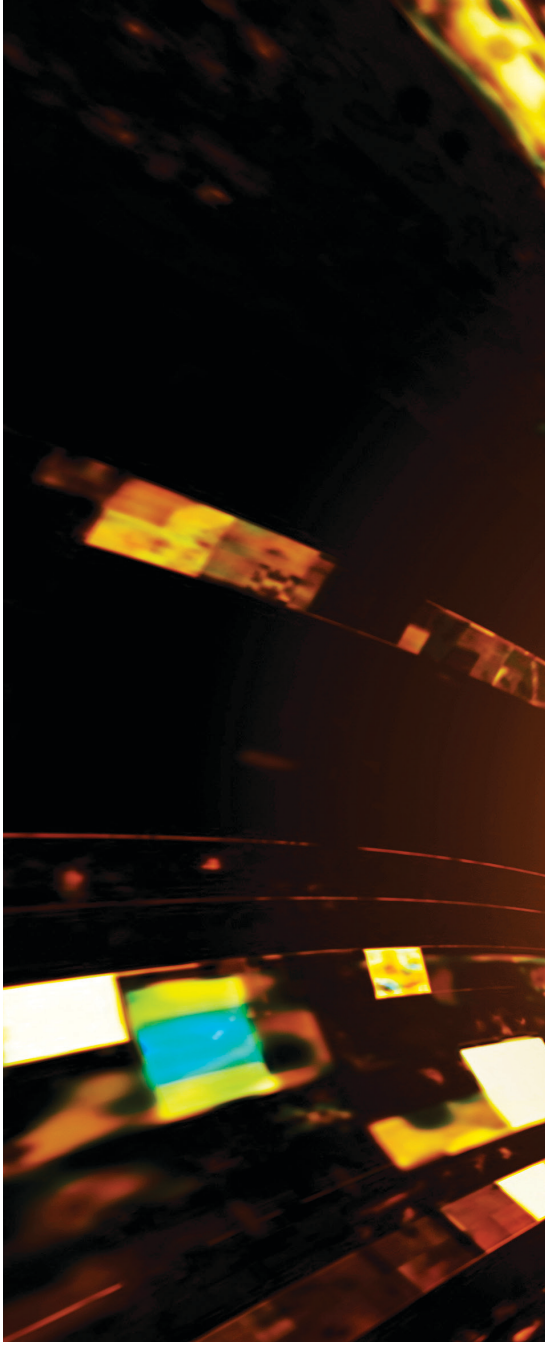
DOI:10.1145/3363294

Demystifying the uses of a powerful tool for uncertain information.

BY YAN PEI, SWARNENDU BISWAS, DONALD S. FUSSELL, AND KESHAV PINGALI

An Elementary Introduction to Kalman Filtering



KALMAN FILTERING IS a state estimation technique used in many application areas such as spacecraft navigation, motion planning in robotics, signal processing, and wireless sensor networks because of its ability to extract useful information from noisy data and its small computational and memory requirements.[12,20,27–29] Recent work has used Kalman filtering in controllers for computer systems.[5,13,14,23]

Although many introductions to Kalman filtering are available in the literature,[1–4,6–11,17,21,25,29] they are usually focused on particular applications such as robot motion or state estimation in linear systems, making it difficult to see how to apply Kalman filtering to other problems. Other presentations derive Kalman filtering as an application of Bayesian inference, assuming that noise is Gaussian. This leads to the common misconception that Kalman filtering can be applied only if noise is Gaussian.[15]

Abstractly, Kalman filtering can be seen as a particular approach to combining approximations of an unknown value to produce a better approximation. Suppose we use two devices of different

» key insights

- This article presents an elementary derivation of Kalman filtering, a classic state estimation technique.
- Understanding Kalman filtering is useful for more principled control of computer systems.
- Kalman filtering is used as a black box by many computer scientists.

IMAGE BY A LUNA BLUE

designs to measure the temperature of a CPU core. Because devices are usually noisy, the measurements are likely to differ from the actual temperature of the core. As the devices are of different designs, let us assume that noise affects the two devices in unrelated ways (this is formalized here using the notion of correlation). Therefore, the measurements x_1 and x_2 are likely to be different from each other and from the actual core temperature x_c. A natural question is the following: is there a way to combine the information in the noisy measurements x_1 and x_2 to obtain a good approximation of the actual temperature x_c?

One ad hoc solution is to use the formula $0.5*x_1+0.5*x_2$ to take the average of the two measurements, giving them equal weight. Formulas of this sort are called *linear estimators* because they use a weighted sum to fuse values; for our temperature problem, their general form is $\beta*x_1+\alpha*x_2$. In this presentation, we use the term *estimate* to refer to both a noisy measurement and a value computed by an estimator, as both are approximations of unknown values of interest.

Suppose we have additional information about the two devices, say the second one uses more advanced temperature sensing. Because we would have more confidence in the second measurement, it seems reasonable that we should discard the first one, which is equivalent to using the linear estimator $0.0*x_1 + 1.0*x_2$. Kalman filtering tells us that in general, this intuitively reasonable linear estimator is not "optimal;" paradoxically, there is useful information even in the measurement from the lower quality device, and the optimal estimator is one in which the weight given to each measurement is proportional to the confidence we have in the device producing that measurement. Only if we have no confidence whatever in the first device should we discard its measurement.

The goal of this article[a] is to present the abstract concepts behind Kalman filtering in a way that is accessible to most computer scientists while clarifying the key assumptions, and then show how the problem of state estimation in linear systems can be solved as an

a An extended version of this article that includes additional background material and proofs is available.[30]

application of these general concepts. First, the informal ideas discussed here are formalized using the notions of distributions and random samples from distributions. Confidence in estimates is quantified using the variances and covariances of these distributions.[b] Two algorithms are described next. The first one shows how to fuse estimates (such as core temperature measurements) optimally, given a reasonable definition of optimality. The second algorithm addresses a problem that arises frequently in practice: estimates are vectors (for example, the position and velocity of a robot), but only a part of the vector can be measured directly; in such a situation, how can an estimate of the entire vector be obtained from an estimate of just a part of that vector? The best linear unbiased estimator (BLUE) is used to solve this problem.[16,19,26] It is shown that the Kalman filter can be derived in a straightforward way by using these two algorithms to solve the problem of state estimation in linear systems. The extended Kalman filter and unscented Kalman filter, which extended Kalman filtering to nonlinear systems, are described briefly at the end of the article.

Formalizing Estimates

Scalar estimates. To model the behavior of devices producing noisy temperature measurements, we associate each device i with a *random variable* that has a *probability density function* (pdf) $p_i(x)$ such as the ones shown in Figure 1 (the x-axis in this figure represents temperature). Random variables need not be Gaussian.[c] Obtaining a measurement from device i corresponds to drawing a random sample from the distribution for that device. We write $x_1 \sim p_1(\mu_1, \sigma_1^2)$ to denote that x_i is a random variable with pdf p_i whose mean and variance are μ_i and σ_i^2, respectively; following convention, we use x_i to represent a random sample from this distribution as well.

Means and variances of distributions model different kinds of inaccuracies in measurements. Device i is said to have a *systematic error* or *bias* in its measurements if the mean μ_i of its distribution is not equal to the actual temperature x_c (in general, to the value being estimated, which is known as *ground truth*); otherwise, the instrument is unbiased. Figure 1 shows pdfs for two devices that have different amounts of systematic error. The variance σ_i^2 on the other hand is a measure of the *random error* in the measurements. The impact of random errors can be mitigated by taking many measurements with a given device and averaging their values, but this approach will not reduce systematic error.

In the formulation of Kalman filtering, it is assumed that measuring devices do not have systematic errors. However, we do not have the luxury of taking many measurements of a given state, so we must take into account the impact of random error on a single measurement. Therefore, confidence in a device is modeled formally by the variance of the distribution associated with that device; the smaller the variance, the higher our confidence in the measurements made by the device. In Figure 1, the fact we have less confidence in the first device has been illustrated by making p_1 more spread out than p_2, giving it a larger variance.

The informal notion that noise should affect the two devices in "unrelated ways" is formalized by requiring that the corresponding random variables be *uncorrelated*. This is a weaker condition than requiring them to be *independent*, as explained in our online appendix (http://dl.acm.org/citation.cfm?doid=3363294&picked=formats). Suppose we are given the measurement made by one of the devices (say x_1) and we have to guess what the other measurement (x_2) might be. If knowing x_1 does not give us any new information about what x_2 might be, the random variables are independent. This is expressed formally by the equation $p(x_2|x_1) = p(x_2)$; intuitively, knowing the value of x_1 does not change the pdf for the possible values of x_2. If the random variables are only uncorrelated, knowing x_1 might give us new information about x_2 such as restricting its possible values but the mean of $x_2|x_1$ will still be μ_2. Using expectations, this can be written as $E[x_2|x_1] = E[x_2]$, which is equivalent to requiring that $E[(x_1-\mu_1)(x_2-\mu_2)]$, the covariance between the two variables, be equal to zero. This is obviously a weaker condition than independence.

