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:

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

This density over the variable x is characterized by two sets of parameters: The mean μ and the covariance Σ. The mean μ 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, that is, they posses 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 forming its own mode in the posterior.

The representation of a Gaussian by its mean and covariance is called the *moments representation.* 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 representation, called *canonical representation*, or sometimes *natural representation*. Both representations, the moments and the canonical representations, are functionally equivalent in that a bijective mapping exists that transforms one into the other (and back). However, they lead to filter algorithms with orthogonal computational characteristics.

This chapter introduces the two basic Gaussian filter algorithms.

- Section 3.2 describes the Kalman filter, which implements the Bayes filter using the moments representation for a restricted class of problems with linear dynamics and measurement functions.
- The Kalman filter is extended to nonlinear problems in Section 3.3, which describes the extended Kalman filter.
- Section 3.4 describes the information filter, which is the dual of the Kalman filter using the canonical representation 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 (KF)*. The Kalman filter was invented in the 1950s by Rudolph Emil Kalman, as a technique for filtering and prediction in linear systems. 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 representation: At time t, the belief is represented by the the mean μ_t and the covariance Σ_t. Posteriors are Gaussian if the following three properties hold, in addition to the Markov assumptions of the Bayes filter.

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

$$x_t \quad = \quad A_t x_{t-1} + B_t u_t + \varepsilon_t \; . \tag{3.2}$$

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, that is, they are of the form

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

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 ε_t in (3.2) is a Gaussian random vector that models the randomness in 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.

Equation (3.2) defines the state transition probability $p(x_t \mid u_t, x_{t-1})$. 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:

$$\begin{aligned} & p(x_t \mid u_t, x_{t-1}) \\ &= \det(2\pi R_t)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right\} \end{aligned} \tag{3.4}$$

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

$$z_t = C_t x_t + \delta_t . \tag{3.5}$$

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

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

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 μ_t, Σ_t

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

3. Finally, the initial belief $bel(x_0)$ must be normal distributed. We will denote the mean of this belief by μ_0 and the covariance by Σ_0:

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

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 (Section 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 μ_t and the covariance Σ_t. The input of the Kalman filter is the belief at time $t-1$, represented by μ_{t-1} and Σ_{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 μ_t and Σ_t.

In Lines 2 and 3, the predicted belief $\bar{\mu}$ and $\bar{\Sigma}$ is calculated representing the belief $\overline{bel}(x_t)$ 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 μ_{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 $\overline{bel}(x_t)$ 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. 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). 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 $O(d^{2.8})$ for a matrix of size $d \times d$. Each iteration of the Kalman filter algorithm, as stated here, is lower bounded by (approximately) $O(k^{2.8})$, where k is the dimension of the measurement vector z_t. This (approximate) cubic complexity stems from the matrix inversion in Line 4. 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.

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 next state transition is stochastic. Lines 2 and 3 of the Kalman filter provides 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. Next, the

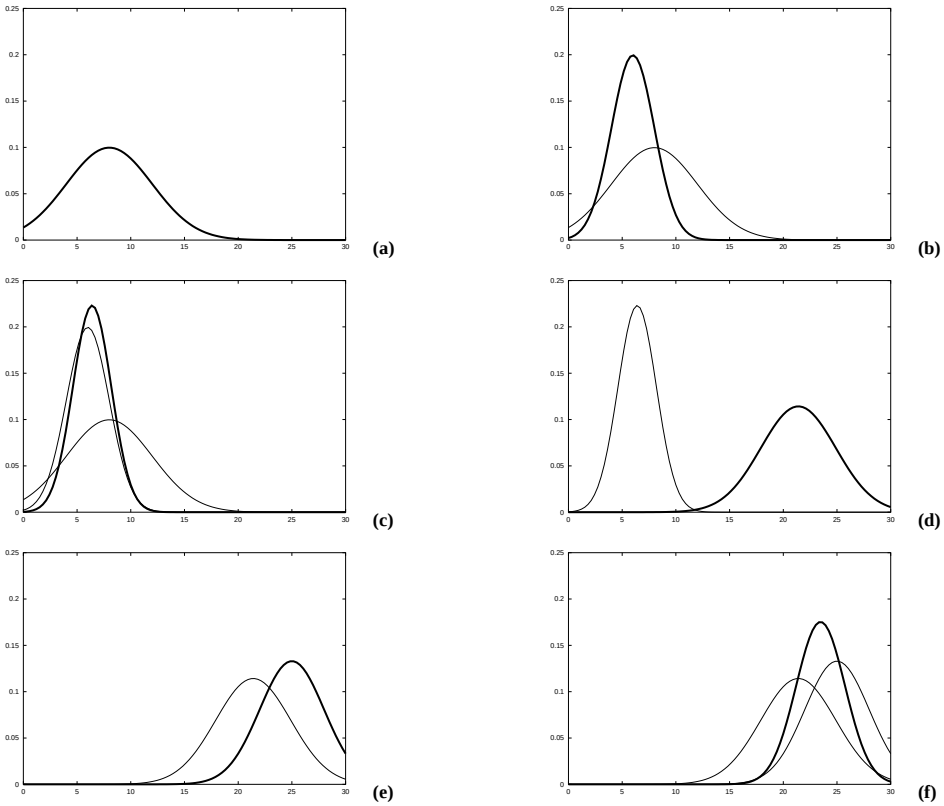

Figure 3.2 Illustration of Kalman filters: (a) initial belief, (b) a measurement (in bold) with the associated uncertainty, (c) belief after integrating the measurement into the belief using the Kalman filter algorithm, (d) belief after motion to the right (which introduces uncertainty), (e) a new measurement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

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.

Part 1: Prediction. Our derivation begins with Lines 2 and 3 of the algorithm, in which the belief $\overline{bel}(x_t)$ is calculated from the belief one time step earlier, $bel(x_{t-1})$. Lines 2 and 3 implement the update step described in Equation (2.61), restated here for the reader's convenience:

$$\overline{bel}(x_t) = \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\sim \mathcal{N}(x_t; A_t x_{t-1} + B_t u_t, R_t)} \underbrace{bel(x_{t-1})}_{\sim \mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})} dx_{t-1} \tag{3.7}$$

The "prior" belief $bel(x_{t-1})$ is represented by the mean μ_{t-1} and the covariance Σ_{t-1}. The state transition probability $p(x_t \mid x_{t-1}, u_t)$ was given in (3.4) as a normal distribution over x_t with mean $A_t x_{t-1} + B_t u_t$ and covariance R_t. As we shall show now, the outcome of (3.7) is again a Gaussian with mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$ as stated in Table 3.1.

We begin by writing (3.7) in its Gaussian form:

$$\begin{aligned} \overline{bel}(x_t) &= \eta \int \exp\left\{-\tfrac{1}{2}\,(x_t - A_t\,x_{t-1} - B_t\,u_t)^T\,R_t^{-1}\,(x_t - A_t\,x_{t-1} - B_t\,u_t)\right\} \\ &\qquad \exp\left\{-\tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})\right\}\,dx_{t-1}\,. \end{aligned} \tag{3.8}$$

In short, we have

$$\overline{bel}(x_t) = \eta \int \exp\{-L_t\}\,dx_{t-1} \tag{3.9}$$

with

$$\begin{aligned} L_t &= \tfrac{1}{2}\,(x_t - A_t\,x_{t-1} - B_t\,u_t)^T\,R_t^{-1}\,(x_t - A_t\,x_{t-1} - B_t\,u_t) \\ &\quad + \tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\,\Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}). \end{aligned} \tag{3.10}$$

Notice that L_t is quadratic in x_{t-1}; it is also quadratic in x_t.

