Notes on Kalman Filtering

Brian Borchers and Rick Aster

November 7, 2011

Introduction

Data Assimilation is the problem of merging model predictions with actual measurements of a system to produce an optimal estimate of the current state of the system and/or predictions of the future state of the system. For example, weather forecasters run massive computational models that predict winds, temperature, etc. As time progresses, it is important to incorporate available weather observations into the mathematical model. Since these weather observations are noisy, the problem of incorporating the observations into the model is inherently statistical in nature.

Data Assimilation is becoming a very hot topic in many areas of science, including atmospheric physics, oceanography, and hydrology. In the next few lectures, we'll introduce Kalman filtering, which is one of the simplest approaches to data assimilation. The Kalman filter was introduced in a 1960 paper by R. E. Kalman.

The Model Of The System

Consider a discrete time dynamical system governed by the equation

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

Here, x_k, u_{k-1}, and w_{k-1} are vectors and the subscripts refer to the time steps rather than indexing elements of the vectors. The state of the system at time k is given by the vector x_k. Deterministic inputs to the system at time $k-1$ are given by u_{k-1}. Random noise affecting the system at time $k-1$ is given by w_{k-1}. We'll assume that w_{k-1} has a multivariate normal distribution with mean 0 and covariance matrix Q.

We'll obtain a vector of measurements z_k at time k, where z_k is given by

$$z_k = Hx_k + v_k. \tag{2}$$

Here v_k represents random noise in the observation z_k. We'll assume that v_k is normally distributed with mean 0 and covariance matrix R.

The matrices A, B, H, Q, and R are all assumed to be known, although the Kalman filter can be extended to simultaneously estimate these matrices along with x_k. For now, our goal is to estimate x_k and predict x_{k+1}, x_{k+2}, ..., as accurately as possible given z_1, z_2, ..., z_k.

The estimate that we will obtain will come in the form of a multivariate normal distribution with a specified mean $\hat{x}_k$ and covariance matrix $\hat{P}_k$. We will want to measure the "tightness" of this multivariate normal distribution. A convenient measure of the tightness of an MVN distribution with covariance matrix C is

$$\mathrm{trace}(C) = C_{1,1} + C_{2,2} + \ldots + C_{n,n}. \tag{3}$$

$$\mathrm{trace}(C) = Var(X_1) + Var(X_2) + \ldots + Var(X_n). \tag{4}$$

Example 1 For example, x_k might be a six element vector containing the position (3 coordinates) and velocity (3 coordinates) of an aircraft at time k. The vector u_{k-1} might represent control inputs (thrust, elevator, rudder, etc.) to the aircraft at time $k-1$, and w_{k-1} might represent the effects of turbulence on the aircraft. We may be using a very simple radar to observe the aircraft, so that we get measurements of the position, z, but not the velocity of the aircraft at each moment in time. These measurements of the aircraft's position might also be noisy.

In many cases, the system that we're interested in is described by a system of differential equations in continuous time:

$$x'(t) = Ax(t) + Bu(t). \tag{5}$$

We can discretize this system of equations using time steps of length Δt, to get

$$x(t + \Delta t) = x(t) + \Delta t x'(t). \tag{6}$$

$$x(t + \Delta t) = x(t) + \Delta t(Ax(t) + Bu(t)). \tag{7}$$

Letting $x_k = x(t + \Delta t)$ and $x_{k-1} = x(t)$, we get

$$x_k = (I + \Delta t A)x_{k-1} + \Delta t B u_{k-1}. \tag{8}$$

In many practical applications of Kalman filtering the mathematical model of the system consists of an even more complicated system of partial differential equations. Such systems are commonly discretized using finite difference or finite element methods. Rather than diving into the details of the numerical analysis used in discretizing PDE's, we will simply assume that our problem has been cast in the form of (1).