Although the discussion in this section has focused on measurements, the same formalization can be used for estimates produced by an estimator. Lemma 1(i) shows how the mean and variance of a linear combination of pairwise uncorrelated random variables can be computed from the means and variances of the random variables.[18] The mean and variance can be used to quantify bias and random errors for the estimator as in the case of measurements.

An *unbiased estimator* is one whose mean is equal to the unknown value being estimated and it is preferable to a biased estimator with the same variance. Only unbiased estimators are considered in this article. Furthermore, an unbiased estimator with a smaller variance is preferable to one with a larger variance as we would have more confidence in the estimates it produces. As a step toward generalizing this discussion to estimators that produce vector estimates, we refer to the variance of an unbiased scalar estimator as the *mean square error* of that estimator or *MSE* for short.

Lemma 1(ii) asserts that if a random variable is pairwise uncorrelated with a set of random variables, it is uncorrelated with any linear combination of those variables.

Lemma 1. *Let* $x_1 \sim p_1(\mu_1, \sigma_1^2), .., x_n \sim p_n(\mu_n, \sigma_n^2)$ *be a set of pairwise uncorrelated random variables. Let* $y = \sum_{i=1}^{n} \alpha_i x_i$ *be a random variable that is a linear combination of the* x_i*'s.*

(i) The mean and variance of y are:

$$\mu_y = \sum_{i=1}^{n} \alpha_i \mu_i \quad (1)$$

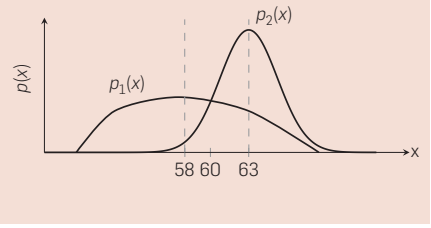
$$\sigma_y^2 = \sum_{i=1}^{n} \alpha_i^2 \sigma_i^2 \quad (2)$$

(ii) If random variable x_{n+1} *is pair-wise uncorrelated with* $x_1, .., x_n$*, it is uncorrelated with y.*

b Basic concepts such as probability density function, mean, expectation, variance and covariance are introduced in the online appendix.

c The role of Gaussians in Kalman filtering is discussed later in the article.

Figure 1. Using pdfs to model devices with systematic and random errors. Ground truth is 60°C. Dashed lines are means of pdfs.



Vector estimates. In some applications, estimates are vectors. For example, the state of a mobile robot might be represented by a vector containing its position and velocity. Similarly, the vital signs of a person might be represented by a vector containing his temperature, pulse rate, and blood pressure. Here, we denote a vector by a boldfaced lowercase letter, and a matrix by an uppercase letter.

The covariance matrix $\sum_{xx}$ of a random variable $\mathbf{x}$ is the matrix $E[(\mathbf{x} - \boldsymbol{\mu}_x)(\mathbf{x} - \boldsymbol{\mu}_x)^T]$, where $\boldsymbol{\mu}_x$ is the mean of $\mathbf{x}$. Intuitively, entry (i,j) of this matrix is the covariance between the i and j components of vector $\mathbf{x}$; in particular, entry (i,i) is the variance of the i^{th} component of $\mathbf{x}$. A random variable $\mathbf{x}$ with a pdf p whose mean is $\boldsymbol{\mu}_x$ and covariance matrix is $\sum_{xx}$ is written as $\mathbf{x} \sim p(\boldsymbol{\mu}_x, \sum_{xx})$. The inverse of the covariance matrix $(\sum_{xx}^{-1})$ is called the precision or *information* matrix.

Uncorrelated random variables. The cross-covariance matrix $\sum_{vw}$ of two random variables $\mathbf{v}$ and $\mathbf{w}$ is the matrix $E[(\mathbf{v}-\boldsymbol{\mu}_v)(\mathbf{w}-\boldsymbol{\mu}_w)^T]$. Intuitively, element (i,j) of this matrix is the covariance between elements $\mathbf{v}(i)$ and $\mathbf{w}(j)$. If the random variables are uncorrelated, all entries in this matrix are zero, which is equivalent to saying that every component of $\mathbf{v}$ is uncorrelated with every component of $\mathbf{w}$. Lemma 2 generalizes Lemma 1.

Lemma 2. *Let* $\boldsymbol{x}_1 \sim p_1(\boldsymbol{\mu}_1, \sum_1)$, ..., $\boldsymbol{x}_n \sim p_n(\boldsymbol{\mu}_n, \sum_n)$ *be a set of pairwise uncorrelated random variables of length m. Let* $\boldsymbol{y} = \sum_{i=1}^{n} A_i \boldsymbol{x}_i$.

(i) The mean and covariance matrix of $\boldsymbol{y}$ *are the following:*

$$\boldsymbol{\mu}_y = \sum_{i=1}^{n} A_i \boldsymbol{\mu}_i \quad (3)$$

$$\sum_{yy} = \sum_{i=1}^{n} A_i \sum_i A_i^T \quad (4)$$

(ii) If random variable $\boldsymbol{x}_{n+1}$ *is pairwise uncorrelated with* $\boldsymbol{x}_1$, .., $\boldsymbol{x}_n$, *it is uncorrelated with* $\boldsymbol{y}$.

The *MSE* of an unbiased estimator $\mathbf{y}$ is $E[(\mathbf{y}-\boldsymbol{\mu}_y)^T(\mathbf{y}-\boldsymbol{\mu}_y)]$, which is the sum of the variances of the components of $\mathbf{y}$; if $\mathbf{y}$ has length 1, this reduces to variance as expected. The *MSE* is also the sum of the diagonal elements of $\sum_{yy}$ (this is called the *trace* of $\sum_{yy}$).

An unbiased estimator is one whose mean is equal to the unknown value being estimated and it is preferable to a biased estimator with the same variance.

Fusing Scalar Estimates

We now consider the problem of choosing the optimal values of the parameters α and β in the linear estimator $\beta * x_1 + \alpha * x_2$ for fusing two estimates x_1 and x_2 from uncorrelated scalar-valued random variables.

The first reasonable requirement is that if the two estimates x_1 and x_2 are equal, fusing them should produce the same value. This implies that $\alpha+\beta=1$. Therefore, the linear estimators of interest are of the form

$$y_\alpha(x_1, x_2) = (1-\alpha) * x_1 + \alpha * x_2 \quad (5)$$

If x_1 and x_2 in Equation 5 are considered to be unbiased estimators of some quantity of interest, then y_α is an unbiased estimator for any value of α. How should optimality of such an estimator be defined? One reasonable definition is that the optimal value of α *minimizes the variance of* y_α as this will produce the highest-confidence fused estimates.

Theorem 1. *Let* $x_1 \sim p_1(\mu_1, \sigma_1^2)$ *and* $x_2 \sim p_2(\mu_2, \sigma_2^2)$ *be uncorrelated random variables. Consider the linear estimator* $y_\alpha(x_1,x_2)=(1-\alpha)*x_1+\alpha*x_2$. *The variance of the estimator is minimized for* $\alpha = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$.

The proof is straightforward and is given in the online appendix. The variance (*MSE*) of y_α can be determined from Lemma 1:

$$\sigma_y^2(\alpha) = (1-\alpha)^2 * \sigma_1^2 + \alpha^2 * \sigma_2^2 \quad (6)$$

Setting the derivative of $\sigma_y^2(\alpha)$ with respect to α to zero and solving the resulting equation yield the required result.

