is uncertain. Based on this uncertainty, we model our initial belief by a Gaussian with variance $\sigma^2_{\text{init}} = 900m^2$.

To find out more about our location, we query a GPS receiver. The GPS tells us our location is $z_{\text{GPS}} = 1,100m$. This GPS receiver is known to have an error variance of $\sigma^2_{\text{init}} = 100m^2$.

(a) Write the probability density functions of the prior $p(x)$ and the measurement $p(z \mid x)$.

(b) Using Bayes rule, what is the posterior $p(x \mid z)$? Can you prove it to be Gaussian?

(c) How likely was the measurement $x_{\text{GPS}} = 1,100m$ given our prior, and knowledge of the error probability of our GPS receiver?

Hint: This is an exercise in manipulating quadratic expressions.

5. Derive Equations (2.18) and (2.19) from (2.17) and the laws of probability stated in the text.

6. Prove Equation (2.25). What are the implications of this equality?

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:

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

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; 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

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.

GAUSSIAN POSTERIOR

The Kalman filter represents beliefs by the moments parameterization: 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 state transition 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:

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

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

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 ε_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 $p(x_t \mid u_t, x_{t-1})$. 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:

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

```
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 $p(z_t \mid x_t)$ must also be *linear* in its arguments, with added Gaussian noise:

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

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:

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

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

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

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 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.

KALMAN GAIN

INNOVATION

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, 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_tC_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.

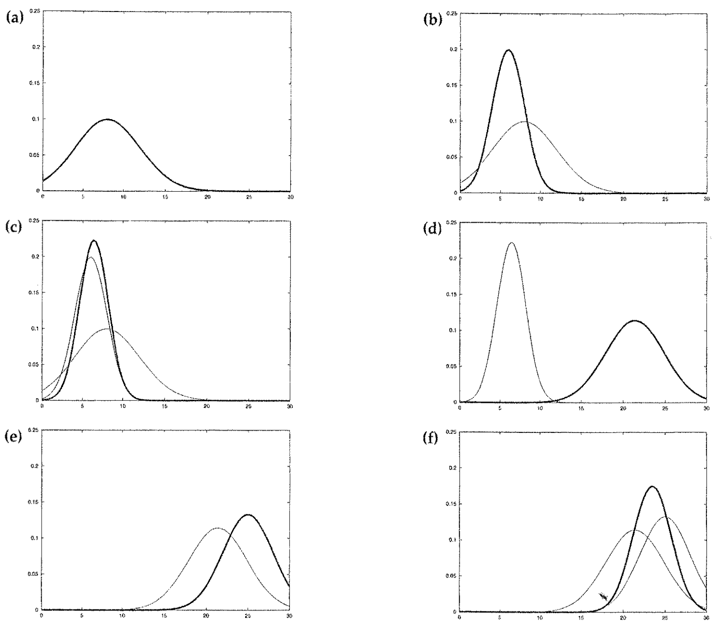

Figure 3.2 Illustration of Kalman filters: (a) initial belief, (b) a measurement (in bold) with the associated uncertainty, (c) belief after integrating the measurement into the belief using the Kalman filter algorithm, (d) belief after motion to the right (which introduces uncertainty), (e) a new measurement 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 $\overline{bel}(x_t)$ is calculated from the belief one time step earlier, $bel(x_{t-1})$. Lines 2 and 3 implement the update step described in Equation (2.41), restated here for the reader's convenience:

$$\overline{bel}(x_t) \;=\; \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\sim\mathcal{N}(x_t; A_t x_{t-1} + B_t u_t, R_t)} \quad \underbrace{bel(x_{t-1})}_{\sim\mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})} \quad dx_{t-1} \tag{3.8}$$

The 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 (3.8) is again a Gaussian with mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$ as stated in Table 3.1.

We begin by writing (3.8) in its Gaussian form:

$$\begin{aligned} &\overline{bel}(x_t) \\ &\quad = \eta \int \exp\left\{-\tfrac{1}{2}\,(x_t - A_t\, x_{t-1} - B_t\, u_t)^T\, R_t^{-1}\,(x_t - A_t\, x_{t-1} - B_t\, u_t)\right\} \\ &\qquad\qquad \exp\left\{-\tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})\right\}\; dx_{t-1} \end{aligned} \tag{3.9}$$

In short, we have

$$\overline{bel}(x_t) \;=\; \eta \int \exp\{-L_t\}\; dx_{t-1} \tag{3.10}$$

with

$$\begin{aligned} L_t \;=\;& \tfrac{1}{2}\,(x_t - A_t\, x_{t-1} - B_t\, u_t)^T\, R_t^{-1}\,(x_t - A_t\, x_{t-1} - B_t\, u_t) \\ &+ \tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\, \Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \end{aligned} \tag{3.11}$$

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

Expression (3.10) contains an integral. Solving this integral requires us to reorder the terms in this interval, in a way that might appear counterintuitive at first. In particular, we will decompose L_t into two functions, $L_t(x_{t-1}, x_t)$ and $L_t(x_t)$:

$$L_t \;=\; L_t(x_{t-1}, x_t) + L_t(x_t) \tag{3.12}$$

This decomposition will simply be the result of reordering the terms in L_t. A key goal of this decomposition step shall be that the variables in L_t are partitioned into two sets, of which only one will depend on the variable x_{t-1}. The other, $L_t(x_t)$, will not depend on x_{t-1}. As a result, we will be able to move the latter variables out of the integral over the variable x_{t-1}.

This is illustrated by the following transformation:

$$\begin{aligned} \overline{bel}(x_t) \;&=\; \eta \int \exp\{-L_t\}\; dx_{t-1} \\ &=\; \eta \int \exp\{-L_t(x_{t-1}, x_t) - L_t(x_t)\}\; dx_{t-1} \\ &=\; \eta\; \exp\{-L_t(x_t)\} \int \exp\{-L_t(x_{t-1}, x_t)\}\; dx_{t-1} \end{aligned} \tag{3.13}$$

Of course, there exist many ways to decompose L_t into two sets that would meet this criterion. The key insight is that we will choose $L_t(x_{t-1}, x_t)$ such that the value of the integral in (3.13) does not depend on x_t. If we succeed in defining such a function $L_t(x_{t-1}, x_t)$, the entire integral over $L_t(x_{t-1}, x_t)$ will simply become a constant relative to the problem of estimating the belief distribution over x_t. Constants are usually captured in the normalization constant η, so under our decomposition we will be able to subsume this constant into η (now for a different actual value of η as above):

$$\overline{bel}(x_t) \;=\; \eta\; \exp\{-L_t(x_t)\} \tag{3.14}$$

Thus, our decomposition would make it possible to eliminate the integral from the belief (3.10). The result is just a normalized exponential over a quadratic function, which turns out to be a Gaussian.

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:

$$\frac{\partial L_t}{\partial x_{t-1}} \;=\; -A_t^T\, R_t^{-1}\,(x_t - A_t\, x_{t-1} - B_t\, u_t) + \Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \tag{3.15}$$

$$\frac{\partial^2 L_t}{\partial x_{t-1}^2} \;=\; A_t^T\, R_t^{-1}\, A_t + \Sigma_{t-1}^{-1} \;=:\; \Psi_t^{-1} \tag{3.16}$$

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:

$$A_t^T\, R_t^{-1}\,(x_t - A_t\, x_{t-1} - B_t\, u_t) \;=\; \Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \tag{3.17}$$

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

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

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

$$(3.19)\quad L_t(x_{t-1}, x_t) = \tfrac{1}{2}(x_{t-1} - \Psi_t\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}])^T\,\Psi^{-1}\\ (x_{t-1} - \Psi_t\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}])$$