The Kalman Filter

We have two sources of information that can help us in estimating the state of the system at time k. First, we can use the equations that describe the dynamics of the system. Substituting $w_{k-1} = 0$ into (1), we might reasonably estimate

$$\hat{x}_k = Ax_{k-1} + Bu_{k-1} \tag{9}$$

A second useful source of information is our observation z_k. We might pick $\hat{x}_k$ so as to minimize $\|z_k - Hx_k\|$. There's an obvious trade-off between these two methods of estimating x_k. The Kalman filter produces a weighted combination of these two estimates that is optimal in the sense that it minimizes the uncertainty of the resulting estimate.

We'll begin the estimation process with an initial guess for the state of the system at time 0. Since we want to keep track of the uncertainty in our estimates, we'll have to specify the uncertainty in our initial guess. We describe this by using a multivariate normal distribution

$$x_0 \sim N(\hat{x}_0, \hat{P}_0). \tag{10}$$

In the *prediction* step, we are given an estimate $\hat{x}_{k-1}$ of the state of the system at time $k-1$, with associated covariance matrix $\hat{P}_{k-1}$. We substitute the mean value of $w_{k-1} = 0$ into (1) to obtain the estimate

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}. \tag{11}$$

The minus superscript is used to distinguish this estimate from the final estimate that we get after including the observation z_k. The covariance of our new estimate is

$$\hat{P}_k^- = Cov(\hat{x}_k^-). \tag{12}$$

$$\hat{P}_k^- = Cov(A\hat{x}_{k-1} + Bu_{k-1} + w_{k-1}). \tag{13}$$

The Bu_{k-1} term is not random, so its covariance is zero. The covariance of w_{k-1} is Q. The covariance of $A\hat{x}_{k-1}$ is $A\, Cov(\hat{x}_{k-1})A^T$. Thus

$$\hat{P}_k^- = ACov(\hat{x}_{k-1})A^T + Q. \tag{14}$$

$$\hat{P}_k^- = A\hat{P}_{k-1}A^T + Q. \tag{15}$$

We could simply repeat this process for x_1, x_2, ... If no observations of the system are available, that would be an appropriate way to estimate the system state.

In the *update* step, we modify the prediction estimate to include the observation.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{16}$$

$$\hat{x}_k = (I - K_kH)\hat{x}_k^- + K_kz_k. \tag{17}$$

Here the factor K_k is called the Kalman gain. It adjusts the relative influence of z_k and $\hat{x}_k^-$. We will soon show that

$$K_k = \hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1} \tag{18}$$

is optimal in the sense that it minimizes the trace of $\hat{P}_k$.

The covariance of our updated estimate is

$$\hat{P}_k = Cov(\hat{x}_k). \tag{19}$$

$$\hat{P}_k = Cov((I - K_k H)\hat{x}_k^- + K_k z_k). \tag{20}$$

$$\hat{P}_k = (I - K_k H)\hat{P}_k^-(I - K_k H)^T + K_k Cov(z_k) K_k^T. \tag{21}$$

Since $Cov(z_k) = R$,

$$\hat{P}_k = (I - K_k H)\hat{P}_k^-(I - K_k H)^T + K_k R K_k^T. \tag{22}$$

This simplifies to

$$\hat{P}_k = \hat{P}_k^- - K_k H \hat{P}_k^- - \hat{P}_k^- H^T K_k^T + K_k(H\hat{P}_k^- H^T)K_k^T + K_k R K_k^T. \tag{23}$$

$$\hat{P}_k = \hat{P}_k^- - K_k H \hat{P}_k^- - \hat{P}_k^- H^T K_k^T + K_k(H\hat{P}_k^- H^T + R)K_k^T. \tag{24}$$

We want to minimize the trace of $\hat{P}_k$. Using vector calculus, it can be shown that

$$\frac{\partial \text{trace}(\hat{P}_k)}{\partial K_k} = -2(H\hat{P}_k^-)^T + 2K_k(H\hat{P}_k^- H^T + R). \tag{25}$$