In the literature, the optimal value of α is called the *Kalman gain K*. Substituting K into the linear fusion model, we get the optimal linear estimator $y(x_1, x_2)$:

$$y(x_1,x_2) = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} * x_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} * x_2 \quad (7)$$

As a step toward fusion of $n>2$ estimates, it is useful to rewrite this as follows:

$$y(x_1,x_2) = \frac{\frac{1}{\sigma_1^2}}{\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2}} * x_1 + \frac{\frac{1}{\sigma_2^2}}{\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2}} * x_2 \quad (8)$$

Substituting the optimal value of α into Equation 6, we get

$$\sigma_y^2 = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} \quad (9)$$

The expressions for y and σ_y^2 are complicated because they contain the reciprocals of variances. If we let ν_1 and ν_2 denote the precisions of the two distributions, the expressions for y and ν_y can be written more simply as follows:

$$y(x_1, x_2) = \frac{\nu_1}{\nu_1 + \nu_2} * x_1 + \frac{\nu_2}{\nu_1 + \nu_2} * x_2 \quad (10)$$

$$\nu_y = \nu_1 + \nu_2 \quad (11)$$

These results say the weight we should give to an estimate is proportional to the confidence we have in that estimate, and that we have more confidence in the fused estimate than in the individual estimates, which is intuitively reasonable. To use these results, we need only the variances of the distributions. In particular, the pdfs p_i, which are usually not available in applications, are not needed, and the proof of Theorem 1 does not require these pdfs to have the same mean.

Fusing multiple scalar estimates. These results can be generalized to optimally fuse multiple pairwise uncorrelated estimates x_1, x_2, ..., x_n. Let $y_{n,\alpha}(x_1, .., x_n)$ denote the linear estimator for fusing the n estimates given parameters α_1, .., α_n, which we denote by α (the notation $y_\alpha(x_1, x_2)$ introduced previously can be considered to be an abbreviation of $y_{2,\alpha}(x_1, x_2)$).

Theorem 2. *Let $x_i \sim p_i(\mu_i, \sigma_i^2)$ for $(1 \le i \le n)$ be a set of pairwise uncorrelated random variables. Consider the linear estimator $y_{n,\alpha}(x_1, .., x_n) = \sum_{i=1}^{n} \alpha_i x_i$ where $\sum_{i=1}^{n} \alpha_i = 1$. The variance of the estimator is minimized for*

$$\alpha_i = \frac{\frac{1}{\sigma_i^2}}{\sum_{j=1}^{n} \frac{1}{\sigma_j^2}}$$

The minimal variance is given by the following expression:

$$\sigma_{y_n}^2 = \frac{1}{\sum_{j=1}^{n} \frac{1}{\sigma_j^2}} \quad (12)$$

As before, these expressions are more intuitive if the variance is replaced with precision: the contribution of x_i to the value of $y_n(x_1, .., x_n)$ is proportional to x_i's confidence.

Kalman filtering can be seen as a particular approach to combining approximations of an unknown value to produce a better approximation.

$$y_n(x_1, .., x_n) = \sum_{i=1}^{n} \frac{\nu_i}{\nu_1 + \ldots + \nu_n} * x_i \quad (13)$$

$$\nu_{y_n} = \sum_{i=1}^{n} \nu_i \quad (14)$$

Equations 13 and 14 generalize Equations 10 and 11.

Incremental fusing is optimal. In many applications, the estimates x_1, x_2, ..., x_n become available successively over a period of time. Although it is possible to store all the estimates and use Equations 13 and 14 to fuse all the estimates from scratch whenever a new estimate becomes available, it is possible to save both time and storage if one can do this fusion incrementally. We show that just as a sequence of numbers can be added by keeping a running sum and adding the numbers to this running sum one at a time, a sequence of $n>2$ estimates can be fused by keeping a "running estimate" and fusing estimates from the sequence one at a time into this running estimate without any loss in the quality of the final estimate. In short, we want to show that $y_n(x_1,..,x_n)=y_2(y_2(..y_2(x_1,x_2)...),x_n)$. A little bit of algebra shows that if $n>2$, Equations 13 and 14 for the optimal linear estimator and its precision can be expressed as shown in Equations 15 and 16.

$$y_n(x_1, .., x_n) = \frac{\nu_n}{\nu_{y_{n-1}} + \nu_n} x_n + \frac{\nu_{y_{n-1}}}{\nu_{y_{n-1}} + \nu_n} y_{n-1}(x_1, \ldots, x_{n-1}) \quad (15)$$

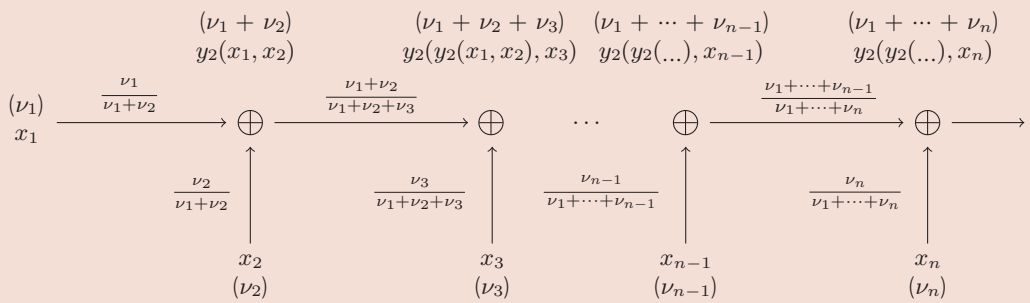
$$\nu_{y_n} = \nu_{y_{n-1}} + \nu_n \quad (16)$$

This shows that $y_n(x_1, .., x_n) = y_2(y_{n-1}(x_1, .., x_{n-1}), x_n)$. Using this argument recursively gives the required result.[d]

To make the connection to Kalman filtering, it is useful to derive the same result using a pictorial argument. Figure 2 shows the process of incrementally fusing the n estimates. In this picture, time progresses from left to right, the precision of each estimate is shown in parentheses next to it, and the weights on the edges are the weights from Equation 10. The contribution of each x_i to the final value $y_2(y_2(...), x_n)$ is given by the product of the weights on the path from x_i to the final value, and this product is obviously equal to the weight of x_i in

d We thank Mani Chandy for showing us this approach to proving the result.

Figure 2. Dataflow graph for incremental fusion.



Equation 13, showing that incremental fusion is optimal.

Summary. The results in this section can be summarized informally as follows. *When using a linear estimator to fuse uncertain scalar estimates, the weight given to each estimate should be inversely proportional to the variance of the random variable from which that estimate is obtained. Furthermore, when fusing n>2 estimates, estimates can be fused incrementally without any loss in the quality of the final result.* These results are often expressed formally in terms of the Kalman gain K, as shown in Figure 3; the equations can be applied recursively to fuse multiple estimates. Note that if $\nu_1 \gg \nu_2$, $K \approx 0$ and $y(x_1,x_2) \approx x_1$; conversely if $\nu_1 \ll \nu_2$, $K \approx 1$ and $y(x_1,x_2) \approx x_2$.

Fusing Vector Estimates

The results for fusing scalar estimates can be extended to vectors by replacing *variances* with *covariance matrices*.

For vectors, the linear estimator is $\boldsymbol{y}_{n,A}(\mathbf{x}_1,\mathbf{x}_2,..,\mathbf{x}_n)=\sum_{i=1}^{n} A_i\mathbf{x}_i$ where $\sum_{i=1}^{n} A_i = I$. Here A stands for the matrix parameters $(A_1,...,A_n)$. All the vectors $(\mathbf{x}_i)$ are assumed to be of the same length. To simplify notation, we omit the subscript n in $\mathbf{y}_{n,A}$ in the discussion here as it is obvious from the context.

Optimality. The parameters A_1, ..., A_n in the linear data fusion model are chosen to minimize $MSE(\mathbf{y}_A)$ which is $E[(\mathbf{y}_A-\boldsymbol{\mu}_{yA})^{\mathrm{T}}(\mathbf{y}_A-\boldsymbol{\mu}_{yA})]$.

Theorem 3 generalizes Theorem 2 to the vector case. The proof of this theorem is given in the appendix. Comparing Theorems 2 and 3, we see that the expressions are similar, the main difference being that the role of variance in the scalar case is played by the covariance matrix in the vector case.

