An Elementary Introduction to Kalman Filtering

Yan Pei
University of Texas at Austin
ypei@cs.utexas.edu

Swarnendu Biswas
University of Texas at Austin
sbiswas@ices.utexas.edu

Donald S. Fussell
University of Texas at Austin
fussell@cs.utexas.edu

Keshav Pingali
University of Texas at Austin
pingali@cs.utexas.edu

ABSTRACT

Kalman filtering is a classic state estimation technique used widely in engineering applications such as statistical signal processing and control of vehicles. It is now being used to solve problems in computer systems, such as controlling the voltage and frequency of processors to minimize energy while meeting throughput requirements.

Although there are many presentations of Kalman filtering in the literature, they are usually focused on particular problem domains such as linear systems with Gaussian noise or robot navigation, which makes it difficult to understand the general principles behind Kalman filtering. In this paper, we first present the general statistical ideas behind Kalman filtering at a level accessible to anyone with a basic knowledge of probability theory and calculus, and then show how these abstract concepts can be applied to state estimation problems in linear systems. This separation of abstract concepts from applications should make it easier to apply Kalman filtering to other problems in computer systems.

KEYWORDS

Kalman filtering, data fusion, uncertainty, noise, state estimation, covariance, BLUE estimators, linear systems

1 INTRODUCTION

Kalman filtering is a state estimation technique invented in 1960 by Rudolf E. Kálmán [14]. It is used in many areas including spacecraft navigation, motion planning in robotics, signal processing, and wireless sensor networks [11, 17, 21–23] because of its small computational and memory requirements, and its ability to extract useful information from noisy data. Recent work shows how Kalman filtering can be used in controllers for computer systems [4, 12, 13, 19].

Although many presentations of Kalman filtering exist in the literature [1–3, 5–10, 16, 18, 23], they are usually focused on particular applications like robot motion or state estimation in linear systems with Gaussian noise. This can make it difficult to see how to apply Kalman filtering to other problems. The goal of this paper is to present the abstract statistical ideas behind Kalman filtering independently of particular applications, and then show how these ideas can be applied to solve particular problems such as state estimation in linear systems.

Abstractly, Kalman filtering can be viewed as an algorithm for combining imprecise estimates of some unknown value to obtain a more precise estimate of that value. We use informal methods similar to Kalman filtering in everyday life. When we want to buy a house for example, we may ask a couple of real estate agents to give us independent estimates of what the house is worth. For now, we use the word "independent" informally to mean that the two agents are not allowed to consult with each other. If the two estimates are different, as is likely, how do we combine them into a single value to make an offer on the house? This is an example of a *data fusion problem.*

One solution to our real-estate problem is to take the average of the two estimates; if these estimates are x_1 and x_2, they are combined using the formula $0.5{*}x_1 + 0.5{*}x_2$. This gives equal weight to the estimates. Suppose however we have additional information about the two agents; perhaps the first one is a novice while the second one has a lot of experience in real estate. In that case, we may have more confidence in the second estimate, so we may give it more weight by using a formula such as $0.25{*}x_1 + 0.75{*}x_2$. In general, we can consider a *convex combination* of the two estimates, which is a formula of the form $(1-\alpha){*}x_1 + \alpha{*}x_2$, where $0 \leq \alpha \leq 1$; intuitively, the more confidence we have in the second estimate, the closer α should be to 1. In the extreme case, when α=1, we discard the first estimate and use only the second estimate.

The expression $(1-\alpha){*}x_1 + \alpha{*}x_2$ is an example of a *linear estimator* [15]. The statistics behind Kalman filtering tell us how to pick the optimal value of α: the weight given to an estimate should be proportional to the confidence we have in that estimate, which is intuitively reasonable.

To quantify these ideas, we need to formalize the concepts of *estimates* and *confidence* in estimates. Section 2 describes the model used in Kalman filtering. Estimates are modeled as random samples from certain *distributions*, and confidence in estimates is quantified in terms of the *variances* and *covariances* of these distributions.

Sections 3-5 develop the two key statistical ideas behind Kalman filtering.

(1) How should uncertain estimates be fused optimally?
Section 3 shows how to fuse *scalar* estimates such as house prices optimally. It is also shown that the problem of fusing more than two estimates can be reduced to the problem of fusing two estimates at a time, without any loss in the quality of the final estimate.
Section 4 extends these results to estimates that are *vectors*, such as state vectors representing the estimated position and velocity of a robot or spacecraft. The math is more complicated than in the scalar case but the basic ideas remain the same, except that instead of working with variances of scalar estimates, we must work with covariance matrices of vector estimates.

(2) In some applications, estimates are vectors but only a part of the vector may be directly observable. For example, the state of a spacecraft may be represented by its position and velocity, but only its position may be directly observable. In such cases, how do we obtain a complete estimate from a partial estimate?
Section 5 introduces the *Best Linear Unbiased Estimator (BLUE)*, which is used in Kalman filtering for this purpose. It can be seen as a generalization of the ordinary least squares (OLS) method to problems in which data comes from distributions rather than being a set of discrete points.

Section 6 shows how these results can be used to solve state estimation problems for linear systems, which is the usual context for presenting Kalman filters. First, we consider the problem of state estimation when the entire state is observable, which can be solved using the data fusion results from Sections 3 and 4. Then we consider the more complex problem of state estimation when the state is only partially observable, which requires in addition the BLUE estimator from Section 5. Section 6.3 illustrates Kalman filtering with a concrete example.

2 FORMALIZATION OF ESTIMATES

This section makes precise the notions of *estimates* and *confidence* in estimates, which were introduced informally in Section 1.

2.1 Scalar estimates

One way to model the behavior of an agent producing scalar estimates such as house prices is through a *probability distribution function* (usually shortened to *distribution*) like the ones shown in Figure 1 in which the x-axis represents values that can be assigned to the house, and the y-axis represents the probability density. Each agent has its own distribution, and obtaining an estimate from an agent corresponds to drawing a random sample x_i from the distribution for agent i.

Most presentations of Kalman filters assume distributions are Gaussian but we assume that we know only the mean μ_i and the variance σ_i^2 of each distribution p_i. We write $x_i : p_i{\sim}(\mu_i, \sigma_i^2)$ to denote that x_i is a random sample drawn from distribution p_i which has a mean of μ_i and a variance of σ_i^2. The reciprocal of the variance of a distribution is sometimes called the *precision* of that distribution.

The informal notion of "confidence in the estimates made by an agent" is quantified by the variance of the distribution from which the estimates are drawn. An experienced agent making high-confidence estimates is modeled by a distribution with a smaller variance than one used to model an inexperienced agent; notice that in Figure 1, the inexperienced agent is "all over the map."

This approach to modeling confidence in estimates may seem nonintuitive since there is no mention of how close the estimates made by an agent are to the actual value of the house. In particular, an agent can make estimates that are very far off from the actual value of the house but as long as his estimates fall within a narrow range of values, we would still say that we have high confidence in his estimates. In statistics, this is explained by making a distinction between *accuracy* and *precision*. Accuracy is a measure of how close an estimate of a quantity is to the true value of that quantity (the true value is sometimes called the *ground truth*). Precision on the other hand is a measure of how close the estimates are to each other, and is defined without reference to ground truth. A metaphor that is often used to explain this difference is shooting at a bullseye. In this case, ground truth is provided by the center of the bullseye. A precise shooter is one whose shots are clustered closely together even if they may be far from the bullseye. In contrast, the shots of an accurate but not precise shooter would be scattered widely in a region surrounding the bullseye. For the problems considered in this paper, there may be no ground truth, and confidence in estimates is related to precision, not accuracy.

The informal notion of getting *independent* estimates from the two agents is modeled by requiring that estimates x_1 and x_2 be *uncorrelated*; that is, $E[(x_1 - \mu_1)(x_2 - \mu_2)] = 0$. This is not the same thing as requiring them to be independent random variables, as explained in Appendix 8.1. Lemma 2.1 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.

Lemma 2.1. *Let $x_1{:}p_1{\sim}(\mu_1, \sigma_1^2), ..., x_n{:}p_n{\sim}(\mu_n, \sigma_n^2)$ 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.*

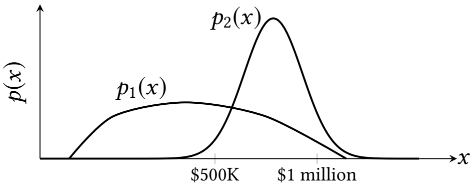


Figure 1: Distributions.