Expression (3.9) contains an integral. To solve this integral in closed form, we will now decompose L_t into two functions, $L_t(x_{t-1}, x_t)$ and $L_t(x_t)$:

$$L_t = L_t(x_{t-1}, x_t) + L_t(x_t) \tag{3.11}$$

so that all terms containing x_{t-1} are collected in $L_t(x_{t-1}, x_t)$. This decomposition will allow us to move $L_t(x_t)$ outside the integration, since its value does not depend on the integration variable x_{t-1}:

$$\begin{aligned} \overline{bel}(x_t) &= \eta \int \exp\{-L_t\}\; dx_{t-1} \\ &= \eta \int \exp\{-L_t(x_{t-1}, x_t) - L_t(x_t)\}\; dx_{t-1} \\ &= \eta\,\exp\{-L_t(x_t)\} \int \exp\{-L_t(x_{t-1}, x_t)\}\; dx_{t-1} \end{aligned} \tag{3.12}$$

Furthermore, we will choose $L_t(x_{t-1}, x_t)$ such that the value of the integral in (3.12) does not depend on x_t. Thus, the integral will simply become a constant relative to the problem of estimating the belief distribution over x_t. The resulting distribution over x_t will be entirely defined through $L_t(x_t)$.

$$\overline{bel}(x_t) = \eta\,\exp\{-L_t(x_t)\} \tag{3.13}$$

Let us now perform this decomposition. We are seeking a function $L_t(x_{t-1}, x_t)$ quadratic in x_{t-1}. (This function will also depend on x_t, but that shall not concern us at this point.) To determine the coefficients of this quadratic, we calculate the first two derivatives of L_t:

$$\frac{\partial L_t}{\partial x_{t-1}} = -A_t^T\,R_t^{-1}\,(x_t - A_t\,x_{t-1} - B_t\,u_t) + \Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \tag{3.14}$$

$$\frac{\partial^2 L_t}{\partial x_{t-1}^2} \quad = \quad A_t^T \; R_t^{-1} \; A_t + \Sigma_{t-1}^{-1} \quad =: \quad \Psi_t^{-1} \tag{3.15}$$

Ψ_t defines the curvature of $L_t(x_{t-1}, x_t)$. Setting the first derivative of L_t to 0 gives us the mean:

$$A_t^T \; R_t^{-1} \; (x_t - A_t \; x_{t-1} - B_t \; u_t) \quad = \quad \Sigma_{t-1}^{-1} \; (x_{t-1} - \mu_{t-1}) \tag{3.16}$$

This expression is now solved for x_{t-1}

$$\begin{aligned}
&\iff \quad A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) - A_t^T \; R_t^{-1} \; A_t \; x_{t-1} \quad = \quad \Sigma_{t-1}^{-1} \; x_{t-1} - \Sigma_{t-1}^{-1} \; \mu_{t-1} \\
&\iff \quad A_t^T \; R_t^{-1} \; A_t \; x_{t-1} + \Sigma_{t-1}^{-1} \; x_{t-1} \quad = \quad A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) + \Sigma_{t-1}^{-1} \; \mu_{t-1} \\
&\iff \quad (A_t^T \; R_t^{-1} \; A_t \; + \Sigma_{t-1}^{-1}) \; x_{t-1} \quad = \quad A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) + \Sigma_{t-1}^{-1} \; \mu_{t-1} \\
&\iff \quad \Psi_t^{-1} \; x_{t-1} \quad = \quad A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) + \Sigma_{t-1}^{-1} \; \mu_{t-1} \\
&\iff \quad x_{t-1} \quad = \quad \Psi_t \; [A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) + \Sigma_{t-1}^{-1} \; \mu_{t-1}]
\end{aligned} \tag{3.17}$$

Thus, we now have a quadratic function $L_t(x_{t-1}, x_t)$, defined as follows:

$$\begin{aligned}
L_t(x_{t-1}, x_t) \quad = \quad & \tfrac{1}{2}(x_{t-1} - \Psi_t \; [A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) + \Sigma_{t-1}^{-1} \; \mu_{t-1}])^T \; \Psi^{-1} \\
& (x_{t-1} - \Psi_t \; [A_t^T \; R_t^{-1} \; (x_t - B_t \; u_t) + \Sigma_{t-1}^{-1} \; \mu_{t-1}])
\end{aligned} \tag{3.18}$$

Clearly, this is not the only quadratic function satisfying our decomposition in (3.11). However, $L_t(x_{t-1}, x_t)$ is of the common quadratic form of the negative exponent of a normal distribution. In fact the function

$$\det(2\pi\Psi)^{-\frac{1}{2}} \; \exp\{-L_t(x_{t-1}, x_t)\} \tag{3.19}$$

is a valid probability density function (PDF) for the variable x_{t-1}. As the reader easily verifies, this function is of the form defined in (3.1). We know from (2.5) that PDFs integrate to 1. Thus, we have

$$\int \det(2\pi\Psi)^{-\frac{1}{2}} \; \exp\{-L_t(x_{t-1}, x_t)\} \; dx_{t-1} \quad = \quad 1 \tag{3.20}$$

From this it follows that

$$\int \exp\{-L_t(x_{t-1}, x_t)\}\, dx_{t-1} \quad = \quad \det(2\pi\Psi)^{\frac{1}{2}}\ . \tag{3.21}$$

The important thing to notice is that the value of this integral is *independent* of x_t, our target variable. Thus, for our problem of calculating a distribution over x_t, this integral is constant. Subsuming this constant into the normalizer η, we get the following expression for Equation (3.12):

$$\begin{aligned} \overline{bel}(x_t) &= \eta\ \exp\{-L_t(x_t)\} \int \exp\{-L_t(x_{t-1}, x_t)\}\ dx_{t-1} \\ &= \eta\ \exp\{-L_t(x_t)\} \end{aligned} \tag{3.22}$$

Notice that the normalizers η left and right of the equal sign are *not* the same. This decomposition establishes the correctness of (3.13).

It remains to determine the function $L_t(x_t)$, which is the difference of L_t, defined in (3.10), and $L_t(x_{t-1}, x_t)$, defined in (3.18):

$$\begin{aligned} L_t(x_t) &= L_t - L_t(x_{t-1}, x_t) \\ &= \tfrac{1}{2}\ (x_t - A_t\ x_{t-1} - B_t\ u_t)^T\ R_t^{-1}\ (x_t - A_t\ x_{t-1} - B_t\ u_t) \\ &\quad + \tfrac{1}{2}\ (x_{t-1} - \mu_{t-1})^T\ \Sigma_{t-1}^{-1}\ (x_{t-1} - \mu_{t-1}) \\ &\quad -\tfrac{1}{2}(x_{t-1} - \Psi_t\ [A_t^T\ R_t^{-1}\ (x_t - B_t\ u_t) + \Sigma_{t-1}^{-1}\ \mu_{t-1}])^T\ \Psi^{-1} \\ &\quad (x_{t-1} - \Psi_t\ [A_t^T\ R_t^{-1}\ (x_t - B_t\ u_t) + \Sigma_{t-1}^{-1}\ \mu_{t-1}]) \end{aligned} \tag{3.23}$$

Let us quickly verify that $L_t(x_t)$ indeed does not depend on x_{t-1}. To do so, we substitute back $\Psi_t = (A_t^T\ R_t^{-1}\ A_t + \Sigma_{t-1}^{-1})^{-1}$, and multiply out the terms above. For the reader's convenience, terms that contain x_{t-1} are underlined (doubly if they are quadratic in x_{t-1}).

$$\begin{aligned} L_t(x_t) &= \underline{\underline{\tfrac{1}{2}\ x_{t-1}^T A_t^T\ R_t^{-1}\ A_t\ x_{t-1}}}\ \underline{- x_{t-1}^T A_t^T\ R_t^{-1}\ (x_t - B_t\ u_t)} \\ &\qquad + \tfrac{1}{2}\ (x_t - B_t\ u_t)^T\ R_t^{-1}\ (x_t - B_t\ u_t) \\ &\quad \underline{\underline{+\tfrac{1}{2}\ x_{t-1}^T\ \Sigma_{t-1}^{-1}\ x_{t-1}}}\ \underline{- x_{t-1}^T\ \Sigma_{t-1}^{-1}\ \mu_{t-1}}\ + \tfrac{1}{2}\ \mu_{t-1}^T\ \Sigma_{t-1}^{-1}\ \mu_{t-1} \\ &\quad \underline{\underline{-\tfrac{1}{2}\ x_{t-1}^T\ (A_t^T\ R_t^{-1}\ A_t + \Sigma_{t-1}^{-1})\ x_{t-1}}} \end{aligned}$$

$$\begin{array}{l} \underline{+x_{t-1}^T\,[A_t^T\;R_t^{-1}\;(x_t - B_t\;u_t) + \Sigma_{t-1}^{-1}\;\mu_{t-1}]} \\ -\frac{1}{2}\;[A_t^T\;R_t^{-1}\;(x_t - B_t\;u_t) + \Sigma_{t-1}^{-1}\;\mu_{t-1}]^T\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1} \\ \qquad [A_t^T\;R_t^{-1}\;(x_t - B_t\;u_t) + \Sigma_{t-1}^{-1}\;\mu_{t-1}] \end{array} \tag{3.24}$$

