

Kalman Filter

Wim van Drongelen, 2013

1. Introduction

Getting to understand a system can be quite a challenge. One approach is to create a model, an abstraction of the system. The idea is that this abstraction captures the system's important features. On the other hand one might attempt to learn the system's behavior by measuring its 'spontaneous' output or its input-output relationship. A pessimist would say that all models are wrong since they only represent part of the real thing, and all measurements are wrong too because they are also incomplete and noisy as well. Unfortunately, the pessimist is correct with this assessment. The Kalman approach represents a more optimistic view; this technique is to optimally combine the not so perfect information obtained via both modeling and measurements in order to gain the best possible knowledge about a system's state.

The basic idea is that we perform multiple (probably regular) measurements of a system's output and that we use these measurements and our knowledge of the system to reconstruct the state of the system. Practically, the Kalman filter process consists of two steps:

- (1) We make an *a priori* estimate of the state of the system before a measurement is made.
- (2) Subsequently, after we have made the measurement, we compute a new *a posteriori* estimate by fusing the *a priori* prediction with this latest measurement.

In our examples we will assume trivial dynamics: i.e., our prediction of the next state is the current state. Of course, when your system dynamics is more complex, the prediction step 1 must reflect this, but it wouldn't change step 2, i.e. the procedure of the assimilation/fusion of a prediction with a measurement.

For more in depth treatment of the Kalman filter, see the texts of Brown and Hwang (1997), Maybeck (1982), and especially for neuroscience applications Schiff (2012). The websites from Welch and Bishop listed in the references are also very useful.

2. Introductory Terminology

This section briefly summarizes of the most important terminology required to follow the development of the Kalman filter procedure in Section 3.

2.1. Normal or Gaussian Distribution

The noise components in the Kalman filter approach described in Section 3 are Gaussian white noise terms with zero mean. Recall that the PDF of the normal or Gaussian distribution is

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

μ – mean (zero in our case), and σ – standard deviation

Note: There are Kalman filter versions that deal with different types of noise processes.

2.2. Minimum Mean-Square Error (MMSE)

An optimal fit of a series of predictions with a series of observations can be achieved in several ways. Using the MMSE approach, one computes the difference between an estimate \hat{y} and a target value y . The square of this difference, $(\hat{y} - y)^2$ gives an impression how well the two fit. In case of a series of N estimates and target values, one gets a quantification of the error E by determining the sum of the squares of their differences $E = \sum_{i=1}^N (\hat{y}_i - y_i)^2$, and a common technique is to minimize this expression with respect to the parameter of interest. For example if our model is $\hat{y}_i = ax_i$, then a is the parameter we need to find. The minimum of E with respect to parameter a can be found by differentiating the expression for the sum of squares with respect to

this parameter and setting this derivative equal to zero, i.e. $\frac{\partial E}{\partial a} = 0$. In Section 3, the Kalman filter is developed by using the MMSE technique. For another example see Section 5.2.2 in van Drongelen (2007).

2.3. Recursive Algorithm

Assume we measure the resting potential of a neuron based on a series of measurements: z_1, z_2, \dots, z_n . It seems reasonable to use the estimated mean \hat{m}_n of the n measurements as the estimate of the resting potential:

$$\hat{m}_n = \frac{z_1 + z_2 + \dots + z_n}{n}. \quad (2)$$

This equation isn't an optimal basis for the development of an algorithm. For each new measurement we need more memory: i.e. we need n memory locations for measurements z_1 - z_n . In addition we need to complete all measurements before we have the estimate.

With a recursive approach (as is used in the Kalman filter algorithm) one estimates the resting potential after each measurement using updates for each measurement

$$1^{\text{st}} \text{ measurement: } \hat{m}_1 = z_1 \quad (3)$$

$$2^{\text{nd}} \text{ measurement: } \hat{m}_2 = \frac{1}{2} \hat{m}_1 + \frac{1}{2} z_2 \quad (4)$$

$$3^{\text{rd}} \text{ measurement: } \hat{m}_3 = \frac{2}{3} \hat{m}_2 + \frac{1}{3} z_3 \quad (5)$$

Etc.

Now we update the estimate at each measurement and we only need memory for the previous estimate and the new measurement to make a new estimate.

