Kalman Filter Tutorial

Irene Markelić

No Institute Given

"What does chance ever do for us?"
Paley, William

1 Motivating Example

Imagine you have a robot that is supposed to navigate from an arbitrary position on a 1d line to a goal position B. You could follow the simple strategy to let the robot move into the direction of B until its distance to B is zero. In this case it is reasonable to use the position of the robot as the **state** X . The position can be just a scalar indicating the distance in meters to B. Easy! But: The robot can only perceive information about its current state (its distance to B) from its own sensors. (Nobody external tells the robot where it is..unless you have GPS). Lets say the robot has an IR sensor that measures its distance to B. Now: Somebody places the robot on an arbitrary position on the line. The first thing the robot does, is to make a **measurement**. Measurements are usually denoted by z. Let's say the robot measures $z = 2m$. From experience it might be known that the sensor is not so good, in other words the measurements are noisy. You might know that the sensor has an error that is most of the time zero, but sometimes (with a certain frequency) in the range between +/-1m. You can model that as a RV e with a Gaussian probability densitiy function. Now the best you can do is to maintain an estimate or **belief** about your current position which is 2m + e. The robot then moves toward B by putting some known voltage on its motors which causes a translation in the direction of B. This motor **control data** (its actions) is usually denoted by u. We can relate u to the robot's state (because we know that a high voltage leads to a large translation..) and we can summarize this relationship in a matrix C. Thus, Cu describes the effect of the robot's actions to its state. Therefore, the robot can use the previously measured state at time $t-1$ and Cu to make a **prediction** about the next state that it is about to measure, i.e. its *expectation*. Again: The robot has two sources of information to estimate its own state: 1) its own actions under the assumption that the robot knows how they change the world and 2) its measurements. What is needed is a tool to combine these two stochastic quantities together to achieve an optimal state estimation. A Kalman filter is an example of such a tool, and in the following we are going to cover the necessary knowledge to understand the Kalman filter equations.

Note, that this example describes a problem known by all of us. There is a system that does not KNOW in which state it is, it can only make some measurements that are noisy. That can be expressed in terms of Markov Models, i.e. the system can be described by a Markov Process. The system does not know

in which state it is, the states are *hidden*, that is, we are dealing with a HMM. Only some observable states exsist, from which the true state must be deduced. How do humans do that?

2 Vocabulary

First some vocabulary:

- state: denoted x, x_t resp., to denote time dependence
- measurement: z_t
- control data u_t
- belief: Since it is not possible to know the true state it must be inferred from noisy measurements. Therefore, we can only have a blief about the true state. It is prepresented by a by a **conditional probability distribution**.
- conditional probability distribution: is assigned to a random variable.
- prediction: If you calculate the posterior *before* incorporating the measurement, just after the action (control data).
- state evolution of a dynamic stochastic system
- system evolution of a dynamic stochastic system

3 Useful Statistics

How can we represent uncertain information and work with it? How can we combine several sources of uncertain information? Is the combination of two uncertain information more uncertain or more certain? To answer these questions we will have to consult statistics.

3.1 Random Variable and Probability Distribution

Random Variable and its assigned probability density fucntion can be used to describe random events/experiments, in more general terms: Knowledge that is uncertain. In the following we will list some of its properties:

- A random variable, RV, is a function X that maps from the sample space (the space Ω of all possible events ω) to the real numbers:

$$X(\omega) \mapsto \mathbb{R} \tag{1}$$

  - E.g. a toin coss can be described by the random variable X. The possible events are $\Omega = heads, tails$. If we toss the coin and the event is *heads* then X maps to 1, and to 0 else. Thus, the value of a random variable is stochastic, we cannot know it, before.
  - Another example, if we roll a dice 5 times and X is "how often do we obtain a 2", then X(A) can take the values $0, 1, 2, 3, 4, 5$.