The mean and variance of y are the following:

$$\mu_y = \sum_{i=1}^{n} \alpha_i \mu_i \tag{1}$$

$$\sigma_y^2 = \sum_{i=1}^{n} \alpha_i^2 \sigma_i^2 \tag{2}$$

Proof. Equation 1 follows from the fact that expectation is a linear operator:

$$\mu_y = E[y] = E[\Sigma_{i=1}^{n} \alpha_i x_i] = \Sigma_{i=1}^{n} \alpha_i E[x_i] = \Sigma_{i=1}^{n} \alpha_i \mu_i.$$

Equation 2 follows from linearity of the expectation operator and the fact that the estimates are pairwise uncorrelated:

$$\begin{aligned}\sigma_y^2 &= E[(y-\mu_y)^2] = E[\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} \alpha_i\alpha_j (x_i-\mu_i)(x_j-\mu_j)] \\ &= \Sigma_{i=1}^{n}\Sigma_{j=1}^{n} \alpha_i\alpha_j E[(x_i-\mu_i)(x_j-\mu_j)]\end{aligned}$$

Since the variables $x_1, \ldots, x_n$ are pairwise uncorrelated, $E[(x_i-\mu_i)(x_j-\mu_j)] = 0$ if $i \neq j$, from which the result follows. □

2.2 Vector estimates

In some applications, estimates are vectors. For example, the state of a robot moving along a single dimension 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. In this paper, we denote a vector by a boldfaced lowercase letter, and a matrix by an uppercase letter. The covariance matrix of a random variable $\mathbf{x}$:$p(\mathbf{x})$ with mean $\boldsymbol{\mu}_x$ is the matrix $E[(\mathbf{x}-\boldsymbol{\mu}_x)(\mathbf{x}-\boldsymbol{\mu}_x)^{\mathrm{T}}]$.

Estimates: An estimate $\mathbf{x}_i$ is a random sample drawn from a distribution with mean $\boldsymbol{\mu}_i$ and covariance matrix Σ_i, written as $\mathbf{x}_i : p_i{\sim}(\boldsymbol{\mu}_i, \Sigma_i)$. The inverse of the covariance matrix Σ_i^{-1} is called the precision or information matrix. Note that if the dimension of $\mathbf{x}_i$ is one, the covariance matrix reduces to variance.

Uncorrelated estimates: Estimates $\mathbf{x}_i$ and $\mathbf{x}_j$ are uncorrelated if $E[(\mathbf{x}_i-\boldsymbol{\mu}_i)(\mathbf{x}_j-\boldsymbol{\mu}_j)^{\mathrm{T}}] = \mathbf{0}$. This is equivalent to saying that every component of $\mathbf{x}_i$ is uncorrelated with every component of $\mathbf{x}_j$.

Lemma 2.2 generalizes Lemma 2.1.

Lemma 2.2. *Let $\boldsymbol{x}_1$:$p_1{\sim}(\boldsymbol{\mu}_1, \Sigma_1), ..., \boldsymbol{x}_n$:$p_n{\sim}(\boldsymbol{\mu}_n, \Sigma_n)$ be a set of pairwise uncorrelated random vectors of length m. Let $\boldsymbol{y} = \sum_{i=1}^{n} A_i \boldsymbol{x}_i$. Then, the mean and variance of $\boldsymbol{y}$ are the following:*

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

$$\Sigma_y = \sum_{i=1}^{n} A_i \Sigma_i A_i^{\mathrm{T}} \tag{4}$$

Proof. The proof is similar to the proof of Lemma 2.1.

Equation 3 follows from the linearity of the expectation operator.

Equation 4 can be proved as follows:

$$\begin{aligned}\Sigma_y &= E[(\mathbf{y}-\boldsymbol{\mu}_y)(\mathbf{y}-\boldsymbol{\mu}_y)^{\mathrm{T}}] \\ &= E[\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} A_i(\mathbf{x}_i-\boldsymbol{\mu}_i)(\mathbf{x}_j-\boldsymbol{\mu}_j)^{\mathrm{T}} A_j^{\mathrm{T}}] \\ &= \Sigma_{i=1}^{n}\Sigma_{j=1}^{n} A_i E[(\mathbf{x}_i-\boldsymbol{\mu}_i)(\mathbf{x}_j-\boldsymbol{\mu}_j)^{\mathrm{T}}] A_j^{\mathrm{T}}\end{aligned}$$

The variables $\mathbf{x}_1, ..\mathbf{x}_n$ are pairwise uncorrelated, therefore $E[(\mathbf{x}_i-\boldsymbol{\mu}_i)(\mathbf{x}_j-\boldsymbol{\mu}_j)] = 0$ if $(i \neq j)$, from which the result follows. □

3 FUSING SCALAR ESTIMATES

Section 3.1 discusses the problem of fusing two scalar estimates. Section 3.2 generalizes this to the problem of fusing $n{>}2$ scalar estimates. Section 3.3 shows that fusing $n{>}2$ estimates can be done iteratively by fusing two estimates at a time without any loss of quality in the final estimate.

3.1 Fusing two scalar estimates

We now consider the problem of choosing the optimal value of the parameter α in the formula $y_\alpha(x_1, x_2){=}(1{-}\alpha){*}x_1 + \alpha{*}x_2$ for fusing uncorrelated estimates x_1 and x_2. How should optimality be defined? One reasonable definition is that the optimal value of α *minimizes the variance of* $y_\alpha(x_1, x_2)$; since confidence in an estimate is inversely proportional to the variance of the distribution from which the estimates are drawn, this definition of optimality will produce the highest-confidence fused estimates. The variance of $y_\alpha(x_1, x_2)$ is called the *mean square error* (*MSE*) of that estimator, and it obviously depends on α; the minimum value of this variance as α is varied is called the *minimum mean square error error* (*MMSE*) below.

Theorem 3.1. *Let x_1:$p_1(x){\sim}(\mu_1, \sigma_1^2)$ and x_2:$p_2(x){\sim}(\mu_2, \sigma_2^2)$ be uncorrelated estimates, and suppose they are fused using the formula $y_\alpha(x_1, x_2) = (1{-}\alpha){*}x_1 + \alpha{*}x_2$. The value of MSE$(y_\alpha)$ is minimized for $\alpha = \frac{\sigma_1^2}{{\sigma_1}^2 + {\sigma_2}^2}$.*

Proof. From Lemma 2.1,

$$\sigma_y^2(\alpha) = (1-\alpha)^2 {*} {\sigma_1}^2 + \alpha^2 {*} {\sigma_2}^2. \tag{5}$$

Differentiating $\sigma_y^2(\alpha)$ with respect to α and setting the derivative to zero proves the result. In the literature, this optimal value of α is called the *Kalman gain K*. □

Substituting K into the linear fusion model, we get the optimal linear estimator $\widehat{y}(x_1, x_2)$:

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

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

$$\widehat{y}(x_1, x_2) = \frac{\frac{1}{\sigma_1^2}}{\frac{1}{{\sigma_1}^2} + \frac{1}{{\sigma_2}^2}} * x_1 + \frac{\frac{1}{\sigma_2^2}}{\frac{1}{{\sigma_1}^2} + \frac{1}{{\sigma_2}^2}} * x_2 \tag{6}$$

Substituting K into Equation 5 gives the following expression for the variance of $\widehat{y}$:

$$\sigma_{\widehat{y}}^2 = \frac{1}{\frac{1}{{\sigma_1}^2} + \frac{1}{{\sigma_2}^2}} \tag{7}$$

The expressions for $\widehat{y}$ and $\sigma_{\widehat{y}}$ 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 $\widehat{y}$ and $\nu_{\widehat{y}}$ can be written more simply as follows:

$$\widehat{y}(x_1, x_2) = \frac{\nu_1}{\nu_1 + \nu_2} * x_1 + \frac{\nu_2}{\nu_1 + \nu_2} * x_2 \tag{8}$$

$$\nu_{\widehat{y}} = \nu_1 + \nu_2 \tag{9}$$

These results say that the weight we should give to an estimate is proportional to the confidence we have in that estimate, which is intuitively reasonable. Note that if μ_1=μ_2, the expectation $E[y_\alpha]$ is μ_1(=μ_2) regardless of the value α. In this case, y_α is said to be an *unbiased estimator*, and the optimal value of α is the one that minimizes the variance of the unbiased estimator.

