An Introduction to the Kalman Filter

by
Greg Welch[1]
and
Gary Bishop[2]

TR 95-041
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175

Abstract

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.

The purpose of this paper is to provide a practical introduction to the discrete Kalman filter. This introduction includes a description and some discussion of the basic discrete Kalman filter, a derivation, description and some discussion of the extended Kalman filter, and a relatively simple (tangible) example with real numbers & results.

---

[1] welch@cs.unc.edu, http://www.cs.unc.edu/~welch
[2] gb@cs.unc.edu, http://www.cs.unc.edu/~gb

1   The Discrete Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [Kalman60]. Since that time, due in large part to advances in digital computing, the *Kalman filter* has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. A very "friendly" introduction to the general idea of the Kalman filter can be found in Chapter 1 of [Maybeck79], while a more complete introductory discussion can be found in [Sorenson70], which also contains some interesting historical narrative. More extensive references include [Gelb74; Grewal93; Maybeck79; Lewis86; Brown92; Jacobs93].

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

$$x_{k+1} = A_k x_k + B u_k + w_k, \tag{1.1}$$

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

$$z_k = H_k x_k + v_k. \tag{1.2}$$

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

$$p(w) \sim N(0, Q), \tag{1.3}$$

$$p(v) \sim N(0, R). \tag{1.4}$$

The $n \times n$ matrix A in the difference equation (1.1) relates the state at time step k to the state at step $k+1$, in the absence of either a driving function or process noise. The $n \times l$ matrix B relates the control input $u \in \Re^l$ to the state x. The $m \times n$ matrix H in the measurement equation (1.2) relates the state to the measurement z_k.

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

$$P_k^- = E[e_k^- e_k^{-T}], \tag{1.5}$$

and the *a posteriori* estimate error covariance is

$$P_k = E[e_k e_k^T].$$
(1.6)

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 $H_k \hat{x}_k^-$ as shown below in (1.7). Some justification for (1.7) is given in "The Probabilistic Origins of the Filter" found below.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-)$$
(1.7)

The difference $(z_k - H_k \hat{x}_k^-)$ in (1.7) is called the measurement *innovation*, or the *residual*. The residual reflects the discrepancy between the predicted measurement $H_k \hat{x}_k^-$ 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 (1.7) is chosen to be the *gain* or *blending factor* that minimizes the *a posteriori* error covariance (1.6). This minimization can be accomplished by first substituting (1.7) into the above definition for e_k, substituting that into (1.6), 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 [Maybeck79; Brown92; Jacobs93]. One form of the resulting K that minimizes (1.6) is given by[1]

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$= \frac{P_k^- H_k^T}{H_k P_k^- H_k^T + R_k} \qquad .$$
(1.8)

Looking at (1.8) we see that as the measurement error covariance R_k approaches zero, the gain K weights the residual more heavily. Specifically,

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

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_k approaches zero, the actual measurement z_k is "trusted" more and more, while the predicted measurement $H_k \hat{x}_k^-$ 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 $H_k \hat{x}_k^-$ is trusted more and more.

---

[1] All of the Kalman filter equations can be algebraically manipulated into to several forms. Equation (1.8) represents the Kalman gain in one popular form.

The Probabilistic Origins of the Filter

The justification for (1.7) 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 (1.7) reflects the mean (the first moment) of the state distribution— it is normally distributed if the conditions of (1.3) and (1.4) are met. The *a posteriori* estimate error covariance (1.6) reflects the variance of the state distribution (the second non-central moment). In other words,

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

For more details on the probabilistic origins of the Kalman filter, see [Maybeck79; Brown92; Jacobs93].

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 1-1.