– The value of a RV is not known a priori. But it is possible to make statements about how probable it is that the RV takes a certain value. This is done by the probability density function.

$$f(x) = P(X = x) \qquad \text{probability density function, pdf} \quad (2)$$

$$F(x) = P(X \leq b) = \int_{-\infty}^{b} f(x)dx \qquad \text{distribution function} \quad (3)$$

3.2 Total Probability, Conditional Probability, and Bayes Formula

After we now have learned how uncertain knowledge can be represented, we want to turn to the question how can we combine several sources of uncertain information? Is the combination of two uncertain information more uncertain or more certain? We will find that the Bayes Theorem is very helpful for answering these questions.
In order to derive the Bayes Theorem intuitively let's first have a look at a decision tree:

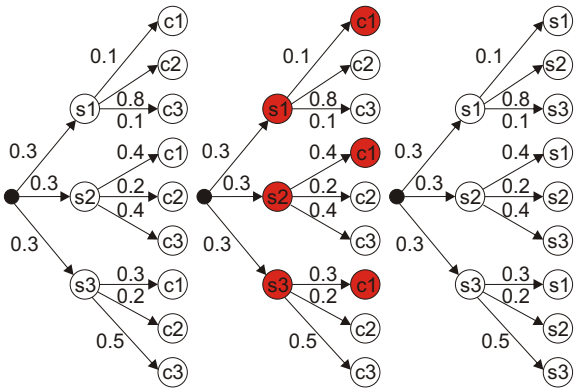
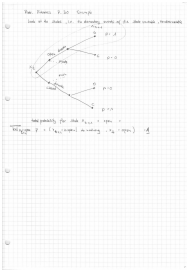


Fig. 1. A probabilistic decision tree.

The decision tree is a convenient form to display multi-stage random experiments. That is, experiments whose outcome is not deterministic. In Fig. 1 the starting point is the black starting circle on the left, from there the tree branches into three possible outcomes of the first experiment, s_1, s_2 and s_3. One can view these as different urns that contain black, white and red colored balls. The edges leading to each outcome are labeled with a probability that this event occurrs. E.g. if the experiment is to randomly select one of the three urns than each particular urn is selected with the given probability. After an urn has been selected the next random experiment is conductet. This might be to draw one ball out of the selected urn and we might be interested in the probability of



selecting a certain color. Since there are balls of three colors in each urn the tree branches into three different outcomes, that are labeled $c1$, $c2$ and $c3$. We can now ask two questions:

- 1) How probable is it to select a ball of a certain color? E.g. a black ball, and assume that it is denoted by $c1$?
- 2) Given that at the end of this two-level experiment we have a black ball, how probable is it, that it was drawn from a particular urn, let's say urn $s2$?

The first question can be formalized as follows:

$$P(c_1) = P(s_1 \cap c_1) \cup P(s_2 \cap c_1) \cup P(s_3 \cap c_1) \Rightarrow \tag{4}$$
$$= P(s_1 \cap c_1) + P(s_2 \cap c_1) + P(s_3 \cap c_1) \Rightarrow \tag{5}$$
$$= P(s_1)P(c_1|s_1) + P(s_2)P(c_1|s_2) + P(s_3)P(c_1|s_3) \Rightarrow \tag{6}$$
$$= \sum_i P(s_i)P(c_1|s_i). \text{ Total Probability} \tag{7}$$

(Concerning row 2, we can sum the probabilities because these events are excluding each other.) This is the **total probability** for the event c_1 to occurr. Furthermore, since:

$$P(s_1 \cap c_1) = P(s_1)P(c_1|s_1) \Rightarrow \tag{8}$$
$$P(c_1|s_1) = \frac{P(s_1 \cap c_1)}{P(s_1)} \text{ Conditional Probability} \tag{9}$$

This is the **conditional probability**. The second question can be formalized as follows:

$$P(s_1|c_1) \tag{10}$$

This can be calculated by relating the probability of c_1 and s_1 occurring together to the total probability of c_1. Therefore:

$$P(s_1|c_1) = \frac{P(s_1 \cap c_1)}{P(c_1)} \Rightarrow \tag{11}$$
$$= \frac{P(s_1)P(s_1|c_1)}{P(c_1)} \Rightarrow \tag{12}$$
$$= \frac{P(s_1)P(s_1|c_1)}{\sum_i P(s_i)P(c_1|s_i)} \text{ Bayes Formula} \tag{13}$$

