at step k:

$$f(\vec{z}|\mu) = \frac{1}{(2\pi|C_\mu|)^{n_\mu/2}} e^{-\frac{1}{2}(\vec{z} - H_\mu \hat{\vec{x}}_\mu^-)^T C_\mu^{-1} (\vec{z} - H_\mu \hat{\vec{x}}_\mu)} , \tag{5.2}$$

1. Recall that "white" means that the spectral density is constant and the autocorrelation is a delta function. In other words, the output of the noise source at any one instant in time is independent from that at any other instant in time.

where

$$C_\mu = H_\mu P_\mu^- H_\mu^T + R_\mu .$$

We have omitted the subscript k for clarity. Note again that the state vector $\hat{x}_\mu$ and error covariance matrix P_μ^- are the *a priori* (predicted) versions at step k, already computed at each filter prediction step using equation (4.9) and equation (4.10). In other words, the density is conditioned on the model μ and all of its associated *a priori* (predicted) parameters.

5.2.2 Fixed Multiple Models

We begin with the case where we believe that there is *one* correct model for the process, and that the model is fixed or does not change over time, however we don't know what that model is. Over time, as the filter reaches a steady state, we want to converge on a choice for the single most likely model. For this approach let us assume that the *correct* model M is one of r possible *known* fixed models,

$$M \in \{\mu_j\}_{j=1}^r .$$

The Probability of a Particular Model Given the Measurements

Given a new measurement $\hat{z}$ at time step k, and associated *a priori* state and covariance estimates from equation (4.9) and equation (4.10), we can use equation (5.2) to compute the recursive probability $p_j(k)$ that candidate model μ_j is the correct model at that time:

$$p_j(k) = \frac{f(\hat{z}|\mu_j) p_j(k-1)}{\sum_{h=1}^{r} f(\hat{z}|\mu_h) p_h(k-1)} . \tag{5.3}$$

One would initialize $p_j(0)$ with some *a priori* estimate of the probability that μ_j is the correct model. For example, one could consider all models equally likely to begin with, and set

$$p_j(0) = \frac{1}{r}, \; j = 1, 2, \ldots, r .$$

Note that $f(\hat{z}|\mu_j)$ and $p_j(0)$ are scalars, and at every time step k,

$$\sum_{j=1}^{r} p_j(k) = 1 .$$

The Final Model-Conditioned Estimate

The final combined or *model-conditioned* estimate of the state $\hat{x}_k$ and error covariance P_k are computed as a weighted combination of each candidate filter's *a posteriori* state and error covariance estimates. The weight for each candidate model is the model probability given by equation (5.3). The final model-conditioned state estimate is computed as

$$\widehat{x}_k = \sum_{j=1}^{r} p_j(k)\hat{x}_{k,\mu_j}, \text{ and} \tag{5.4}$$

the final model-conditioned error covariance as

$$P_k = \sum_{j=1}^{r} p_j(k)[P_{k,\mu_j} + \varepsilon_{\mu_j}\varepsilon_{\mu_j}^T], \tag{5.5}$$

where $\varepsilon_{\mu_j} = \widehat{x}_k - \hat{x}_{k,\mu_j}$.

The Algorithm

To begin with, one would instantiate r independent Kalman filters, one for each of the r candidate models. Each of these filters would then be run independently, in parallel, with the addition of the necessary individual density and final probability computations.

At each *time update* (see Figure 4.1) one would compute the normal *a priori* Kalman filter elements (see table 4.1), and then

a. using the conditional density function given in equation (5.2), compute the liklihood of the current (actual) measurement $\vec{z}$ for each candidate model μ_j;

b. using the previous probability $p_j(k-1)$ for each candidate model μ_j, use the recursive equation (5.3) to compute the probability $p_j(k)$ that each individual model μ_j is correct;

c. for each candidate model μ_j, compute the *a posteriori* (corrected) state estimate $\hat{x}_{k,\mu_j}$ and error covariance P_{k,μ_j} using equation (4.12) and equation (4.13);

d. given each candidate filter's *a posteriori (corrected)* state estimate $\hat{x}_{k,\mu_j}$, compute the final *model-conditioned* state estimate $\widehat{x}_k$ using equation (5.4); and

e. if desired, given each candidate filter's *a posteriori (corrected)* error covariance estimate P_{k,μ_j}, compute the final model-conditioned error covariance P_k using equation (5.5).