2.4. Data Assimilation

Suppose we have two measurements y_1 and y_2 and we would like to combine/fuse these observations into a single estimate y . Further, the uncertainty of the two measurements is given by their variance (=standard deviation squared), s_1 and s_2 respectively. If we follow the fusion principle used in the Kalman approach, which is based on MMSE, we will get

$$y = \frac{s_2}{s_1 + s_2} y_1 + \frac{s_1}{s_1 + s_2} y_2 \quad (6)$$

for the new estimate, and its variance s can be obtained from

$$\frac{1}{s} = \frac{1}{s_1} + \frac{1}{s_2} \quad (7)$$

By following this procedure, we can obtain an improved estimate based on two separate ones (Fig. 1).

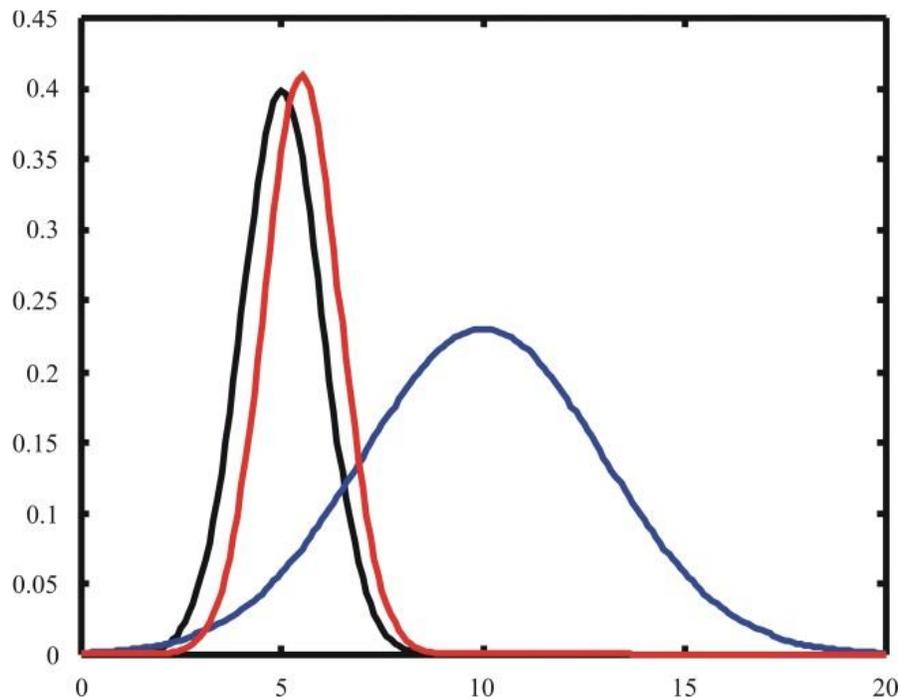


Fig.1

An example of the fusion procedure is depicted in Fig. 1. Here we have two observations (black and blue) associated with two distributions with mean values of 5 and 10 and standard deviations 1 and 3. Using Eqs (6) and (7) we find that the best estimate is reflected by a distribution with a mean of 5.5 and a standard deviation of $\sqrt{0.9} \approx 0.95$ (red).

From this example it is obvious that the measurement with less uncertainty (black curve) has more weight in the estimate while the one with more uncertainty (blue curve) only contributes a little.

2.5. State Model

One way of characterizing the dynamics of a system is to describe how it evolves through a number of states. If we characterize state as a vector of variables in a state space, we can describe

the system's dynamics by its path through state space. A simple example is a moving object. If we take an example where there is no acceleration, its state can be described by the coordinates of its position \vec{x} in three dimensions: i.e., the 3D space is its state space. The dynamics can be seen by plotting position vs. time, which also gives its velocity $d\vec{x}/dt$ (i.e. the time derivative of its position).

Let's take another example, a second order ordinary differential equation (ODE)

$$\ddot{x}_1 + b\dot{x}_1 + cx_1 = 0 \quad (8)$$

Now we define

$$\dot{x}_1 = x_2, \text{ and rewrite the ODE as} \quad (9a)$$

$$\dot{x}_2 + bx_2 + cx_1 = 0 \quad (9b)$$

In this way we rewrote the single 2nd order ODE in (8) as two 1st order ODEs: (9a, b). Now we can define a system state vector

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (10)$$

Note that, unlike the previous example, this vector is not the position of a physical object but a vector that defines the state of the system. The equations in (9a, b) can now be compactly written as

$$\frac{d\vec{x}}{dt} = A\vec{x}, \text{ with } A = \begin{pmatrix} 0 & 1 \\ -c & -b \end{pmatrix} \quad (11)$$

Here, the dynamics is expressed as an operation of matrix A on state vector \vec{x} . This notation can be used as an alternative to any ODE.