Setting the derivative equal to 0,

$$-2(H\hat{P}_k^-)^T + 2K_k(H\hat{P}_k^- H^T + R) = 0. \tag{26}$$

$$K_k = (H\hat{P}_k^-)^T(H\hat{P}_k^- H^T + R)^{-1}. \tag{27}$$

$$K_k = \hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1}. \tag{28}$$

Using this optimal Kalman gain, $\hat{P}_k$ simplifies further.

$$\hat{P}_k = \hat{P}_k^- - K_k H \hat{P}_k^- - \hat{P}_k^- H^T K_k^T + K_k(H\hat{P}_k^- H^T + R)K_k^T. \tag{29}$$

$$\begin{aligned}\hat{P}_k = \hat{P}_k^- - \hat{P}_K^- H^T(H\hat{P}_k^- H^T + R)^{-1} H \hat{P}_k^- - \\ \hat{P}_k^- H^T(\hat{P}_K^- H^T(H\hat{P}_k^- H^T + R)^{-1}))^T + \\ \hat{P}_K^- H^T(H\hat{P}_k^- H^T + R)^{-1}(H\hat{P}_k^- H^T + R)(\hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1})^T\end{aligned} \tag{30}$$

$$\hat{P}_k = \hat{P}_k^- - \hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1} H \hat{P}_k^- \tag{31}$$

$$\hat{P}_k = (I - K_k H)\hat{P}_k^-. \tag{32}$$

The algorithm can be summarized as follows. For $k = 1, 2, \ldots$,

1. Let $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$.
2. Let $\hat{P}_k^- = A\hat{P}_{k-1}A^T + Q$.
3. Let $K_k = \hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1}$.
4. Let $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$.
5. Let $\hat{P}_k = (I - K_k H)\hat{P}_k^-$.

In practice, we may not have an observation at every time step. In that case, we can use predictions at each time step and compute updates steps whenever observations become available.

Example 2 In this example, we'll consider a system governed by the second order differential equation

$$y''(t) + 0.01y'(t) + y(t) = \sin(2t) \tag{33}$$

with the initial conditions $y(0) = 0.1$, $y'(0) = 0.5$.

We must first use a standard trick to convert this second order ordinary differential equation into a system of two first order differential equations. Let

$$x_1(t) = y(t) \tag{34}$$

and

$$x_2(t) = y'(t). \tag{35}$$

The relation between $x_1(t)$ and $x_2(t)$ is

$$x_1'(t) = x_2(t). \tag{36}$$

Also, (33) becomes

$$x_2'(t) = -x_1(t) - 0.01x_2(t) + \sin(2t). \tag{37}$$

This system of two first order equations can be written as

$$x'(t) = Ax(t) + Bu(t) \tag{38}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -0.01 \end{bmatrix}, \tag{39}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{40}$$

and

$$u(t) = \begin{bmatrix} 0 \\ \sin(2t) \end{bmatrix}. \tag{41}$$

This system of differential equations will be discretized using (8) with time steps of $\Delta t = 0.01$. At each time step, the state vector will be randomly perturbed with $N(0, Q)$, noise, where

$$Q = \begin{bmatrix} 0.0005 & 0.0 \\ 0.0 & 0.0005 \end{bmatrix}. \tag{42}$$

We will observe $x_1(t)$ once per second (every 100 time steps.) Thus

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{43}$$

Our observations will have a variance of 0.0005.

For the initial conditions we will begin with the estimate

$$\hat{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{44}$$

and covariance

$$\hat{P}_0 = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}. \tag{45}$$

Figure 1 shows the true state of the system. Figure 2 shows the estimate of the system state using only prediction steps. The dotted lines in this plot are one standard-deviation error bars. The initial uncertainty in $x(t)$ is due to uncertainty in the initial conditions. Later, this uncertainty increases due to the effect of noise on the state of the system.