This is the famous **Bayes Formula**. We can calculate this probability for each outcome, then:

$$P(s_j|c_1) = \frac{P(s_j)P(c_1|s_j)}{\sum_i P(s_i)P(c_1|s_i)}. \tag{14}$$

Furthermore, the Bayes Formula can be generalized to conditional probability density distributions, pdf's: Given two random variables, RV X and Y:

$$f_X(x|Y = y) = \frac{f_Y(y|X = x)f_X(x)}{f_Y(y)} \tag{15}$$

3.3 Importance of Bayes Formula

The Bayes Formula or Theorem looks trivial, however, it is of great importance, what we realize if we think about if for a second. For example you could interpret it like this:

$$Posterior = \frac{Prior \times Likelihood}{\alpha}, \tag{16}$$

where *Prior* would be $P(s_j)$, the *Likelihood* is $P(c_1|s_j)$, and α is a normalizing constant. The Prior represents some uninformed belief, that, after viewing some evidence is corrected. The evidence could be a measurement. Relating to our previous example we could think of the s_i in the decision tree in Fig. 1 as different possible states that the robot. could be in, but that it does not know for sure, they are *unobservable* or *hidden* (like in a HMM). The c_i could represent a measurement. Thus, we can ask (somehow the inverse question): **Given a certain measurement how big is the probability for the robot of being in one out of n different states?** We already stated that belief is given in form of pdf's. The Bayes Formula gives us a nice way of including new knowledge (a pdf) to old knowledge (another pdf). That is, it gives us the tool to reshape our belief in the case of new (unfortunately also uncertain) information. This leads us to a filter, that let's us use all available information - the Bayes Filter. Therefore, we now leave statistics and look at its application:

4 Bayes Filter

Armed with the information from the previous sections we are now ready to look at the Bayes Filter. The Bayes Filter forms the basis for many other filters, like the Kalman or the particle filter. The Bayes Filter Algorithm is the most general algorithm for calculating beliefs [1] and is a *recursive* state estimator (because you maintain a belief, which is a pdf about a state). It is very important to stress that the filter is recursive. Note, that if you make a measurement and you want to calculate a pdf conditioned on this measurement you should actually take *all* previous measurements into account. But if you make the assumption, that the

underlying (unknown) process is *Markovian*, then you imply that the state (x_t) already comprises all information, also the old measurements. Therefore:

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_{t1}), \tag{17}$$

where the notation $1:t$ indicates all e.g. measurements starting from 1 to t. Only this simplification makes the algorithm computationally tractable! Otherwise you would need to carry the entire history of measurements and control data and previous states in a buffer with you. Which would grow and grow and grow..

Algorithm 1 $AlgorithmBayes\_filter(bel(x_{t-1}, u_t, z_t))$

1: **for all** x_t do **do**
2: $\bar{bel}(x_{t)} = \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$
3: $bel(x_t) = \alpha p(z_t|x_t)\bar{bel}(x_t$
4: **end for**
5: **return** $bel(x_t)$

Taken from [1] Let's go over this algorithm. It takes three pdf's and in line 2 it calculates the new pdf (belief) about it's current state after having executed some action u. Note, that what this does is to calculate the total probability for being in each possible situation/state. This step is called the *prediction*. (What you model is the state-transition probability). In line 3 it then incorporates new information, a measurement z into its predicted belief, using Bayes Formula. The outcome is a new pdf describing the robot's belief about its current state, after having done one action and one measurement.

