

# 3

## *Gaussian Filters*

### 3.1 Introduction

This chapter describes an important family of recursive state estimators, collectively called *Gaussian filters*. Historically, Gaussian filters constitute the earliest tractable implementations of the Bayes filter for continuous spaces. They are also by far the most popular family of techniques to date—despite a number of shortcomings.

Gaussian techniques all share the basic idea that beliefs are represented by multivariate normal distributions. We already encountered a definition of the multivariate normal distribution in Equation (2.4), which is restated here for convenience:

$$(3.1) \quad p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\}$$

This density over the variable  $x$  is characterized by two sets of parameters: The mean  $\mu$  and the covariance  $\Sigma$ . The mean  $\mu$  is a vector that possesses the same dimensionality as the state  $x$ . The covariance is a quadratic matrix that is symmetric and positive-semidefinite. Its dimension is the dimensionality of the state  $x$  squared. Thus, the number of elements in the covariance matrix depends quadratically on the number of elements in the state vector.

The commitment to represent the posterior by a Gaussian has important ramifications. Most importantly, Gaussians are unimodal; they possess a single maximum. Such a posterior is characteristic of many tracking problems in robotics, in which the posterior is focused around the true state with a small margin of uncertainty. Gaussian posteriors are a poor match for many global estimation problems in which many distinct hypotheses exist, each of which forms its own mode in the posterior.

The parameterization of a Gaussian by its mean and covariance is called

MOMENTS  
PARAMETERIZATION

CANONICAL  
PARAMETERIZATION

Moments

the *moments parameterization*. This is because the mean and covariance are the first and second moments of a probability distribution; all other moments are zero for normal distributions. In this chapter, we will also discuss an alternative parameterization, called *canonical parameterization*, or sometimes *natural parameterization*. Both parameterizations, the moments and the canonical parameterizations, are functionally equivalent in that a bijective mapping exists that transforms one into the other. However, they lead to filter algorithms with somewhat different computational characteristics. As we shall see, the canonical and the natural parameterizations are best thought of as duals: what appears to be computationally easy in one parameterization is involved in the other, and vice versa.

This chapter introduces the two basic Gaussian filter algorithms.

- Chapter 3.2 describes the Kalman filter, which implements the Bayes filter using the moments parameterization for a restricted class of problems with linear dynamics and measurement functions.
- The Kalman filter is extended to nonlinear problems in Chapter 3.3, which describes the extended Kalman filter.
- Chapter 3.4 describes a different nonlinear Kalman filter, known as unscented Kalman filter.
- Chapter 3.5 describes the information filter, which is the dual of the Kalman filter using the canonical parameterization of Gaussians.

## 3.2 The Kalman Filter

### 3.2.1 Linear Gaussian Systems

Probably the best studied technique for implementing Bayes filters is the *Kalman filter*, or (*KF*). The Kalman filter was invented by Swerling (1958) and Kalman (1960) as a technique for filtering and prediction in *linear Gaussian systems*, which will be defined in a moment. The Kalman filter implements belief computation for continuous states. It is not applicable to discrete or hybrid state spaces.

The Kalman filter represents beliefs by the moments parameterization: At time  $t$ , the belief is represented by the the mean  $\mu_t$  and the covariance  $\Sigma_t$ . Posteriors are *Gaussian* if the following three properties hold, in addition to the Markov assumptions of the Bayes filter.

GAUSSIAN POSTERIOR

1. The state transition probability  $p(x_t | u_t, x_{t-1})$  must be a *linear* function in its arguments with added Gaussian noise. This is expressed by the following equation:

$$(3.2) \quad x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

Here  $x_t$  and  $x_{t-1}$  are state vectors, and  $u_t$  is the control vector at time  $t$ . In our notation, both of these vectors are vertical vectors. They are of the form

$$(3.3) \quad x_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{n,t} \end{pmatrix} \quad \text{and} \quad u_t = \begin{pmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{m,t} \end{pmatrix}$$

$A_t$  and  $B_t$  are matrices.  $A_t$  is a square matrix of size  $n \times n$ , where  $n$  is the dimension of the state vector  $x_t$ .  $B_t$  is of size  $n \times m$ , with  $m$  being the dimension of the control vector  $u_t$ . By multiplying the state and control vector with the matrices  $A_t$  and  $B_t$ , respectively, the state transition function becomes *linear* in its arguments. Thus, Kalman filters assume linear system dynamics.

The random variable  $\varepsilon_t$  in (3.2) is a Gaussian random vector that models the uncertainty introduced by the state transition. It is of the same dimension as the state vector. Its mean is zero, and its covariance will be denoted  $R_t$ . A state transition probability of the form (3.2) is called a *linear Gaussian*, to reflect the fact that it is linear in its arguments with additive Gaussian noise. Technically, one may also include a constant additive term in (3.2), which is here omitted since it plays no role in the material to come.