Clearly, this is not the only quadratic function satisfying our decomposition in (3.12). 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

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

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

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

From this it follows that

$$(3.22)\quad \int \exp\{-L_t(x_{t-1}, x_t)\}\ dx_{t-1} = \det(2\pi\Psi)^{\frac{1}{2}}$$

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 (3.13):

$$(3.23)\quad \overline{bel}(x_t) = \eta\ \exp\{-L_t(x_t)\} \int \exp\{-L_t(x_{t-1}, x_t)\}\ dx_{t-1}\\ = \eta\ \exp\{-L_t(x_t)\}$$

This decomposition establishes the correctness of (3.14). Notice once again that the normalizers η are *not* the same in both lines.

It remains to determine the function $L_t(x_t)$, which is the difference of L_t, defined in (3.11), and $L_t(x_{t-1}, x_t)$, defined in (3.19):

$$(3.24)\quad L_t(x_t) = L_t - L_t(x_{t-1}, x_t)\\ = \tfrac{1}{2}\,(x_t - A_t\,x_{t-1} - B_t\,u_t)^T\,R_t^{-1}\,(x_t - A_t\,x_{t-1} - B_t\,u_t)\\ + \tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\,\Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1})\\ - \tfrac{1}{2}(x_{t-1} - \Psi_t\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}])^T\,\Psi^{-1}\\ (x_{t-1} - \Psi_t\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}])$$

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}).

$$(3.25)\quad L_t(x_t) = \underline{\underline{\tfrac{1}{2}\,x_{t-1}^T A_t^T\,R_t^{-1}\,A_t\,x_{t-1}}} \underline{- x_{t-1}^T A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t)}\\ + \tfrac{1}{2}\,(x_t - B_t\,u_t)^T\,R_t^{-1}\,(x_t - B_t\,u_t)\\ \underline{\underline{+\tfrac{1}{2}\,x_{t-1}^T\,\Sigma_{t-1}^{-1}\,x_{t-1}}} \underline{- x_{t-1}^T\,\Sigma_{t-1}^{-1}\,\mu_{t-1}} + \tfrac{1}{2}\,\mu_{t-1}^T\,\Sigma_{t-1}^{-1}\,\mu_{t-1}\\ \underline{\underline{-\tfrac{1}{2}\,x_{t-1}^T\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})\,x_{t-1}}}\\ \underline{+x_{t-1}^T\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}]}\\ -\tfrac{1}{2}\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}]^T\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\\ [A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}]$$

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 $L_t(x_{t-1}, x_t)$.

$$(3.26)\quad L_t(x_t) = +\tfrac{1}{2}\,(x_t - B_t\,u_t)^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \tfrac{1}{2}\,\mu_{t-1}^T\,\Sigma_{t-1}^{-1}\,\mu_{t-1}\\ -\tfrac{1}{2}\,[A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}]^T\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\\ [A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}]$$

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 computing the first and second derivatives of $L_t(x_t)$ with respect to x_t:

$$(3.27)\quad \frac{\partial L_t(x_t)}{\partial x_t} = R_t^{-1}\,(x_t - B_t\,u_t) - R_t^{-1}\,A_t\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\\ [A_t^T\,R_t^{-1}\,(x_t - B_t\,u_t) + \Sigma_{t-1}^{-1}\,\mu_{t-1}]\\ = [R_t^{-1} - R_t^{-1}\,A_t\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1} A_t^T\,R_t^{-1}]\,(x_t - B_t\,u_t)\\ - R_t^{-1}\,A_t\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\,\Sigma_{t-1}^{-1}\,\mu_{t-1}$$

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

$$(3.28)\quad R_t^{-1} - R_t^{-1}\,A_t\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1} A_t^T\,R_t^{-1} = (R_t + A_t\,\Sigma_{t-1}\,A_t^T)^{-1}$$

Hence the desired derivative is given by the following expression:

$$(3.29)\quad \frac{\partial L_t(x_t)}{\partial x_t} = (R_t + A_t\,\Sigma_{t-1}\,A_t^T)^{-1}\,(x_t - B_t\,u_t)\\ -R_t^{-1}\,A_t\,(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\,\Sigma_{t-1}^{-1}\,\mu_{t-1}$$

Inversion Lemma. For any invertible quadratic matrices R and Q and any matrix P with appropriate dimensions, 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. Define $\Psi = (Q^{-1} + P^T\,R^{-1}\,P)^{-1}$. It suffices to show that

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

This is shown through a series of transformations:

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

Table 3.2 The (specialized) inversion lemma, sometimes called the *Sherman/Morrison formula.*

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

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

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

$$\begin{aligned}
x_t \;&=\; B_t\,u_t + \underbrace{(R_t + A_t\,\Sigma_{t-1}\,A_t^T)R_t^{-1}\,A_t}_{A_t + A_t\,\Sigma_{t-1}\,A_t^T R_t^{-1}\,A_t}\,\underbrace{(A_t^T\,R_t^{-1}\,A_t + \Sigma_{t-1}^{-1})^{-1}\,\Sigma_{t-1}^{-1}}_{(\Sigma_{t-1}A_t^T\,R_t^{-1}\,A_t + I)^{-1}}\,\mu_{t-1} \\
&=\; B_t\,u_t + A_t\,\underbrace{(I + \Sigma_{t-1}\,A_t^T R_t^{-1}\,A_t)\,(\Sigma_{t-1}A_t^T\,R_t^{-1}\,A_t + I)^{-1}}_{=I}\,\mu_{t-1}
\end{aligned} \tag{3.31}$$

$$= \;B_t\,u_t + A_t\,\mu_{t-1}$$

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 $L_t(x_t)$:

$$\frac{\partial^2 L_t(x_t)}{\partial x_t^2} \;=\; (A_t\,\Sigma_{t-1}\,A_t^T + R_t)^{-1} \tag{3.32}$$

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 (2.38) and restated here in annotated form:

$$bel(x_t) \;=\; \eta\;\underbrace{p(z_t \mid x_t)}_{\sim\mathcal{N}(z_t;C_t x_t,Q_t)}\;\underbrace{\overline{bel}(x_t)}_{\sim\mathcal{N}(x_t;\bar{\mu}_t,\bar{\Sigma}_t)} \tag{3.33}$$

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

$$bel(x_t) \;=\; \eta\;\exp\{-J_t\} \tag{3.34}$$

with

$$J_t \;=\; \tfrac{1}{2}\,(z_t - C_t x_t)^T\,Q_t^{-1}\,(z_t - C_t x_t) + \tfrac{1}{2}\,(x_t - \bar{\mu}_t)^T\,\bar{\Sigma}_t^{-1}\,(x_t - \bar{\mu}_t) \tag{3.35}$$

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:

$$\frac{\partial J}{\partial x_t} = -C_t^T\, Q_t^{-1}\,(z_t - C_t x_t) + \bar\Sigma_t^{-1}\,(x_t - \bar\mu_t) \tag{3.36}$$

$$\frac{\partial^2 J}{\partial x_t^2} = C_t^T\, Q_t^{-1}\, C_t + \bar\Sigma_t^{-1} \tag{3.37}$$

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

$$\Sigma_t = (C_t^T\, Q_t^{-1}\, C_t + \bar\Sigma_t^{-1})^{-1} \tag{3.38}$$

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):

$$C_t^T\, Q_t^{-1}\,(z_t - C_t\mu_t) = \bar\Sigma_t^{-1}\,(\mu_t - \bar\mu_t) \tag{3.39}$$

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

$$\begin{aligned} & C_t^T\, Q_t^{-1}\,(z_t - C_t\mu_t) \\ &= C_t^T\, Q_t^{-1}\,(z_t - C_t\,\mu_t + C_t\,\bar\mu_t - C_t\,\bar\mu_t) \\ &= C_t^T\, Q_t^{-1}\,(z_t - C_t\bar\mu_t) - C_t^T\, Q_t^{-1}\, C_t\,(\mu_t - \bar\mu_t) \end{aligned} \tag{3.40}$$

Substituting this back into (3.39) gives us

$$C_t^T\, Q_t^{-1}\,(z_t - C_t\bar\mu_t) = \underbrace{(C_t^T\, Q_t^{-1}\, C_t + \bar\Sigma_t^{-1})}_{=\Sigma_t^{-1}}\,(\mu_t - \bar\mu_t) \tag{3.41}$$

and hence we have

$$\Sigma_t\, C_t^T\, Q_t^{-1}\,(z_t - C_t\,\bar\mu_t) = \mu_t - \bar\mu_t \tag{3.42}$$

We now define the *Kalman gain* as

$$K_t = \Sigma_t\, C_t^T\, Q_t^{-1} \tag{3.43}$$

and obtain

$$\mu_t = \bar\mu_t + K_t\,(z_t - C_t\,\bar\mu_t) \tag{3.44}$$

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

The Kalman gain, as defined in (3.43), 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 express K_t in terms of covariances other than Σ_t. It begins with the definition of K_t in (3.43):