3.2 Fusing multiple scalar estimates

The approach in Section 3.1 can be generalized to optimally fuse multiple pairwise uncorrelated estimates $x_1, x_2, ..., x_n$. Let $y_\alpha(x_1, .., x_n)$ denote the linear estimator given parameters $\alpha_1, .., \alpha_n$, which we denote by α.

Theorem 3.2. *Let pairwise uncorrelated estimates $x_i (1 \leq i \leq n)$ drawn from distributions $p_i(x) \sim (\mu_i, \sigma_i^2)$ be fused using the linear model $y_\alpha(x_1, .., x_n) = \sum_{i=1}^n \alpha_i x_i$ where $\sum_{i=1}^n \alpha_i = 1$. The value of MSE$(y_\alpha)$ is minimized for*

$$\alpha_i = \frac{\frac{1}{\sigma_i^2}}{\sum_{i=1}^n \frac{1}{\sigma_i^2}}.$$

Proof. From Lemma 2.1, $\sigma_y^2(\alpha) = \sum_{i=1}^n {\alpha_i}^2 {\sigma_i}^2$. To find the values of α_i that minimize the variance σ_y^2 under the constraint that the α_i's sum to 1, we use the method of Lagrange multipliers. Define

$$f(\alpha_1, ..., \alpha_n) = \sum_{i=1}^n {\alpha_i}^2 {\sigma_i}^2 + \lambda(\sum_{i=1}^n \alpha_i - 1)$$

where λ is the Lagrange multiplier. Taking the partial derivatives of f with respect to each α_i and setting these derivatives to zero, we find $\alpha_1 \sigma_1^2 = \alpha_2 \sigma_2^2 = ... = \alpha_n \sigma_n^2 = -\lambda/2$. From this, and the fact that sum of the α_i's is 1, the result follows. □

The variance is given by the following expression:

$$\sigma^2{}_{\widehat{y}} = \frac{1}{\sum_{i=1}^n \frac{1}{\sigma_i^2}}. \tag{10}$$

As in Section 3.1, these expressions are more intuitive if the variance is replaced with precision.

$$\widehat{y}(x_1, .., x_n) = \sum_{i=1}^n \frac{\nu_i}{\nu_1 + ... + \nu_n} * x_i \tag{11}$$

$$\nu_{\widehat{y}} = \sum_{i=1}^n \nu_i \tag{12}$$

Equations 11 and 12 generalize Equations 8 and 9.

3.3 Incremental fusing is optimal

In many applications, the estimates $x_1, x_2, ..., x_n$ become available successively over a period of time. While it is possible to store all the estimates and use Equations 11 and 12 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. In this section, 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 $\widehat{y}(x_1, x_2, ..., x_n) = \widehat{y}(\widehat{y}(\widehat{y}(x_1, x_2), x_3)..., x_n)$. Note that this is not the same thing as showing $\widehat{y}$, interpreted as a binary function, is associative.

Figure 2 shows the process of incrementally fusing estimates. Imagine that time progresses from left to right in this picture. Estimate x_1 is available initially, and the other estimates x_i become available in succession; the precision of each estimate is shown in parentheses next to each estimate. When the estimate x_2 becomes available, it is fused with x_1 using Equation 8. In Figure 2, the labels on the edges connecting x_1 and x_2 to $\widehat{y}(x_1, x_2)$ are the weights given to these estimates in Equation 8. When estimate x_3 becomes available,

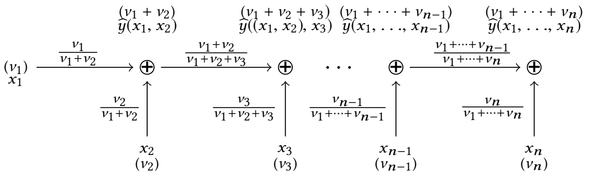

(v_1+v_2)
$\widehat{y}(x_1,x_2)$
$(v_1+v_2+v_3)$
$\widehat{y}(\widehat{y}(x_1,x_2),x_3)$
$(v_1+\cdots+v_{n-1})$
$\widehat{y}(x_1,\ldots,x_{n-1})$
$(v_1+\cdots+v_n)$
$\widehat{y}(x_1,\ldots,x_n)$
(v_1)
x_1
$\frac{v_1}{v_1+v_2}$
$\frac{v_1+v_2}{v_1+v_2+v_3}$
$\frac{v_1+\cdots+v_{n-1}}{v_1+\cdots+v_n}$
$\frac{v_2}{v_1+v_2}$
$\frac{v_3}{v_1+v_2+v_3}$
$\frac{v_{n-1}}{v_1+\cdots+v_{n-1}}$
$\frac{v_n}{v_1+\cdots+v_n}$
x_2
(v_2)
x_3
(v_3)
x_{n-1}
(v_{n-1})
x_n
(v_n)

Figure 2: Dataflow graph for incremental fusion.

it is fused with $\widehat{y}(x_1, x_2)$ using Equation 8; as before, the labels on the edges correspond to the weights given to $\widehat{y}(x_1, x_2)$ and x_3 when they are fused to produce $\widehat{y}(\widehat{y}(x_1, x_2), x_3)$.

The contribution of x_i to the final value $\widehat{y}(\widehat{y}(\widehat{y}(x_0, x_1), x_2)..., x_n)$ is given by the product of the weights on the path from x_i to the final value in Figure 2. As shown below, this product has the same value as the weight of x_i in Equation 11, showing that incremental fusion is optimal.

$$\frac{v_i}{v_1 + \ldots + v_i} * \frac{v_1 + \ldots + v_i}{v_1 + \ldots + v_{i+1}} * \ldots * \frac{v_1 + \ldots + v_{n-1}}{v_1 + \ldots + v_n} = \frac{v_i}{v_1 + \ldots + v_n}$$

3.4 Summary

The main result in this section can be summarized informally as follows. *When using a linear model to fuse uncertain scalar estimates, the weight given to each estimate should be inversely proportional to the variance of that estimate. Furthermore, when fusing n>2 estimates, estimates can be fused incrementally without any loss in the quality of the final result.* More formally, the results in this section for fusing scalar estimates are often expressed in terms of the Kalman gain, as shown below; these equations can be applied recursively to fuse multiple estimates.

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

$$K = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \frac{v_2}{v_1 + v_2} \tag{13}$$

$$\widehat{y}(x_1, x_2) = x_1 + K(x_2 - x_1) \tag{14}$$

$$\mu_{\widehat{y}} = \mu_1 + K(\mu_2 - \mu_1) \tag{15}$$

$$\sigma_{\widehat{y}}^2 = {\sigma_1}^2 - K{\sigma_1}^2 \quad \text{or} \quad v_{\widehat{y}} = v_1 + v_2 \tag{16}$$

4 FUSING VECTOR ESTIMATES

This section addresses the problem of fusing estimates when the estimates are vectors. Although the math is more complicated, the conclusion is that the results in Section 3 for fusing scalar estimates can be extended to the vector case simply by replacing *variances* with *covariance matrices*, as shown in this section.

4.1 Fusing multiple vector estimates

For vectors, the linear data fusion model is

$$\mathbf{y}_A(\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_n) = \sum_{i=1}^{n} A_i \mathbf{x}_i \quad \text{where} \quad \sum_{i=1}^{n} A_i = I. \tag{17}$$

Here A stands for the matrix parameters $(A_1, ..., A_n)$. All the vectors are assumed to be of the same length.

Optimality: The *MSE* in this case is the expected value of the two-norm of $(\mathbf{y}_A - \boldsymbol{\mu}_{y_A})$, which is $E[(\mathbf{y}_A - \boldsymbol{\mu}_{y_A})^{\mathrm{T}}(\mathbf{y}_A - \boldsymbol{\mu}_{y_A})]$. Note that if the vectors have length 1, this reduces to variance. The parameters $A_1, ..., A_n$ in the linear data fusion model are chosen to minimize this *MSE*.