Time Update
("Predict")

Measurement Update
("Correct")
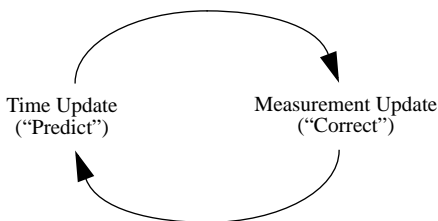
Figure 1-1. The ongoing discrete Kalman filter cycle. The *time update* projects the current state estimate ahead in time. The *measurement update* adjusts the projected estimate by an actual measurement at that time.

The specific equations for the time and measurement updates are presented below in Table 1-1 and Table 1-2.

Table 1-1: Discrete Kalman filter time update equations.

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B u_k \qquad (1.9)$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \qquad (1.10)$$

Again notice how the time update equations in Table 1-1 project the state and covariance estimates from time step k to step $k+1$. A_k and B are from (1.1), while Q_k is from (1.3). Initial conditions for the filter are discussed in the earlier references.

Table 1-2: Discrete Kalman filter measurement update equations.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \qquad (1.11)$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-) \qquad (1.12)$$

$$P_k = (I - K_k H_k) P_k^- \qquad (1.13)$$

The first task during the measurement update is to compute the Kalman gain, K_k. Notice that the equation given here as (1.11) is the same as (1.8). 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 (1.12). Again (1.12) is simply (1.7) repeated here for completeness. The final step is to obtain an *a posteriori* error covariance estimate via (1.13).
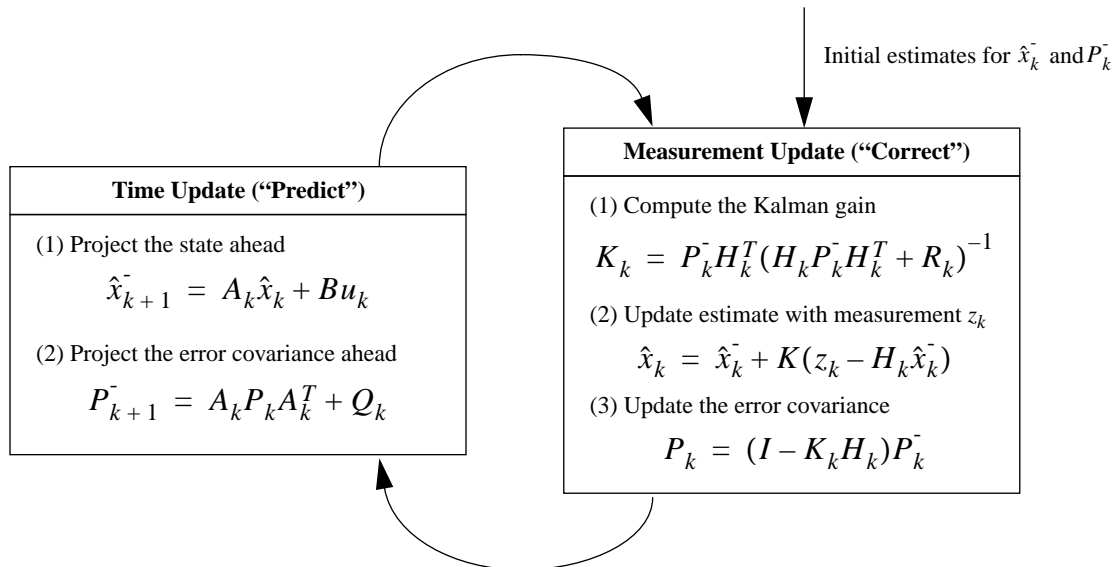
After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the very appealing features of the Kalman filter—it makes practical implementations much more feasible than (for example) an implementation of a Wiener filter [Brown92] 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 1-2 below offers a complete picture of the operation of the filter, combining the high-level diagram of Figure 1-1 with the equations from Table 1-1 and Table 1-2.

Filter Parameters and Tuning

In the actual implementation of the filter, each of the measurement error covariance matrix R_k and the process noise Q_k (given by (1.4) and (1.3) respectively) might be measured prior to operation of the filter. In the case of the measurement error covariance R_k in particular this makes sense— because we need to be able to measure the process (while operating the filter) we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement error.

