Optimization-Based Control

Richard M. Murray
Control and Dynamical Systems
California Institute of Technology

Version v2.2d (22 Feb 2022)

Chapter 7

Sensor Fusion

In this chapter we consider the problem of combining the data from different sensors to obtain an estimate of a (common) dynamical system. Unlike the previous chapters, we focus here on discrete-time processes, leaving the continuous-time case to the exercises. We begin with a summary of the input/output properties of discrete-time systems with stochastic inputs, then present the discrete-time Kalman filter, and use that formalism to formulate and present solutions for the sensor fusion problem. Some advanced methods of estimation and fusion are also summarized at the end of the chapter that demonstrate how to move beyond the linear, Gaussian process assumptions.

Prerequisites. The material in this chapter is designed to be reasonably self-contained, so that it can be used without covering Sections 5.3–5.4 or Chapter 6 of this supplement. We assume rudimentary familiarity with discrete-time linear systems, at the level of the brief descriptions in Chapters 3 and 7 of FBS2e, and discrete-time random processes as described in Section 5.2 of these notes.

7.1 Discrete-Time Stochastic Systems

We begin with a concise overview of stochastic system in discrete time, echoing our development of continuous-time random systems described in Chapter 5. We consider systems of the form

$$X[k+1] = AX[k] + Bu[k] + FV[k], \qquad Y[k] \quad = CX[k] + W[k], \tag{7.1}$$

where $X \in \mathbb{R}^n$ represents the state, $u \in \mathbb{R}^m$ represents the (deterministic) input, $V \in \mathbb{R}^q$ represents process disturbances, $Y \in \mathbb{R}^p$ represents the system output and $V \in \mathbb{R}^p$ represents measurement noise.

As in the case of continuous-time systems, we are interested in the response of the system to the random input $V[k]$. We will assume that V is a Gaussian process with zero mean and correlation function $r_V(k, k+d)$ (or correlation matrix $R_V(k, k+d)$ if V is vector valued). As in the continuous case, we say that a random process is white noise if $r_V(k, k+d) = r_V \delta(d)$ with $\delta(d) = 1$ if $d = 0$ and 0 otherwise. (Note that in the discrete-time case, white noise has finite covariance.)

To compute the response $Y[k]$ of the system, we look at the properties of the state vector $X[k]$. For simplicity, we take $u = 0$ (since the system is linear, we can always add it back in by superposition). Note first that the state at time $k+d$ can be written as

$$\begin{aligned} X[k+d] &= AX[k+d-1] + FV[x+l-1] \\ &= A(AX[k+d-2] + FV[x+l-2]) + FV[x+l-1] \\ &= A^d X[k] + \sum_{j=1}^{d} A^{j-1} FV[k+d-j]. \end{aligned}$$

The mean of the state at time k is given by

$$\mathbb{E}(X[k]) = A^k \mathbb{E}(E[0]) + \sum_{j=1}^{k} A^{j-1} F \mathbb{E}(V[k-j]) = A^k \mathbb{E}(X[0]).$$

To compute the covariance $R_X(k, k+d)$, we start by computing $R_X(k, k+1)$:

$$\begin{aligned} R_X(k, k+1) &= \mathbb{E}(X[k] X^\mathsf{T}[k+1]) \\ &= \mathbb{E}((A^k x[0] + A^{k-1} F w[0] + \cdots + ABw[k-2] + B[k-1]) \cdot \\ &\qquad (A^{k+1} x[0] + A^k B w[0] + \cdots + Bw[k])^\mathsf{T}) \end{aligned}$$

Performing a similar calculation for $R_X(k, k+d)$, it can be shown that

$$\begin{aligned} R_X(k, k+d) = \big(A^k P[0] (A^\mathsf{T})^k + A^{k-1} F R_V[0] F^\mathsf{T} (A^\mathsf{T})^{k-1} + \ldots \\ + F R_V[k] F^\mathsf{T}\big) (A^\mathsf{T})^d =: P[k] (A^\mathsf{T})^d, \end{aligned} \tag{7.2}$$

where

$$P[k+1] = AP[k]A^\mathsf{T} + F R_V[k] F^\mathsf{T}. \tag{7.3}$$

The matrix $P[k]$ is the covariance of the state matrix and we see that its value can be computed *recursively* starting with $P[0] = \mathbb{E}(X[0] X^\mathsf{T}[0])$ and then applying equation (7.3). Equations (7.2) and (7.3) are the equivalent of Proposition 5.2 for continuous-time processes. If we additionally assume that V is stationary and focus on the steady state response, we obtain the following.