Figure 3 shows the Kalman filter estimates including observations of $x_1(t)$ once per second. Although the initial uncertainty is quite high, the Kalman filter quickly "learns" the actual state of the system and then tracks it quite closely. Each circle on the $x_1(t)$ plot represents an observation of the system. Notice that when an observation is obtained the Kalman estimate "jumps" to incorporate the new observation. Also note that although we only observe $x_1(t)$, the Kalman filter also manages to track $x_2(t)$. This happens because the system of differential equations connects $x_1(t)$ and $x_2(t)$. Figure 4 shows the true state of the system and the Kalman filter estimate on the same plot.

Figure 5 shows the differences between the system state and the simple prediction. Figure 6 shows the difference between the system state and the Kalman prediction. Notice that the Kalman filter produced much tighter estimates of both $x_1(t)$ and $x_2(t)$ using only a few observations of $x_1(t)$.

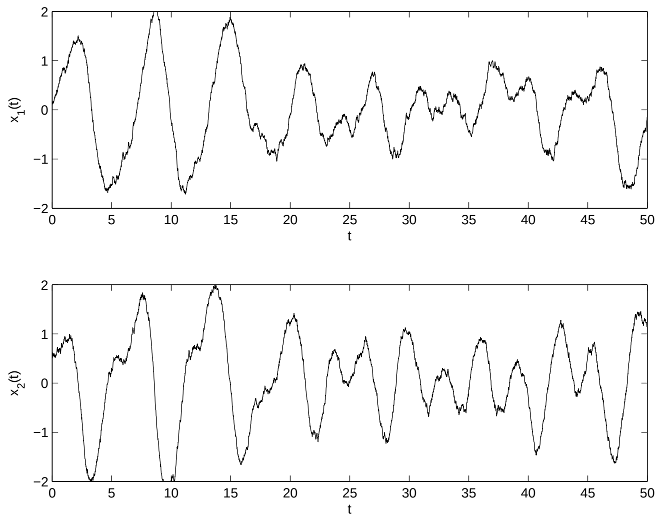

Figure 1: Plot of the system state $x(t)$.

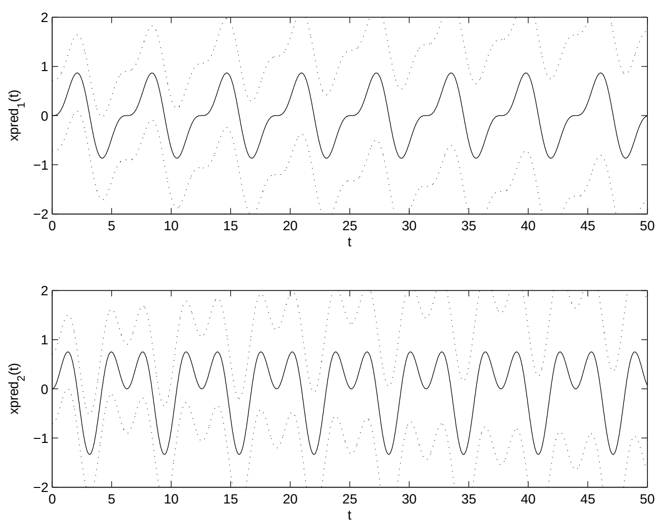

Figure 2: Estimate of $x(t)$ using prediction steps only.