Theorem 3. *Let $\mathbf{x}_i \sim p_i(\boldsymbol{\mu}_i, \Sigma_i)$ for $(1 \le i \le n)$ be a set of pairwise uncorrelated random variables. Consider the linear estimator $\boldsymbol{y}_A(\boldsymbol{x}_1,..,\boldsymbol{x}_n)=\sum_{i=1}^{n} A_i\boldsymbol{x}_i$, where $\sum_{i=1}^{n} A_i = I$. The value of $MSE(\boldsymbol{y}_A)$ is minimized for*

$$A_i = \Big(\sum_{j=1}^{n} \Sigma_j^{-1}\Big)^{-1} \Sigma_i^{-1} \tag{23}$$

Therefore the optimal estimator is

$$\boldsymbol{y}(\mathbf{x}_1,\ldots,\mathbf{x}_n) = \Big(\sum_{j=1}^{n} \Sigma_j^{-1}\Big)^{-1} \sum_{i=1}^{n} \Sigma_i^{-1}\mathbf{x}_i \tag{24}$$

The covariance matrix of $\mathbf{y}$ can be computed by using Lemma 2.

$$\Sigma_{\mathbf{yy}} = \Big(\sum_{j=1}^{n} \Sigma_j^{-1}\Big)^{-1} \tag{25}$$

In the vector case, precision is the inverse of a covariance matrix, denoted by N. Equations 26–27 use precision to express the optimal estimator and its variance and generalize Equations 13–14 to the vector case.

$$\mathbf{y}(\mathbf{x}_1,\ldots,\mathbf{x}_n) = N_{\mathbf{y}}^{-1} \sum_{i=1}^{n} N_i\mathbf{x}_i \tag{26}$$

$$N_{\mathbf{y}} = \sum_{j=1}^{n} N_j \tag{27}$$

As in the scalar case, fusion of $n>2$ vector estimates can be done incrementally without loss of precision. The proof is similar to the scalar case and is omitted.

Figure 3. Optimal fusion of scalar estimates.

$$x_1 \sim p_1(\mu_1,\sigma_1^2), \quad x_2 \sim p_2(\mu_2,\sigma_2^2)$$

$$K = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} = \frac{\nu_2}{\nu_1+\nu_2} \tag{17}$$

$$y(x_1,x_2) = x_1 + K(x_2 - x_1) \tag{18}$$

$$\sigma_y^2 = (1-K)\sigma_1^2 \quad \text{or} \quad \nu_y = \nu_1+\nu_2 \tag{19}$$

Figure 4. Optimal fusion of vector estimates.

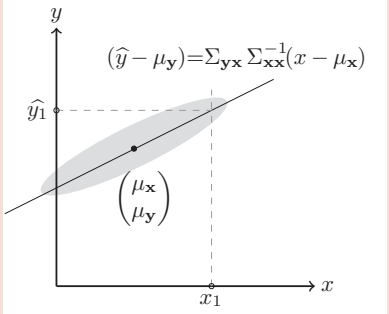
$$\mathbf{x}_1 \sim p_1(\boldsymbol{\mu}_1,\Sigma_1), \quad \mathbf{x}_2 \sim p_2(\boldsymbol{\mu}_2,\Sigma_2)$$

$$K = \Sigma_1(\Sigma_1+\Sigma_2)^{-1} = (N_1+N_2)^{-1}N_2 \tag{20}$$

$$\mathbf{y}(\mathbf{x}_1,\mathbf{x}_2) = \mathbf{x}_1 + K(\mathbf{x}_2-\mathbf{x}_1) \tag{21}$$

$$\Sigma_{\mathbf{yy}} = (I-K)\Sigma_1 \quad \text{or} \quad N_{\mathbf{y}} = N_1+N_2 \tag{22}$$

Figure 5. BLUE line corresponding to Equation (31).



There are several equivalent expressions for the Kalman gain for the fusion of two estimates. The following one, which is easily derived from Equation 23, is the vector analog of Equation 17:

$$K = \Sigma_1(\Sigma_1 + \Sigma_2)^{-1} \quad (28)$$

The covariance matrix of the optimal estimator $\mathbf{y}(\mathbf{x}_1, \mathbf{x}_2)$ is the following.

$$\Sigma_{yy} = \Sigma_1(\Sigma_1 + \Sigma_2)^{-1}\Sigma_2 \quad (29)$$
$$= K\Sigma_2 = \Sigma_1 - K\Sigma_1 \quad (30)$$

Summary. The results in this section can be summarized in terms of the Kalman gain K as shown in Figure 4.

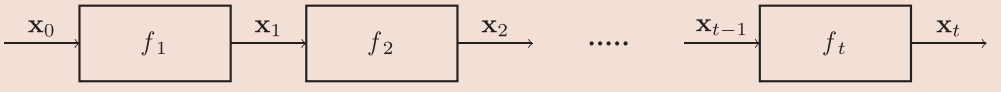
Best Linear Unbiased Estimator (BLUE)

In some applications, the state of the system is represented by a vector but only part of the state can be measured directly, so it is necessary to estimate the hidden portion of the state corresponding to a measured value of the visible state. This section describes an estimator called the *best linear unbiased estimator* (BLUE)[16,19,26] for doing this.

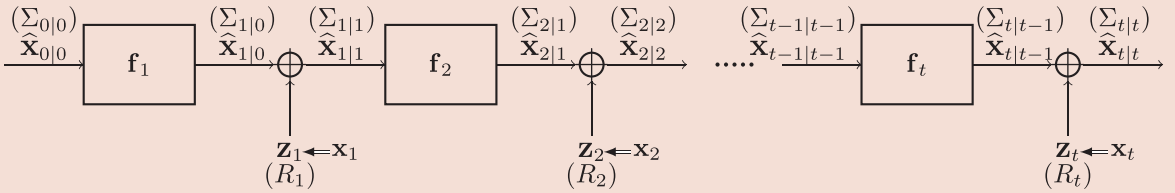
Consider the general problem of determining a value for vector **y** given a value for a vector **x**. If there is a functional relationship between **x** and **y** (say $\mathbf{y}=F(\mathbf{x})$ and F is given), it is easy to compute **y** given a value for **x** (say $\mathbf{x}_1$).

In our context, however, **x** and **y** are random variables, so such a precise functional relationship will not hold. Figure 5 shows an example in which x and y are scalar-valued random variables. The gray ellipse in this figure, called a *confidence ellipse*, is a projection of the joint distribution of x and y onto the (x, y) plane that shows where some large proportion of the (x, y) values are likely to be. Suppose x takes the value x_1. Even within the confidence ellipse, there are many points (x_1, y), so we cannot associate a single value of y with x_1. One possibility is to compute the mean of the y values associated with x_1 (that is,

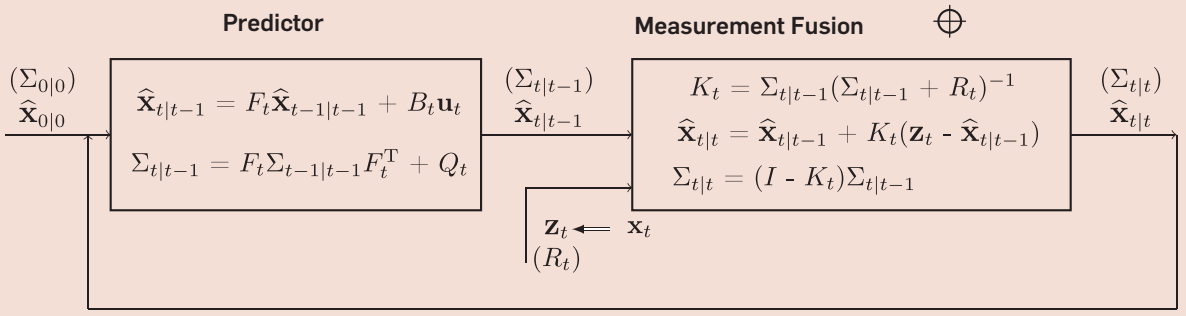
Figure 6. State estimation using Kalman filtering.



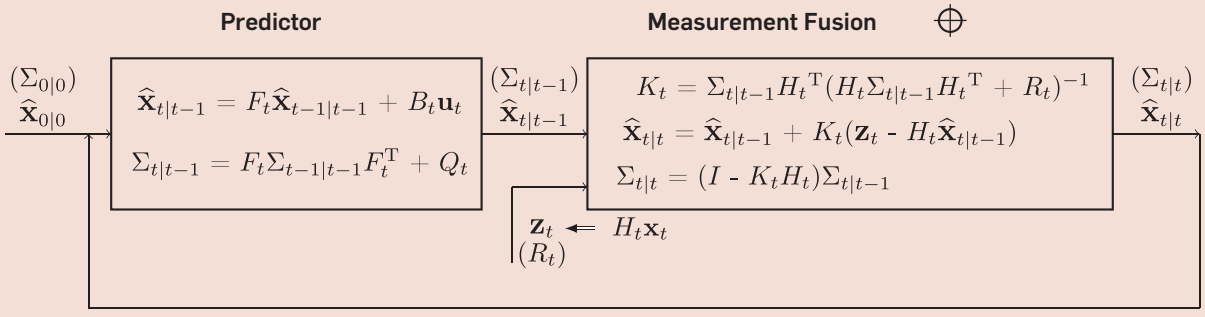
(a) Discrete-time dynamical system.