$$\begin{aligned} K_t &= \Sigma_t\, C_t^T\, Q_t^{-1} \\ &= \Sigma_t\, C_t^T\, Q_t^{-1}\, \underbrace{(C_t\,\bar\Sigma_t\, C_t^T + Q_t)\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1}}_{=I} \\ &= \Sigma_t\,(C_t^T\, Q_t^{-1}\, C_t\,\bar\Sigma_t\, C_t^T + C_t^T\, \underbrace{Q_t^{-1}\, Q_t}_{=I})\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1} \\ &= \Sigma_t\,(C_t^T\, Q_t^{-1}\, C_t\,\bar\Sigma_t\, C_t^T + C_t^T)\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1} \\ &= \Sigma_t\,(C_t^T\, Q_t^{-1}\, C_t\,\bar\Sigma_t\, C_t^T + \underbrace{\bar\Sigma_t^{-1}\,\bar\Sigma_t}_{=I}\, C_t^T)\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1} \\ &= \Sigma_t\,\underbrace{(C_t^T\, Q_t^{-1}\, C_t + \bar\Sigma_t^{-1})}_{=\Sigma_t^{-1}}\,\bar\Sigma_t\, C_t^T\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1} \\ &= \underbrace{\Sigma_t\,\Sigma_t^{-1}}_{=I}\,\bar\Sigma_t\, C_t^T\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1} \\ &= \bar\Sigma_t\, C_t^T\,(C_t\,\bar\Sigma_t\, C_t^T + Q_t)^{-1} \end{aligned} \tag{3.45}$$

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 (3.38) 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 (3.38):

$$(\bar\Sigma_t^{-1} + C_t^T\, Q_t^{-1}\, C_t)^{-1} = \bar\Sigma_t - \bar\Sigma_t\, C_t^T\,(Q_t + C_t\,\bar\Sigma_t\, C_t^T)^{-1}\, C_t\,\bar\Sigma_t \tag{3.46}$$

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

$$\begin{aligned} \Sigma_t &= (C_t^T\, Q_t^{-1}\, C_t + \bar\Sigma_t^{-1})^{-1} \\ &= \bar\Sigma_t - \bar\Sigma_t\, C_t^T\,(Q_t + C_t\,\bar\Sigma_t\, C_t^T)^{-1}\, C_t\,\bar\Sigma_t \\ &= [I - \underbrace{\bar\Sigma_t\, C_t^T\,(Q_t + C_t\,\bar\Sigma_t\, C_t^T)^{-1}}_{=K_t,\ \text{see Eq. (3.45)}}\, C_t]\,\bar\Sigma_t \\ &= (I - K_t\, C_t)\,\bar\Sigma_t \end{aligned} \tag{3.47}$$

This completes our correctness proof, in that it shows the correctness of line 6 of our Kalman filter algorithm.

3.3 The Extended Kalman Filter

3.3.1 Why Linearize?

The assumptions that observations are linear functions of the state and that the next state is a linear function of the previous state are crucial for the correctness of the Kalman filter. The observation that any linear transformation of a Gaussian random variable results in another Gaussian random variable played an important role in the derivation of the Kalman filter algorithm. The efficiency of the Kalman filter is then due to the fact that the parameters of the resulting Gaussian can be computed in closed form.

Throughout this and the following chapters, we will illustrate properties of different density representations using the transformation of a one-dimensional Gaussian random variable. Figure 3.3a illustrates the *linear* transformation of such a random variable. The graph on the lower right shows the density of the random variable $X \sim \mathcal{N}(x; \mu, \sigma^2)$. Let us assume that X is passed through the linear function $y = ax + b$, shown in the upper right graph. The resulting random variable, Y, is distributed according to a Gaussian with mean $a\mu + b$ and variance $a^2\sigma^2$. This Gaussian is illustrated by the gray area in the upper left graph of Figure 3.3a. The reader may notice that this example is closely related to the next state update of the Kalman filter, with $X = x_{t-1}$ and $Y = x_t$ but without an additive noise variable; see also Equation (3.2).

Unfortunately, state transitions and measurements are rarely linear 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 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.