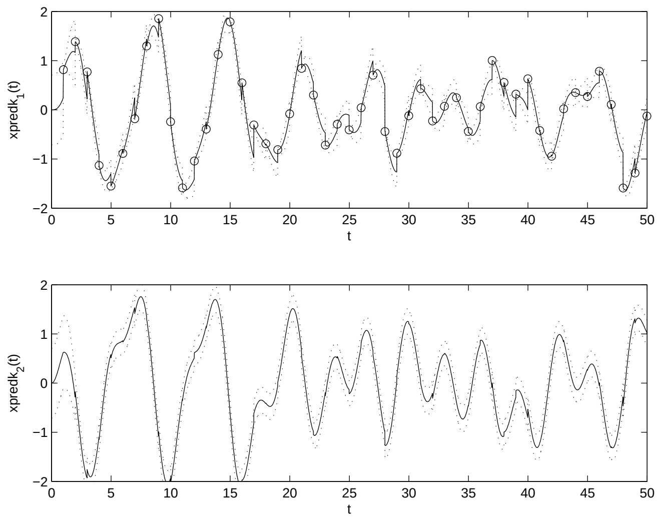

Figure 3: Kalman estimates of the $x_1(t)$ and $x_2(t)$.

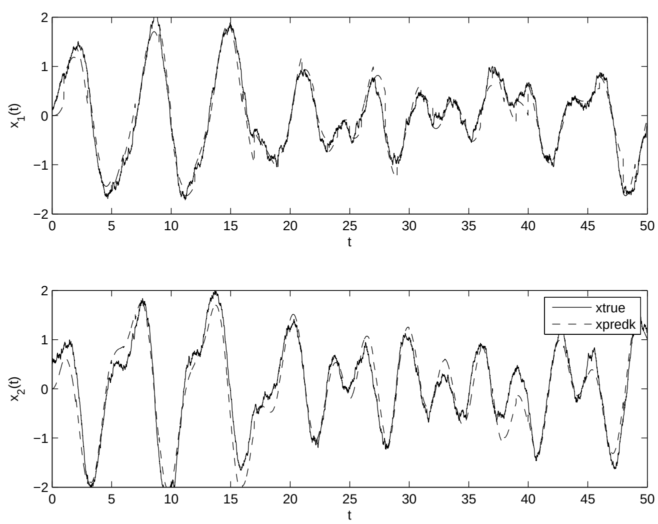

Figure 4: Kalman estimate versus the true values of $x_1(t)$ and $x_2(t)$.

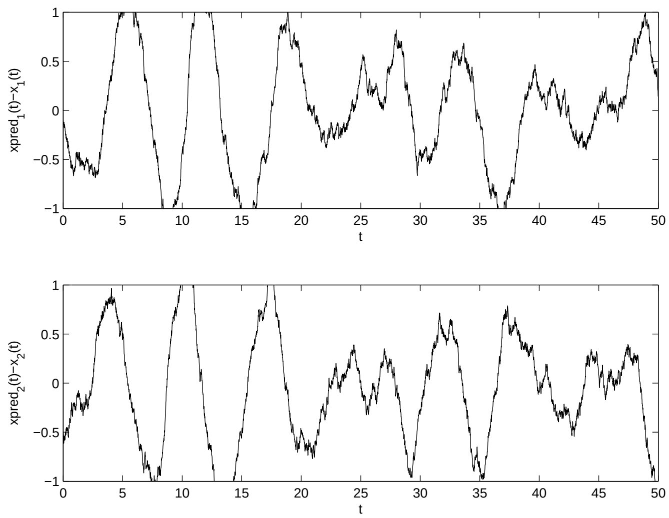

Figure 5: Difference between the system state and prediction estimate.

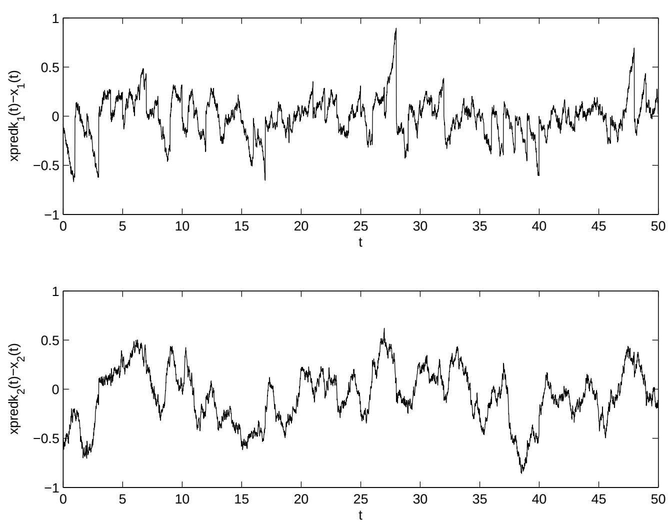

Figure 6: Difference between the system state and Kalman estimate.