It is now easily seen that all terms that contain x_{t-1} cancel out. This should come at no surprise, since it is a consequence of our construction of $L_t(x_{t-1}, x_t)$.

$$\begin{array}{rcl} L_t(x_t) & = & +\frac{1}{2}\;(x_t - B_t\;u_t)^T\;R_t^{-1}\;(x_t - B_t\;u_t)\;+\frac{1}{2}\;\mu_{t-1}^T\;\Sigma_{t-1}^{-1}\;\mu_{t-1} \\ & & -\frac{1}{2}\;[A_t^T\;R_t^{-1}\;(x_t - B_t\;u_t) + \Sigma_{t-1}^{-1}\;\mu_{t-1}]^T\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1} \\ & & \qquad [A_t^T\;R_t^{-1}\;(x_t - B_t\;u_t) + \Sigma_{t-1}^{-1}\;\mu_{t-1}] \end{array} \tag{3.25}$$

Furthermore, $L_t(x_t)$ is quadratic in x_t. This observation means that $\overline{bel}(x_t)$ is indeed normal distributed. The mean and covariance of this distribution are of course the minimum and curvature of $L_t(x_t)$, which we now easily obtain by by computing the first and second derivatives of $L_t(x_t)$ with respect to x_t:

$$\begin{array}{rcl} \frac{\partial L_t(x_t)}{\partial x_t} & = & R_t^{-1}\;(x_t - B_t\;u_t)\;\; - R_t^{-1}\;A_t\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1} \\ & & [A_t^T\;R_t^{-1}\;(x_t - B_t\;u_t) + \Sigma_{t-1}^{-1}\;\mu_{t-1}] \\ & = & [R_t^{-1} - R_t^{-1}\;A_t\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1} A_t^T\;R_t^{-1}]\;(x_t - B_t\;u_t) \\ & & - R_t^{-1}\;A_t\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1}\;\Sigma_{t-1}^{-1}\;\mu_{t-1} \end{array} \tag{3.26}$$

The *inversion lemma* stated (and proved) in Table 3.2 allows us to express the first factor as follows:

$$R_t^{-1} - R_t^{-1}\;A_t\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1} A_t^T\;R_t^{-1} \;\; = \;\; (R_t + A_t\;\Sigma_{t-1}\;A_t^T)^{-1} \tag{3.27}$$

Hence the desired derivative is given by the following expression:

$$\begin{array}{rcl} \frac{\partial L_t(x_t)}{\partial x_t} & = & (R_t + A_t\;\Sigma_{t-1}\;A_t^T)^{-1}\;(x_t - B_t\;u_t) \\ & & -R_t^{-1}\;A_t\;(A_t^T\;R_t^{-1}\;A_t + \Sigma_{t-1}^{-1})^{-1}\;\Sigma_{t-1}^{-1}\;\mu_{t-1} \end{array} \tag{3.28}$$

Inversion Lemma. For any invertible quadratic matrices R and Q and any matrix P with appropriate dimension, the following holds true

$$(R + P\,Q\,P^T)^{-1} \;=\; R^{-1} - R^{-1}\,P\,(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,R^{-1}$$

assuming that all above matrices can be inverted as stated.

Proof. It suffices to show that

$$(R^{-1} - R^{-1}\,P\,(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,R^{-1})\,(R + P\,Q\,P^T) \;=\; I$$

This is shown through a series of transformations:

$$\begin{aligned}
&= \underbrace{R^{-1}\,R}_{=I} + R^{-1}\,P\,Q\,P^T \;-\; R^{-1}\,P\,(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,\underbrace{R^{-1}\,R}_{=I} \\
&\quad - R^{-1}\,P\,(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,R^{-1}\,P\,Q\,P^T \\
&= I \;+\; R^{-1}\,P\,Q\,P^T \;-\; R^{-1}\,P\,(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T \\
&\quad - R^{-1}\,P\,(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,R^{-1}\,P\,Q\,P^T \\
&= I \;+\; R^{-1}\,P\,[Q\,P^T \;-\; (Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T \\
&\quad - (Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,R^{-1}\,P\,Q\,P^T] \\
&= I \;+\; R^{-1}\,P\,[Q\,P^T \;-\; (Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,\underbrace{Q^{-1}\,Q}_{=I}\,P^T \\
&\quad - (Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,P^T\,R^{-1}\,P\,Q\,P^T] \\
&= I \;+\; R^{-1}\,P\,[Q\,P^T \;-\; \underbrace{(Q^{-1} + P^T\,R^{-1}\,P)^{-1}\,(Q^{-1} + P^T\,R^{-1}\,P)}_{=I}\,Q\,P^T] \\
&= I \;+\; R^{-1}\,P\,[\underbrace{Q\,P^T \;-\; Q\,P^T}_{=0}] \;=\; I
\end{aligned}$$

Table 3.2 The (specialized) inversion lemma.

The minimum of $L_t(x_t)$ is attained when the first derivative is zero.

$$(R_t + A_t\,\Sigma_{t-1}\,A_t^T)^{-1}\,(x_t - B_t\,u_t) \;=\; R_t^{-1}\,A_t\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\,\Sigma_{t-1}^{-1}\,\mu_{t-1} \tag{3.29}$$

Solving this for the target variable x_t gives us the surprisingly compact result

$$\begin{aligned} x_t &= B_t\ u_t + \underbrace{(R_t + A_t\ \Sigma_{t-1}\ A_t^T) R_t^{-1}\ A_t}_{A_t + A_t\ \Sigma_{t-1}\ A_t^T R_t^{-1}\ A_t} \underbrace{(A_t^T\ R_t^{-1}\ A_t + \Sigma_{t-1}^{-1})^{-1}\ \Sigma_{t-1}^{-1}}_{(\Sigma_{t-1} A_t^T\ R_t^{-1}\ A_t + I)^{-1}}\ \mu_{t-1} \\ &= B_t\ u_t + A_t\ \underbrace{(I + \Sigma_{t-1}\ A_t^T R_t^{-1}\ A_t)\ (\Sigma_{t-1} A_t^T\ R_t^{-1}\ A_t + I)^{-1}}_{= I}\ \mu_{t-1} \\ &= B_t\ u_t + A_t\ \mu_{t-1} \end{aligned} \tag{3.30}$$

Thus, the mean of the belief $\overline{bel}(x_t)$ after incorporating the motion command u_t is $B_t\ u_t + A_t\ \mu_{t-1}$. This proves the correctness of Line 2 of the Kalman filter algorithm in Table 3.1. Line 3 is now obtained by calculating the second derivative of $L_t(x_t)$:

$$\frac{\partial^2 L_t(x_t)}{\partial x_t^2} = (R_t + A_t\ \Sigma_{t-1}\ A_t^T)^{-1} \tag{3.31}$$

This is the curvature of the quadratic function $L_t(x_t)$, whose inverse is the covariance of the belief $\overline{bel}(x_t)$.

To summarize, we showed that the prediction steps in Lines 2 and 3 of the Kalman filter algorithm indeed implement the Bayes filter prediction step. To do so, we first decomposed the exponent of the belief $\overline{bel}(x_t)$ into two functions, $L_t(x_{t-1}, x_t)$ and $L_t(x_t)$. Then we showed that $L_t(x_{t-1}, x_t)$ changes the predicted belief $\overline{bel}(x_t)$ only by a constant factor, which can be subsumed into the normalizing constant η. Finally, we determined the function $L_t(x_t)$ and showed that it results in the mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$ of the Kalman filter prediction $\overline{bel}(x_t)$.