EXTENDED KALMAN FILTER

The *extended Kalman filter*, or *EKF*, relaxes one of these assumptions: the linearity assumption. Here the assumption is that the state transition probability and the measurement probabilities are governed by *nonlinear* functions g and h, respectively:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad (3.48)$$

$$z_t = h(x_t) + \delta_t \quad (3.49)$$

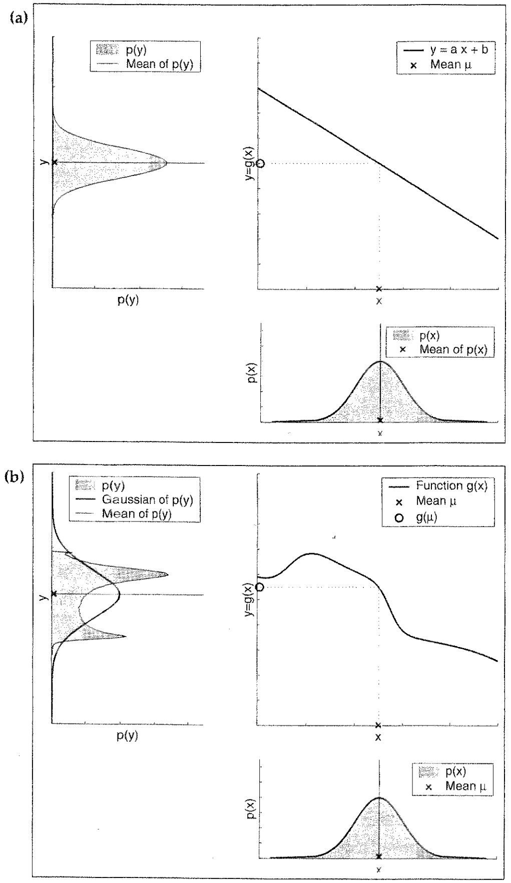

Figure 3.3 (a) Linear and (b) nonlinear transformation of a Gaussian random variable. The lower right plots show the density of the original random variable, X. This random variable is passed through the function displayed in the upper right graphs (the transformation of the mean is indicated by the dotted line). The density of the resulting random variable Y is plotted in the upper left graphs.

This model strictly generalizes the linear Gaussian model underlying Kalman filters, as 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, and the Bayes filter does not possess a closed-form solution.

Figure 3.3b illustrates the impact of a nonlinear transformation on a Gaussian random variable. The graphs on the lower right and upper right plot the random variable X and the nonlinear function g, respectively. The density of the transformed random variable, $Y = g(X)$, is indicated by the gray area in the upper left graph of Figure 3.3b. Since this density cannot be computed in closed form, it was estimated by drawing 500,000 samples according to $p(x)$, passing them through the function g, and then histogramming over the range of g. As can be seen, Y is not a Gaussian because the nonlinearities in g distort the density of X in ways that destroy its Gaussian shape.

The extended Kalman filter (EKF) calculates a Gaussian approximation to the true belief. The dashed curve in the upper left graph of Figure 3.3b shows the Gaussian approximation to the density of the random variable Y. Accordingly, EKFs represent the belief $bel(x_t)$ at time t 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. The goal of the EKF is thus shifted from computing the exact posterior to efficiently estimating its mean and covariance. However, since these statistics cannot be computed in closed form, the EKF has to resort to an additional approximation.

3.3.2 Linearization Via Taylor Expansion

The key idea underlying the EKF approximation is called *linearization*. Figure 3.4 illustrates the basic concept. Linearization approximates the nonlinear function g by a linear function that is tangent to g at the mean of the Gaussian (dashed line in the upper right graph). Projecting the Gaussian through this linear approximation results in a Gaussian density, as indicated by the dashed line in the upper left graph. The solid line in the upper left graph represents the mean and covariance of the Monte-Carlo approximation. The mismatch between these two Gaussians indicates the error caused by the linear approximation of g.

The key advantage of the linearization, however, lies in its efficiency.

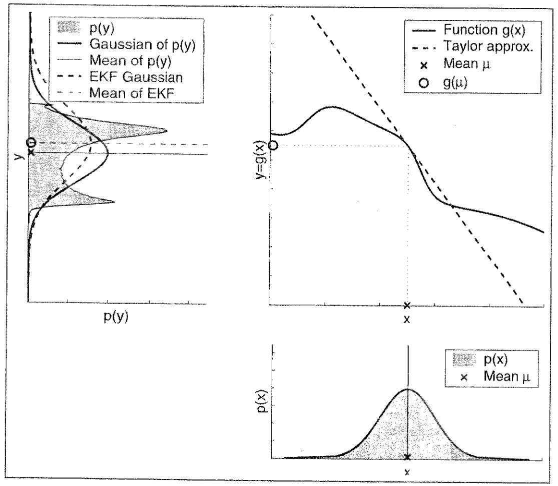

Figure 3.4 Illustration of linearization applied by the EKF. Instead of passing the Gaussian through the nonlinear function g, it is passed through a linear approximation of g. The linear function is tangent to g at the mean of the original Gaussian. The resulting Gaussian is shown as the dashed line in the upper left graph. The linearization incurs an approximation error, as indicated by the mismatch between the linearized Gaussian (dashed) and the Gaussian computed from the highly accurate Monte-Carlo estimate (solid).