Theorem 4.1 generalizes Theorem 3.2 to the vector case. The proof of this theorem uses matrix derivatives and is given in Appendix 8.3 since it is not needed for understanding the rest of this paper. What is important is to compare Theorems 4.1 and 3.2 and notice 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 4.1. *Let pairwise uncorrelated estimates* $\boldsymbol{x}_i(1{\le}i{\le}n)$ *drawn from distributions* $p_i(x){=}(\boldsymbol{\mu}_i, \Sigma_i)$ *be fused using the linear model* $\boldsymbol{y}_A(\boldsymbol{x}_1, .., \boldsymbol{x}_n) = \sum_{i=1}^{n} A_i \boldsymbol{x}_i$, *where* $\sum_{i=1}^{n} A_i = I$. *The* $MSE(\boldsymbol{y}_A)$ *is minimized for*

$$A_i = (\sum_{j=1}^{n} \Sigma_j^{-1})^{-1} \Sigma_i^{-1}. \tag{18}$$

The covariance matrix of the optimal estimator $\widehat{\mathbf{y}}$ can be determined by substituting the optimal A_i values into the expression for Σ_y in Lemma 2.2.

$$\Sigma_{\widehat{\mathbf{y}}} = (\sum_{j=1}^{n} \Sigma_j^{-1})^{-1} \tag{19}$$

In the vector case, precision is the inverse of a covariance matrix, denoted by N. Equations 20–21 use precision to express the optimal estimator and its variance, and generalize Equations 11–12 to the vector case.

$$\widehat{\mathbf{y}}(\mathbf{x}_1, ..., \mathbf{x}_n) = \sum_{i=1}^{n} (\sum_{j=1}^{n} N_j)^{-1} N_i \mathbf{x}_i \tag{20}$$

$$N_{\widehat{y}} = \sum_{j=1}^{n} N_j \tag{21}$$

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 one in Section 3.3, and is omitted.

There are several equivalent expressions for the Kalman gain for the fusion of two estimates. The following one, which is easily derived from Equation 18, is the vector analog of Equation 13:

$$K = \Sigma_1(\Sigma_1 + \Sigma_2)^{-1} \tag{22}$$

The covariance matrix of $\widehat{\mathbf{y}}$ can be written as follows.

$$\Sigma_{\widehat{\mathbf{y}}} = \Sigma_1(\Sigma_1 + \Sigma_2)^{-1}\Sigma_2 \tag{23}$$

$$= K\Sigma_2 = \Sigma_1 - K\Sigma_1 \tag{24}$$

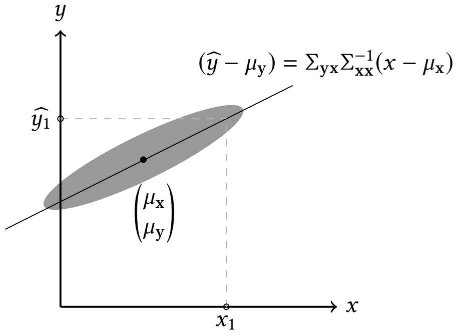
4.2 Summary

The results in this section can be summarized in terms of the Kalman gain K as follows:

$$\boxed{\begin{aligned} &\mathbf{x}_1 : p_1\sim(\boldsymbol{\mu}_1, \Sigma_1), \quad \mathbf{x}_2 : p_2\sim(\boldsymbol{\mu}_2, \Sigma_2) \\ &K = \Sigma_1(\Sigma_1 + \Sigma_2)^{-1} = (N_1 + N_2)^{-1}N_2 && (25) \\ &\widehat{\mathbf{y}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 + K(\mathbf{x}_2 - \mathbf{x}_1) && (26) \\ &\boldsymbol{\mu}_y = \boldsymbol{\mu}_1 + K(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) && (27) \\ &\Sigma_{\widehat{y}} = \Sigma_1 - K\Sigma_1 \quad \text{or} \quad N_{\widehat{y}} = N_1 + N_2 && (28) \end{aligned}}$$

5 BEST LINEAR UNBIASED ESTIMATOR (BLUE)

In some applications, estimates are vectors but only part of the vector may be given to us directly, and it is necessary to estimate the hidden portion. This section introduces a statistical method called the *Best Linear Unbiased Estimator* (BLUE).



y
$(\widehat{y} - \mu_y) = \Sigma_{yx}\Sigma_{xx}^{-1}(x - \mu_x)$
$\widehat{y_1}$
$\begin{pmatrix}\mu_x \\ \mu_y\end{pmatrix}$
x
x_1

Figure 3: BLUE line corresponding to Equation 29

Consider the general problem of determining a value for vector $\mathbf{y}$ given a value for a vector $\mathbf{x}$. If there is a functional relationship between $\mathbf{x}$ and $\mathbf{y}$ (say $\mathbf{y}$=$F(\mathbf{x})$ and F is given), it is easy to compute $\mathbf{y}$ given a value for $\mathbf{x}$. In our context however, $\mathbf{x}$ and $\mathbf{y}$ are random variables so such a precise functional relationship will not hold. The best we can do is to estimate the likely value of $\mathbf{y}$, given a value of $\mathbf{x}$ and the information we have about how $\mathbf{x}$ and $\mathbf{y}$ are correlated.

Figure 3 shows an example in which x and y are scalar-valued random variables. The grey 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. For a given value x_1, there are in general many points (x_1, y) that lie within this ellipse, but these y values are clustered around the line shown in the figure so the value $\widehat{y_1}$ 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. Given a set of discrete points (x_i, y_i) where each x_i and y_i are scalars, OLS determines the "best" linear relationship between these points, where best is defined as minimizing the square error between the predicted and actual values of y_i. This relation can then be used to predict a value for y given a value for x. The BLUE estimator presented below generalizes this to the case when x and y are vectors, and are random variables obtained from distributions instead of a set of discrete points.

5.1 Computing BLUE

Let $\mathbf{x}$:$p_x\sim(\mu_x, \Sigma_{xx})$ and $\mathbf{y}$:$p_y\sim(\mu_y, \Sigma_{yy})$ be random variables. Consider a linear estimator $\widehat{\mathbf{y}}_{A,\mathbf{b}}(\mathbf{x})$=$A\mathbf{x}$+$\mathbf{b}$. How should we choose A and $\mathbf{b}$? As in the OLS approach, we can pick values that minimize the *MSE* between random variable $\mathbf{y}$ and the estimate $\widehat{\mathbf{y}}$.

$$\begin{aligned} MSE_{A,\mathbf{b}}(\mathbf{y}, \widehat{\mathbf{y}}) &= E[(\mathbf{y} - \widehat{\mathbf{y}})^{\mathrm{T}}(\mathbf{y} - \widehat{\mathbf{y}})] \\ &= E[(\mathbf{y} - (A\mathbf{x} + \mathbf{b}))^{\mathrm{T}}(\mathbf{y} - (A\mathbf{x} + \mathbf{b}))] \\ &= E[\mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{y}^{\mathrm{T}}(A\mathbf{x} + \mathbf{b}) + (A\mathbf{x} + \mathbf{b})^{\mathrm{T}}(A\mathbf{x} + \mathbf{b})] \end{aligned}$$

Setting the partial derivatives of $MSE_{A,\mathbf{b}}(\mathbf{y}, \widehat{\mathbf{y}})$ with respect to $\mathbf{b}$ and A to zero, we find that $\mathbf{b} = (\mu_y - A\mu_x)$, and $A = \Sigma_{yx}\Sigma_{xx}^{-1}$, where Σ_{yx} is the covariance between $\mathbf{y}$ and $\mathbf{x}$. Therefore, the best linear estimator is

$$\widehat{\mathbf{y}} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(\mathbf{x} - \mu_x) \tag{29}$$

This is an unbiased estimator because the mean of $\widehat{\mathbf{y}}$ is equal to μ_y. Note that if $\Sigma_{yx} = 0$ (that is, $\mathbf{x}$ and $\mathbf{y}$ are uncorrelated), the best estimate of $\mathbf{y}$ is just μ_y, so knowing the value of $\mathbf{x}$ does not give us any additional information about $\mathbf{y}$ as one would expect. In Figure 3, this corresponds to the case when the BLUE line is parallel to the x-axis. At the other extreme, suppose that $\mathbf{y}$ and $\mathbf{x}$ are functionally related so $\mathbf{y} = C\mathbf{x}$. In that case, it is easy to see that $\Sigma_{yx} = C\Sigma_{xx}$, so $\widehat{y} = C\mathbf{x}$ as expected. In Figure 3, this corresponds to the case when the confidence ellipse shrinks down to the BLUE line.