Take Home Message The structure of the algorithm is inherent to all such (stochastic recursive state estimators) filters. It is a predictor-corrector structure. Why is that important? What can we do with it? Why is this great? Because it lets us carry knowledge from the past to the present and because of the Markovian assumption and its resulting recursive structure without having to carry a lot of data with us. It let's us incorporate knowledge about how our actions alter our states into our belief about the future state. Thus, what the robot perceives is not independent from our actions. If we have already certain expectations about what is supposed to be sensed we can save processing time because we do not need to consider information that is totally unlikely to be of relevance. It helps us to make sense of noisy measurements. If you do not do this, you have to process all information in every timestep, this is like a pure top-down processing. After each processing you throw all gained information away and start again.

Drawback of the Bayes Filter Algorithm: If you have states whose spaces are not finit you have the problem of conducting the integration in line 2 in closed form. Also the multiplication. But I do not see why that is difficult? Therefore, this type of algorithm is implemented in many different flavors that deal with these shortcomings in different ways.

5 The Kalman Filter

The Kalman Filter is a tractable instance of a Bayes Filter. It is characterised by that it represents the belief by Gaussians. The good thing is, that this makes it possible to calculate beliefs in time polynomilal in the dimension of the state space [1]. There are other filters that also rely on the Gaussian assumption, and in general they are called *Gaussian Filters*. Therefore, first we need to look at Gaussian functions:

5.1 Gaussian Distributions

The Gaussian or normal Distribution is defined by its mean μ= expected value and variance σ^2. That means, whenever you model anything using normal distributions, you are only considering these first two moments and you throw away all the finer information, which is a bit crude. On the other hand it is nice to use Gaussians, because they are easy to compute with. (Remark: In GSL library there are many functions, that automatically create many distributions and draw random numbers from such distributions.
http://www.gnu.org/software/gsl/manua/html_node/index.html#Top)

The normal distribution in the uni- and multivariate (http://www.gseis.ucla.edu/courses/ed231a1/notes4/mat31.html a nice link about bivariate Gaussians) case are given below:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2}(\frac{x-\mu)}{\sigma})^2} \qquad \text{univariate normal distribution} \tag{18}$$

$$f(X) = \frac{1}{\sqrt{2\pi^n}\sqrt{|\Sigma|}} e^{\frac{-1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)} \quad \text{multivariate normal distribution.} \tag{19}$$

Covariance Matrix In the multivariate case μ is a vector and Σ is the covariance matrix, which is like a generalization of the variance. The covariance matrix is a crude description of the shape and orientation of the sample that the Gaussian describes. The covariance matrix completely specifies the distribution apart from the position of μ. The first value in the diagonal describes the variance on the First Principal Component. (A principal component may be shown to minimize the sum of the squares of the distances of point to the Principal Comomnent measured perpendicaularly to this PC.) That is the direction where the variance is largest. The Second Principal Component is orthogonal to the first and the variance on this component is given by the second diagonal value in the covariance matrix. These two components describe the directions where the sample has the largest and the smallest variance. It is possible to just look at the covariance matrix and draw the orientation of a two dimensional normal distribution. E.g. if you consider the example covariance matrix in equation 20 where the first entry is 1, then the spread on the first Principle Component is 1. If the second diagonal

element is 0.5, then the total shape is elliptic. If the off-diagonal elemnts are e.g. -0.5, then the ellipse is rotated clockwise, comp. Fig. 2 (If you diagonalize the covariance matrix, you get another covariance matrix which contains the first two Eigenvalues of the distribution in its diagonal.) (There is a nice animation of this in: www.aiaccess.net/English/Glossaries/GlosMod/Flash/e_gm_fla_covariance_matrix.htm)

$$\begin{pmatrix} 1 & -0.5 \\ -0.5 & 0.5 \end{pmatrix} \tag{20}$$

More properties of covariance are listed in the Appendix.(End of info about

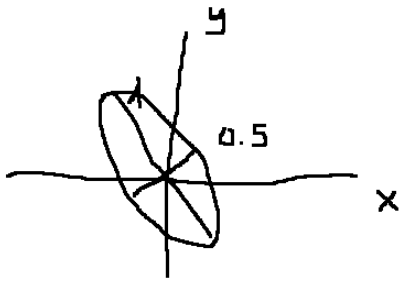


Fig. 2. How to read the covariance matrix.