(b) Dynamical system with uncertainty.



(c) Implementation of the dataflow diagram (b).



(d) Implementation of the dataflow diagram (b) for systems with partial observability.

the expectation $E[y|x=x_1]$) and return this as the estimate for y if $x=x_1$. This requires knowing the joint distribution of x and y, which may not always be available.

In some problems, we can assume that there is an unknown linear relationship between **x** and **y** and that uncertainty comes from noise. Therefore, we can use a technique similar to the ordinary least squares (OLS) method to estimate this linear relationship, and use it to return the best estimate of y for any given value of x. In Figure 5, we see that although there are many points (x_1, y), the y values are clustered around the line as shown in the figure so the value $\hat{y}_1$ is a reasonable estimate for the value of y corresponding to x_1. This line, called the *best linear unbiased estimator* (BLUE), is the analog of ordinary least squares (OLS) for distributions.

Computing BLUE. Consider the estimator $\hat{\mathbf{y}}_{A,\mathbf{b}}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$. We choose A and **b** so that this is an unbiased estimator with minimal *MSE*. The "^" over the **y** is notation that indicates that we are computing an estimate for **y**.

Theorem 4. *Let*

$$\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix} \sim p\left(\begin{pmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{pmatrix}, \begin{pmatrix}\Sigma_{xx} & \Sigma_{xy}\\ \Sigma_{yx} & \Sigma_{yy}\end{pmatrix}\right).$$

The estimator $\hat{\boldsymbol{y}}_{A,\boldsymbol{b}}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ *for estimating the value of* $\boldsymbol{y}$ *for a given value of* $\boldsymbol{x}$ *is an unbiased estimator with minimal MSE if*

$$\boldsymbol{b} = \boldsymbol{\mu}_y - A(\boldsymbol{\mu}_x)$$

$$A = \Sigma_{yx}\Sigma_{xx}^{-1}$$

The proof of Theorem 4 is straight forward. For an unbiased estimator, $E[\hat{\mathbf{y}}] = E[\mathbf{y}]$. This implies that $\mathbf{b}=\boldsymbol{\mu}_y - A(\boldsymbol{\mu}_x)$ so an unbiased estimator is of the form $\hat{\mathbf{y}}_A(\mathbf{x}) = \boldsymbol{\mu}_y + A(\mathbf{x} - \boldsymbol{\mu}_x)$. Note this is equivalent to asserting the BLUE line must pass through the point $(\boldsymbol{\mu}_x, \boldsymbol{\mu}_y)$. Setting the derivative of $MSE_A(\hat{\mathbf{y}}_A)$ with respect to A to zero[22] and solving for A, we find that the best linear unbiased estimator is

$$\hat{\mathbf{y}} = \boldsymbol{\mu}_y + \Sigma_{yx}\Sigma_{xx}^{-1}(\mathbf{x} - \boldsymbol{\mu}_x) \qquad (31)$$

This equation can be understood intuitively as follows. If we have no information about **x** and **y**, the best we can do is the estimate $(\boldsymbol{\mu}_x, \boldsymbol{\mu}_y)$, which lies on the BLUE line. However, if we know that **x** has a particular value $\mathbf{x}_1$, we can use the correlation between **y** and **x** to estimate a better value for **y** from the difference $(\mathbf{x}_1 - \boldsymbol{\mu}_x)$. Note that if $\Sigma_{yx} = 0$ (that is, **x** and **y** are uncorrelated), the best estimate of **y** is just $\boldsymbol{\mu}_y$, so knowing the value of **x** does not give us any additional information about **y** as one would expect. In Figure 5, this corresponds to the case when the BLUE line is parallel to the x-axis. At the other extreme, suppose that **y** and **x** are functionally related so $\mathbf{y} = C\mathbf{x}$. In that case, it is easy to see that $\Sigma_{yx} = C\Sigma_{xx}$, so $\hat{\mathbf{y}} = C\mathbf{x}$ as expected. In Figure 5, this corresponds to the case when the confidence ellipse shrinks down to the BLUE line.

Equation 31 is a generalization of ordinary least squares in the sense that if we compute the relevant means and variances of a set of discrete data (x_i, y_i) and substitute into Equation 31, we get the same line that is obtained by using OLS.

Kalman Filters for Linear Systems

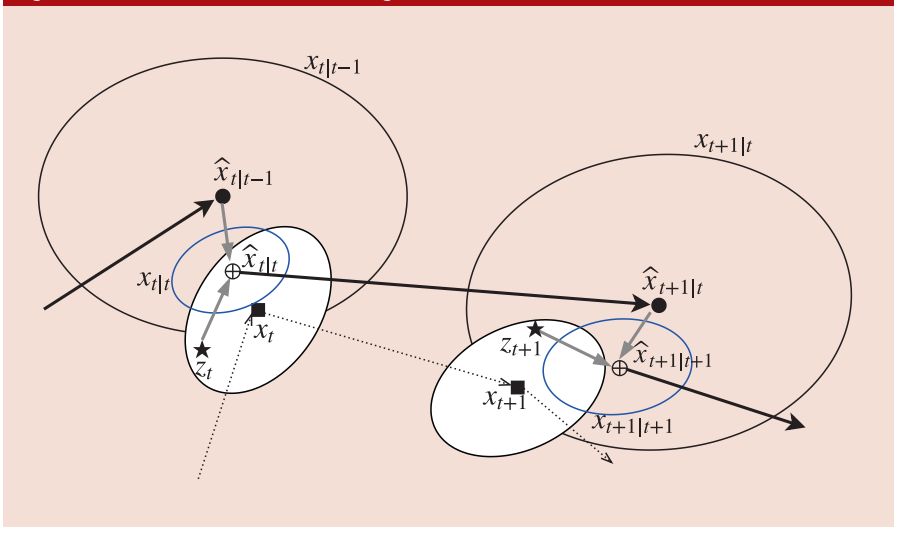
We now apply the algorithms for optimal fusion of vector estimates (Figure 4) and the BLUE estimator (Theorem 4) to the particular problem of state estimation in linear systems, which is the classical application of Kalman filtering.

Figure 6a shows how the evolution of the state of such a system over time can be computed if the initial state $\mathbf{x}_0$ and the model of the system dynamics are known precisely. Time advances in discrete steps. The state of the system at any time step is a function of the state of the system at the previous time step and the control inputs applied to the system during that interval. This is usually expressed by an equation of the form $\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{u}_t)$ where $\mathbf{u}_t$ is the control input. The function f_t is nonlinear in the general case, and can be different for different steps. If the system is linear, the relation for state evolution over time can be written as $\mathbf{x}_t = F_t\mathbf{x}_{t-1} + B_t\mathbf{u}_t$, where F_t and B_t are time-dependent matrices that can be determined from the physics of the system. Therefore, if the initial state $\mathbf{x}_0$ is known exactly and the system dynamics are modeled perfectly by the F_t and B_t matrices, the evolution of the state over time can be computed precisely as shown in Figure 6a.

In general, however, we may not know the initial state exactly, and the system dynamics and control inputs may not be known precisely. These inaccuracies may cause the state computed by the model to diverge unacceptably from the actual state over time. To avoid this, we can make measurements of the state after each time step. If these measurements were exact, there would of course be no need to model the system dynamics. However, in general, the measurements themselves are imprecise.

Kalman filtering was invented to solve the problem of state estimation in such systems. Figure 6b shows the dataflow of the computation, and we use it to introduce standard terminology. An estimate of the initial state, denoted by $\hat{\mathbf{x}}_{0|0}$, is assumed to be available. At each time step t=1, 2, .., the system model is used to provide an estimate of the state at time t using information from time t–1. This step is called *prediction* and the estimate that it provides is called the

Figure 7. Illustration of Kalman filtering.



a priori estimate and denoted by $\hat{\mathbf{x}}_{t|t-1}$. The *a priori* estimate is then fused with $\mathbf{z}_t$, the state estimate obtained from the measurement at time t, and the result is the a posteriori state estimate at time t, denoted by $\hat{\mathbf{x}}_{t|t}$. This *a posteriori* estimate is used by the model to produce the *a priori* estimate for the next time step and so on. As described here, the *a priori* and *a posteriori* estimates are the means of certain random variables; the covariance matrices of these random variables are shown within parentheses each estimate in Figure 6b, and these are used to weight estimates when fusing them.

We first present the state evolution model and *a priori* state estimation. Then we discuss how state estimates are fused if an estimate of the entire state can be obtained by measurement. Finally, we discuss how to address this problem when only a portion of the state can be measured directly.