Convergence of the Mode Estimates

As described in (Bar-Shalom and Li 1993), the final mode-conditioned state estimate will converge to agree with one of the models, if one of the models is the correct one. In any case, it will converge to some constant mode represented by a fixed weighting of the individual multiple models.

If the actual mode is not constant, i.e. if the process can be switching or varying between different models, one can use various ad hoc methods to prevent convergence on a single mode. For example,

a. One can impose an artificial lower bound on the model probabilities,

b. impose a finite memory (sliding window) on the likelihood function, or

c. impose an exponential decay on the likelihood function.

A problem with using ad hoc means of varying the blending of fixed multiple models is that the error in the incorrect models (at any moment) can grow unbounded, i.e. the incorrect filters can get lost. Thus the filters might have to be re-initialized.

5.2.3 Dynamic Multiple Model Method

The multiple-model approach described in Section 5.2.2 is appropriate for systems where we believe there is *one* correct model for the process, and that the model is *fixed*. However there are situations where the choice from a set of candidate models varies continuously while the filter is in operation.´In such a case one cannot make a fixed *a priori* choice of filter parameters. Instead one could operate a set of candidate filters in parallel (similar to Section 5.2.2) and use a continuously varying model-conditioned combination of the candidate state and error covariance estimates.

The *dynamic multiple model approach* is virtually identical to the *fixed* approach outlined in Section 5.2.2, with the exception of the model probability given by equation (5.3). In the dynamic case we do not want the probabilities to converge to fixed values, but to remain free to change with each new measurement. Given a new measurement $\vec{z}$ at time

step k, and associated *a priori* state and covariance estimates from equation (4.9) and equation (4.10), one could compute the probability $p_j(k)$ that candidate model μ_j is the correct model at that time simply using

$$p_j(k) = \frac{f(\vec{z}|\mu_j)}{\sum_{h=1}^{r} f(\vec{z}|\mu_h)}. \tag{5.6}$$

The algorithm would remain the same as in Section 5.2.2, except that in step (b) one would use equation (5.6) instead of equation (5.3).

5.3 Hybrid or Multi-Sensor Fusion

Stochastic estimation tools such as the Kalman filter can be used to combine or *fuse* information from different "mediums or sensors for *hybrid systems*. The basic idea is to use the Kalman filter to weight the different mediums most heavily in the circumstances where they each perform best, thus providing more accurate and stable estimates than a system based on any one medium alone. In particular, the *indirect feedback* Kalman filter shown in Figure 5.1 (also called a *complementary* or *error-state* Kalman filter) is often used to combined the two mediums (Maybeck 1979). In such a configuration, the Kalman filter is used to estimate the *difference* between the current inertial and optical (or acoustic) outputs, i.e. it continually estimates the *error* in the inertial estimates by using the optical system as a second (redundant) reference. This error estimate is then used to correct the inertial estimates. The *tuning* of the Kalman filter parameters (see "Parameter Estimation or Tuning" on page 35) then adjusts the weight of the correction as a function of frequency. By slightly modifying the Kalman filter, adaptive velocity response can be incorporated also. This can be accomplished by adjusting (in real time) the expected optical measurement error as a function of the magnitude of velocity. The dashed line in Figure 5.1 also indicates the use of inertial estimates to help a image-based optical system to prevent tracking of moving scene objects (i.e. unrelated motion in the environment).

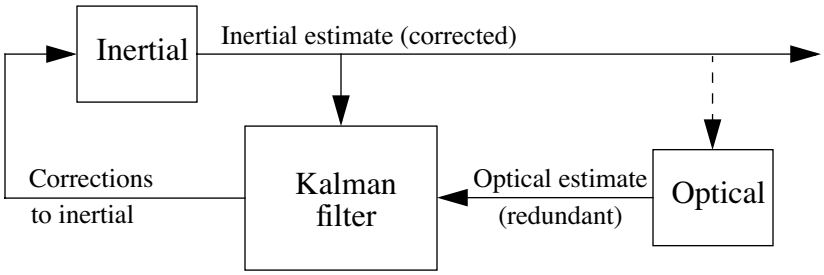

Figure 5.1: The Kalman filter used in an *indirect-feedback* configuration to optimally weight inertial and optical information.