2.6. Bayesian Analysis

In the Kalman filter approach we update the probability distribution function (i.e. the mean and variance) of the state x of a system based on measurement z , i.e. $p(x|z)$. Conditional probabilities and distributions are the topic of Bayesian statistics and therefore the Kalman approach is a form of Bayesian analysis.

3. Derivation of a Kalman Filter for a Simple Case

In this section, we derive a simple version of a Kalman filter application following the procedure of Section 5.5 in Brown and Hwang (1997). They show a derivation for a vector observation, here I show the equivalent simplification for a scalar.

NOTES:

- To facilitate comparison, the equations in this section are numbered as the (much more complicated) equivalents in the original text (Brown and Hwang, 1997).
- All equations with a border are used in the example algorithm of the Matlab example in Appendix 1 and are also shown in the flowchart in Fig. 2.

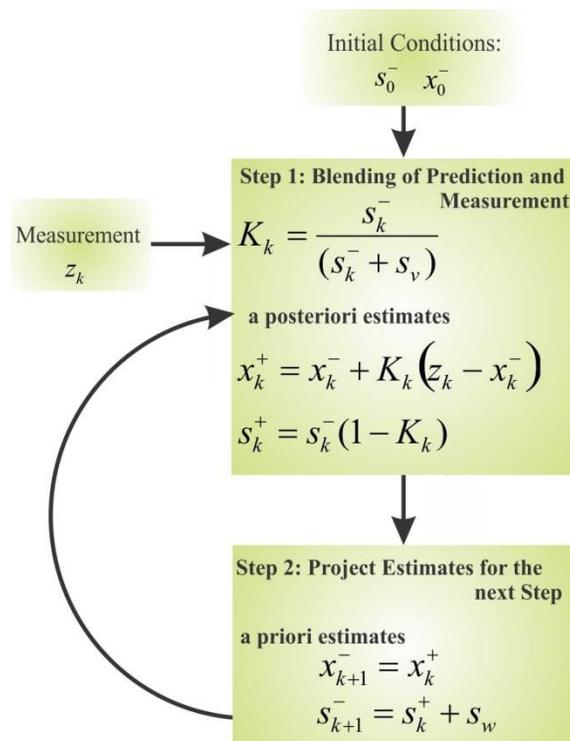


Fig. 2

Flowchart of the Kalman filter procedure derived in this section and implemented in the Matlab script Kalman_Vm (Appendix 1). In the Matlab script, steps 1-2 are indicated. Step 1: Eq (5.5.17), Eq (5.5.8), Eq (5.5.22); Step 2: Eqs (5.5.23) and (5.5.25). The equations in the diagram are derived in the text.

We start with a simple process x , for example a neuron's resting potential: i.e. its membrane potential in the absence of overt stimulation. We assume that the update rule for the state of the process is

$$x_{k+1} = x_k + w \quad (5.5.1)$$

The process measurement is z and is defined (as simple as possible) as

$$z_k = x_k + v \quad (5.5.2)$$

The Gaussian white noise variables w and v model the process and measurement noise respectively, and (for the sake of this example) they are assumed stationary and ergodic. The mean for both is zero and the associated variance values are s_w and s_v . The covariance of w and v is zero because they are independent.

Now assume we have an *a priori* estimate x_k^- . In most texts authors include a 'hat': i.e. \hat{x}_k^- to indicate one is dealing with an estimate. To simplify notation, hats to denote estimates are omitted here. The *a priori* error we make with this estimate is $x_k - x_k^-$ and the variance s_k^- associated with this error is given by

$$s_k^- = E[(x_k - x_k^-)^2] \quad (5.5.7)$$

Now we perform a measurement z_k and assimilate this measurement with our *a priori* estimate to obtain an *a posteriori* estimate x_k^+ , i.e.

$$\boxed{x_k^+ = x_k^- + K_k(z_k - x_k^-)} \quad (5.5.8)$$

The factor K_k in Equation (5.5.8) is the blending factor that determines to what extent the measurement and *a priori* estimate affect the new *a posteriori* estimate.