In the case of Q_k, often times the choice is less deterministic. For example, this noise source is often used to represent the uncertainty in the process model (1.1). Sometimes a very poor model can be used simply by "injecting" enough uncertainty via the selection of Q_k. Certainly in this case one would hope that the measurements of the process would be 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_k and R_k. The tuning is usually performed off-line, frequently with the help of another (distinct) Kalman filter.

Initial estimates for $\hat{x}_k^-$ and P_k^-

Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B u_k$$

(2) Project the error covariance ahead

$$P_{k+1}^- = A_k P_k A_k^T + Q_k$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = (I - K_k H_k) P_k^-$$

Figure 1-2. A complete picture of the operation of the Kalman filter, combining the high-level diagram of Figure 1-1 with the equations from Table 1-1 and Table 1-2.

In closing we note that under conditions where Q_k and R_k .are constant, both the estimation error covariance P_k and the Kalman gain K_k will stabilize quickly and then remain constant (see the filter update equations in Figure 1-2). 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 [Grewal93].

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_k is sometimes changed dynamically during filter operation 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 a case Q_k can be used to model not only the uncertainty in the model, but also the uncertainty of the user's intentions.

2   The Extended Kalman Filter (EKF)

The Process to be Estimated

As described above in section , 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[1].

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 . 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

$$x_{k+1} = f(x_k, u_k, w_k),\qquad(2.1)$$

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

$$z_k = h(x_k, v_k),\qquad(2.2)$$

where the random variables w_k and v_k again represent the process and measurement noise as in (1.3) and (1.4). In this case the *non-linear* function $f(\bullet)$ in the difference equation (2.1) relates the state at time step k to the state at step $k+1$. It includes as parameters any driving function u_k and the zero-mean process noise w_k. The *non-linear* function $h(\bullet)$ in the measurement equation (2.2) relates the state x_k to the measurement z_k.

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

$$\tilde{x}_{k+1} = f(\hat{x}_k, u_k, 0)\qquad(2.3)$$

and

$$\tilde{z}_k = h(\tilde{x}_k, 0),\qquad(2.4)$$

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

---

[1] 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 very 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].

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 (2.3) and (2.4),

$$x_{k+1} \approx \tilde{x}_{k+1} + A(x_k - \hat{x}_k) + Ww_k, \tag{2.5}$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k. \tag{2.6}$$

where

- x_{k+1} and z_k are the actual state and measurement vectors,

- $\tilde{x}_{k+1}$ and $\tilde{z}_k$ are the approximate state and measurement vectors from (2.3) and (2.4),

- $\hat{x}_k$ 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 (1.3) and (1.4).

- A is the Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to x, that is

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

- W is the Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to w,

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

- H is the Jacobian matrix of partial derivatives of $h(\bullet)$ 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 $h(\bullet)$ 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,

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k, \tag{2.7}$$

and the measurement residual,

$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k. \tag{2.8}$$

Remember that in practice one does not have access to x_k in (2.7), 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 (2.8), it is the actual measurement that one is using to estimate x_k. Using (2.7) and (2.8) we can write governing equations for an *error process* as

$$\tilde{e}_{x_{k+1}} \approx A(x_k - \hat{x}_k) + \varepsilon_k, \tag{2.9}$$

$$\tilde{e}_{z_k} \approx H\tilde{e}_{x_k} + \eta_k, \tag{2.10}$$

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 (1.3) and (1.4) respectively.