Part 2: Measurement Update. We will now derive the measurement update in Lines 4, 5, and 6 (Table 3.1) of our Kalman filter algorithm. We begin with the general Bayes filter mechanism for incorporating measurements, stated in Equation (2.58) and restated here in annotated form:

$$bel(x_t) = \eta\ \underbrace{p(z_t \mid x_t)}_{\sim \mathcal{N}(z_t; C_t x_t, Q_t)}\ \underbrace{\overline{bel}(x_t)}_{\sim \mathcal{N}(x_t; \bar{\mu}_t, \bar{\Sigma}_t)} \tag{3.32}$$

The mean and covariance of $\overline{bel}(x_t)$ are obviously given by $\bar{\mu}_t$ and $\bar{\Sigma}_t$. The measurement probability $p(z_t \mid x_t)$ was defined in (3.6) to be normal as well, with mean $C_t\ x_t$

and covariance Q_t. Thus, the product is given by an exponential

$$bel(x_t) \quad = \quad \eta \ \exp\{-J_t\} \tag{3.33}$$

with

$$J_t \quad = \quad \tfrac{1}{2} \ (z_t - C_t x_t)^T \ Q_t^{-1} \ (z_t - C_t x_t) + \tfrac{1}{2} \ (x_t - \bar{\mu}_t)^T \ \bar{\Sigma}_t^{-1} \ (x_t - \bar{\mu}_t) \tag{3.34}$$

This function is quadratic in x_t, hence $bel(x_t)$ is a Gaussian. To calculate its parameters, we once again calculate the first two derivatives of J_t with respect to x_t:

$$\frac{\partial J}{\partial x_t} \quad = \quad -C_t^T \ Q_t^{-1} \ (z_t - C_t x_t) + \bar{\Sigma}_t^{-1} \ (x_t - \bar{\mu}_t) \tag{3.35}$$

$$\frac{\partial^2 J}{\partial x_t^2} \quad = \quad C_t^T \ Q_t^{-1} \ C_t + \bar{\Sigma}_t^{-1} \tag{3.36}$$

The second term is the inverse of the covariance of $bel(x_t)$:

$$\Sigma_t \quad = \quad (C_t^T \ Q_t^{-1} \ C_t + \bar{\Sigma}_t^{-1})^{-1} \tag{3.37}$$

The mean of $bel(x_t)$ is the minimum of this quadratic function, which we now calculate by setting the first derivative of J_t to zero (and substituting μ_t for x_t):

$$C_t^T \ Q_t^{-1} \ (z_t - C_t \mu_t) \quad = \quad \bar{\Sigma}_t^{-1} \ (\mu_t - \bar{\mu}_t) \tag{3.38}$$

The expression on the left of the equal sign can be transformed as follows:

$$\begin{aligned} & C_t^T \ Q_t^{-1} \ (z_t - C_t \mu_t) \\ & \quad = \quad C_t^T \ Q_t^{-1} \ (z_t - C_t \ \mu_t + C_t \ \bar{\mu}_t - C_t \ \bar{\mu}_t) \\ & \quad = \quad C_t^T \ Q_t^{-1} \ (z_t - C_t \bar{\mu}_t) - C_t^T \ Q_t^{-1} \ C_t \ (\mu_t - \bar{\mu}_t) \end{aligned} \tag{3.39}$$

Substituting this back into (3.38) gives us

$$C_t^T \ Q_t^{-1} \ (z_t - C_t \bar{\mu}_t) \quad = \quad \underbrace{(C_t^T \ Q_t^{-1} \ C_t + \bar{\Sigma}_t^{-1})}_{= \Sigma_t^{-1}} \ (\mu_t - \bar{\mu}_t) \tag{3.40}$$

and hence we have

$$\Sigma_t \, C_t^T \, Q_t^{-1} \, (z_t - C_t \, \bar{\mu}_t) \quad = \quad \mu_t - \bar{\mu}_t \tag{3.41}$$

We now define the Kalman gain as

$$K_t \quad = \quad \Sigma_t \, C_t^T \, Q_t^{-1} \tag{3.42}$$

and obtain

$$\mu_t \quad = \quad \bar{\mu}_t + K_t \, (z_t - C_t \, \bar{\mu}_t) \tag{3.43}$$

This proves the correctness of Line 5 in the Kalman filter algorithm in Table 3.1.

The Kalman gain, as defined in (3.42), is a function of Σ_t. This is at odds with the fact that we utilize K_t to calculate Σ_t in Line 6 of the algorithm. The following transformation shows us how to expresses K_t in terms of covariances other than Σ_t. It begins with the definition of K_t in (3.42):

$$\begin{aligned}
K_t &= \Sigma_t \, C_t^T \, Q_t^{-1} \\
&= \Sigma_t \, C_t^T \, Q_t^{-1} \, \underbrace{(C_t \, \bar{\Sigma}_t \, C_t^T + Q_t) \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1}}_{= I} \\
&= \Sigma_t \, (C_t^T \, Q_t^{-1} \, C_t \, \bar{\Sigma}_t \, C_t^T + C_t^T \, \underbrace{Q_t^{-1} \, Q_t}_{= I}) \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1} \\
&= \Sigma_t \, (C_t^T \, Q_t^{-1} \, C_t \, \bar{\Sigma}_t \, C_t^T + C_t^T) \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1} \\
&= \Sigma_t \, (C_t^T \, Q_t^{-1} \, C_t \, \bar{\Sigma}_t \, C_t^T + \underbrace{\bar{\Sigma}_t^{-1} \, \bar{\Sigma}_t}_{= I} \, C_t^T) \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1} \\
&= \Sigma_t \, \underbrace{(C_t^T \, Q_t^{-1} \, C_t + \bar{\Sigma}_t^{-1})}_{= \Sigma_t^{-1}} \, \bar{\Sigma}_t \, C_t^T \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1} \\
&= \underbrace{\Sigma_t \, \Sigma_t^{-1}}_{= I} \, \bar{\Sigma}_t \, C_t^T \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1} \\
&= \bar{\Sigma}_t \, C_t^T \, (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1}
\end{aligned} \tag{3.44}$$

This expression proves the correctness of Line 4 of our Kalman filter algorithm. Line 6 is obtained by expressing the covariance using the Kalman gain K_t. The advantage

of the calculation in Table 3.1 over the definition in Equation (3.37) lies in the fact that we can avoid inverting the state covariance matrix. This is essential for applications of Kalman filters to high-dimensional state spaces.

Our transformation is once again carried out using the *inversion lemma*, which was already stated in Table 3.2. Here we restate it using the notation of Equation (3.37):

$$(\bar{\Sigma}_t^{-1} + C_t^T\ Q_t^{-1}\ C_t)^{-1} \quad = \quad \bar{\Sigma}_t - \bar{\Sigma}_t\ C_t^T\ (Q_t + C_t\ \bar{\Sigma}_t\ C_t^T)^{-1}\ C_t\ \bar{\Sigma}_t \tag{3.45}$$

This lets us arrive at the following expression for the covariance:

$$\begin{aligned}
\Sigma_t &= (C_t^T\ Q_t^{-1}\ C_t + \bar{\Sigma}_t^{-1})^{-1} \\
&= \bar{\Sigma}_t - \bar{\Sigma}_t\ C_t^T\ (Q_t + C_t\ \bar{\Sigma}_t\ C_t^T)^{-1}\ C_t\ \bar{\Sigma}_t \\
&= [I - \underbrace{\bar{\Sigma}_t\ C_t^T\ (Q_t + C_t\ \bar{\Sigma}_t\ C_t^T)^{-1}}_{= K_t, \text{ see Eq. (3.44)}}\ C_t]\ \bar{\Sigma}_t \\
&= (I - K_t\ C_t)\ \bar{\Sigma}_t
\end{aligned} \tag{3.46}$$

This proves the correctness of Line 6 of our Kalman filter algorithm.

3.3 THE EXTENDED KALMAN FILTER

The assumptions of linear state transitions and linear measurements with added Gaussian noise are rarely fulfilled in practice. For example, a robot that moves with constant translational and rotational velocity typically moves on a circular trajectory, which cannot be described by linear next state transitions. This observation, along with the assumption of unimodal beliefs, renders plain Kalman filters, as discussed so far, inapplicable to all but the most trivial robotics problems.

The extended Kalman filter (EKF) overcomes one of these assumptions: the linearity assumption. Here the assumption is that the next state probability and the measurement probabilities are governed by nonlinear functions g and h, respectively:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{3.47}$$

$$z_t = h(x_t) + \delta_t\ . \tag{3.48}$$

This model strictly generalizes the linear Gaussian model underlying Kalman filters, postulated in Equations (3.2) and (3.5). The function g replaces the matrices A_t and B_t in (3.2), and h replaces the matrix C_t in (3.5). Unfortunately, with arbitrary functions g and h, the belief is no longer a Gaussian. In fact, performing the belief update exactly is usually impossible for nonlinear functions g and h, in the sense that the Bayes filter does not possess a closed-form solution.

The extended Kalman filter (EKF) calculates an approximation to the true belief. It represents this approximation by a Gaussian. In particular, the belief $bel(x_t)$ at time t is represented by a mean μ_t and a covariance Σ_t. Thus, the EKF inherits from the Kalman filter the basic belief representation, but it differs in that this belief is only approximate, not exact as was the case in Kalman filters.