State evolution model and prediction. The evolution of the state over time is described by a series of random variables $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$

- The random variable $\mathbf{x}_0$ captures the likelihood of different initial states.
- The random variables at successive time steps are related by the following linear model:

$$\mathbf{x}_t = F_t\mathbf{x}_{t-1} + B_t\,\mathbf{u}_t + \mathbf{w}_t \qquad (32)$$

Here, $\mathbf{u}_t$ is the control input, which is assumed to be deterministic, and $\mathbf{w}_t$ is a zero-mean noise term that models all the uncertainty in the system. The covariance matrix of $\mathbf{w}_t$ is denoted by Q_t, and the noise terms in different time steps are assumed to be uncorrelated to each other (such as, $E[\mathbf{w}_i\mathbf{w}_j]=0$ if $i\neq j$) and to $\mathbf{x}_0$.

For estimation, we have a random variable $\mathbf{x}_{0|0}$ that captures our belief about the likelihood of different states at time t=0, and two random variables $\mathbf{x}_{t|t-1}$ and $\mathbf{x}_{t|t}$ at each time step t = 1, 2, ... that capture our beliefs about the likelihood of different states at time t before and after fusion with the measurement, respectively. The mean and covariance matrix of a random variable $\mathbf{x}_{i|j}$ are denoted by $\hat{\mathbf{x}}_{i|j}$ and $\sum_{i|j}$, respectively. We assume $E[\hat{\mathbf{x}}_{0|0}] = E[\mathbf{x}_0]$ (no bias).

Prediction essentially uses $\mathbf{x}_{t-1|t-1}$ as a proxy for $\mathbf{x}_{t-1}$ in Equation 32 to determine $\mathbf{x}_{t|t-1}$ as shown in Equation 33.

Figure 8. Computation of a posteriori estimate.

(i) The *a priori* estimate of the observable part of the state is $H_t\widehat{\mathbf{x}}_{t|t-1}$ and its covariance is $H_t\Sigma_{t|t-1}H_t^{\mathrm{T}}$. Using Equation 21 to fuse it with the measurement gives the *a posteriori* estimate of the observable part of the state: $H_t\widehat{\mathbf{x}}_{t|t} = H_t\widehat{\mathbf{x}}_{t|t-1} + H_t\Sigma_{t|t-1}H_t^{\mathrm{T}}(H_t\Sigma_{t|t-1}H_t^{\mathrm{T}} + R_t)^{-1}(\mathbf{z}_t - H_t\widehat{\mathbf{x}}_{t|t-1})$. Let

$$K_t{=}\Sigma_{t|t-1}H_t^{\mathrm{T}}(H_t\Sigma_{t|t-1}H_t^{\mathrm{T}} + R_t)^{-1}.$$

The *a posteriori* estimate of the observable state can be written in terms of K_t as follows:

$$H_t\widehat{\mathbf{x}}_{t|t} = H_t\widehat{\mathbf{x}}_{t|t-1} + H_tK_t(\mathbf{z}_t - H_t\widehat{\mathbf{x}}_{t|t-1}) \qquad (36)$$

(ii) The *a priori* estimate of the hidden state is $C_t\widehat{\mathbf{x}}_{t|t-1}$. The covariance between the hidden portion $C_t\mathbf{x}_{t|t-1}$ and the observable portion $H_t\mathbf{x}_{t|t-1}$ is $C_t\Sigma_{t|t-1}H_t^{\mathrm{T}}$. The difference between the *a priori* estimate and *a posteriori* estimate of $H_t\mathbf{x}$ is $H_tK_t(\mathbf{z}_t{-}H\widehat{\mathbf{x}}_{t|t-1})$. Therefore the *a posteriori* estimate of the hidden portion of the state is obtained directly from Equation 31:

$$\begin{aligned} C_t\widehat{\mathbf{x}}_{t|t} &= C_t\widehat{\mathbf{x}}_{t|t-1} + (C_t\Sigma_{t|t-1}H_t^{\mathrm{T}})(H_t\Sigma_{t|t-1}H_t^{\mathrm{T}})^{-1}H_tK_t(\mathbf{z}_t - H_t\widehat{\mathbf{x}}_{t|t-1}) \\ &= C_t\widehat{\mathbf{x}}_{t|t-1} + C_tK_t(\mathbf{z}_t - H_t\widehat{\mathbf{x}}_{t|t-1}) \end{aligned} \qquad (37)$$

(iii) Putting the *a posteriori* estimates (36) and (37) together,

$$\begin{pmatrix} H_t \\ C_t \end{pmatrix}\widehat{\mathbf{x}}_{t|t} = \begin{pmatrix} H_t \\ C_t \end{pmatrix}\widehat{\mathbf{x}}_{t|t-1} + \begin{pmatrix} H_t \\ C_t \end{pmatrix}K_t(\mathbf{z}_t - H_t\widehat{\mathbf{x}}_{t|t-1}) \qquad (38)$$

Since $\begin{pmatrix} H_t \\ C_t \end{pmatrix}$ is invertible, it can be canceled from the left and right hand sides, giving the equation

$$\widehat{\mathbf{x}}_{t|t} = \widehat{\mathbf{x}}_{t|t-1} + K_t(\mathbf{z}_t - H_t\widehat{\mathbf{x}}_{t|t-1}) \qquad (39)$$

(iv) To compute $\Sigma_{t|t}$, Equation 39 can be rewritten as $\widehat{\mathbf{x}}_{t|t} = (I - K_tH_t)\widehat{\mathbf{x}}_{t|t-1} + K_t\mathbf{z}_t$. Since $\mathbf{x}_{t|t-1}$ and $\mathbf{z}_t$ are uncorrelated, it follows from Lemma 2 that

$$\Sigma_{t|t} = (I - K_tH_t)\Sigma_{t|t-1}(I - K_tH_t)^{\mathrm{T}} + K_tR_tK_t^{\mathrm{T}} \qquad (40)$$

Substituting the value of K_t and simplifying, we get

$$\Sigma_{t|t} = (I - K_tH_t)\Sigma_{t|t-1} \qquad (41)$$

$$\mathbf{x}_{t|t-1} = F_t\mathbf{x}_{t-1|t-1} + B_t\mathbf{u}_t + \mathbf{w}_t \quad (33)$$

For state estimation, we need only the mean and covariance matrix of $\mathbf{x}_{t|t-1}$. The predictor box in Figure 6 computes these values; the covariance matrix is obtained from Lemma 2 under the assumption that $\mathbf{w}_t$ is uncorrelated with $\mathbf{x}_{t-1|t-1}$, which is justified here.

Fusing complete observations of the state. If the entire state can be measured at each time step, the imprecise measurement at time t is modeled as follows:

$$\mathbf{z}_t = \mathbf{x}_t + \mathbf{v}_t \quad (34)$$

where $\mathbf{v}_t$ is a zero-mean noise term with covariance matrix R_t. The noise terms in different time steps are assumed to be uncorrelated with each other (such as, $E[\mathbf{v}_i\mathbf{v}_j]$ is zero if $i \neq j$) as well as with $\mathbf{x}_{0|0}$ and all $\mathbf{w}_k$. A subtle point here is that $\mathbf{x}_t$ in this equation is the actual state of the system at time t (that is, a particular realization of the random variable $\mathbf{x}_t$), so variability in $\mathbf{z}_t$ comes only from $\mathbf{v}_t$ and its covariance matrix R_t.

Therefore, we have two imprecise estimates for the state at each time step $t = 1, 2, \ldots$, the *a priori* estimate from the predictor ($\hat{\mathbf{x}}_{t|t-1}$) and the one from the measurement ($\mathbf{z}_t$). If $\mathbf{z}_t$ is uncorrelated to $\mathbf{x}_{t|t-1}$, we can use Equations 20–22 to fuse the estimates as shown in Figure 6c.

The assumptions that (i) $\mathbf{x}_{t-1|t-1}$ is uncorrelated with $\mathbf{w}_t$, which is used in prediction, and (ii) $\mathbf{x}_{t|t-1}$ is uncorrelated with $\mathbf{z}_t$, which is used in fusion, are easily proved to be correct by induction on t, using Lemma 2(ii). Figure 6b gives the intuition: $\mathbf{x}_{t|t-1}$ for example is an affine function of the random variables $\mathbf{x}_{0|0}, \mathbf{w}_1, \mathbf{v}_1, \mathbf{w}_2, \mathbf{v}_2, \ldots, \mathbf{w}_t$, and is therefore uncorrelated with $\mathbf{v}_t$ (by assumption about $\mathbf{v}_t$ and Lemma 2(ii)) and hence with $\mathbf{z}_t$.