Substituting the optimal values of A and $\mathbf{b}$ into the expression for *MSE* gives us the minimum *MSE* (MMSE):

$$MMSE(\mathbf{y}, \widehat{\mathbf{y}}) = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy} \tag{30}$$

Intuitively, knowing the value of $\mathbf{x}$ permits us to reduce the uncertainty in the value of $\mathbf{y}$ by an additive term that depends on how strongly $\mathbf{y}$ and $\mathbf{x}$ are correlated.

It is easy to show that Equation 29 is a generalization of ordinary least squares in the sense that if we compute means and variances of a set of discrete data (x_i, y_i) and substitute into Equation 29, we get the same line that is obtained by using OLS.

6 KALMAN FILTERS FOR LINEAR SYSTEMS

In this section, we apply the algorithms developed in Sections 3-5 to the particular problem of estimating the state of *linear systems*, which is the classical application of Kalman filtering.

Figure 4(a) shows how the evolution over time of the state of such a system 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 $\mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t)$ 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 $\mathbf{x}_{t+1} = F_t\mathbf{x}_t + B_t\mathbf{u}_t$, 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 with time can be computed precisely.

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 computed and actual states to diverge unacceptably over time. To avoid this, we can make measurements of the state after each time step. If these measurements were exact and the entire state could be observed after each time step, there would of course be no need to model the system dynamics. However, in general, (i) the measurements themselves are imprecise, and (ii) some components of the state may not be directly observable by measurements.

6.1 Fusing complete observations of the state

If the entire state can be observed through measurements, we have two imprecise estimates for the state after each time step, one from the model of the system dynamics and one from the measurement. If these estimates are uncorrelated and their covariance matrices are known, we can use Equations 25:28 to fuse these estimates and compute the covariance matrix of this fused estimate. The fused estimate of the state is fed into the model of the system to compute the estimate of the state and its covariance at the next time step, and the entire process is repeated.

Figure 4(b) shows the dataflow diagram of this computation. For each state $\mathbf{x}_{t+1}$ in the precise computation of Figure 4(a), there are three random variables in Figure 4(b): the estimate from the model of the dynamical system, denoted by $\mathbf{x}_{t+1|t}$, the estimate from the measurement, denoted by $\mathbf{z}_{t+1}$, and the fused estimate, denoted by $\mathbf{x}_{t+1|t+1}$. Intuitively, the notation $\mathbf{x}_{t+1|t}$ stands for the estimate of the state at time $(t+1)$ *given the information at time* t, and it is often referred to as the *a priori* estimate. Similarly, $\mathbf{x}_{t+1|t+1}$ is the corresponding estimate *given the information available at time* $(t+1)$, which includes information from the measurement, and is often referred to as the *a posteriori* estimate. To set up this computation, we introduce the following notation.

- The initial state is denoted by $\mathbf{x}_0$ and its covariance by $\Sigma_{0|0}$.
- Uncertainty in the system model and control inputs is represented by making $\mathbf{x}_{t+1|t}$ a random variable and introducing a zero-mean noise term $\mathbf{w}_t$ into the state evolution equation, which becomes

$$\mathbf{x}_{t+1|t} = F_t\mathbf{x}_{t|t} + B_t\mathbf{u}_t + \mathbf{w}_t \tag{31}$$

  The covariance matrix of $\mathbf{w}_t$ is denoted by Q_t and $\mathbf{w}_t$ is assumed to be uncorrelated with $\mathbf{x}_{t|t}$.
- The imprecise measurement at time $t+1$ is modeled by a random variable $\mathbf{z}_{t+1} = \mathbf{x}_{t+1} + \mathbf{v}_{t+1}$ where $\mathbf{v}_{t+1}$ is a noise term. $\mathbf{v}_{t+1}$ has a covariance matrix R_{t+1} and is uncorrelated to $\mathbf{x}_{t+1|t}$.

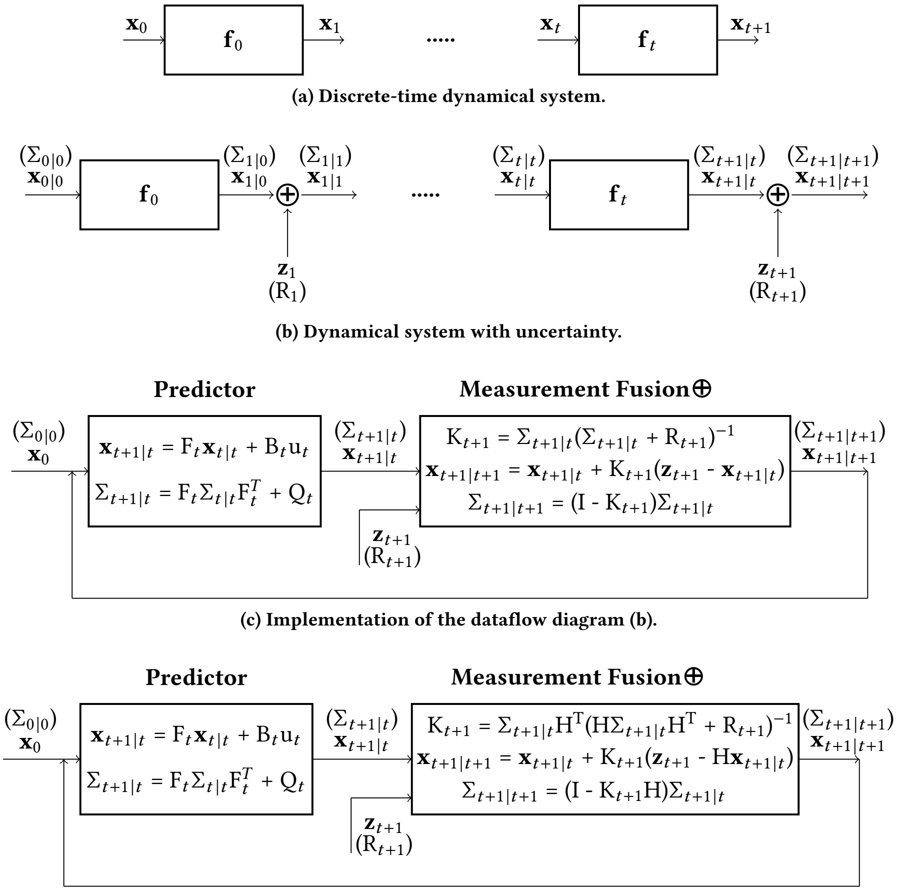
Examining Figure 4(c), we see that if we can compute $\Sigma_{t+1|t}$, the covariance matrix of $\mathbf{x}_{t+1|t}$, from $\Sigma_{t|t}$, we have everything we need to implement the vector data fusion technique described in Section 4. This can be done by applying Equation 4, which tells us that $\Sigma_{t+1|t} = F_t\Sigma_{t|t}F_t^{\mathrm{T}} + Q_t$. This equation propagates uncertainty in the input of f_t to its output.

Figure 4(c) puts all the pieces together. Although the computations appear to be complicated, the key thing to note is that they are a direct application of the vector data fusion technique of Section 4 to the special case of linear systems.

6.2 Fusing partial observations of the state

If some components of the state cannot be measured directly, the prediction phase remains unchanged from Section 6.1 but the fusion phase is different and can be understood intuitively in terms of the following steps.