covariance)

Since the normal distribution is specified by its mean and variance it is also written in the following form:

$$X \sim N(\mu, \Sigma), \tag{21}$$

where X is a RV having the normal distribution as pdf. The following properties of normal distributions are of interest:

–

$$X \sim N(\mu, \Sigma), Y = AX \qquad \Rightarrow Y \sim N(A\mu, A\Sigma A^T) \tag{22}$$

$$X \sim N(\mu, \Sigma), Y = AX + c \qquad \Rightarrow Y \sim N(A\mu + c, A\Sigma A^T) \tag{23}$$

– Normalized product of two Gaussian distributions is a Gaussian.
– The normalized convolution of two Gaussians is a Gaussian.

Now after we have reviewed some properties auf Gaussians we turn back to the Kalman Filter: A typical situation for applying this filter is shown in Fig. 3.

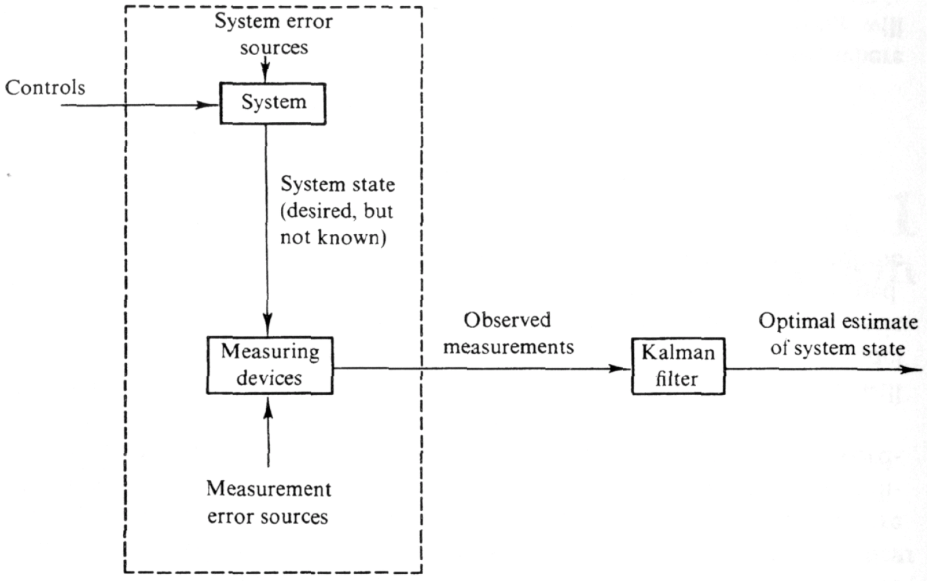


Fig. 3. A typical KF application, image stolen from [2]. Can you observe the connection to Bayes Theorem?

In the Kalman Filter the belief about being in a certain state is described by a pdf with a normal distribution that is assigned to the state, which is modeled as a RV, X. I do understand, that the covariance of the belief describes the uncertainty, and that we must find a way of how to propagate the covariance and the mean over time. What I do not understand is the importance and meaning of P, the covariance of the error of our estimate. The error is also modeled by a pdf? Evtl. because it is important to have an estimate about the precision of your estimate? If you believe that you cannot predict your system's state well, you should put more emphasis on the measurements you make and vv. The covariance of X is the expected value of the expression on p. 108.