Notice that the equations (2.9) and (2.10) are linear, and that they closely resemble the difference and measurement equations (1.1) and (1.2) from the discrete Kalman filter. This motivates us to use the actual measurement residual $\tilde{e}_{z_k}$ in (2.8) and a second (hypothetical) Kalman filter to estimate the prediction error $\tilde{e}_{x_k}$ given by (2.9). This estimate, call it $\hat{e}_k$, could then be used along with (2.7) to obtain the *a posteriori* state estimates for the original non-linear process as

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k. \tag{2.11}$$

The random variables of (2.9) and (2.10) 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

$$\hat{e}_k = K_k \tilde{e}_{z_k}. \tag{2.12}$$

By substituting (2.12) back into (2.11) and making use of (2.8) we see that we do not actually need the second (hypothetical) Kalman filter:

$$\begin{aligned} \hat{x}_k &= \tilde{x}_k + K_k \tilde{e}_{z_k} \\ &= \tilde{x}_k + K_k (z_k - \tilde{z}_k) \end{aligned} \tag{2.13}$$

Equation (2.13) can now be used for the measurement update in the extended Kalman filter, with $\tilde{x}_k$ and $\tilde{z}_k$ coming from (2.3) and (2.4), and the Kalman gain K_k coming from (1.11) with the appropriate substitution for the measurement error covariance.

The complete set of EKF equations is shown below in Table 2-1 and Table 2-2. Note that we have substituted $\hat{x}_{k+1}^-$ for $\tilde{x}_{k+1}$ to remain consistent with the earlier "super minus" 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 2-1: EKF time update equations.

$$\hat{x}_{k+1}^- = f(\hat{x}_k, u_k, 0) \tag{2.14}$$

$$P_{k+1}^- = A_k P_k A_k^T + W_k Q_k W_k^T \tag{2.15}$$

As with the basic discrete Kalman filter, the time update equations in Table 2-1 project the state and covariance estimates from time step k to step $k+1$. Again $f(\bullet)$ in (2.14) comes from (2.3), A_k and W are the process Jacobians at step k, and Q_k is the process noise covariance (1.3) at step k.
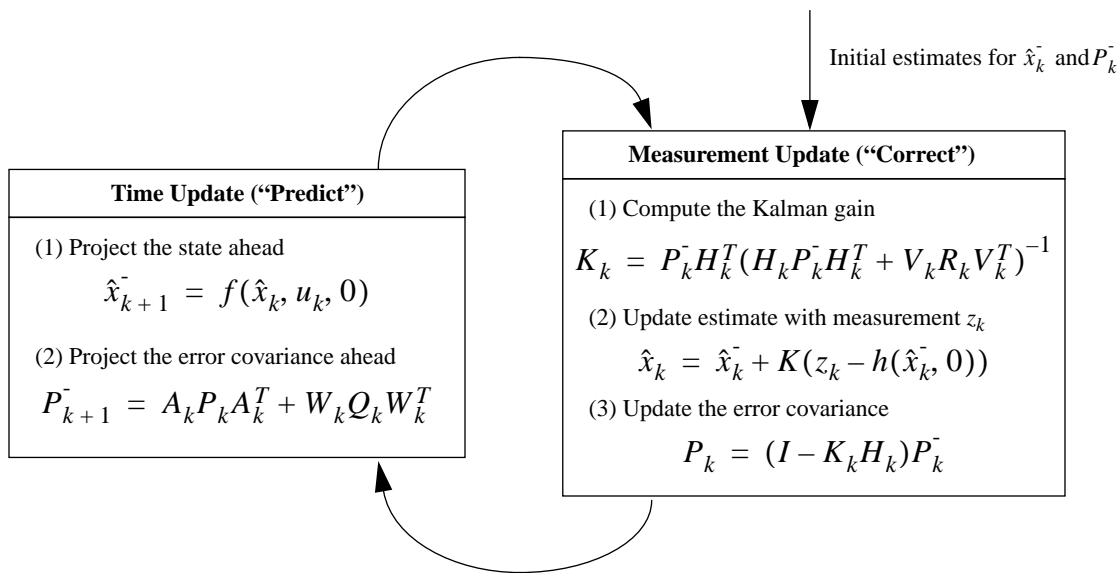
Table 2-2: EKF measurement update equations.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{2.16}$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - h(\hat{x}_k^-, 0)) \tag{2.17}$$

$$P_k = (I - K_k H_k) P_k^- \tag{2.18}$$

As with the basic discrete Kalman filter, the measurement update equations in Table 2-2 correct the state and covariance estimates with the measurement z_k. Again $h(\bullet)$ in (2.17) comes from (2.4), H_k and V are the measurement Jacobians at step k, and R_k is the measurement noise covariance (1.4) at step k.