3.3.1 Linearization Via Taylor Expansion

The key idea underlying the EKF is called *linearization*. Figure **??** illustrates the basic concept. Suppose we are given a nonlinear next state function g. A Gaussian projected through this function is typically non-Gaussian. This is because nonlinearities in g distort the belief in ways that destroys its nice Gaussian shape, as illustrated in the figure. Linearization approximates g by a linear function that is tangent to g at the mean of the Gaussian. By projecting the Gaussian through this linear approximation, the posterior is Gaussian. In fact, once g is linearized, the mechanics of belief propagation are equivalent to those of the Kalman filter. The same argument applies to the multiplication of Gaussians when a measurement function h is involved. Again, the EKF approximates h by a linear function tangent to h, thereby retaining the Gaussian nature of the posterior belief.

There exist many techniques for linearizing nonlinear functions. EKFs utilize a method called (first order) *Taylor expansion*. Taylor expansion construct a linear approximation to a function g from g's value and slope. The slope is given by the partial derivative

$$g'(u_t, x_{t-1}) \quad := \quad \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}} \tag{3.49}$$

Clearly, both the value of g and its slope depend on the argument of g. A logical choice for selecting the argument is to chose the state deemed most likely at the time of linearization. For Gaussians, the most likely state is the mean of the posterior μ_{t-1}. In other words, g is approximated by its value at μ_{t-1} (and at u_t), and the linear

extrapolation is achieved by a term proportional to the gradient of g at μ_{t-1} and u_t:

$$\begin{aligned} g(u_t, x_{t-1}) &\approx g(u_t, \mu_{t-1}) + \underbrace{g'(u_t, \mu_{t-1})}_{=: G_t} (x_{t-1} - \mu_{t-1}) \\ &= g(u_t, \mu_{t-1}) + G_t \, (x_{t-1} - \mu_{t-1}) \end{aligned} \tag{3.50}$$

Written as Gaussian, the next state probability is approximated as follows:

$$\begin{aligned} & p(x_t \mid u_t, x_{t-1}) \\ & \quad \approx \det (2\pi R_t)^{-\frac{1}{2}} \exp \left\{ -\tfrac{1}{2} \, [x_t - g(u_t, \mu_{t-1}) - G_t \, (x_{t-1} - \mu_{t-1})]^T \right. \\ & \qquad \left. R_t^{-1} \, [x_t - g(u_t, \mu_{t-1}) - G_t \, (x_{t-1} - \mu_{t-1})] \right\} \end{aligned} \tag{3.51}$$

Notice that G_t is a matrix of size $n \times n$, with n denoting the dimension of the state. This matrix is often called the *Jacobian*. The value of the Jacobian depends on u_t and μ_{t-1}, hence it differs for different points in time.

EKFs implement the exact same linearization for the measurement function h. Here the Taylor expansion is developed around $\bar{\mu}_t$, the state deemed most likely by the robot at the time it linearizes h:

$$\begin{aligned} h(x_t) &\approx h(\bar{\mu}_t) + \underbrace{h'(\bar{\mu}_t)}_{=: H_t} (x_t - \bar{\mu}_t) \\ &= h(\bar{\mu}_t) + H_t \, (x_t - \bar{\mu}_t) \end{aligned} \tag{3.52}$$

with $h'(x_t) = \frac{\partial h(x_t)}{\partial x_t}$. Written as a Gaussian, we have

$$\begin{aligned} p(z_t \mid x_t) = & \det (2\pi Q_t)^{-\frac{1}{2}} \exp \left\{ -\tfrac{1}{2} \, [z_t - h(\bar{\mu}_t) - H_t \, (x_t - \bar{\mu}_t)]^T \right. \\ & \left. Q_t^{-1} \, [z_t - h(\bar{\mu}_t) - H_t \, (x_t - \bar{\mu}_t)] \right\} \end{aligned} \tag{3.53}$$

3.3.2 The EKF Algorithm

Table 3.3 states the EKF algorithm. In many ways, this algorithm is similar to the Kalman filter algorithm stated in Table 3.1. The most important differences are summarized by the following table:

1: **Algorithm Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):**
2: $\bar{\mu}_t = g(u_t, \mu_{t-1})$
3: $\bar{\Sigma}_t = G_t \; \Sigma_{t-1} \; G_t^T + R_t$
4: $K_t = \bar{\Sigma}_t \; H_t^T (H_t \; \bar{\Sigma}_t \; H_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
6: $\Sigma_t = (I - K_t \; H_t) \; \bar{\Sigma}_t$
7: return μ_t, Σ_t

Table 3.3 The extended Kalman filter (EKF) algorithm.

	Kalman filter	EKF
state prediction (Line 2)	$A_t \; \mu_{t-1} + B_t \; u_t$	$g(u_t, \mu_{t-1})$
measurement prediction (Line 5)	$C_t \; \bar{\mu}_t$	$h(\bar{\mu}_t)$

That is, the linear predictions in Kalman filters are replaced by their nonlinear generalizations in EKFs. Moreover, EKFs use Jacobians G_t and H_t instead of the corresponding linear system matrices A_t, B_t, and C_t in Kalman filters. The Jacobian G_t corresponds to the matrices A_t and B_t, and the Jacobian H_t corresponds to C_t. A detailed example for extended Kalman filters will be given in Chapter **??**.

3.3.3 Mathematical Derivation of the EKF

The mathematical derivation of the EKF parallels that of the Kalman filter in Section 3.2.4, and hence shall only be sketched here. The prediction is calculated as follows (cf. (3.7)):

$$\overline{bel}(x_t) \quad = \quad \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\sim \mathcal{N}(x_t; g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}), R_t)} \quad \underbrace{bel(x_{t-1})}_{\sim \mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})} \quad dx_{t-1} \tag{3.54}$$

This distribution is the EKF analog of the prediction distribution in the Kalman filter, stated in (3.7). The Gaussian $p(x_t \mid x_{t-1}, u_t)$ can be found in Equation (3.51). The

function L_t is given by (cf. (3.10))

$$\begin{aligned} L_t &= \tfrac{1}{2}\,(x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1}))^T \\ &\quad R_t^{-1}\,(x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})) \\ &\quad + \tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\,\Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \end{aligned} \tag{3.55}$$

which is quadratic in both x_{t-1} and x_t, as above. As in (3.11), we decompose L_t into $L_t(x_{t-1}, x_t)$ and $L_t(x_t)$:

$$\begin{aligned} &L_t(x_{t-1}, x_t) \\ &\quad = \tfrac{1}{2}\,(x_{t-1} - \Phi_t\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t, \mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}])^T\,\Phi^{-1} \\ &\qquad (x_{t-1} - \Phi_t\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t, \mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}]) \end{aligned} \tag{3.56}$$

with

$$\Phi_t = (G_t^T\,R_t^{-1}\,G_t + \Sigma_{t-1}^{-1})^{-1} \tag{3.57}$$

and hence

$$\begin{aligned} L_t(x_t) &= \tfrac{1}{2}\,(x_t - g(u_t, \mu_{t-1}) + G_t\mu_{t-1})^T\,R_t^{-1}\,(x_t - g(u_t, \mu_{t-1}) + G_t\mu_{t-1}) \\ &\quad + \tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\,\Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \\ &\quad - \tfrac{1}{2}\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t, \mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}]^T \\ &\quad \Phi_t\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t, \mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}] \end{aligned} \tag{3.58}$$

As the reader easily verifies, setting the first derivative of $L_t(x_t)$ to zero gives us the update $\mu_t = g(u_t, \mu_{t-1})$, in analogy to the derivation in Equations (3.26) through (3.30). The second derivative is given by $(R_t + G_t\,\Sigma_{t-1}\,G_t^T)^{-1}$ (see (3.31)).

The measurement update is also derived analogously to the Kalman filter in Section 3.2.4. In analogy to (3.32), we have for the EKF

$$bel(x_t) = \eta \underbrace{p(z_t \mid x_t)}_{\sim\mathcal{N}(z_t; h(\bar\mu_t) + H_t\,(x_t - \bar\mu_t), Q_t)} \underbrace{\overline{bel}(x_t)}_{\sim\mathcal{N}(x_t; \bar\mu_t, \bar\Sigma_t)} \tag{3.59}$$

using the linearized next state transition function from (3.52). This leads to the exponent (see (3.34)):

$$\begin{aligned} J_t &= \tfrac{1}{2}\,(z_t - h(\bar{\mu}_t) - H_t\,(x_t - \bar{\mu}_t))^T\,Q_t^{-1}\,(z_t - h(\bar{\mu}_t) - H_t\,(x_t - \bar{\mu}_t)) \\ &\quad + \tfrac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t) \end{aligned} \tag{3.60}$$

The resulting mean and covariance is given by

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \tag{3.61}$$

$$\Sigma_t = (I - K_t\,H_t)\,\bar{\Sigma}_t \tag{3.62}$$

with the Kalman gain

$$K_t = \bar{\Sigma}_t\,H_t^T\,(H_t\,\bar{\Sigma}_{t-1}\,H_t^T + Q_t)^{-1} \tag{3.63}$$

The derivation of these equations is analogous to Equations (3.35) through (3.46).