Proposition 7.1 (Steady state response to white noise)**.** *For a discrete-time, time-invariant, linear system driven by white noise, the correlation matrices for the state and output converge in steady state to*

$$R_X(d) = R_X(k, k+d) = PA^d, \qquad R_Y(d) = C R_X(d) C^\mathsf{T},$$

where P satisfies the algebraic equation

$$APA^\mathsf{T} + F R_V F^\mathsf{T} = 0 \qquad P > 0. \tag{7.4}$$

7.2 Kalman Filters in Discrete Time (FBS2e)

We now consider the optimal estimator in discrete time. This material is presented in FBS2e in slightly simplified (but consistent) form.

Consider a discrete time, linear system with input, having dynamics

$$\begin{aligned} X[k+1] &= AX[k] + Bu[k] + FV[k], \\ Y[k] &= CX[k] + W[k], \end{aligned} \tag{7.5}$$

where $V[k]$ and $W[k]$ are Gaussian, white noise processes satisfying

$$\begin{aligned} &\mathbb{E}(V[k]) = 0 && \mathbb{E}(W[k]) = 0 \\ &\mathbb{E}(V[k]V^{\mathsf{T}}[j]) = \begin{cases} 0 & k \neq j \\ R_V & k = j \end{cases} && \mathbb{E}(W[k]W^{\mathsf{T}}[j]) = \begin{cases} 0 & k \neq j \\ R_W & k = j \end{cases} \\ &\mathbb{E}(V[k]W^{\mathsf{T}}[j]) = 0. \end{aligned} \tag{7.6}$$

We assume that the initial condition is also modeled as a Gaussian random variable with

$$\mathbb{E}(X[0]) = x_0 \qquad \mathbb{E}(X[0]X^{\mathsf{T}}[0]) = P[0]. \tag{7.7}$$

We wish to find an estimate $\hat{X}[k]$ that gives the minimum mean square error (MMSE) for $\mathbb{E}((\hat{X}[k] - X[k])(\hat{X}[k] - X[k])^{\mathsf{T}})$ given the measurements $\{Y[l] : 0 \leq l \leq k\}$. We consider an observer of the form

$$\hat{X}[k+1] = A\hat{X}[k] + Bu[k] - L[k](C\hat{X}[k] - Y[k]). \tag{7.8}$$

The following theorem summarizes the main result.

Theorem 7.2. *Consider a random process $X[k]$ with dynamics* (7.5) *and noise processes and initial conditions described by equations* (7.6) *and* (7.7)*. The observer gain L that minimizes the mean square error is given by*

$$L[k] = AP[k]C^{\mathsf{T}}(R_W + CP[k]C^{\mathsf{T}})^{-1},$$

where

$$\begin{aligned} P[k+1] &= (A - LC)P[k](A - LC)^{\mathsf{T}} + FR_VF^{\mathsf{T}} + LR_WL^{\mathsf{T}} \\ P[0] &= \mathbb{E}(X[0]X^{\mathsf{T}}[0]). \end{aligned} \tag{7.9}$$

Proof. We wish to minimize the mean square of the error, $\mathbb{E}((\hat{X}[k] - X[k])(\hat{X}[k] - X[k])^{\mathsf{T}})$. We will define this quantity as $P[k]$ and then show that it satisfies the recursion given in equation (7.9). Let $E[k] = C\hat{X}[k] - Y[k]$ be the residual between the measured output and the estimated output. By definition,

$$\begin{aligned} P[k+1] &= \mathbb{E}(E[k+1]E^{\mathsf{T}}[k+1]) \\ &= (A - LC)P[k](A - LC)^{\mathsf{T}} + FR_VF^{\mathsf{T}} + LR_WL^{\mathsf{T}} \\ &= AP[k]A^{\mathsf{T}} - AP[k]C^{\mathsf{T}}L^{\mathsf{T}} - LCP[k]A^{\mathsf{T}} + \\ &\qquad L(R_W + CP[k]C^{\mathsf{T}})L^{\mathsf{T}} + FR_VF^{\mathsf{T}}. \end{aligned}$$