The basic operation of the EKF is the same as the linear discrete Kalman filter as shown in Figure 1-1. Figure 2-1 below offers a complete picture of the operation of the EKF, combining the high-level diagram of Figure 1-1 with the equations from Table 2-1 and Table 2-2.

Initial estimates for $\hat{x}_k^-$ and P_k^-

Time Update ("Predict")

(1) Project the state ahead
$$\hat{x}_{k+1}^- = f(\hat{x}_k, u_k, 0)$$

(2) Project the error covariance ahead
$$P_{k+1}^- = A_k P_k A_k^T + W_k Q_k W_k^T$$

Measurement Update ("Correct")

(1) Compute the Kalman gain
$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

(2) Update estimate with measurement z_k
$$\hat{x}_k = \hat{x}_k^- + K(z_k - h(\hat{x}_k^-, 0))$$

(3) Update the error covariance
$$P_k = (I - K_k H_k) P_k^-$$

Figure 2-1. A complete picture of the operation of the *extended* Kalman filter, combining the high-level diagram of Figure 1-1 with the equations from Table 2-1 and Table 2-2.

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 $h(\bullet)$, 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 $h(\bullet)$, then as you might expect the filter will quickly diverge. The control theory term to describe this situation is *unobservable*.

3   A Kalman Filter in Action: 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.

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

$$x_{k+1} = A_k x_k + B u_k + w_k$$
$$= x_k + w_k \quad ,$$

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

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

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.)

The Filter Equations and Parameters

Our time update equations are

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

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

and our measurement update equations are

$$K_k = P_k^-(P_k^- + R)^{-1}$$

$$= \frac{P_k^-}{P_k^- + R}, \tag{3.1}$$

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

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

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 $\hat{x}_k^- = 0$.

Similarly we need to choose an initial value for P_k, i.e. P_k^-. If we were absolutely certain that our initial state estimate $\hat{x}_k^- = 0$ was correct, we would let $P_k^- = 0$. However given the uncertainty in our initial estimate $\hat{x}_k^-$, choosing $P_k^- = 0$ 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 $P_k^- \neq 0$ and the filter would eventually converge. We'll start our filter with $P_k^- = 1$.

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 3-1 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.
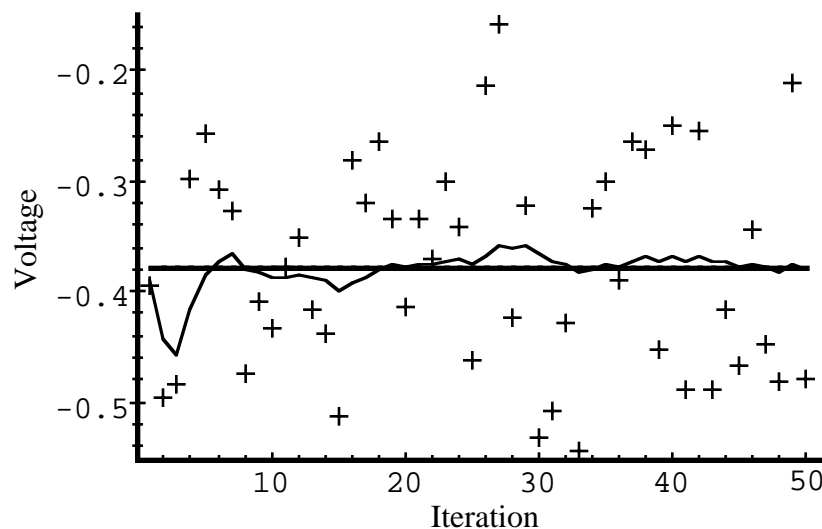

Figure 3-1. The first simulation: $R = (0.1)^2 = 0.01$. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve.