Figure 7 shows the computation pictorially using confidence ellipses to illustrate uncertainty. The dotted arrows at the bottom of the figure show the evolution of the state, and the solid arrows show the computation of the *a priori* estimates and their fusion with measurements.

Fusing partial observations of the state. In some problems, only a portion of the state can be measured directly. The observable portion of the state is specified formally using a full row-rank matrix H_t called the *observation matrix*: if the state is $\mathbf{x}$, what is observable is $H_t\mathbf{x}$. For example, if the state vector has two components and only the first component is observable, H_t can be [1 0]. In general, the H_t matrix can specify a linear relationship between the state and the observation, and it can be time-dependent. The imprecise measurement model introduced in Equation 34 becomes:

$$\mathbf{z}_t = H_t\mathbf{x}_t + \mathbf{v}_t \quad (35)$$

The hidden portion of the state can be specified using a matrix C_t, which is an orthogonal complement of H_t. For example, if H_t = [1 0], one choice for C_t is [0 1].

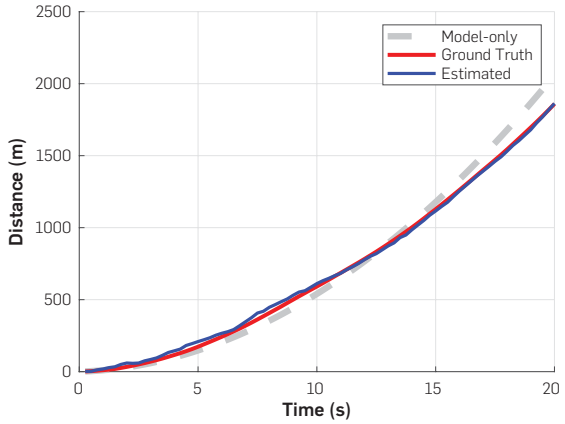
Figure 6d shows the computation for this case. The fusion phase can be understood intuitively in terms of the following steps.

i. The observable part of the *a priori* estimate of the state ($H_t\hat{\mathbf{x}}_{t|t-1}$) is fused with the measurement ($\mathbf{z}_t$), using Equations 20–22. The quantity ($\mathbf{z}_t - H_t\hat{\mathbf{x}}_{t|t-1}$) is called the *innovation*. The result is the *a posteriori* estimate of the observable state ($H_t\hat{\mathbf{x}}_{t|t}$).
ii. The BLUE of Theorem 4 is used to obtain the *a posteriori* estimate of the hidden state ($C_t\hat{\mathbf{x}}_{t|t}$) by adding to the *a priori* estimate of the hidden state ($C_t\hat{\mathbf{x}}_{t|t-1}$) a value obtained from the product of the covariance between $H_t\mathbf{x}_{t|t-1}$ and $C_t\mathbf{x}_{t|t-1}$ and the difference between $H_t\hat{\mathbf{x}}_{t|t-1}$ and $H_t\hat{\mathbf{x}}_{t|t}$.
iii. The *a posteriori* estimates of the observable and hidden portions of the state are composed to produce the *a posteriori* estimate of the entire state ($\hat{\mathbf{x}}_{t|t-1}$).

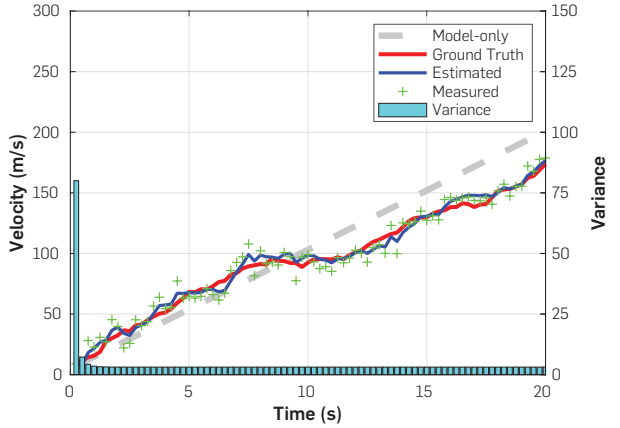
The actual implementation produces the final result directly without going through these steps as shown in Figure 6d, but these incremental steps are useful for understanding how all this works, and their implementation is shown in more detail in Figure 8.

Figure 6d puts all this together. In the literature, this dataflow is referred to as Kalman filtering. Unlike in Equations 18 and 21, the

Figure 9. Estimates of the object's state over time.



(a) Evolution of state: Distance



(b) Evolution of state: Velocity

Kalman gain is not a dimensionless value here. If $H_t = I$, the computations in Figure 6d reduce to those of Figure 6c as expected.

Equation 39 shows that the *a posteriori* state estimate is a linear combination of the *a priori* state estimate ($\hat{\mathbf{x}}_{t|t-1}$) and the measurement ($\mathbf{z}_t$). The optimality of this linear unbiased estimator is shown in the Appendix. It was shown earlier that incremental fusion of scalar estimates is optimal. The dataflow of Figures 6(c,d) computes the *a posteriori* state estimate at time t by incrementally fusing measurements from the previous time steps, and this incremental fusion can be shown to be optimal using a similar argument.

Example: falling body. To demonstrate the effectiveness of the Kalman filter, we consider an example in which an object falls from the origin at time t=0 with an initial speed of 0 m/s and an expected constant acceleration of 9.8 m/s^2 due to gravity. Note that acceleration in reality may not be constant due to factors such as wind, and air friction.

The state vector of the object contains two components, one for the distance from the origin $s(t)$ and one for the velocity $v(t)$. We assume that only the velocity state can be measured at each time step. If time is discretized in steps of 0.25 seconds, the difference equation for the dynamics of the system is easily shown to be the following:

$$\begin{pmatrix} v_n \\ s_n \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0.25 & 1 \end{pmatrix} \begin{pmatrix} v_{n-1} \\ s_{n-1} \end{pmatrix} + \begin{pmatrix} 0 & 0.25 \\ 0 & 0.5 \times 0.25^2 \end{pmatrix} \begin{pmatrix} 0 \\ 9.8 \end{pmatrix}$$

where we assume $\begin{pmatrix} v_0 \\ s_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\Sigma_0 = \begin{pmatrix} 80 & 0 \\ 0 & 10 \end{pmatrix}$.

The gray lines in Figure 9 show the evolution of velocity and distance with time according to this model. Because of uncertainty in modeling the system dynamics, the actual evolution of the velocity and position will be different in practice. The red lines in Figure 9 show one trajectory for this evolution, corresponding to a Gaussian noise term with covariance $\begin{pmatrix} 2 & 2.5 \\ 2.5 & 4 \end{pmatrix}$ in Equation 32 (because this noise term is random, there are many trajectories for the evolution, and we are just showing one of them). The red lines correspond to "ground truth" in our example.

The green points in Figure 9b show the noisy measurements of velocity at different time steps, assuming the noise is modeled by a Gaussian with variance 8. The blue lines show the *a posteriori* estimates of the velocity and position. It can be seen that the *a posteriori* estimates track the ground truth quite well even when the ideal system model (the gray lines) is inaccurate and the measurements are noisy. The cyan bars in the right figure show the variance of the velocity at different time steps. Although the initial variance is quite large, application of Kalman filtering is able to reduce it rapidly in few time steps.

We have shown that Kalman filtering for state estimation in linear systems can be derived from two elementary ideas: optimal linear estimators for fusing uncorrelated estimates and best linear unbiased estimators for correlated variables.

Discussion. We have shown that Kalman filtering for state estimation in linear systems can be derived from two elementary ideas: optimal linear estimators for fusing uncorrelated estimates and best linear unbiased estimators for correlated variables. This is a different approach to the subject than the standard presentations in the literature. One standard approach is to use Bayesian inference. The other approach is to assume that the *a posteriori* state estimator is a linear combination of the form $A_t\hat{\mathbf{x}}_{t|t-1} + B_t\mathbf{z}_t$, and then find the values of A_t and B_t that produce an unbiased estimator with minimum *MSE*. We believe that the advantage of the presentation given here is that it exposes the concepts and assumptions that underlie Kalman filtering.