Letting $R_\epsilon = (R_W + CP[k]C^\mathsf{T})$, we have

$$\begin{aligned}P[k+1] &= AP[k]A^\mathsf{T} - AP[k]C^\mathsf{T}L^\mathsf{T} - LCP[k]A^\mathsf{T} + LR_\epsilon L^\mathsf{T} + FR_VF^\mathsf{T} \\ &= AP[k]A^\mathsf{T} + \big(L - AP[k]C^\mathsf{T}R_\epsilon^{-1}\big)R_\epsilon\big(L - AP[k]C^\mathsf{T}R_\epsilon^{-1}\big)^\mathsf{T} \\ &\quad - AP[k]C^\mathsf{T}R_\epsilon^{-1}CP[k]A^\mathsf{T} + FR_VF^\mathsf{T}.\end{aligned}$$

In order to minimize this expression, we choose $L = AP[k]C^\mathsf{T}R_\epsilon^{-1}$ and the theorem is proven. □

Note that the Kalman filter has the form of a *recursive* filter: given $P[k] = \mathbb{E}(E[k]E[k]^\mathsf{T})$ at time k, can compute how the estimate and covariance *change*. Thus we do not need to keep track of old values of the output. Furthermore, the Kalman filter gives the estimate $\hat{X}[k]$ *and* the covariance $P[k]$, so we can see how reliable the estimate is. It can also be shown that the Kalman filter extracts the maximum possible information about output data. It can be shown that for the Kalman filter the correlation matrix for the error is

$$R_E[j,k] = R\delta_{jk}.$$

In other words, the error is a white noise process, so there is no remaining dynamic information content in the error.

In the special case when the noise is stationary (R_V, R_W constant) and *if* $P[k]$ converges, then the observer gain is constant:

$$L = APC^\mathsf{T}(R_W + CPC^\mathsf{T}),$$

where P satisfies

$$P = APA^\mathsf{T} + FR_VF^\mathsf{T} - APC^\mathsf{T}\big(R_W + CPC^\mathsf{T}\big)^{-1}CPA^\mathsf{T}.$$

We see that the optimal gain depends on both the process noise and the measurement noise, but in a nontrivial way. Like the use of LQR to choose state feedback gains, the Kalman filter permits a systematic derivation of the observer gains given a description of the noise processes. The solution for the constant gain case is solved by the `dlqe` command in MATLAB.

7.3 Predictor-Corrector Form

The Kalman filter can be written in a two step form by separating the correction step (where we make use of new measurements of the output) and the prediction step (where we compute the expected state and covariance at the next time instant).

We make use of the notation $\hat{X}[k|j]$ to represent the estimated state at time instant k given the information up to time j (where typically $j = k-1$). Using this notation, the filter can be solved using the following algorithm:

Step 0: Initialization.

$$\begin{aligned}k &= 1 \\ \hat{X}[0|0] &= \mathbb{E}(X[0]) \\ P[0|0] &= \mathbb{E}(X[0]X^\mathsf{T}[0])\end{aligned}$$

Step 1: Prediction. Update the estimates and covariance matrix to account for all data taken up to time $k-1$:

$$\begin{aligned}\hat{X}[k|k-1] &= A\hat{X}[k-1|k-1] + Bu[k-1]\\ P[k|k-1] &= AP[k-1|k-1]A^{\mathsf{T}} + FR_V[k-1]F^{\mathsf{T}}\end{aligned}$$

Step 2: Correction. Correct the estimates and covariance matrix to account for the data taken at time step k:

$$\begin{aligned}L[k] &= P[k|k-1]C^{\mathsf{T}}(R_W + CP[k|k-1]C^{\mathsf{T}})^{-1},\\ \hat{X}[k|k] &= \hat{X}[k|k-1] + L[k](Y[k] - C\hat{X}[k|k-1]),\\ P[k|k] &= P[k|k-1] - L[k]CP[k|k-1].\end{aligned}$$

Step 3: Iterate. Set k to $k+1$ and repeat steps 1 and 2.

Note that the correction step reduces the covariance by an amount related to the relative accuracy of the measurement, while the prediction step increases the covariance by an amount related to the process disturbance.

This form of the discrete-time Kalman filter is convenient because we can reason about the estimate in the case when we do not obtain a measurement on every iteration of the algorithm. In this case, we simply update the prediction step (increasing the covariance) until we receive new sensor data, at which point we call the correction step (decreasing the covariance).