The *a posteriori* error we make with this estimate is $x_k - x_k^+$ and the variance s_k^+ associated with this error is given by

$$s_k^+ = E[(x_k - x_k^+)^2] \quad (5.5.9)$$

Substitution of Eq (5.5.2) in Eq (5.5.8) gives $x_k^+ = x_k^- + K_k(x_k + v - x_k^-)$. If we plug this result in Eq (5.5.9), we get:

$$s_k^+ = E\left[\left((x_k - x_k^-) - K_k(x_k + v - x_k^-)\right)^2\right] \quad (5.5.10)$$

We can expand this expression and take the Expectations. Recall that $x_k - x_k^-$ is the *a priori* estimation error and uncorrelated with measurement noise v . The outcome of this is (in case this isn't obvious, details of these steps are summarized in Appendix 2):

$$s_k^+ = s_k^- - 2K_k s_k^- + K_k^2 s_k^- + K_k^2 s_v$$

This is usually written as:

$$s_k^+ = s_k^- (1 - K_k)^2 + K_k^2 s_v, \quad (5.5.11)$$

or

$$s_k^+ = s_k^- - 2K_k s_k^- + (s_k^- + s_v) K_k^2$$

To optimize the *a posteriori* estimate using optimal blending in Eq (5.5.8), we must minimize the variance s_k^+ (= square of the estimated error) with respect to K_k , i.e. we differentiate the expression for s_k^+ with respect to K_k and set the result equal to zero:

$$-2s_k^- + 2(s_k^- + s_v)K_k = 0$$

This generates the expression for an optimal K_k ,

$$K_k = \frac{s_k^-}{(s_k^- + s_v)}. \quad (5.5.17)$$

NOTE: At this point it is interesting to evaluate the result. In the update Equation (5.5.8) you can

see what this optimized result means: $x_k^+ = x_k^- + \frac{s_k^-}{(s_k^- + s_v)}(z_k - x_k^-)$. If the *a priori* estimate is

unreliable, indicated by its large variance $s_k^- \rightarrow \infty$, we will ignore the estimate and we believe

more of the measurement, i.e. $x_k^+ \rightarrow z_k$. On the other hand, if the measurement is completely

unreliable, $s_v \rightarrow \infty$, we ignore the measurement and believe the *a priori* estimate: $x_k^+ \rightarrow x_k^-$.

Now we proceed and use the result in (5.5.17) to obtain an expression for s_v :

$$s_v = \frac{s_k^- - K_k s_k^-}{K_k}$$

Substitution of this result in (5.5.11) gives an expression to find the *a posteriori* error based on an optimized blending factor K_k :

$$s_k^+ = s_k^- (1 - K_k)^2 + \frac{K_k^2 s_k^- - K_k^3 s_k^-}{K_k}$$

With a little bit of algebra we get

$$s_k^+ = s_k^- - 2K_k s_k^- + K_k^2 s_k^- + K_k s_k^- - K_k^2 s_k^-,$$

which we simplify to:

$$\boxed{s_k^+ = s_k^- (1 - K_k)} \quad (5.5.22)$$

Here it should be noted that depending on how one simplifies Eq (5.5.11), one might get different expressions. Some work better than others, depending on the situation. We use Eq (5.5.22) here because of its simplicity.

Lastly we need to use the information to make a projection towards the next time step occurring at t_{k+1} (Fig. 2, Step 2). This procedure depends on the model for the process we are measuring from! Since we employ a simple model in this example, Eq (5.5.1), our *a priori* estimate of x is simply the *a posteriori* estimate of the previous time step:

$$\boxed{x_{k+1}^- = x_k^+} \quad (5.5.23)$$

Here we ignore the noise (w) term in Equation (5.5.1) since the Expectation of w is zero. In many applications this step can become more complicated or even an extremely complex procedure depending on the dynamics between t_k and t_{k+1} .

The *a priori* estimate of the variance at t_{k+1} is based on the error of the prediction in Eq (5.5.23),

$$e_{k+1}^- = x_{k+1} - x_{k+1}^-.$$

If we substitute Equations (5.5.1) and (5.5.23) in this expression, we get $e_{k+1}^- = x_k + w - x_k^+$.