– (This description uses the same notation as in **??**.) The Kalman Filter in its pure form assumes a *linear* state transition probability, i.e.:

$$p(x_t|u_t, x_{t-1}) \qquad \text{the state transition probability is described by:} \tag{24}$$

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon \qquad \text{the description of the state evolution} \tag{25}$$

where: A is a $n \times n$ matrix prescribing the system's evolution, B is a $m \times n$ matrix describing the effect that u the control data has on the system's state, ϵ is gaussian noise. The mean of ϵ is zero and its covariance will be denoted by R_t. Note, that line 1 in equation 25 is describing a normally distributed

pdf. The mean of the distribution is $A_t x_{t-1} + B_t u_t$ and the covariance is R_t. If you plug this in the formula for the multivariate normal distribution you obtain:

$$p(x_t|u_t, x_{t-1}) = \frac{1}{\sqrt{2\pi R_t}} e^{\frac{-1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)} \quad (26)$$

- The measurement probability $p(t_t|x_t)$:

$$z_t = C_t x_t + \delta t, \quad (27)$$

where C is a $k \times n$ matrix, with k being the dimension of the measurement vector z_t. The vector δ_t describes measurement noise with mean zero and covariance Q_t.
- the initial belief must also be normallydistributed.

These three assumptions are sufficient to ensure that the posterior belief is always a Gaussian for any point in time.

Algorithm 3 $AlgorithmKalman\_filter(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$

1: $\bar{\mu} = A_t \mu_{t-1} + B_t u_t$
2: $\bar{\Sigma} = A_t \Sigma_{t-1} A_t^T + R_t$
3: $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
4: $\mu_t = \bar{\mu_t} + K_t(z_t - C_t \bar{\mu_t})$
5: $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
6: **return** μ_t, Σ_t

The mathematical derivation of this goes over several pages in a textbook. Important is to note, that this is structurally equal to that of the presented Bayes filter. In the first two lines the information about the state-transition is incorporated into a given initial belief. In the remaining lines the update step is conducted, i.e. the information from a measurement in incorporated. In line 4 the so-called Kalman Gain K is calculated which is based on *innovation*. That means that the error between the predicted measurement $(C_t \bar{\mu_t})$and the actual measurement z_t is calculated and based on that difference it is decided how much to "trust" new measurements. In Fig. **??** the effect of the filter is shown.

5.2 Advantages and Disadvantages

- + recursive and gaussian distributions = efficient!
- + uses normal distributions for belief progagation, that is easy to use!
- - uses normal distributions for belief propagation, that is not very precise and sometimes not appropriate. When is it inappropriate? Since Gaussians are unimodal distributions it is good for, e.g. tracking. But if the modeled problem must maintain different hypothesis your are better off with a multimodal distribution.

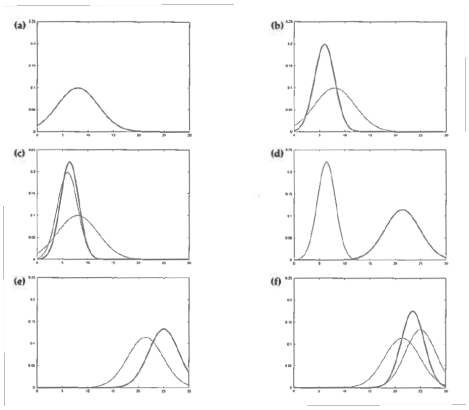


Fig. 4. Shows the effect of the KF, image taken from **??**.

- - only for linear applications.
- - only applicable if the exact system dynamics are known! How to get the system dynamics? → system identification.

5.3 Some later added stuff

Marginalization: $P(A) = P(A \cap B) + P(A \cap \bar{B})$

The Kalman filter's heart is the Kalman Gain. You want to filter a value, let's say. Then what to trust, the measurement or the prediction? The correction is done in this way:

$$\hat{x}_j = \hat{x}_{\bar{j}} + k(Residual) = \tag{28}$$

$$\hat{x}_j = \hat{x}_{\bar{j}} + k(z_j - h\hat{x}_{\bar{j}}) = \tag{29}$$

$$\tag{30}$$

The belief, that is the a posteriori estimate is corrected in accordance to the a priori estimate + Kalman Gain times the difference between the actual measurement and the a prori estimate. Now: You choose k, the Kalman Gain, such that it minimizes the uncertainty about your result. That makes sense, at the end you want to return THE value that you are most sure of. In other words, choose k, s.t. it minimize the expected value of the difference between the a posterior estimate and the measurement:

$$p_{\bar{j}} = E\{\hat{x}_j - \hat{x}_{\bar{j}}\} \tag{31}$$

$$(x_j - \hat{x}_j)^2 = (x_j - \hat{x}_{\bar{j}} + k(z_j - h\hat{x}_{\bar{j}}))^2 \tag{32}$$