The following lemma will be useful in the sequel:

Lemma 7.3. *The optimal gain $L[k]$ satisfies*

$$L[k] = P[k|k]C^{\mathsf{T}}R_W^{-1}$$

Proof. $L[k]$ is defined as

$$L[k] = P[k|k-1]C^{\mathsf{T}}(R_W + CP[k|k-1]C^{\mathsf{T}})^{-1}.$$

Multiplying through by the inverse term on the right and expanding, we have

$$\begin{aligned}L[k](R_W + CP[k|k-1]C^{\mathsf{T}}) &= P[k|k-1]C^{\mathsf{T}},\\ L[k]R_W + L[k]CP[k|k-1]C^{\mathsf{T}} &= P[k|k-1]C^{\mathsf{T}},\end{aligned}$$

and hence

$$\begin{aligned}L[k]R_W &= P[k|k-1]C^{\mathsf{T}} - L[k]CP[k|k-1]C^{\mathsf{T}},\\ &= (I - L[k]C)P[k|k-1]C^{\mathsf{T}} = P[k|k]C^{\mathsf{T}}.\end{aligned}$$

The desired results follows by multiplying on the right by R_W^{-1}. □

7.4 Sensor Fusion

We now return to the main topic of the chapter: sensor fusion. Consider the situation described in Figure 7.1, where we have an input/output dynamical system

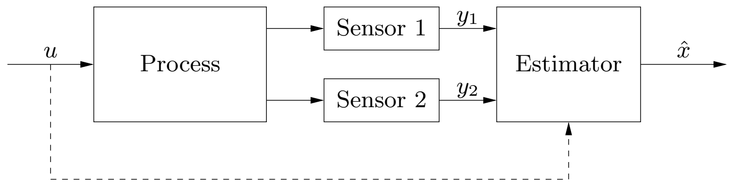


Figure 7.1: Sensor fusion. Multiple sensors report on data from a single process. The estimator fusions this information from the sensors to obtain an estimate of the state of the system. Depending on the use case, the input (dashed line) may not be avaliable to the estimator.

with multiple sensors capable of taking measurements. The problem of sensor fusion involves deciding how to best combine the measurements from the individual sensors in order to accurately estimate the process state X. Since different sensors may have different noise characteristics, evidently we should combine the sensors in a way that places more weight on sensors with lower noise. In addition, in some situations we may have different sensors available at different times, so that not all information is available on each measurement update.

While sensor fusion can be used for estimation of the state of a system being controlled, another common application is to sense the state of a system in the environment. A difference for this use case is that the input to the system in the environment is often not available, requiring the estimator to use a model for the system in which the input is modeled by a random process.

Sensor weighting

To gain more insight into how the sensor data are combined, we investigate the functional form of $L[k]$. Suppose that each sensor takes a measurement of the form

$$Y^i = C^i X + W^i, \qquad i = 1, \dots, p,$$

where the superscript i corresponds to the specific sensor. Let W^i be a zero mean, white noise process with covariance $\sigma_i^2 = R_{W^i}(0)$. It follows from Lemma 7.3 that

$$L[k] = P[k|k] C^\mathsf{T} R_W^{-1}.$$

First note that if $P[k|k]$ is small, indicating that our estimate of X is close to the actual value (in the MMSE sense), then $L[k]$ will be small due to the leading $P[k|k]$ term. Furthermore, the characteristics of the individual sensors are contained in the different σ_i^2 terms, which only appears in R_W. Expanding the gain matrix, we have

$$L[k] = P[k|k] C^\mathsf{T} R_W^{-1}, \qquad R_W^{-1} = \begin{bmatrix} 1/\sigma_1^2 & & \\ & \ddots & \\ & & 1/\sigma_p^2 \end{bmatrix}.$$

We see from the form of R_W^{-1} that each sensor is inversely weighted by its covariance. Thus noisy sensors ($\sigma_i^2 \gg 1$) will have a small weight and require averaging over many iterations before their data can affect the state estimate. Conversely, if $\sigma_i^2 \ll 1$, the data is "trusted" and is used with higher weight in each iteration.

Information filters

An alternative formulation of the Kalman filter is to make use of the inverse of the covariance matrix, called the *information matrix*, to represent the error of the estimate. It turns out that writing the state estimator in this form has several advantages both conceptually and when implementing distributed computations. This form of the Kalman filter is known as the *information filter*.