Most presentations in the literature also begin by assuming that the noise terms $\mathbf{w}_t$ in the state evolution equation and $\mathbf{v}_t$ in the measurement are Gaussian. Although some presentations[1,10] use properties of Gaussians to derive the results in Figure 3, these results do not depend on distributions being Gaussians. Gaussians however enter the picture in a deeper way if one considers *nonlinear* estimators. It can be shown that if the noise terms are not Gaussian, there may be nonlinear estimators whose *MSE* is lower than that of the linear estimator presented in Figure 6d. However, if the noise is Gaussian, this linear estimator is as good as any unbiased nonlinear estimator (that is, the linear estimator is a *minimum variance unbiased estimator*

(MVUE)). This result is proved using the Cramer-Rao lower bound.[24]

Extension to Nonlinear Systems

The *extended Kalman filter* (EKF) and *unscented Kalman filter* (UKF) are heuristic approaches to using Kalman filtering for nonlinear systems. The state evolution and measurement equations for nonlinear systems with additive noise can be written as follows; in these equations, f and h are nonlinear functions.

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t \quad (42)$$

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{v}_t \quad (43)$$

Intuitively, the EKF constructs linear approximations to the nonlinear functions f and h and applies the Kalman filter equations, whereas the UKF constructs approximations to probability distributions and propagates these through the nonlinear functions to construct approximations to the posterior distributions.

EKF. Examining Figure 6d, we see that the *a priori* state estimate in the predictor can be computed using the system model: $\hat{\mathbf{x}}_{t|t-1} = f(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t)$. However, as the system dynamics and measurement equations are nonlinear, it is not clear how to compute the co-variance matrices for the *a priori* estimate and the measurement. In the EKF, these matrices are computed by linearizing Equations 42 and 43 using the Taylor series expansions for the nonlinear functions f and h. This requires computing the following *Jacobians*,[e] which play the role of F_t and H_t in Figure 6d.

$$F_t = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t} \qquad H_t = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{t|t-1}}$$

The EKF performs well in some applications such as navigation systems and GPS.[28]

UKF. When the system dynamics and observation models are highly nonlinear, the unscented Kalman filter (UKF)[15] can be an improvement over the EKF. The UKF is based on the *unscented transformation*, which is a method for computing the statistics of a random variable $\mathbf{x}$ that undergoes a nonlinear transformation ($\mathbf{y} = g(\mathbf{x})$). The random variable $\mathbf{x}$ is sampled using a carefully chosen set of *sigma points* and these sample points are propagated through the nonlinear function g. The statistics of $\mathbf{y}$ are estimated using a weighted sample mean and covariance of the posterior sigma points. The UKF tends to be more robust and accurate than the EKF but has higher computation overhead due to the sampling process.

e The Jacobian matrix is the matrix of all first order partial derivatives of a vector-valued function.

Conclusion

In this article, we have shown that two concepts—optimal linear estimators for fusing uncorrelated estimates and best linear unbiased estimators for correlated variables—provide the underpinnings for Kalman filtering. By combining these ideas, standard results on Kalman filtering for linear systems can be derived in an intuitive and straightforward way that is simpler than other presentations of this material in the literature. This approach makes clear the assumptions that underlie the optimality results associated with Kalman filtering and should make it easier to apply Kalman filtering to problems in computer systems.

Acknowledgments

This research was supported by NSF grants 1337281, 1406355, and 1618425, and by DARPA contracts FA8750-16-2-0004 and FA8650-15-C-7563. We are indebted to K. Mani Chandy (Caltech), Ivo Babuska (UT Austin,) and Augusto Ferrante (Padova, Italy) for their valuable feedback.

References

1. Babb, T. How a Kalman filter works, in pictures | bzarg. 2018. https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/. Accessed: 2018-11-30
2. Balakrishnan, A.V. *Kalman Filtering Theory*. Optimization Software, Inc., Los Angeles, CA, USA, 1987.
3. Barker, A.L., Brown, D.E., Martin, W.N. *Bayesian Estimation and the Kalman Filter*. Technical Report. Charlottesville, VA, USA, 1994.
4. Becker, A. Kalman filter overview. https://www.kalmanfilter.net/default.aspx. 2018. Accessed: 2018-11-08.
5. Bergman, K. Nanophotonic interconnection networks in multicore embedded computing. In *2009 IEEE/LEOS Winter Topicals Meeting Series* (2009), 6–7.
6. Cao, L., Schwartz, H.M. Analysis of the Kalman filter based estimation algorithm: An orthogonal decomposition approach. *Automatica 1*, 40 (2004), 5–19.
7. Chui, C.K., Chen, G. *Kalman Filtering: With Real-Time Applications*, 5th edn. Springer Publishing Company, Incorporated, 2017.
8. Eubank, R.L. *A Kalman Filter Primer (Statistics: Textbooks and Monographs)*. Chapman & Hall/CRC, 2005.
9. Evensen, G. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
10. Faragher, R. Understanding the basis of the Kalman filter via a simple and intuitive derivation. *IEEE Signal Process. Mag. 5*, 29 (2012), 128–132.
11. Grewal, M.S., Andrews, A.P. *Kalman Filtering: Theory and Practice with MATLAB*, 4th edn. Wiley-IEEE Press, 2014.
12. Hess, A.-K., Rantzer, A. Distributed Kalman filter algorithms for self-localization of mobile devices. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '10, 2010, 191–200.
13. Imes, C., Kim, D.H.K., Maggio, M., Hoffmann, H. POET: A portable approach to minimizing energy under soft real-time constraints. In *21st IEEE Real-Time and Embedded Technology and Applications Symposium*, April 2015, 75–86.
14. Imes, C., Hoffmann, H. Bard: A unified framework for managing soft timing and power constraints. In *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)*, 2016.
15. Julier, S.J., Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE 3*, 92 (2004), 401–422.
16. Kitanidis, P.K. Unbiased minimum-variance linear state estimation. *Automatica 6*, 23 (1987), 775–778.
17. Lindquist, A., Picci, G. *Linear Stochastic Systems*. Springer-Verlag, 2017.
18. Maybeck, P.S. *Stochastic Models, Estimation, and Control*, volume 3. Academic Press, 1982.
19. Mendel, J.M. *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Pearson Education, 1995.
20. Nagarajan, K., Gans, N., Jafari, R. Modeling human gait using a Kalman filter to measure walking distance. In *Proceedings of the 2nd Conference on Wireless Health*, WH '11 (New York, NY, USA, 2011). ACM, 34:1–34:2.
21. Nakamura, E.F., Loureiro, A.A.F., Frery, A.C. Information fusion for wireless sensor networks: methods, models, and classifications. *ACM Comput. Surv. 3*, 39 (2007).
22. Petersen, K.B., Pedersen, M.S. The Matrix Cookbook. http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274. November 2012. Version 20121115.
23. Pothukuchi, R.P., Ansari, A., Voulgaris, P., Torrellas, J. Using multiple input, multiple output formal control to maximize resource efficiency in architectures. In *Proceedings of the 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* (2016), IEEE, 658–670.
24. Rao, C.R. Information and the accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37 (1945), 81–89.
25. Rhudy, M.B., Salguero, R.A., Holappa, K. A Kalman filtering tutorial for undergraduate students. *Int. J. Comp. Sci. Eng. Surv.* (1), 8 (2017).
26. Sengupta, S.K. Fundamentals of statistical signal processing: Estimation theory. *Technometrics* (4), 37 (1995), 465–466.
27. Souza, É.L., Nakamura, E.F., Pazzi, R.W. Target tracking for sensor networks: A survey. *ACM Comput. Surv.* (2), 49 (2016), 30:1–30:31.
28. Thrun, S., Burgard, W., Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
29. Welch, G., Bishop, G. *An Introduction to the Kalman Filter*. Technical Report. Chapel Hill, NC, USA, 1995.
30. Pei, Y., Biswas, S., Fussell, D.S., Pingali, K. An Elementary Introduction to Kalman Filtering. *ArXiv e-prints*. October 2017.

The online appendix for this article can be found at http://dl.acm.org/citation.cfm?doid=3363294&picked=formats

Yan Pei (ypei@cs.utexas.edu) is a graduate research assistant in the Department of Computer Science at the University of Texas, Austin, TX, USA.

Swarnendu Biswas (swarnendu@cse.iitk.ac.in) is an assistant professor in the Department of Computer Science and Engineering at the Indian Institute of Technology, Kanpur, India.

Donald S. Fussell (fussell@cs.utexas.edu) is the Trammell Crow Regents Professor in the Department of Computer Science at the University of Texas, Austin, TX, USA.

Keshav Pingali (pingali@cs.utexas.edu) is a professor in the Department of Computer Science at the University of Texas, Austin, and the W.A."Tex" Moncrief Chair of Computing in the UT Oden Institute of Computational Engineering and Science, Austin, TX, USA.