When considering the choice for P_k^- above, we mentioned that the choice was not critical as long as $P_k^- \neq 0$ because the filter would eventually converge. Below in Figure 3-2 we have plotted the value of P_k^- versus the iteration. By the 50th iteration, it has settled from the initial (rough) choice of 1 to approximately 0.0002 (Volts^2).
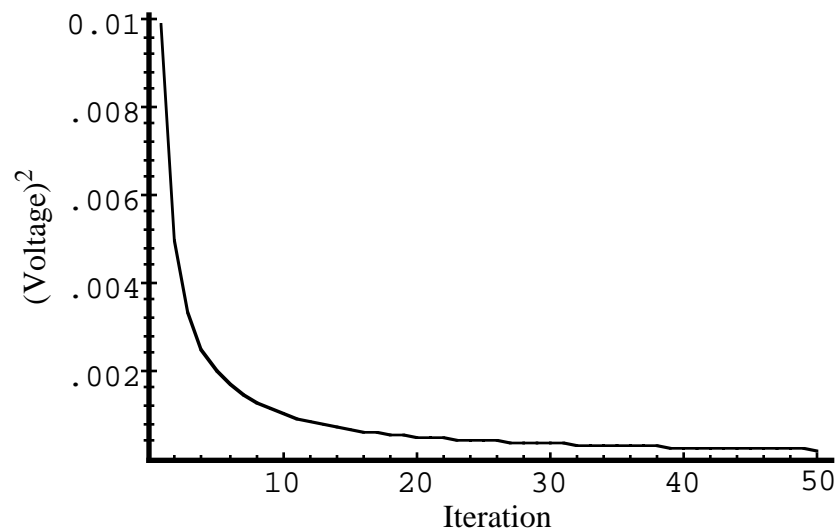
Figure 3-2. After 50 iterations, our initial (rough) error covariance P_k^- choice of 1 has settled to about 0.0002 (Volts^2).

In section under the topic "Filter Parameters and Tuning" we briefly discussed changing or "tuning" the parameters Q and R to obtain different filter performance. In Figure 3-3 and Figure 3-4 below we can see what happens when R is increased or decreased by a factor of 100 respectively. In Figure 3-3 the filter was told that the measurement variance was 100 times greater (i.e. $R = 1$) so it was "slower" to believe the measurements.
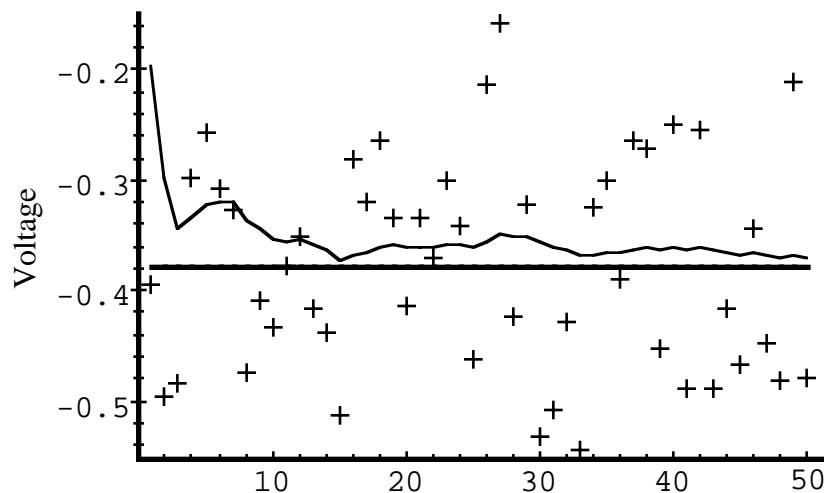
Figure 3-3. Second simulation: $R = 1$. The filter is slower to respond to the measurements, resulting in reduced estimate variance.