To find the k that minimizes this expression you need to derive it wrt k and set equal 0. Doing some calculations leads to:

$$k = \frac{hp\bar{j}}{h^2 p_{\bar{j}} + R} \tag{33}$$

Together with equation 28 it can be seen that the correction to the estimate is big, whenever k is big. That means, in that case we do not trust the prediction, but rather the measurement. And that we trust the measurement more then the prediction when k is small. We see, that the value of k depends on to variables, which is the measurement error covariance R and the expected value of the a priori estimate, $p_{\bar{j}} = E\{(e_{\bar{j}})^2\}$ and $e_{\bar{j}} = x_j - \hat{x}_{\bar{j}}$ Thus, you see that when the expected value of the a priori estimate is large, then k is large and we will rely more on the measurement than the prediction. On the other hand, if R, the measurement error covariance is high, then k will be small an we will rely more on the prediction than the measurement.

6 Scalar Example

Here, I use the KF to estimate a constant value. This is the implementation of the example from the tutorial "An introduction to the Kalman Filter" Bishop and Welsh p.31 On this webpage they also treat this problem.
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/WELCH/kalman.3.html
This is the code that I wrote in scilab.
makeNoiseSignalForKFTesting.sce (Arbeit/trunk/MyCode/ScilabCode)
testKF.sce
Task: Filter a noisy signal which is a constant of value -0.37727. The noise it is corrupted with is white and has standard deviation of 0.1. First make the signal. Then do the filtering on the signal. Initial values: This is the result:

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
2. Maybeck, P.S.: Stochastic Models, Estimation, and Control, Vol. 1. Academic Press (1979)
3. Simon, D.: Optimal State Estimation. Wiley (2006)
4. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transaction of the ASMEJournal of Basic Engineering (1960) 33–45

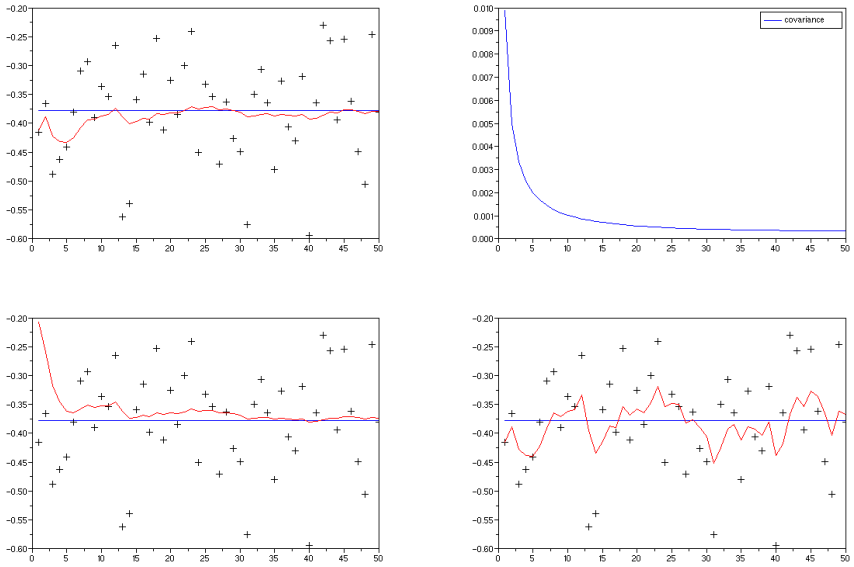


Fig. 5. From top to bottom, a) from left to right: R = 0.1 b) the associated covariance d) R=1, slow to trust measurements d) R=0.00001 small, fast to trust measurements