The Extended Kalman Filter

The Extended Kalman Filter (EKF) extends the Kalman filtering concept to problems with nonlinear dynamics. Our new equation for the time evolution of the system state will be of the form

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{46}$$

where w_{k-1} is a random perturbation of the system. This time, we'll assume that w_{k-1} has a multivariate normal distribution with mean 0 and covariance matrix Q_{k-1}. That is, the covariance is allowed to be time dependent.

Our new measurement model will be

$$z_k = h(x_k, v_k) \tag{47}$$

where v_k is a multivariate normal $N(0, R_k)$ noise vector.

The prediction step is a straight forward generalization of what we have previously done in the Kalman filter.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0). \tag{48}$$

We'll also introduce a new notation for the predicted observation

$$\hat{z}_k^- = h(\hat{x}_k^-, 0). \tag{49}$$

In general, for a nonlinear function f, $\hat{x}_k^-$ will not have a multivariate normal distribution. However, we can reasonably hope that $f(x, u, w)$ will be approximately linear for relatively small changes in x and w, so that $\hat{x}_k^-$ will be at least approximately normally distributed.

We linearize $f(x, u, w)$ around $(\hat{x}_{k-1}, u_{k-1}, 0)$ as

$$f(x, u, w) = f(\hat{x}_{k-1}, u_{k-1}, 0) + A_{k-1}(x - \hat{x}_{k-1}) + W_{k-1}(w - 0) \tag{50}$$

where A and W are matrices of partial derivatives of f with respect to x and w. Note that since u_{k-1} is assumed to be known exactly, we don't need to linearize in the u variable. The entries in A_{k-1} and W_{k-1} are given by

$$A_{i,j,k-1} = \frac{\partial f_i(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial x_j}. \tag{51}$$

$$W_{i,j,k-1} = \frac{\partial f_i(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial w_j}. \tag{52}$$

Using this linearization, we end up with an approximate covariance matrix for $\hat{x}_k^-$,

$$\hat{P}_k^- = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + W_{k-1}Q_{k-1}W_{k-1}^T. \tag{53}$$

Similarly, we can linearize $h()$. Let

$$H_{i,j,k} = \frac{\partial h_i(\hat{x}_k^-, 0)}{\partial x_j}. \tag{54}$$

$$V_{i,j,k} = \frac{\partial h_i(\hat{x}_k^-, 0)}{\partial v_j}. \tag{55}$$

Now, let

$$\hat{e}_{x_k}^- = x_k - \hat{x}_k^- \tag{56}$$

and

$$\hat{e}_{z_k}^- = z_k - \hat{z}_k^-. \tag{57}$$

We don't actually know x_k, but we do expect $x_k - \hat{x}_k^-$ to be relatively small. Thus we can use our linearization of $f()$ to derive an approximation for $\hat{e}_{x_k}^-$.

$$\hat{e}_{x_k}^- = f(x_{k-1}, u_{k-1}, w_{k-1}) - f(\hat{x}_{k-1}, u_{k-1}, 0). \tag{58}$$

By the linearization,

$$\hat{e}_{x_k}^- \approx A_{k-1}(x_{k-1} - \hat{x}_{k-1}) + \epsilon_k \tag{59}$$

where ϵ_k accounts for the effect of the random w_{k-1}. The distribution of ϵ_k is $N(0, W_{k-1}Q_{k-1}W_{k-1}^T)$. Similarly,

$$\hat{e}_{z_k}^- = h(x_k, v_k) - h(\hat{x}_k^-, 0). \tag{60}$$

By the linearization this is approximately

$$\hat{e}_{z_k}^- \approx H\hat{e}_{x_k}^- + \eta_k \tag{61}$$

where η_k has an $N(0, V_k R_k V_k^T)$ distribution.