3.3.4 Practical Considerations

The EKF has become just about the most popular tool for state estimation in robotics. Its strength lies in its simplicity and in its computational efficiency. As was the case for the Kalman filter, each update requires time $O(k^{2.8} + n^2)$, where k is the dimension of the measurement vector z_t, and n is the dimension of the state vector x_t. Other algorithms, such as the particle filter discussed further below, may require time exponential in n.

The EKF owes its computational efficiency to the fact that it represents the belief by a multivariate Gaussian distribution. A Gaussian is a unimodal distribution, which can be thought of as a single guess, annotated with an uncertainty ellipse. In many practical problems, Gaussians are robust estimators. Applications of the Kalman filter to state spaces with 1,000 dimensions or more will be discussed in later chapters of this book. EKFs have been applied with great success to a number of state estimation problems that violate the underlying assumptions.

Sometimes, one might want to pursue multiple distinct hypotheses. For example, a robot might have two distinct hypotheses as to where it is, but the arithmetic mean of

these hypotheses is not a likely contender. Such situations require multi-modal representations for the posterior belief. EKFs, in the form described here, are incapable of representing such multimodal beliefs. A common extension of EKFs is to represent posteriors using *mixtures*, or *sums*, of Gaussians. A mixture of J Gaussians may be of the form (cf. (**??**)):

$$bel(x) \quad = \quad \sum_j a_j \ \det\left(2\pi\Sigma_{j,t}\right)^{-\frac{1}{2}} \ \exp\left\{-\tfrac{1}{2}(x_t - \mu_{j,t})^T \ \Sigma_{j,t}^{-1} \ (x_t - \mu_{j,t})\right\} \tag{3.64}$$

where a_j are mixture parameters with $a_j \geq 0$ and $\sum_j a_j = 1$. EKFs that utilize such mixture representations are called *multi-hypothesis (extended) Kalman filters*, of MHEKF.

An important limitation of the EKF arises from the fact that it approximates state transitions and measurements using linear Taylor expansions. In virtually all robotics problems, these functions are nonlinear. The goodness of this approximation depends on two main factors. First, it depends on the degree of nonlinearity of the functions that are being approximated. If these functions are approximately linear, the EKF approximation may generally be a good one, and EKFs may approximate the posterior belief with sufficient accuracy. However, sometimes, the functions are not only nonlinear, but are also multi-modal, in which case the linearization may be a poor approximation. The goodness of the linearization also depends on the degree of uncertainty. The less certain the robot, the wider its Gaussian belief, and the more it is affected by nonlinearities in the state transition and measurement functions. In practice, when applying EKFs it is therefore important to keep the uncertainty of the state estimate small.

We also note that Taylor series expansion is only one way to linearize. Two other approaches have often been found to yield superior results. One is the *unscented Kalman filter*, which probes the function to be linearized at selected points and calculates a linearized approximation based on the outcomes of these probes. Another is known as *moments matching*, in which the linearization is calculated in a way that preserves the true mean and the true covariance of the posterior distribution (which is not the case for EKFs). Both techniques are relatively recent but appear to be superior to the EKF linearization.

3.4 THE INFORMATION FILTER

The dual of the Kalman filter is the information filter. Just like the KF and its nonlinear version, the EKF, the information filter (IF) represents the belief by a Gaussian. Thus, the standard information filter is subject to the same assumptions underlying the Kalman filter. The key difference between the KF and the IF arises from the way the Gaussian belief is represented. Whereas in the Kalman filter family of algorithms, Gaussians are represented by their moments (mean, covariance), information filters represent Gaussians in their canonical representation, which is comprised of an information matrix and an information vector. The difference in representation leads to different update equations. In particular, what is computationally complex in one representation happens to be simple in the other (and vice versa). The canonical and the moments representations are often considered *dual* to each other, and thus are the IF and the KF.

3.4.1 Canonical Representation

The canonical representation of a multivariate Gaussian is given by a matrix Ω and a vector ξ. The matrix Ω is the inverse of the covariance matrix:

$$\Omega = \Sigma^{-1} . \tag{3.65}$$

Ω is called the *information matrix*, or sometimes the *precision matrix*. The vector ξ is called the *information vector*. It is defined as

$$\xi = \Sigma^{-1} \mu . \tag{3.66}$$

It is easy to see that Ω and ξ are a complete parameterization of a Gaussian. In particular, the mean and covariance of the Gaussian can easily be obtained from the canonical representation by the inverse of (3.65) and (3.66):

$$\Sigma = \Omega^{-1} \tag{3.67}$$

$$\mu = \Omega^{-1} \xi \tag{3.68}$$

The canonical representation is often derived by multiplying out the exponent of a Gaussian. In (3.1), we defined the multivariate normal distribution as follows:

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{3.69}$$

A straightforward sequence of transformations leads to the following parameterization:

$$\begin{aligned} p(x) &= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \tfrac{1}{2}\mu^T\Sigma^{-1}\mu\right\} \\ &= \underbrace{\det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}\mu^T\Sigma^{-1}\mu\right\}}_{\text{const.}} \exp\left\{-\tfrac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\} \end{aligned} \tag{3.70}$$

The term labeled "const." does not depend on the target variable x. Hence, it can be subsumed into the normalizer η.

$$p(x) = \eta \exp\left\{-\tfrac{1}{2}x^T\ \Sigma^{-1}\ x + x^T\ \Sigma^{-1}\ \mu\right\} \tag{3.71}$$

This form motivates the parameterization of a Gaussian by its canonical parameters Ω and ξ.

$$p(x) = \eta \exp\left\{-\tfrac{1}{2}x^T\ \Omega\ x + x^T\ \xi\right\} \tag{3.72}$$

In many ways, the canonical representation is more elegant than the moments representation. In particular, the negative logarithm of the Gaussian (which plays an essential role in information theory) is a quadratic function in the canonical parameters Ω and ξ:

$$-\log p(x) = \text{const.} + \tfrac{1}{2}x^T\ \Omega\ x - x^T\ \xi \tag{3.73}$$

Here "const." is a constant. The reader may notice that we cannot use the symbol η to denote this constant, since negative logarithms of probabilities do not normalize to 1. The negative logarithm of our distribution $p(x)$ is quadratic in x, with the quadratic term parameterized by Ω and the linear term by ξ. In fact, for Gaussians, Ω must

1: **Algorithm Information_filter($\xi_{t-1}, \Omega_{t-1}, u_t, z_t$):**
2: $\bar{\Omega}_t = (A_t\ \Omega_{t-1}^{-1}\ A_t^T + R_t)^{-1}$
3: $\bar{\xi}_t = \bar{\Omega}_t(A_t\ \Omega_{t-1}^{-1}\ \xi_{t-1} + B_t\ u_t)$
4: $\Omega_t = C_t^T\ Q_t^{-1}\ C_t + \bar{\Omega}_t$
5: $\xi_t = C_t^T\ Q_t^{-1}\ z_t + \bar{\xi}_t$
6: return ξ_t, Ω_t

Table 3.4 The information filter (IF) algorithm.

be positive semidefinite, hence $-\log p(x)$ is a quadratic distance function with mean $\mu = \Omega^{-1}\ \xi$. This is easily verified by setting the first derivative of (3.73) to zero:

$$\frac{\partial[-\log p(x)]}{\partial x} \;=\; 0 \;\iff\; \Omega x - \xi \;=\; 0 \;\iff\; x = \Omega^{-1}\xi \tag{3.74}$$

The matrix Ω determines the rate at which the distance function increases in the different dimensions of the variable x. A quadratic distance that is weighted by a matrix Ω is called *Mahalanobis distance*.

3.4.2 The Information Filter Algorithm

Table 3.4 states the update algorithm known as information filter. Its input is a Gaussian in its canonical representation ξ_{t-1} and Ω_{t-1}, representing the belief at time $t-1$. Just like all Bayes filters, its input includes the control u_t and the measurement z_t. The output are the parameters ξ_t and Ω_t of the updated Gaussian.