(i) The portion of the *a priori* state estimate corresponding to the observable part is fused with the measurement, using the techniques developed in Sections 3-4. The result is the *a posteriori estimate of the observable state.*


$\mathbf{x}_0$
$\mathbf{f}_0$
$\mathbf{x}_1$
.....
$\mathbf{x}_t$
$\mathbf{f}_t$
$\mathbf{x}_{t+1}$
$(\Sigma_{0|0})$ $\mathbf{x}_{0|0}$
$(\Sigma_{1|0})$ $\mathbf{x}_{1|0}$
$(\Sigma_{1|1})$ $\mathbf{x}_{1|1}$
$\oplus$
$\mathbf{z}_1$ (R_1)
$(\Sigma_{t|t})$ $\mathbf{x}_{t|t}$
$(\Sigma_{t+1|t})$ $\mathbf{x}_{t+1|t}$
$(\Sigma_{t+1|t+1})$ $\mathbf{x}_{t+1|t+1}$
$\mathbf{z}_{t+1}$ (R_{t+1})
Predictor
Measurement Fusion$\oplus$
$(\Sigma_{0|0})$ $\mathbf{x}_0$
$\mathbf{x}_{t+1|t} = \mathrm{F}_t\mathbf{x}_{t|t} + \mathrm{B}_t\mathrm{u}_t$
$\Sigma_{t+1|t} = \mathrm{F}_t\Sigma_{t|t}\mathrm{F}_t^T + \mathrm{Q}_t$
$\mathrm{K}_{t+1} = \Sigma_{t+1|t}(\Sigma_{t+1|t} + \mathrm{R}_{t+1})^{-1}$
$\mathbf{x}_{t+1|t+1} = \mathbf{x}_{t+1|t} + \mathrm{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{x}_{t+1|t})$
$\Sigma_{t+1|t+1} = (\mathrm{I} - \mathrm{K}_{t+1})\Sigma_{t+1|t}$
Predictor
Measurement Fusion$\oplus$
$\mathrm{K}_{t+1} = \Sigma_{t+1|t}\mathrm{H}^\mathrm{T}(\mathrm{H}\Sigma_{t+1|t}\mathrm{H}^\mathrm{T} + \mathrm{R}_{t+1})^{-1}$
$\mathbf{x}_{t+1|t+1} = \mathbf{x}_{t+1|t} + \mathrm{K}_{t+1}(\mathbf{z}_{t+1} - \mathrm{H}\mathbf{x}_{t+1|t})$
$\Sigma_{t+1|t+1} = (\mathrm{I} - \mathrm{K}_{t+1}\mathrm{H})\Sigma_{t+1|t}$

(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.

Figure 4: State estimation using Kalman filtering

(ii) The BLUE estimator in Section 5 is used to obtain the *a posteriori estimate of the hidden state* from the *a posteriori* estimate of the observable state.

(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.*

The actual implementation produces the final result directly without going through these steps, as shown in Figure 4(d) but these incremental steps are useful for understanding how all this works.

6.2.1 Example: 2D state. Figure 5 illustrates these steps for a two-dimensional problem in which the state-vector has two components, and only the first component can be measured directly. We use the simplified notation below to focus on the essentials.

- *a priori* state estimate: $\mathbf{x}_i = \begin{pmatrix} h_i \\ c_i \end{pmatrix}$
- covariance matrix of *a priori* estimate: $\Sigma_i = \begin{pmatrix} \sigma_h^2 & \sigma_{hc} \\ \sigma_{ch} & \sigma_c^2 \end{pmatrix}$
- *a posteriori* state estimate: $\mathbf{x}_o = \begin{pmatrix} h_o \\ c_o \end{pmatrix}$
- measurement: z
- variance of measurement: r^2

The three steps discussed above for obtaining the *a posteriori* estimate involve the following calculations, shown pictorially in Figure 5.

(i) The *a priori* estimate of the observable state is h_i. The *a posteriori* estimate is obtained from Equation 14.

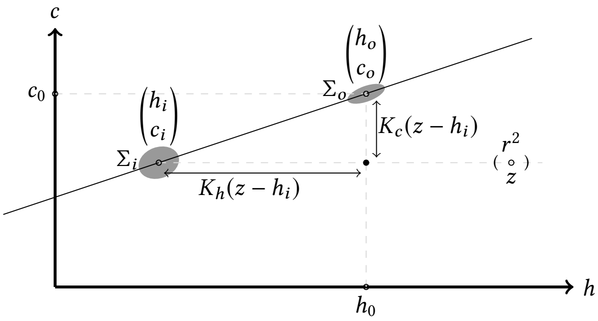
$$h_o = h_i + \frac{\sigma_h^2}{(\sigma_h^2+r^2)}(z-h_i) = h_i + K_h(z-h_i)$$

(ii) The *a priori* estimate of the hidden state is c_i. The *a posteriori* estimate is obtained from Equation 29.

$$c_o = c_i + \frac{\sigma_{hc}}{\sigma_h^2} * \frac{\sigma_h^2}{(\sigma_h^2 + r^2)}(z - h_i) = c_i + \frac{\sigma_{hc}}{(\sigma_h^2 + r^2)}(z - h_i)$$
$$= c_i + K_c(z - h_i)$$

(iii) Putting these together, we get
$$\begin{pmatrix} h_o \\ c_o \end{pmatrix} = \begin{pmatrix} h_i \\ c_i \end{pmatrix} + \begin{pmatrix} K_h \\ K_c \end{pmatrix}(z - h_i)$$

As a step towards generalizing this result in Section 6.2.2, it is useful to rewrite this expression using matrices. Define $H = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $R = \begin{pmatrix} r^2 \end{pmatrix}$. Then $\mathbf{x}_o = \mathbf{x}_i + K(z - H\mathbf{x}_i)$ where $K = \Sigma_i * H^{\mathrm{T}}(H\Sigma_i H^{\mathrm{T}} + R)^{-1}$.



c
c_0
$\begin{pmatrix} h_o \\ c_o \end{pmatrix}$
$\begin{pmatrix} h_i \\ c_i \end{pmatrix}$
Σ_o
Σ_i
$K_c(z - h_i)$
r^2
z
$K_h(z - h_i)$
h
h_0

Figure 5: Computing the *a posteriori* estimate when part of the state is not observable

6.2.2 *General case.* In general, suppose that the observable portion of a state estimate $\mathbf{x}$ is given by $H\mathbf{x}$ where H is a full row-rank matrix. Let C be a basis for the orthogonal complement of H (in the 2D example in Section 6.2.1, $H = (1 \;\; 0)$ and $C = (0 \;\; 1)$). The covariance between $C\mathbf{x}$ and $H\mathbf{x}$ is easily shown to be $C\Sigma H^{\mathrm{T}}$ where Σ is the covariance matrix of $\mathbf{x}$. The three steps for computing the *a posteriori* estimate from the *a priori* estimate in the general case are the following.

(i) The *a priori* estimate of the observable state is $H\mathbf{x}_{t+1|t}$. The *a posteriori* estimate is obtained from Equation 14:

$$H\mathbf{x}_{t+1|t+1} = H\mathbf{x}_{t+1|t} + H\Sigma_{t+1|t}H^{\mathrm{T}}(H\Sigma_{t+1|t}H^{\mathrm{T}} + R_{t+1})^{-1}(\mathbf{z}_{t+1} - H\mathbf{x}_{t+1|t})$$

Let $K_{t+1} = \Sigma_{t+1|t}H^{\mathrm{T}}(H\Sigma_{t+1|t}H^{\mathrm{T}} + R_{t+1})^{-1}$. The *a posteriori* estimate of the observable state can be written in terms of K_{t+1} as follows:

$$H\mathbf{x}_{t+1|t+1} = H\mathbf{x}_{t+1|t} + HK_{t+1}(\mathbf{z}_{t+1} - H\mathbf{x}_{t+1|t}) \tag{32}$$

(ii) The *a priori* estimate of the hidden state is $C\mathbf{x}_{t+1|t}$. The *a posteriori* estimate is obtained from Equation 29:

$$C\mathbf{x}_{t+1|t+1} = C\mathbf{x}_{t+1|t} + (C\Sigma_{t+1|t}H^{\mathrm{T}})(H\Sigma_{t+1|t}H^{\mathrm{T}})^{-1}HK_{t+1}(\mathbf{z}_{t+1} - H\mathbf{x}_{t+1|t})$$
$$= C\mathbf{x}_{t+1|t} + CK_{t+1}(\mathbf{z}_{t+1} - H\mathbf{x}_{t+1|t}) \tag{33}$$

(iii) Putting the *a posteriori* estimates (32) and (33) together,

$$\begin{pmatrix} H \\ C \end{pmatrix}\mathbf{x}_{t+1|t+1} = \begin{pmatrix} H \\ C \end{pmatrix}\mathbf{x}_{t+1|t} + \begin{pmatrix} H \\ C \end{pmatrix}K_{t+1}(\mathbf{z}_{t+1} - H\mathbf{x}_{t+1|t})$$

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

$$\mathbf{x}_{t+1|t+1} = \mathbf{x}_{t+1|t} + K_{t+1}(\mathbf{z}_{t+1} - H\mathbf{x}_{t+1|t}) \tag{34}$$

To compute $\Sigma_{t+1|t+1}$, the covariance of $\mathbf{x}_{t+1|t+1}$, note that $\mathbf{x}_{t+1|t+1} = (I - K_{t+1}H)\mathbf{x}_{t+1|t} + K_{t+1}\mathbf{z}_{t+1}$. Since $\mathbf{x}_{t+1|t}$ and $\mathbf{z}_{t+1}$ are uncorrelated, it follows from Lemma 2.2 that

$$\Sigma_{t+1|t+1} = (I - K_{t+1}H)\Sigma_{t+1|t}(I - K_{t+1}H)^T + K_{t+1}R_{t+1}K_{t+1}^T$$

Substituting the value of K_{t+1} and simplifying, we get

$$\Sigma_{t+1|t+1} = (I - K_{t+1}H)\Sigma_{t+1|t} \tag{35}$$

Figure 4(d) puts all this together. Note that if the entire state is observable, $H = I$ and the computations in Figure 4(d) reduce to those of Figure 4(c) as expected.