Equation (3.2) defines the state transition probability  $p(x_t | u_t, x_{t-1})$ . This probability is obtained by plugging Equation (3.2) into the definition of the multivariate normal distribution (3.1). The mean of the posterior state is given by  $A_t x_{t-1} + B_t u_t$  and the covariance by  $R_t$ :

$$(3.4) \quad p(x_t | u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t) \right\}$$

```

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

```

**Table 3.1** The Kalman filter algorithm for linear Gaussian state transitions and measurements.

2. The measurement probability  $p(z_t | x_t)$  must also be *linear* in its arguments, with added Gaussian noise:

$$(3.5) \quad z_t = C_t x_t + \delta_t$$

Here  $C_t$  is a matrix of size  $k \times n$ , where  $k$  is the dimension of the measurement vector  $z_t$ . The vector  $\delta_t$  describes the measurement noise. The distribution of  $\delta_t$  is a multivariate Gaussian with zero mean and covariance  $Q_t$ . The measurement probability is thus given by the following multivariate normal distribution:

$$(3.6) \quad p(z_t | x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right\}$$

3. Finally, the initial belief  $bel(x_0)$  must be normally distributed. We will denote the mean of this belief by  $\mu_0$  and the covariance by  $\Sigma_0$ :

$$(3.7) \quad bel(x_0) = p(x_0) = \det(2\pi \Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1} (x_0 - \mu_0)\right\}$$

These three assumptions are sufficient to ensure that the posterior  $bel(x_t)$  is always a Gaussian, for any point in time  $t$ . The proof of this non-trivial result can be found below, in the mathematical derivation of the Kalman filter (Chapter 3.2.4).

### 3.2.2 The Kalman Filter Algorithm

The *Kalman filter algorithm* is depicted in Table 3.1. Kalman filters represent the belief  $bel(x_t)$  at time  $t$  by the mean  $\mu_t$  and the covariance  $\Sigma_t$ . The input of the Kalman filter is the belief at time  $t - 1$ , represented by  $\mu_{t-1}$  and  $\Sigma_{t-1}$ . To update these parameters, Kalman filters require the control  $u_t$  and the measurement  $z_t$ . The output is the belief at time  $t$ , represented by  $\mu_t$  and  $\Sigma_t$ .

In lines 2 and 3, the predicted belief  $\bar{\mu}$  and  $\bar{\Sigma}$  is calculated representing the belief  $\bar{bel}(x_t)$  one time step later, but before incorporating the measurement  $z_t$ . This belief is obtained by incorporating the control  $u_t$ . The mean is updated using the deterministic version of the state transition function (3.2), with the mean  $\mu_{t-1}$  substituted for the state  $x_{t-1}$ . The update of the covariance considers the fact that states depend on previous states through the linear matrix  $A_t$ . This matrix is multiplied twice into the covariance, since the covariance is a quadratic matrix.

The belief  $\bar{bel}(x_t)$  is subsequently transformed into the desired belief  $bel(x_t)$  in lines 4 through 6, by incorporating the measurement  $z_t$ . The variable  $K_t$ , computed in line 4 is called *Kalman gain*. It specifies the degree to which the measurement is incorporated into the new state estimate, in a way that will become clearer in Chapter 3.2.4. Line 5 manipulates the mean, by adjusting it in proportion to the Kalman gain  $K_t$  and the deviation of the actual measurement,  $z_t$ , and the measurement predicted according to the measurement probability (3.5). The key concept here is the *innovation*, which is the difference between the actual measurement  $z_t$  and the expected measurement  $C_t \bar{\mu}_t$  in line 5. Finally, the new covariance of the posterior belief is calculated in line 6, adjusting for the information gain resulting from the measurement.

The Kalman filter is computationally quite efficient. For today's best algorithms, the complexity of matrix inversion is approximately  $O(d^{2.4})$  for a matrix of size  $d \times d$ . Each iteration of the Kalman filter algorithm, as stated here, is lower bounded by (approximately)  $O(k^{2.4})$ , where  $k$  is the dimension of the measurement vector  $z_t$ . This (approximate) cubic complexity stems from the matrix inversion in line 4. Even for certain sparse updates discussed in future chapters, it is also at least in  $O(n^2)$ , where  $n$  is the dimension of the state space, due to the multiplication in line 6 (the matrix  $K_t C_t$  may be sparse). In many applications—such as the robot mapping applications discussed in later chapters—the measurement space is much lower dimensional than the state space, and the update is dominated by the  $O(n^2)$  operations.