Since the *a posteriori* error at t_k is $e_k^+ = x_k - x_k^+$, we can plug this into the above expression and we get

$$e_{k+1}^- = e_k^+ + w$$

Using this expression for the *a priori* error, we see that the associated *a priori* variance at t_{k+1} depends on the *a posteriori* error at t_k and the variance of the noise process w :

$$s_{k+1}^- = E[(e_k^+ + w)^2].$$

Because w and the error in the estimate are uncorrelated, we finally have

$$\boxed{s_{k+1}^- = s_k^+ + s_w} \quad (5.5.25)$$

Note:

If we integrate a noise term w (drawn from a zero mean normal distribution with variance σ^2)

over time, i.e. $\int_{t_1}^{t_2} w dt$, we get $\sigma^2(t_2 - t_1)$. So, in case of a large interval between measurements at

t_1 and t_2 , the uncertainty grows with a factor $(t_2 - t_1)$ and, consequently, s_{k+1}^- will become very large, and the prediction has almost no value in the fusion of prediction and measurement.

4. Matlab Example

The example Matlab script `Kalman_Vm.m` (listed in Appendix 1) uses the above derivation to demonstrate how this could be applied to a recording of a membrane potential of a non-stimulated neuron. In this example we assume that the resting potential $V_m = -70$ mV and that the standard deviations of the process noise and the measurement noise are 1 mV and 2 mV respectively. The program follows the flowchart in Fig. 2 and a typical result is depicted in Fig. 3. Due to the random aspect of the procedure, your result will vary slightly from the example shown here.

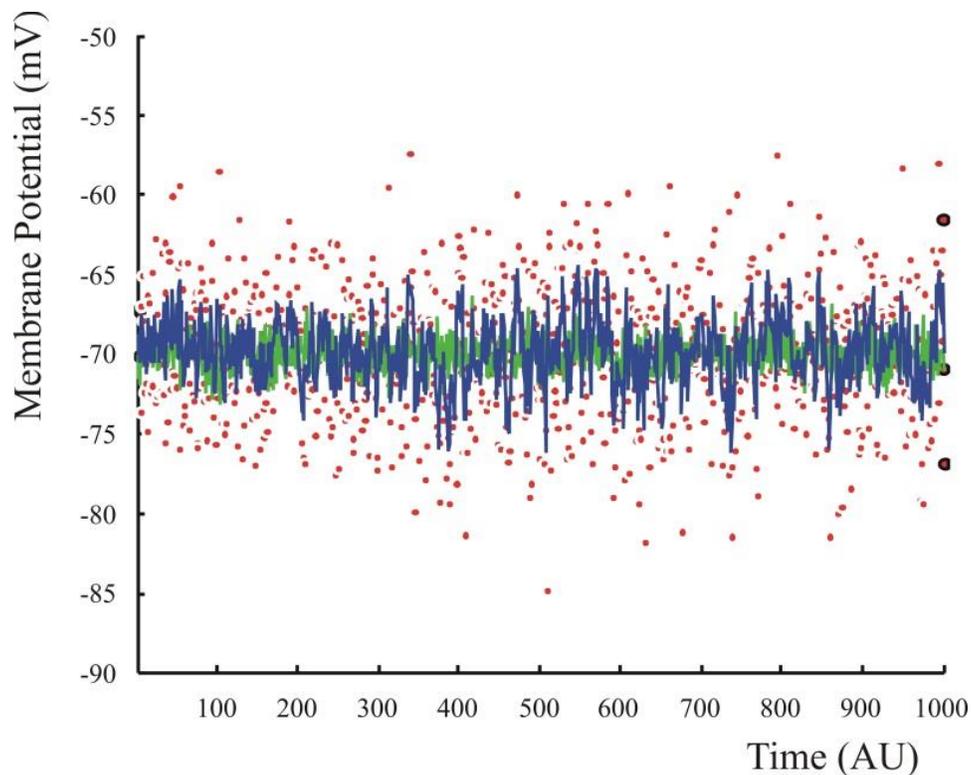


Fig.3

Example of the Kalman filter application created with the attached Matlab script `Kalman_Vm`. The true values are green, the measurements are the red dots and the Kalman filter output is the blue line. Although the result is not perfect, it can be seen that the Kalman filter output is much closer to the real values than the measurements.