We begin by defining the information matrix I and the weighted state estimate $\hat{Z}$:

$$I[k|k] = P^{-1}[k|k], \qquad \hat{Z}[k|k] = P^{-1}[k|k]\hat{X}[k|k].$$

We also make use of the following quantities, which appear in the Kalman filter equations:

$$\Omega^i[k|k] = (C^i)^\mathsf{T} R_{W^i}^{-1}[k|k]C^i, \qquad \Psi^i[k|k] = (C^i)^\mathsf{T} R_{W^i}^{-1}[k|k]C^i\hat{X}[k|k].$$

Using these quantities, we can rewrite the Kalman filter equations as a prediction step

$$\begin{aligned} I[k|k-1] &= \Big(AI^{-1}[k-1|k-1]A^\mathsf{T} + R_W\Big)^{-1}, \\ \hat{Z}[k|k-1] &= I[k|k-1]AI^{-1}[k-1|k-1]\hat{Z}[k-1|k-1] + Bu[k-1] \end{aligned}$$

and a correction step

$$\begin{aligned} I[k|k] &= I[k|k-1] + \sum_{i=1}^{p} \Omega^i[k|k], \\ \hat{Z}[k|k] &= \hat{Z}[k|k-1] + \sum_{i=1}^{p} \Psi^i[k|k]. \end{aligned}$$

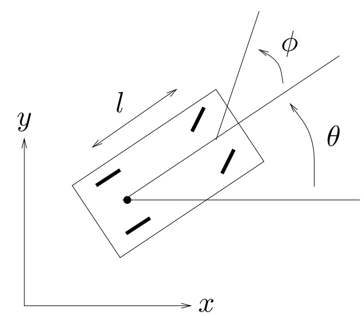
Note that these last equations are in a particularly simple form, with the information matrix being updated by each sensor's Ω^i and similarly the state estimate being updated by each sensor's Ψ^i.

The advantage of using the infomration filter version of the equation is that it allows a simple addition operation for the correction step, corresponding to adding the "information" obtained through the acquisition of new data. We also see the clear relationship between the information content in each sensor channel and the inverse covariance of that sensor, thorugh the definitions of Ω^i and Ψ^i.

Another feature of the information filter formulation is that it allows some efficiencies when implementing distributed estimation across netwworks. In particular, the information carried in the individual sensors can be simply added together through the updates of $I[k|k-1]$. This is helpful especially when the sensors have variable sampling rate and the measurement packets arrive at different times. New information is incorporated whenever it arrives and then a global update of $I[k|k–1]$ at a centralized node is used to integrate all sensor measurements (which can the be rebroadcast out to the sensors). The information form also makes clear how to handle missing data: if no data arrives for a given sensor then no information is added and only the time update is applied, hence the measurement update is skipped.

Exercises

7.1 Consider the problem of estimating the position of an autonomous mobile vehicle using a GPS receiver and an IMU (inertial measurement unit). The continuous time dynamics of the vehicle are given by



$$\dot{x} = \cos\theta\, v$$
$$\dot{y} = \sin\theta\, v$$
$$\dot{\theta} = \frac{1}{\ell}\tan\phi\, v,$$

We assume that the vehicle is disturbance free, but that we have noisy measurements from the GPS receiver and IMU and an initial condition error.

(a) Rewrite the equations of motion in discrete time, assuming that we update the dynamics at a sample time of $h = 0.005$ sec and that we can take $\dot{x}$ to be roughly constant over that period. Run a simulation of your discrete time model from initial condition $(0, 0, 0)$ with constant input $\phi = \pi/8$, $v = 5$ and compare your results with the continuous time model.

(b) Suppose that we have a GPS measurement that is taken every 0.1 seconds and an IMU measurement that is taken every 0.01 seconds. Write a MATLAB program that that computes the discrete time Kalman filter for this system, using the same disturbance, noise and initial conditions as Exercise 6.4.

7.2 Consider a continuous time dynamical system with multiple measurements,

$$\dot{X} = AX + Bu + FV, \qquad Y^i = C^i x + W^i, \quad i = 1, \ldots, q.$$

Assume that the measurement noises W^i are indendendent for each sensor and have zero mean and variance σ_i^2. Show that the optimal estimator for X weights the measurements by the inverse of their covariances.

7.3 Show that if we formulate the optimal estimate using an estimator of the form

$$\hat{X}[k+1] = A\hat{X}[k] + L[k](Y[k+1] - CA\hat{X}[k])$$

that we recover the update law in the predictor-corrector form.