6.3 An example

To illustrate the concepts discussed in this section, we use a simple example in which a car starts from the origin at time t=0 and moves right along the x-axis with an initial speed of 100 m/sec. and a constant deceleration of $5m/sec.^2$ until it comes to rest.

The state-vector of the car has two components, one for the distance from the origin $d(t)$ and one for the speed $v(t)$. 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:

$$\begin{pmatrix} d_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 0.25 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} d_n \\ v_n \end{pmatrix} + \begin{pmatrix} 0 & 0.03125 \\ 0 & 0.25 \end{pmatrix}\begin{pmatrix} 0 \\ -5 \end{pmatrix} \tag{36}$$

where $\begin{pmatrix} d_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 100 \end{pmatrix}$.

The gray lines in Figure 6 show the evolution of velocity and position 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 6 shows one trajectory for this evolution, corresponding to a Gaussian noise term with covariance $\begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$ in Equation 31 (because this noise term is random, there are many trajectories for the evolution, and

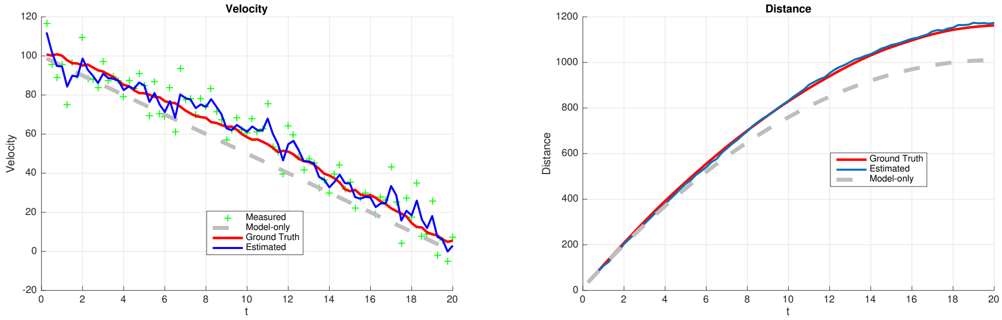


Figure 6: Estimates of the car's state over time

we are just showing one of them). The red lines correspond to "ground truth" in our experiments.

Suppose that only the velocity is directly observable. The green points in Figure 6 show the noisy measurements 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 grey lines) is inaccurate and the measurements are noisy.

6.4 Discussion

Equation 34 shows that the *a posteriori* state estimate is a linear combination of the *a priori* state estimate ($\mathbf{x}_{t+1|t}$) and the estimate of the observable state from the measurement ($\mathbf{z}_{t+1}$). Note however that unlike in Sections 3 and 4, the Kalman gain is not a dimensionless value if the entire state is not observable.

Although the optimality results of Sections 3-5 were used in the derivation of this equation, it does not follow that Equation 34 is the optimal unbiased linear estimator for combining these two estimates. However, this is easy to show using the *MSE* minimization technique that we have used repeatedly in this paper; assume that the *a posteriori* estimator is of the form $K_1\mathbf{x}_{t+1|t}+K_2\mathbf{z}_{t+1}$, and find the values of K_1 and K_2 that produce an unbiased estimator with minimum *MSE*. In fact, this is the point of departure for standard presentations of this material. The advantage of the presentation given here is that it exposes the general statistical concepts 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. Some presentations [1, 9] use properties of Gaussians to derive the results in Sections 3 although as we have seen, 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 4(d) but that if the noise is Gaussian, the 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 [20]. The presentation in this paper is predicated on the belief that this property is not needed to understand what Kalman filtering does.

7 CONCLUSIONS

Kalman filtering is a classic state estimation technique used widely in engineering applications. Although there are many presentations of this technique in the literature, most of them are focused on particular applications like robot motion, which can make it difficult to understand how to apply Kalman filtering to other problems. In this paper, we have shown that two statistical ideas - fusion of uncertain estimates and unbiased linear estimators for correlated variables - provide the conceptual underpinnings for Kalman filtering. By combining these ideas, standard results on Kalman filtering in linear systems can be derived in an intuitive and straightforward way that is simpler than other presentations of this material in the literature.

We believe the approach presented in this paper makes it easier to understand the concepts that underlie Kalman filtering and to apply it to new 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. The authors would like to thank Ivo Babuska (UT Austin), Gianfranco Bilardi (Padova), Augusto

Ferrante (Padova), Scott Niekum (UT Austin) and Peter Stone (UT Austin) for their feedback on this paper.

REFERENCES

[1] Tim Babb. 2015. How a Kalman filter works, in pictures. http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/. (2015).

[2] A. V. Balakrishnan. 1987. *Kalman Filtering Theory*. Optimization Software, Inc., Los Angeles, CA, USA.

[3] Allen L. Barker, Donald E. Brown, and Worthy N. Martin. 1994. *Bayesian Estimation and the Kalman Filter*. Technical Report. Charlottesville, VA, USA.

[4] K. Bergman. 2009. Nanophotonic Interconnection Networks in Multicore Embedded Computing. In *2009 IEEE/LEOS Winter Topicals Meeting Series*. 6–7. https://doi.org/10.1109/LEOSWT.2009.4771628

[5] Liyu Cao and Howard M. Schwartz. 2004. Analysis of the Kalman Filter Based Estimation Algorithm: An Orthogonal Decomposition Approach. *Automatica* 40, 1 (Jan. 2004), 5–19. https://doi.org/10.1016/j.automatica.2003.07.011

[6] Charles K. Chui and Guanrong Chen. 2017. *Kalman Filtering: With Real-Time Applications* (5th ed.). Springer Publishing Company, Incorporated.

[7] R.L. Eubank. 2005. *A Kalman Filter Primer (Statistics: Textbooks and Monographs)*. Chapman & Hall/CRC.

[8] Geir Evensen. 2006. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[9] Rodney Faragher. 2012. Understanding the basis of the Kalman filter via a simple and intuitive derivation. *IEEE Signal Processing Magazine* 128 (September 2012).

[10] Mohinder S. Grewal and Angus P. Andrews. 2014. *Kalman Filtering: Theory and Practice with MATLAB* (4th ed.). Wiley-IEEE Press.

[11] Anne-Kathrin Hess and Anders Rantzer. 2010. 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)*. ACM, New York, NY, USA, 191–200. https://doi.org/10.1145/1755952.1755980

[12] Connor Imes and Henry Hoffmann. 2016. Bard: A Unified Framework for Managing Soft Timing and Power Constraints. In *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)*.

[13] C. Imes, D. H. K. Kim, M. Maggio, and H. Hoffmann. 2015. POET: A Portable Approach to Minimizing Energy Under Soft Real-time Constraints. In *21st IEEE Real-Time and Embedded Technology and Applications Symposium*. 75–86. https://doi.org/10.1109/RTAS.2015.7108419

[14] Rudolph Emil Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82, Series D (1960), 35–45.

[15] Steven M. Kay. 1993. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[16] Anders Lindquist and Giogio Picci. 2017. *Linear stochastic systems*. Springer-Verlag.