5. Kalman Filter: Notation

(See Brown and Hwang (1997), Section 5.5)

$$K_k = \frac{s_k^-}{(s_k^- + s_v)} \quad (5.5.17)$$

$$x_k^+ = x_k^- + K_k(z_k - x_k^-) \quad (5.5.8)$$

$$s_k^+ = s_k^-(1 - K_k) \quad (5.5.22)$$

$$x_{k+1}^- = x_k^+ \quad (5.5.23)$$

$$s_{k+1}^- = s_k^+ + s_w \quad (5.5.25)$$

6. Kalman Filter to Estimate Model Parameters

(See Brown and Hwang (1997), p. 225: Augmented Vector).

The vector-matrix version of the Kalman filter can estimate a series of variables but can also be used to estimate model parameters. Most of the time, the parameter of the model is a constant or changes very slowly so that it can be considered a constant. In this case, one treats the parameter to estimate as a constant with noise, just as we did with x_k in Equation (5.5.1). The noise component allows the algorithm of the Kalman filter to find the best fit of the parameter. In addition, since we don't measure the parameter (most likely because we cannot measure it), we treat it as a hidden variable, i.e. it is part of the model of the state x_k , but it doesn't show up in the measurement z_k . Note that x_k and z_k are vectors here, and that x_k contains the variables as well as the parameters we want to estimate, i.e. x_k is augmented with the parameters.

Appendix 1: Matlab Script that Performs Kalman Filtering

```

% Kalman_Vm
%
% Example using a Kalman filter
% to estimate the membrane potential
% of a neuron
% (Algorithm isn't optimized)
%
% WvD 2013

clear
close all

% Parameters
N=1000;           % number of measurements
Vm=-70;          % the membrane potential (Vm) value
sw=.1^2;         % variance process noise
sv=2^2;          % measurement noise
% Initial Estimates
x_apriori(1)=Vm+90; % first mis-estimate of Vm
% This is the a priori measurement
s_apriori(1)=.1;   % a priori estimate of the error; here variance =.1

% create a simulated measurement vector
% -----
for i=1:N;
    tru(i)=Vm+randn*sw;
    z(i)=tru(i)+randn*sv;
end;

% Perform the Kalman filter
% -----
% Note: Malab indices start at 1, and not
%       at 0 as in most textbook derivations

% Equations are numbered as in the Handout and they
% follow the numbering in Brown and Hwang (1997)
% Introduction to Random Signals and Applied Kalman
% Filtering, Wiley & Sons, New York

for i=1:N;
    % step 1: Blending of Prediction and Measurement
    % Compute blending factor K(i)
    K(i)=s_apriori(i)/(s_apriori(i)+sv); % Eq (5.5.17)
    % Update Estimate using the measurement
    % this is the a posteriori estimate
    x_aposteriori(i)=x_apriori(i)+K(i)*(z(i)-x_apriori(i)); % Eq (5.5.8)
    % Update the error estimate [Note that there
    % are several variants for this procedure;
    % here we use the simplest expression
    s_aposteriori(i)=s_apriori(i)*(1-K(i)); % Eq (5.5.22)
    % step 2: Project Estimates for the next Step
    x_apriori(i+1)=x_aposteriori(i); % Eq (5.5.23)
    s_apriori(i+1)=s_aposteriori(i)+sw; % Eq (5.5.25)
end;

% plot the results
figure;hold;
plot(z,'r.');
plot(tru,'g');
plot(x_aposteriori);
axis([1 N -100 10]);
xlabel ('Time (AU)')
ylabel ('Membrane Potential (mV)')
title ('Kalman Filter Application: true values (green); measurements (red .); Kalman Filter
Output (blue)')

```

Appendix 2: Details of the Steps between Equations (5.5.10) and (5.5.11)

This Appendix gives the details of the steps between equations (5.5.10) and (5.5.11). For convenience, we repeat the equation.

$$s_k^+ = E\left[\left((x_k - x_k^-) - K_k(x_k + v - x_k^-)\right)^2\right] \quad (5.5.10)$$

We can expand this expression and take the Expectations.

If we expand the squared function, we get three terms further evaluated in (1)-(3) below.