The update involves matrices A_t, B_t, C_t, R_t, and Q_t. Those were defined in Section 3.2. The information filter assumes that the state transition and measurement probabilities are governed by the following linear Gaussian equations, originally defined in (3.2) and (3.5):

$$x_t \;=\; A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{3.75}$$

$$z_t \;=\; C_t x_t + \delta_t \tag{3.76}$$

Here R_t and Q_t are the covariances of the zero-mean noise variables ε_t and δ_t, respectively.

Just like the Kalman filter, the information filter is updated in two steps, a prediction step and a measurement update step. The prediction step is implemented in Lines 2 and 3 in Table 3.4. The parameters $\bar{\xi}_t$ and $\bar{\Omega}_t$ describe the Gaussian belief over x_t after incorporating the control u_t, but before incorporating the measurement z_t. The latter is done through Lines 4 and 5. Here the belief is updated based on the measurement z_t.

These two update steps can be vastly different in complexity, especially if the state space possesses many dimensions. The prediction step, as stated in Table 3.4, involves the inversion of two matrices of the size $n \times n$, where n is the dimension of the state space. This inversion requires approximately $O(n^{2.8})$ time. In Kalman filters, the update step is additive and requires at most $O(n^2)$ time; it requires less time if only a subset of variables is affected by a control, or if variables transition independently of each other. These roles are reversed for the measurement update step. Measurement updates are additive in the information filter. They require at most $O(n^2)$ time, and they are even more efficient if measurements carry only information about a subset of all state variables at a time. The measurement update is the difficult step in Kalman filters. It requires matrix inversion whose worst case complexity is $O(n^{2.8})$. This illustrates the dual character of Kalman and information filters.

3.4.3 Mathematical Derivation of the Information Filter

The derivation of the information filter is analogous to that of the Kalman filter. To derive the prediction step (Lines 2 and 3 in Table 3.4), we begin with the corresponding update equations of the Kalman filters, which can be found in Lines 2 and 3 of the algorithm in Table 3.1 and are restated here for the reader's convenience:

$$\bar{\mu}_t = A_t \, \mu_{t-1} + B_t \, u_t \tag{3.77}$$

$$\bar{\Sigma}_t = A_t \, \Sigma_{t-1} \, A_t^T + R_t \tag{3.78}$$

The information filter prediction step follows now directly by substituting the moments μ and Σ by the canonical parameters ξ and Ω according to their definitions in

(3.67) and (3.72):

$$\begin{aligned} \mu_{t-1} &= \Omega_{t-1}^{-1}\ \xi_{t-1} \\ \Sigma_{t-1} &= \Omega_{t-1}^{-1} \end{aligned} \tag{3.79}$$

Substituting these expressions in (3.77) and (3.78) gives us the set of prediction equations

$$\bar{\Omega}_t = (A_t\ \Omega_{t-1}^{-1}\ A_t^T + R_t)^{-1} \tag{3.80}$$

$$\bar{\xi}_t = \bar{\Omega}_t(A_t\ \Omega_{t-1}^{-1}\ \xi_{t-1} + B_t\ u_t) \tag{3.81}$$

These equations are identical to those in Table 3.4. As is easily seen, the prediction step involves two nested inversions of a potentially large matrix. These nested inversions can be avoided when only a small number of state variables is affected by the motion update, a topic which will be discussed later in this book.

The derivation of the measurement update is even simpler. We begin with the Gaussian of the belief at time t, which was provided in Equation (3.34) and is restated here once again:

$$\begin{aligned} & bel(x_t) \\ &= \eta\ \exp\left\{-\tfrac{1}{2}\ (z_t - C_t x_t)^T\ Q_t^{-1}\ (z_t - C_t x_t) - \tfrac{1}{2}\ (x_t - \bar{\mu}_t)^T\ \bar{\Sigma}_t^{-1}\ (x_t - \bar{\mu}_t)\right\} \end{aligned} \tag{3.82}$$

For Gaussians represented in their canonical form this distribution is given by

$$\begin{aligned} & bel(x_t) \\ &= \eta\ \exp\left\{-\tfrac{1}{2}\ x_t^T\ C_t^T\ Q_t^{-1}\ C_t\ x_t + x_t^T\ C_t^T\ Q_t^{-1}\ z_t - \tfrac{1}{2}\ x_t^T\ \bar{\Omega}_t x_t + x_t^T \bar{\xi}_t\right\} \end{aligned} \tag{3.83}$$

which by reordering the terms in the exponent resolves to

$$bel(x_t) = \eta\ \exp\left\{-\tfrac{1}{2}\ x_t^T\ [C_t^T\ Q_t^{-1}\ C_t + \bar{\Omega}_t]\ x_t + x_t^T\ [C_t^T\ Q_t^{-1}\ z_t + \bar{\xi}_t]\right\}$$

We can now read off the measurement update equations, by collecting the terms in the squared brackets:

$$\xi_t = C_t^T\ Q_t^{-1}\ z_t + \bar{\xi}_t \tag{3.84}$$

1: **Algorithm Extended_information_filter($\xi_{t-1}, \Omega_{t-1}, u_t, z_t$):**
2: $\mu_{t-1} = \Omega_{t-1}^{-1}\ \xi_{t-1}$
3: $\bar{\Omega}_t = (G_t\ \Omega_{t-1}^{-1}\ G_t^T + R_t)^{-1}$
4: $\bar{\xi}_t = \bar{\Omega}_t\ g(u_t, \mu_{t-1})$
5: $\bar{\mu}_t = g(u_t, \mu_{t-1})$
6: $\Omega_t = \bar{\Omega}_t + H_t^T\ Q_t^{-1}\ H_t$
7: $\xi_t = \bar{\xi}_t + H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) - H_t\ \bar{\mu}_t]$
8: return ξ_t, Ω_t

Table 3.5 The extended information filter (EIF) algorithm.

$$\Omega_t = C_t^T\ Q_t^{-1}\ C_t + \bar{\Omega}_t \tag{3.85}$$

These equations are identical to the measurement update equations in Lines 4 and 5 of Table 3.4.

3.4.4 The Extended Information Filter Algorithm

The extended version of the information filter is analog to the EKF. Table 3.5 depicts the algorithm. The prediction is realized in Lines 2 through 4, and the measurement update in Lines 5 through 7. These update equations are largely analog to the linear information filter, with the functions g and h (and their Jacobian G_t and H_t) replacing the parameters of the linear model A_t, B_t, and C_t. As before, g and h specify the nonlinear next state function and measurement function, respectively. Those were defined in (3.47) and (3.48) and are restated here:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{3.86}$$
$$z_t = h(x_t) + \delta_t\ . \tag{3.87}$$

Unfortunately, both g and h require a state as an input. This mandates the recovery of a state estimate μ from the canonical parameters. The recovery takes place in Line 2, in which the state μ_{t-1} is calculated from Ω_{t-1} and ξ_{t-1} in the obvious way. Line 5 computes the state $\bar{\mu}_t$ using the equation familiar from the EKF (Line 2 in Table 3.3).

The necessity to recover the state estimate seems at odds with the desire to represent the filter using its canonical parameters. We will revisit this topic when discussing the use of extended information filters in the context of robotic mapping.

3.4.5 Mathematical Derivation of the Extended Information Filter

The extended information filter is easily derived by essentially performing the same linearization that led to the extended Kalman filter above. As in (3.50) and (3.52), EIFs approximate g and h by a Taylor expansion:

$$\begin{aligned} g(u_t, x_{t-1}) &\approx g(u_t, \mu_{t-1}) + G_t\,(x_{t-1} - \mu_{t-1}) && (3.88)\\ h(x_t) &\approx h(\bar{\mu}_t) + H_t\,(x_t - \bar{\mu}_t) && (3.89) \end{aligned}$$

Here G_t and H_t are the Jacobians of g and h at μ_{t-1} and $\bar{\mu}_t$, respectively:

$$\begin{aligned} G_t &= g'(u_t, \mu_{t-1}) && (3.90)\\ H_t &= h'(\bar{\mu}_t) && (3.91) \end{aligned}$$

These definitions are equivalent to those in the EKF. The prediction step is now derived from Lines 2 and 3 of the EKF algorithm (Table 3.3), which are restated here:

$$\begin{aligned} \bar{\Sigma}_t &= G_t\,\Sigma_{t-1}\,G_t^T + R_t && (3.92)\\ \bar{\mu}_t &= g(u_t, \mu_{t-1}) && (3.93) \end{aligned}$$