[17] Kaushik Nagarajan, Nicholas Gans, and Roozbeh Jafari. 2011. Modeling Human Gait Using a Kalman Filter to Measure Walking Distance. In *Proceedings of the 2nd Conference on Wireless Health (WH '11)*. ACM, New York, NY, USA, Article 34, 2 pages. https://doi.org/10.1145/2077546.2077584

[18] Eduardo F. Nakamura, Antonio A. F. Loureiro, and Alejandro C. Frery. 2007. Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications. *ACM Comput. Surv.* 39, 3, Article 9 (Sept. 2007). https://doi.org/10.1145/1267070.1267073

[19] Raghavendra Pradyumna Pothukuchi, Amin Ansari, Petros Voulgaris, and Josep Torrellas. 2016. Using Multiple Input, Multiple Output Formal Control to Maximize Resource Efficiency in Architectures. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*. IEEE, 658–670.

[20] C.R. Rao. 1945. Information and the Accuracy Attainable in the Estimation of Statistical Parameters. *Bulletin of the Calcutta Mathematical Society* 37 (1945), 81–89.

[21] Éfren L. Souza, Eduardo F. Nakamura, and Richard W. Pazzi. 2016. Target Tracking for Sensor Networks: A Survey. *ACM Comput. Surv.* 49, 2, Article 30 (June 2016), 31 pages. https://doi.org/10.1145/2938639

[22] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

[23] Greg Welch and Gary Bishop. 1995. *An Introduction to the Kalman Filter*. Technical Report. Chapel Hill, NC, USA.

8 APPENDIX: RELEVANT RESULTS FROM STATISTICS

8.1 Basic probability theory

For a continuous random variable x, a *probability density function* (pdf) is a function $p(x)$ whose value provides a relative likelihood that the value of the random variable will equal x. The integral of the pdf within a range of values is the probability that the random variable will take a value within that range.

If $g(x)$ is a function of x with pdf $p(x)$, the *expectation* $E[g(x)]$ is defined as the following integral:

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)p(x)dx$$

By definition, the *mean* μ_x of a random variable x is $E[x]$. The *variance* of a discrete random variable x measures the variability of the distribution. For the set of possible values of x, variance (denoted by σ_x^2) is defined by $\sigma_x^2 = E[(x-\mu_x)^2]$. If all outcomes are equally likely, then $\sigma_x^2 = \frac{\Sigma(x_i-\mu_x)^2}{n}$. The standard deviation σ_x is the square root of the variance.

The *covariance* between random variables $x_1 : p_1{\sim}(\mu_1, \sigma_1^2)$ and $x_2 : p_2{\sim}(\mu_2, \sigma_2^2)$ is the expectation $E[(x_1{-}\mu_1){*}(x_2{-}\mu_2)]$. This can also be written as $E[x_1{*}x_2] - \mu_1{*}\mu_2$.

Two random variables are *uncorrelated* or *not correlated* if their covariance is zero.

Covariance and independence of random variables are different but related concepts. Two random variables are independent if knowing the value of one of the variables does not give us any information about the value of the other one. This is written formally as $p(x_1|x_2{=}a_2) = p(x_1)$.

Independent random variables are uncorrelated but random variables can be uncorrelated even if they are not independent. Consider a random variable $u : U$ that is uniformly distributed over the unit circle, and consider random variables $x : [-1, 1]$ and $y : [-1, 1]$ that are the x and y coordinates of points in U. Given a value for x, there are only two possible values for y, so x and y are not independent. However, it is easy to show that they are not correlated.

8.2 Matrix derivatives

If $f(X)$ is a scalar function of a matrix X, the matrix derivative $\frac{\partial f(X)}{\partial X}$ is defined as the matrix

$$\begin{pmatrix} \frac{\partial f(X)}{\partial X(1,1)} & \cdots & \frac{\partial f(X)}{\partial X(1,n)} \\ \cdots & \cdots & \cdots \\ \frac{\partial f(X)}{\partial X(n,1)} & \cdots & \frac{\partial f(X)}{\partial X(n,n)} \end{pmatrix}$$

Lemma 8.1. *Let X be a $m \times n$ matrix, $\boldsymbol{a}$ be a $m \times 1$ vector, $\boldsymbol{b}$ be a $n \times 1$ vector.*

$$\frac{\partial \boldsymbol{a}^T X \boldsymbol{b}}{\partial X} = \boldsymbol{a}\boldsymbol{b}^T \tag{37}$$

$$\frac{\partial \boldsymbol{a}^T X^T X \boldsymbol{b}}{\partial X} = X(\boldsymbol{a}\boldsymbol{b}^T + \boldsymbol{b}\boldsymbol{a}^T) \tag{38}$$

Proof. We sketch the proofs of both parts below.

- Equation 37: In this case, $f(X) = \mathbf{a}^{\mathrm{T}} X \mathbf{b}$. $\frac{\partial f(X)}{\partial X(i,j)} = \mathbf{a}(i)\mathbf{b}(j) = (\mathbf{a}\mathbf{b}^T)(i,j)$.
- Equation 38: In this case, $f(X) = \mathbf{a}^{\mathrm{T}} X^{\mathrm{T}} X \mathbf{b} = (X\mathbf{a})^{\mathrm{T}} X \mathbf{b}$, which is equal to $\sum_{i=1}^{m} \big(\sum_{k=1}^{n} X(i,k){*}a(k)\big){*}\big(\sum_{k=1}^{n} X(i,k){*}b(k)\big)$. Therefore $\frac{\partial f(X)}{\partial X(i,j)} = a(j){*}\sum_{k=1}^{n} X(i,k){*}b(k){+}b(j){*}\sum_{k=1}^{n} X(i,k){*}a(k)$. It is easy to see that this is the same value as the $(i,j)^{th}$ element of $X(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T)$.

□

8.3 Proof of Theorem 4.1

Theorem 4.1, which is reproduced below for convenience, can be proved using matrix derivatives.

Theorem. *Let pairwise uncorrelated estimates $\boldsymbol{x}_i(1{\le}i{\le}n)$ drawn from distributions $p_i(x){=}(\boldsymbol{\mu}_i, \Sigma_i)$ be fused using the linear model $\boldsymbol{y}_A(\boldsymbol{x}_1, .., \boldsymbol{x}_n) = \sum_{i=1}^{n} A_i \boldsymbol{x}_i$, where $\sum_{i=1}^{n} A_i = I$. The $MSE(\boldsymbol{y}_A)$ is minimized for*

$$A_i = (\sum_{\mathrm{j}=1}^{n} \Sigma_j^{-1})^{-1} \Sigma_i^{-1}.$$

Proof. To use the Lagrange multiplier approach of Section 3.2 directly, we can convert the constraint $\sum_{i=1}^{n} A_i = I$ into a set of m^2 scalar equations (for example, the first equation would be $A_1(1,1) + A_2(1,1) + .. + A_n(1,1) = 1$), and then introduce m^2 Lagrange multipliers, which can denoted by $\lambda(1,1), ...\lambda(m,m)$.

This obscures the matrix structure of the problem so it is better to implement this idea implicitly. Let Λ be an $m{\times}m$ matrix in which each entry is one of the scalar Lagrange multipliers we would have introduced in the approach described above. Analogous to the inner product of vectors, we can define the inner product of two matrices as $<A, B> = trace(A^{\mathrm{T}} B)$ (it is easy to see that $<A, B>$ is $\sum_{i=1}^{m} \sum_{j=1}^{m} A(i,j)B(i,j)$). Using this notation, we can formulate the optimization problem using Lagrange multipliers as follows:

$$f(A_1, ..., A_n) = E\Big\{\sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}_i)^{\mathrm{T}} {A_i}^{\mathrm{T}} A_i (\mathbf{x}_i - \boldsymbol{\mu}_i)\Big\} + \Big\langle \Lambda, \big(\sum_{i=1}^{n} A_i - I\big)\Big\rangle$$

Taking the matrix derivative of f with respect to each A_i and setting each derivative to zero to find the optimal values of A_i gives us the equation $E\big\{2A_i(\mathbf{x}_i - \boldsymbol{\mu}_i)(\mathbf{x}_i - \boldsymbol{\mu}_i)^{\mathrm{T}} + \Lambda\big\} = 0$.

This equation can be written as $2A_i\Sigma_i + \Lambda = 0$, which implies

$$A_1\Sigma_1 = A_2\Sigma_2 = \ldots = A_n\Sigma_n = -\frac{\Lambda}{2}$$

Using the constraint that the sum of all A_i equals to the identity matrix I gives us the desired expression for A_i:

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

□