KALMAN GAIN

INNOVATION

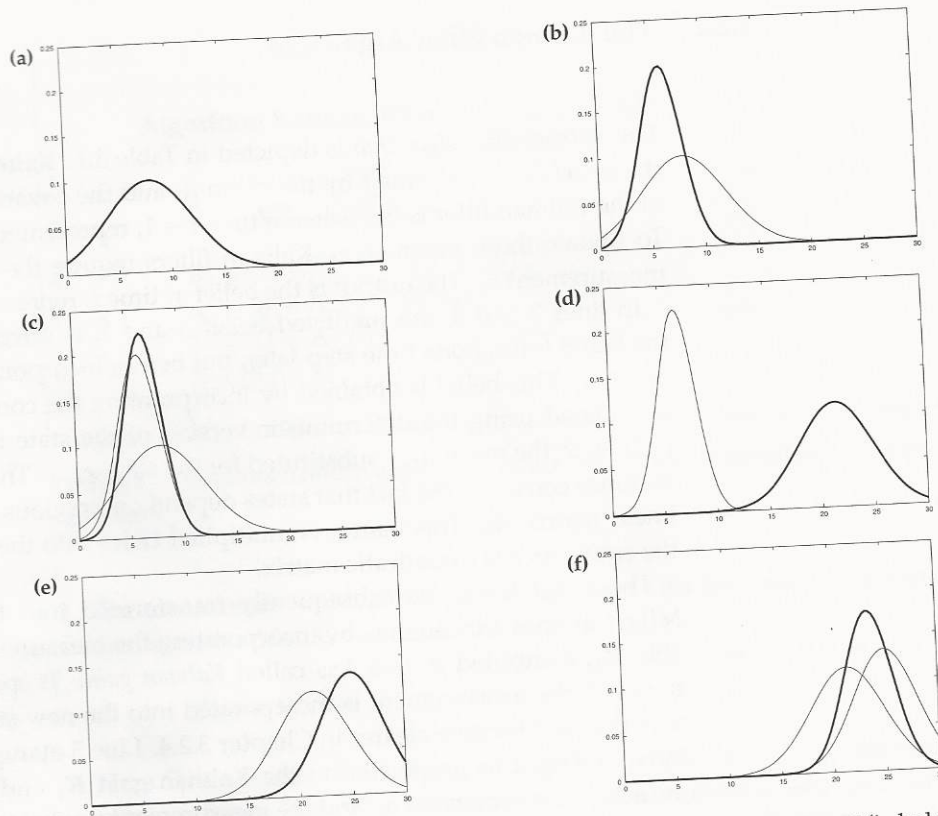


Figure 3.2 Illustration of Kalman filters: (a) initial belief, (b) a measurement (in bold) with the associated uncertainty, (c) belief after integrating the measurement into the belief using the Kalman filter algorithm, (d) belief after motion to the right (which introduces uncertainty), (e) a new measurement with associated uncertainty, and (f) the resulting belief.

### 3.2.3 Illustration

Figure 3.2 illustrates the Kalman filter algorithm for a simplistic one-dimensional localization scenario. Suppose the robot moves along the horizontal axis in each diagram in Figure 3.2. Let the prior over the robot location be given by the normal distribution shown in Figure 3.2a. The robot queries its sensors on its location (e.g., a GPS system), and those return a measurement that is centered at the peak of the bold Gaussian in Figure 3.2b. This bold Gaussian illustrates this measurement: Its peak is the value predicted

by the sensors, and its width (variance) corresponds to the uncertainty in the measurement. Combining the prior with the measurement, via lines 4 through 6 of the Kalman filter algorithm in Table 3.1, yields the bold Gaussian in Figure 3.2c. This belief's mean lies between the two original means, and its uncertainty radius is smaller than both contributing Gaussians. The fact that the residual uncertainty is smaller than the contributing Gaussians may appear counter-intuitive, but it is a general characteristic of information integration in Kalman filters.

Next, assume the robot moves towards the right. Its uncertainty grows due to the fact that the state transition is stochastic. Lines 2 and 3 of the Kalman filter provide us with the Gaussian shown in bold in Figure 3.2d. This Gaussian is shifted by the amount the robot moved, and it is also wider for the reasons just explained. The robot receives a second measurement illustrated by the bold Gaussian in Figure 3.2e, which leads to the posterior shown in bold in Figure 3.2f.

As this example illustrates, the Kalman filter alternates a *measurement update step* (lines 5-7), in which sensor data is integrated into the present belief, with a *prediction step* (or control update step), which modifies the belief in accordance to an action. The update step decreases and the prediction step increases uncertainty in the robot's belief.

### 3.2.4 Mathematical Derivation of the KF

This section derives the Kalman filter algorithm in Table 3.1. The section can safely be skipped at first reading; it is only included for completeness.

Up front, the derivation of the KF is largely an exercise in manipulating quadratic expressions. When multiplying two Gaussians, for example, the exponents add. Since both original exponents are quadratic, so is the resulting sum. The remaining exercise is then to come up with a factorization of the result into a form that makes it possible to read off the desired parameters.

#### Part 1: Prediction

Our derivation begins with lines 2 and 3 of the algorithm, in which the belief  $\bar{bel}(x_t)$  is calculated from the belief one time step earlier,  $bel(x_{t-1})$ . Lines 2 and 3 implement the update step described in Equation (2.41), restated here