In Figure 3-4 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.
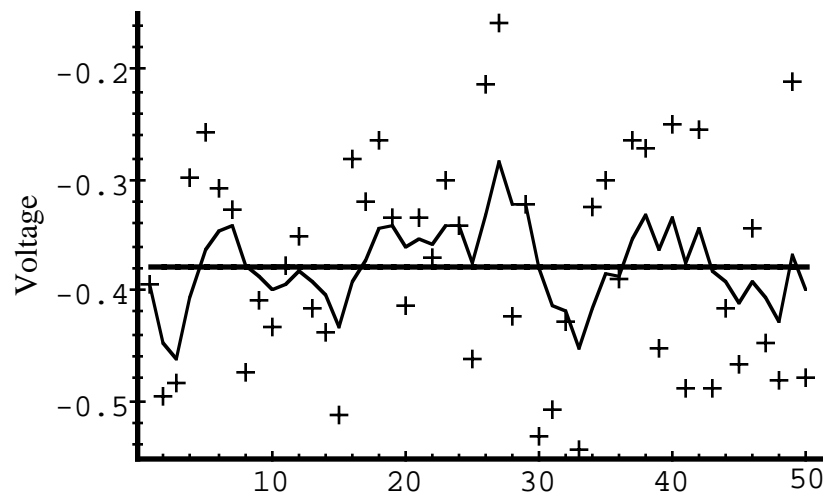
Figure 3-4. Third simulation: $R = 0.0001$. The filter responds to measurements quickly, increasing the estimate variance.

While the estimation of a constant is relatively straight-forward, it clearly demonstrates the workings of the Kalman filter. In Figure 3-3 in particular the Kalman "filtering" is evident as the estimate appears considerably smoother than the noisy measurements.

References

Brown92   Brown, R. G. and P. Y. C. Hwang. 1992. *Introduction to Random Signals and Applied Kalman Filtering, Second Edition*, John Wiley & Sons, Inc.

Gelb74   Gelb, A. 1974. *Applied Optimal Estimation*, MIT Press, Cambridge, MA.

Grewal93   Grewal, Mohinder S., and Angus P. Andrews (1993). Kalman Filtering Theory and Practice. Upper Saddle River, NJ USA, Prentice Hall.

Jacobs93   Jacobs, O. L. R. 1993. *Introduction to Control Theory, 2nd Edition*. Oxford University Press.

Julier   Julier, Simon and Jeffrey Uhlman. "A General Method of Approximating Nonlinear Transformations of Probability Distributions," Robotics Research Group, Department of Engineering Science, University of Oxford [cited 14 November 1995]. Available from http://www.robots.ox.ac.uk/~siju/work/publications/Unscented.zip.

                   Also see: "A New Approach for Filtering Nonlinear Systems" by S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, Proceedings of the 1995 American Control Conference, Seattle, Washington, Pages:1628-1632. Available from http://www.robots.ox.ac.uk/~siju/work/publications/ACC95_pr.zip.

                   Also see Simon Julier's home page at http://www.robots.ox.ac.uk/~siju/.

Kalman60   Kalman, R. E. 1960. "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).

Lewis86   Lewis, Richard. 1986. *Optimal Estimation with an Introduction to Stochastic Control Theory*, John Wiley & Sons, Inc.

Maybeck79   Maybeck, Peter S. 1979. *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc.

Sorenson70   Sorenson, H. W. 1970. "Least-Squares estimation: from Gauss to Kalman," *IEEE Spectrum*, vol. 7, pp. 63-68, July 1970.