Ideally, we could update $\hat{x}_k^-$ to get x_k by

$$x_k = \hat{x}_k^- + \hat{e}_{x_k}^-. \tag{62}$$

Of course, we don't know $\hat{e}_{x_k}^-$, but we can estimate it. Let

$$\hat{e}_{x_k} = K_k(z_k - \hat{z}_k^-) \tag{63}$$

where K_k is a Kalman gain factor to be determined. Then let

$$\hat{x}_k = \hat{x}_k^- + \hat{e}_{x_k}. \tag{64}$$

By a derivation similar to our earlier derivation of the optimal Kalman gain for the linear Kalman filter, it can be shown that the optimal Kalman gain for the EKF is

$$K_k = \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + V_k R_k V_k^T)^{-1}. \tag{65}$$

Using this optimal Kalman gain, the covariance matrix for the updated estimate $\hat{x}_k$ is

$$\hat{P}_k = (I - K_k H_k)\hat{P}_k^-. \tag{66}$$

The EKF algorithm can be summarized as follows. For $k = 1, 2, \ldots$,

1. Let $\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$.
2. Let $\hat{P}_k^- = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + W_{k-1}Q_{k-1}W_{k-1}^T$.
3. Let $K_k = \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + V_k R_k V_k^T)^{-1}$.
4. Let $\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)))$.
5. Let $\hat{P}_k = (I - K_k H_k)\hat{P}_k^-$.

The Ensemble Kalman Filter

A fundamental problem with the EKF is that we must compute the partial derivatives of $f()$ so that they're available for computing the covariance matrix in the prediction step. An alternative approach involves using Monte Carlo simulation. In the Ensemble Kalman Filter (EnKF), we generate a collection of state variables according to the MVN distribution and time $k = 0$, and then follow the evolution of this ensemble through time.

In the following, we'll assume that our system state evolves according to

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{67}$$

and that our observation model is

$$z_k = Hx_k + v_k. \tag{68}$$

As in the extended Kalman filter, we predict x_k with

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0). \tag{69}$$

Instead of using partial derivatives of $f()$ to estimate $\hat{P}_k^-$, we'll use Monte Carlo simulation. Suppose that we're given a collection of random state vectors at time $k-1$, $\hat{x}_{k-1}^i$, $i = 1, 2, \ldots, m$. For each random state vector, we can generate a random $N(0, Q_{k-1})$ vector w_{k-1}^i, and then update the vector with

$$\hat{x}_k^{i,-} = f(\hat{x}_{k-1}^i, u_{k-1}, w_{k-1}^i), \quad i = 1, 2, \ldots, m. \tag{70}$$

Now, we can estimate the covariance matrix $\hat{P}_k^-$ from the vectors $\hat{x}_k^{i,-}$. Let C be this estimate of $\hat{P}_k^-$.

At time k, we obtain a new observation z_k, which is assumed to include MVN $N(0, R)$ noise. By adding $N(0, R)$ noise to z_k, we obtain an ensemble of simulated observations, z_k^i, $i = 1, 2, \ldots, m$.

Next, we update our ensemble of solutions with

$$\hat{x}_k^i = \hat{x}_k^{i,-} + CH^T(HCH^T + R)^{-1}(z_k^i - H\hat{x}_k^{i,-}). \tag{71}$$

This is simply the Kalman filter update, but using the estimated covariance matrix C, and the Monte Carlo simulated observations z_k^i. Finally, we can use the ensemble of states $\hat{x}_k^i$, $i = 1, 2, \ldots, m$ to estimate the mean state, $\hat{x}_k$, and the covariance $\hat{P}_k$.