(1) The first term can be rewritten using Equation (5.5.7), i.e.:

$$E\left[(x_k - x_k^-)^2\right] = s_k^-$$

(2) The second term $E\left[-2(x_k - x_k^-)K_k(x_k + v - x_k^-)\right] =$

$$E\left[-2K_k\left\{(x_k - x_k^-)x_k + (x_k - x_k^-)v - (x_k - x_k^-)x_k^-\right\}\right]$$

The 2nd term in between the curly brackets in the above expression, $(x_k - x_k^-)v$ will vanish when taking the Expectation because $x_k - x_k^-$ is the *a priori* estimation error which is uncorrelated with measurement noise v . Thus, using equation (5.5.7) again, we can simplify

$$E\left[-2K_k\left\{(x_k - x_k^-)(x_k - x_k^-)\right\}\right] = -2K_k s_k^-$$

(3) The third term is $E\left[K_k^2\left\{(x_k + v - x_k^-)(x_k + v - x_k^-)\right\}\right]$. First we focus on the expression in between the curly brackets first and we get

$$x_k^2 + x_k v - x_k x_k^- + v x_k + v^2 - v x_k^- - x_k^- x_k - x_k^- v + x_k^{-2} =$$

$$2v(x_k - x_k^-) + x_k^2 - 2x_k x_k^- + x_k^{-2} + v^2 = 2v(x_k - x_k^-) + (x_k - x_k^-)^2 + v^2$$

When we take Expectations, the first term of the simplified expression will vanish (why?), so we get the following result:

$$E\left[K_k^2(x_k - x_k^-)^2 + K_k^2 v^2\right] = K_k^2 s_k^- + K_k^2 s_v$$

If we now collect all terms from (1)-(3), we can rewrite Eq (5.5.10) as

$$s_k^+ = s_k^- - 2K_k s_k^- + K_k^2 s_k^- + K_k^2 s_v$$

This is usually written as:

$$s_k^+ = s_k^- (1 - K_k)^2 + K_k^2 s_v, \quad (5.5.11)$$

giving us the result in equation (5.5.11)

References

Brown R.G., Hwang P.Y.C. (1997) – Introduction to Random Signals and Applied Kalman Filtering. John Wiley & Sons, New York, NY.

Maybeck PS (1982) – Stochastic Models, Estimation, and Control, Vol. 141, Academic Press, New York.

Schiff S.J. (2012) – Neural Control Engineering: The Emerging Intersection between Control Theory and Neuroscience. MIT Press, Cambridge, MA.

Drongelen W van (2007) – Signal Processing for Neuroscientists: An Introduction to the Analysis of Physiological Signals, Academic Press, Elsevier, Amsterdam

Welch G, Bishop G – An Introduction to the Kalman Filter, University of North Carolina
http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf or a more complete text at
http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf

HOMEWORK

- 1) In this question, the Equations are numbered as in the Handout ‘Kalman Filter.’
Give the expression for estimate \hat{m}_n at the n^{th} measurement, using a recursive approach as in Eqs (3)-(5)

- 2) In this question, the Equations are numbered as in the Handout ‘Kalman Filter.’
Why does (in the Kalman data fusion) adding a measurement always results in a decrease of the uncertainty (=variance) of the estimate? Would this still be true if s_1 or s_2 was infinite (i.e. total uncertainty)? Hint: use Eq (7).

- 3) In this question, the Equations are numbered as in the Handout ‘Kalman Filter.’
Write the following 3rd order ODE in the form of a state vector and matrix operator $\ddot{x}_1 + b\dot{x}_1 + cx_1 = 0$. Now repeat this for $\ddot{x}_1 + b\dot{x}_1 + cx_1 = e$.
Give the details of the procedure, i.e. define the state and matrix A (see Equations (8) – (11)).

- 4) We have a series of measurements y_n and system states x_n . We want to link both using a simple linear model: $y_n = a x_n$. Use the MMSE method to find an optimal estimate for a . Comment on how this result relates to linear regression.

- 5) In this question, unless stated otherwise, the Equations are numbered as in the Handout ‘Kalman Filter.’

a. Show that Eqs (6) and (7) are the same as the Kalman approach to fusion of data.

Hint 1: Combine Eqs (5.5.8) and (5.5.17); add $\frac{s_1}{s_1 + s_2} y_1 - \frac{s_1}{s_1 + s_2} y_1$ to the result and rearrange the terms as in Eq (6).

Hint 2: Combine Eqs (5.5.22) and (5.5.17); rearrange Eq (7) as $s = \frac{s_1 s_2}{s_1 + s_2}$ and compare the results.

b. Discuss the difference of combining variance (noise) between equation (7) in this handout and Equation 3.15 in Van Drongelen (2007).