Substituting Σ_{t-1} by Ω_{t-1}^{-1} and $\bar{\mu}_t$ by $\bar{\Omega}_t^{-1}\bar{\xi}_t$ gives us the prediction equations of the extended information filter:

$$\begin{aligned} \bar{\Omega}_t &= (G_t\,\Omega_{t-1}^{-1}\,G_t^T + R_t)^{-1} && (3.94)\\ \bar{\xi}_t &= \bar{\Omega}_t\,g(u_t, \Omega_{t-1}^{-1}\,\xi_{t-1}) && (3.95) \end{aligned}$$

The measurement update is derived from Equations (3.59) and (3.60). In particular, (3.60) defines the following Gaussian posterior:

$$bel(x_t) = \eta\,\exp\big\{-\tfrac{1}{2}\,(z_t - h(\bar{\mu}_t) - H_t\,(x_t - \bar{\mu}_t))^T\,Q_t^{-1}$$

$$(z_t - h(\bar{\mu}_t) - H_t\ (x_t - \bar{\mu}_t)) - \tfrac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\Big\} \quad (3.96)$$

Multiplying out the exponent and reordering the terms gives us the following expression for the posterior:

$$\begin{aligned} bel(x_t) &= \eta\ \exp\big\{-\tfrac{1}{2}\ x_t^T\ H_t^T\ Q_t^{-1}\ H_t\ x_t + x_t^T\ H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) - H_t\ \bar{\mu}_t] \\ &\quad -\tfrac{1}{2} x_t^T\ \bar{\Sigma}_t^{-1}\ x_t + x_t^T\ \bar{\Sigma}_t^{-1}\ \bar{\mu}_t\big\} \\ &= \eta\ \exp\big\{-\tfrac{1}{2}\ x_t^T\ \big[H_t^T\ Q_t^{-1}\ H_t + \bar{\Sigma}_t^{-1}\big]\ x_t \\ &\quad + x_t^T\ \big[H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) - H_t\ \bar{\mu}_t] + \ \bar{\Sigma}_t^{-1}\ \bar{\mu}_t\big] \end{aligned} \quad (3.97)$$

With $\bar{\Sigma}_t^{-1} = \bar{\Omega}_t$ this expression resolves to the following information form:

$$\begin{aligned} bel(x_t) &= \eta\ \exp\big\{-\tfrac{1}{2}\ x_t^T\ \big[H_t^T\ Q_t^{-1}\ H_t + \bar{\Omega}_t\big]\ x_t \\ &\quad + x_t^T\ \big[H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) - H_t\ \bar{\mu}_t] + \bar{\xi}_t\big] \end{aligned} \quad (3.98)$$

We can now read off the measurement update equations by collecting the terms in the squared brackets:

$$\Omega_t = \bar{\Omega}_t + H_t^T\ Q_t^{-1}\ H_t \quad (3.99)$$

$$\xi_t = \bar{\xi}_t + H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) - H_t\ \bar{\mu}_t] \quad (3.100)$$

3.4.6 Practical Considerations

When applied to robotics problems, the information filter possesses several advantages over the Kalman filter. For example, representing global uncertainty is simple in the information filter: simply set $\Omega = 0$. When using moments, such global uncertainty amounts to a covariance of infinite magnitude. This is especially problematic when sensor measurements carry information about a strict subset of all state variables, a situation often encountered in robotics. Special provisions have to be made to handle such situations in EKFs. Furthermore, the information filter tends to be numerically more stable than the Kalman filter in many of the applications discussed later in this book.

Another advantage of the information filter over the Kalman filter arises from its natural fit for multi-robot problems. Multi-robot problems often involve the integration

of sensor data collected decentrally. Such integration is commonly performed through Bayes rule. When represented in logarithmic form, Bayes rule becomes an addition. As noted above, the canonical parameters of information filters represent a probability in logarithmic form. Thus, information integration is achieved by summing up information from multiple robots. Addition is commutative. Because of this, information filters can often integrate information in arbitrary order, with arbitrary delays, and in a completely decentralized manner. While the same is possible using the moments representation—after all, they represent the same information—the necessary overhead for doing so is much higher. Despite this advantage, the use of information filters in multi-robot systems remains largely under-explored.

These advantages of the information filter are offset by important limitations. A primary disadvantage of the EIF is the need to recover a state estimate in the update step, when applied to nonlinear systems. This step, if implemented as stated here, requires the inversion of the information matrix. Further matrix inversions are required for the prediction step of the information filters. In many robotics problems, the EKF does not involve the inversion of matrices of comparable size. For high dimensional state spaces, the information filter is generally believed to be computationally inferior to the Kalman filter. In fact, this is one of the reasons why the EKF has been vastly more popular than the EIF.

As we will see later in this book, these limitations do not necessarily apply to problems in which the information matrix possess structure. In many robotics problems, the interaction of state variables is local; as a result, the information matrix may be sparse. Such sparseness does *not* translate to sparseness of the covariance.

Information filters can be thought of as graphs, where states are connected whenever the corresponding off-diagonal element in the information matrix is non-zero. Sparse information matrices correspond to sparse graphs; in fact, such graphs are commonly known as Gaussian Markov random fields. A flurry of algorithms exist to perform the basic update and estimation equations efficiently for such fields, under names like "loopy belief propagation." In this book, we will encounter a mapping problem in which the information matrix is (approximately) sparse, and develop an extended information filter that is significantly more efficient than both Kalman filters and non-sparse information filters.

3.5 SUMMARY

In this section, we introduced efficient Bayes filter algorithms that represent the posterior by multivariate Gaussians. We noted that

- Gaussians can be represented in two different ways: The moments representation and the canonical representation. The moments representation consists of the mean (first moment) and the covariance (second moment) of the Gaussian. The canonical, or natural, representation consists of an information matrix and an information vector. Both representations are duals of each other, and each can be recovered from the other via matrix inversion.

- Bayes filters can be implemented for both representations. When using the moments representation, the resulting filter is called Kalman filter. The dual of the Kalman filter is the information filter, which represents the posterior in the canonical representation. Updating a Kalman filter based on a control is computationally simple, whereas incorporating a measurement is more difficult. The opposite is the case for the information filter, where incorporating a measurement is simple, but updating the filter based on a control is difficult.

- For both filters to calculate the correct posterior, three assumptions have to be fulfilled. First, the initial belief must be Gaussian. Second, the state transition probability must be composed of a function that is linear in its argument with added independent Gaussian noise. Third, the same applies to the measurement probability. It must also be linear in its argument, with added Gaussian noise. Systems that meet these assumptions are called linear Gaussian systems.

- Both filters can be extended to nonlinear problems. The technique described in this chapter calculates a tangent to the nonlinear function. Tangents are linear, making the filters applicable. The technique for finding a tangent is called Taylor expansion. Performing a Taylor expansion involves calculating the first derivative of the target function, and evaluating it at a specific point. The result of this operation is a matrix known as the Jacobian. The resulting filters are called "extended."

- The accuracy of Taylor series expansions depends on two factors: The degree of nonlinearity in the system, and the width of the posterior. Extended filters tend to yield good results if the state of the system is known with relatively high accuracy, so that the remaining covariance is small. The larger the uncertainty, the higher the error introduced by the linearization.

- One of the primary advantages of Gaussian filters is computational: The update requires time polynomial in the dimensionality of the state space. This is not

the case of some of the techniques described in the next chapter. The primary disadvantage is their confinement to unimodal Gaussian distributions.

- Within the multivariate Gaussian regime, both filters, the Kalman filter and the information filter, have orthogonal strengths and weaknesses. However, the Kalman filter and its nonlinear extension, the extended Kalman filter, are vastly more popular than the information filter.

The selection of the material in this chapter is based on today's most popular techniques in robotics. There exists a huge number of variations and extensions of the Gaussian filters presented here, that address the various limitations and shortcomings. One of the most apparent limitations of the material presented thus far is the fact that the posterior is represented by a single Gaussian. This confines these filters to situations where the posterior can be described by a unimodal distribution. This is often appropriate in tracking applications, where a robot tracks a state variable with limited uncertainty. When uncertainty grows more global, a single mode can be insufficient, and Gaussians become too crude an approximation to the true posterior belief. This limitation has been well recognized, and even within the Gaussian paradigm extensions exist that can represent multimodal beliefs, e.g., using mixtures of Gaussians. Popular non-Gaussian approaches are described in the next chapter.

3.6 BIBLIOGRAPHICAL REMARKS

Inversion lemma: G.H. Golub, C.F. Van Loan, Matrix Computations, North Oxford Academic, 1986.