In such a configuration, the Kalman filter uses a common *process model*, but a distinct *measurement model* for each of the inertial and optical subsystems.

5.4 Single-Constraint-at-a-Time (SCAAT)

A conventional approach to pose estimation is to collect a group of sensor measurements and then to attempt to simultaneously solve a system of equations that together completely constrain the solution. For example, the 1991 UNC-Chapel Hill wide-area opto-electronic tracking system (Wang 1990; Ward, Azuma et al. 1992) collected a group of diverse measurements for a variety of LEDs and sensors, and then used a method of simultaneous non-linear equations called *Collinearity* to estimate the pose of the head-worn sensor fixture (Azuma and Ward 1991). There was one equation for each measurement, expressing the constraint that a ray from the front principal point of the sensor lens to the LED, must be collinear with a ray from the rear principal point to the intersection with the sensor. Each estimate made use of typically 20 (or more) measurements that together over-constrained the solution.

This *multiple constraint* method had several drawbacks. First, it had a significantly lower estimate rate due to the need to collect multiple measurements per estimate. Second, the system of non-linear equations did not account for the fact that the sensor fixture continued to move throughout the collection of the sequence of measurements. Instead the method effectively assumes that the measurements were taken simultaneously. The violation of this *simultaneity assumption* could introduce significant error during even moderate motion. Finally, the method provided no means to identify or handle unusually noisy individual measurements. Thus, a single erroneous measurement could cause an estimate to jump away from an otherwise smooth track.

In contrast, there is typically nothing about typical solutions to the observer design problem in general (Section 3.2), or the Kalman filter in particular (see “Parameter Estimation or Tuning” on page 35), that dictates the ordering of measurement information. In 1996 we introduced a new approach to tracking with a Kalman filter, an approach that exploits this flexibility in measurement processing. The basic idea is to update the pose estimate as each new measurement is made, rather than waiting to form a complete collection of measurement. Because single measurements under-constrain the mathematical solution, we refer to the approach as *single-constraint-at-a-time* or SCAAT tracking (Welch 1996; Welch and Bishop 1997). The key is that the single measurements provide *some* information about the tracker state, and thus can be used to incrementally improve a previous estimate. We intentionally fuse each individual “insufficient” measurement immediately as it is obtained. With this approach we are able to generate estimates more frequently, with less latency, with improved accuracy, and we are able to estimate the LED positions on-line concurrently while tracking.

This approach is used in our laboratory-based HiBall Tracking System (Welch, Bishop et al. 1999; Welch, Bishop et al. 2001), the commercial version of the same system (3rdTech 2000), and the commercial systems manufactured by Intersense (Foxlin, Harrington et al. 1998; Intersense 2000). For more information see (Welch 1996; Welch and Bishop 1997), the former which is included at the end of this course pack.

A. Bibliography

3rdTech. (2000, July 15). *3rdTech™* , [HTML]. 3rdTech. Available: http://www.3rdtech.com/ [2000, July 19].

Azarbayejani, A., & Pentland, A. (1994). *Recursive estimation of motion, structure, and focal length* (Technical report 243). Cambridge, MA: Massachusetts Institute of Technology (MIT).

Azuma, R. T. (1995). *Predictive Tracking for Augmented Reality*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC USA.

Azuma, R. T., & Bishop, G. (1994). Improving Static and Dynamic Registration in an Optical See-Through HMD, *Computer Graphics* (SIGGRAPH 94 Conference Proceedings ed., pp. 197-204). Orlando, FL USA: ACM Press, Addison-Wesley.

Azuma, R. T., & Ward, M. (1991). *Space-Resection by Collinearity: Mathematics Behind the Optical Ceiling Head-Tracker* (Technical Report 91-048). Chapel Hill, NC USA: University of North Carolina at Chapel Hill.

Bar-Shalom, Y., & Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*: Artec House, Inc.

Brown, R. G., & Hwang, P. Y. C. (1996). *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions* (Third ed.): Wiley & Sons, Inc.

Emura, S., & Tachi, S. (1994a). Compensation of time lag between actual and virtual spaces by multi-sensor integration, *1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (pp. 363-469). Las Vegas, NV: Institute of Electrical and Electronics Engineers.

Emura, S., & Tachi, S. (1994b). *Sensor Fusion based Measurement of Human Head Motion*. Paper presented at the 3rd IEEE International Workshop on Robot and Human Communication (RO-MAN 94 NAGOYA), Nagoya University, Nagoya, Japan.

Foxlin, E., Harrington, M., & Pfeifer, G. (1998). Constellation™ : A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In M. F. Cohen (Ed.), *Computer Graphics* (SIGGRAPH 98 Conference Proceedings ed., pp. 371-378). Orlando, FL USA: ACM Press, Addison-Wesley.

Fuchs (Foxlin), E. (1993). *Inertial Head-Tracking*. Unpublished M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA USA.

Gelb, A. (1974). *Applied Optimal Estimation*. Cambridge, MA: MIT Press.

Grewal, M., S., & Andrews, A., P. (2001). *Kalman Filtering Theory and Practice Using MATLAB* (Second ed.). New York, NY USA: John Wiley & Sons, Inc.

Intersense. (2000). *Intersense IS-900*, [html]. Intersense. Available: http://www.isense.com/ [2000, April 27].

Jacobs, O. L. R. (1993). *Introduction to Control Theory* (Second ed.): Oxford University Press.

Julier, S., J., & Uhlmann, J., K. (1996). *A General Method for Approximating Nonlinear Transformations of Probability Distributions* (Technical Report). Oxford, UK: Robotics Research Group, Department of Engineering Science, University of Oxford.

Kailath, T., Sayed, A., H., & Hassibi, B. (2000). *Linear Estimation*. Upper Saddle River, NJ USA: Prentice Hall.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME—Journal of Basic Engineering, 82*(Series D), 35-45.

Kelly, D. G. (1994). *Introduction to Probability*: Macmillan Publishing Company.

Lewis, F. L. (1986). *Optimal Estimation with an Introductory to Stochastic Control Theory*: John Wiley & Sons, Inc.

Maybeck, P. S. (1979). *Stochastic models, estimation, and control* (Vol. 141).

Mazuryk, T., & Gervautz, M. (1995). Two-Step Prediction and Image Deflection for Exact Head Tracking in Virtual Environments, *Proceedings of EUROGRAPHICS 95* (EUROGRAPHICS 95 ed., Vol. 14 (3), pp. 30-41).

Sorenson, H. W. (1970, July). Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum, 7,* 63-68.

Van Pabst, J. V. L., & Krekel, P. F. C. (1993). Multi Sensor Data Fusion of Points, Line Segments and Surface Segments in 3D Space, *7th International Conference on Image Analysis and Processing—* (pp. 174-182). Capitolo, Monopoli, Italy: World Scientific, Singapore.

Wang, J.-F. (1990). *A real-time optical 6D tracker for head-mounted display systems*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC USA.

Ward, M., Azuma, R. T., Bennett, R., Gottschalk, S., & Fuchs, H. (1992). A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems, *Symposium on Interactive 3D Graphics* (I3D 99 Conference Proceedings ed., pp. 43-52). Cambridge, MA USA: ACM Press, Addison-Wesley.

Welch, G. (1996). *SCAAT: Incremental Tracking with Incomplete Information*. Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

Welch, G., & Bishop, G. (1997). SCAAT: Incremental Tracking with Incomplete Information. In T. Whitted (Ed.), *Computer Graphics* (SIGGRAPH 97 Conference Proceedings ed., pp. 333-344). Los Angeles, CA, USA (August 3 - 8): ACM Press, Addison-Wesley.

Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D. n. (1999). The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (pp. 1-11). University College London, London, United Kingdom (December 20 - 23): ACM SIGGRAPH, Addison-Wesley.

Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D. n. (2001). High-Performance Wide-Area Optical Tracking—The HiBall Tracking System. *Presence: Teleoperators and Virtual Environments, 10*(1).

B. Related Papers

This appendix includes a sample of some relevant papers that we have permission to reproduce.