The Monte-Carlo estimate of the Gaussian was achieved by passing 500,000 points through g followed by the computation of their mean and covariance. The linearization applied by the EKF, on the other hand, only requires determination of the linear approximation followed by the closed-form computation of the resulting Gaussian. In fact, once g is linearized, the mechanics of the EKF's belief propagation are equivalent to those of the Kalman filter.

This technique also is applied 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 con- TAYLOR EXPANSION structs a linear approximation to a function g from g's value and slope. The slope is given by the partial derivative

$$g'(u_t, x_{t-1}) \quad := \quad \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}} \tag{3.50}$$

Clearly, both the value of g and its slope depend on the argument of g. A logical choice for selecting the argument is to choose 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:

$$\begin{aligned} g(u_t, x_{t-1}) &\approx g(u_t, \mu_{t-1}) + \underbrace{g'(u_t, \mu_{t-1})}_{=: G_t} (x_{t-1} - \mu_{t-1}) \\ &= g(u_t, \mu_{t-1}) + G_t \, (x_{t-1} - \mu_{t-1}) \end{aligned} \tag{3.51}$$

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

$$\begin{aligned} & p(x_t \mid u_t, x_{t-1}) \\ &\approx \det(2\pi R_t)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2} \, [x_t - g(u_t, \mu_{t-1}) - G_t \, (x_{t-1} - \mu_{t-1})]^T \right. \\ & \quad \left. R_t^{-1} \, [x_t - g(u_t, \mu_{t-1}) - G_t \, (x_{t-1} - \mu_{t-1})]\right\} \end{aligned} \tag{3.52}$$

JACOBIAN

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:

$$\begin{aligned} h(x_t) &\approx h(\bar{\mu}_t) + \underbrace{h'(\bar{\mu}_t)}_{=: H_t} (x_t - \bar{\mu}_t) \\ &= h(\bar{\mu}_t) + H_t \, (x_t - \bar{\mu}_t) \end{aligned} \tag{3.53}$$

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

$$\begin{aligned} p(z_t \mid x_t) &= \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2} \, [z_t - h(\bar{\mu}_t) - H_t \, (x_t - \bar{\mu}_t)]^T \right. \\ & \quad \left. Q_t^{-1} \, [z_t - h(\bar{\mu}_t) - H_t \, (x_t - \bar{\mu}_t)]\right\} \end{aligned} \tag{3.54}$$

```
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 $\mu_t, \Sigma_t$
```

Table 3.3 The extended Kalman filter algorithm.

3.3.3 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:

	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 7.

3.3.4 Mathematical Derivation of the EKF

The mathematical derivation of the EKF parallels that of the Kalman filter in Chapter 3.2.4, and hence shall only be sketched here. The prediction is calculated as follows (c.f. (3.8)):

$$\overline{bel}(x_t) = \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\sim \mathcal{N}(x_t; g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}), R_t)} \underbrace{bel(x_{t-1})}_{\sim \mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})} dx_{t-1} \tag{3.55}$$

This distribution is the EKF analog of the prediction distribution in the Kalman filter, stated in (3.8). The Gaussian $p(x_t \mid x_{t-1}, u_t)$ can be found in Equation (3.52). The function L_t is given by (c.f. (3.11))

$$L_t = \tfrac{1}{2}\,(x_t - g(u_t,\mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1}))^T \\ R_t^{-1}\,(x_t - g(u_t,\mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})) \\ + \tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\,\Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \tag{3.56}$$

which is quadratic in both x_{t-1} and x_t, as above. As in (3.12), we decompose L_t into $L_t(x_{t-1}, x_t)$ and $L_t(x_t)$:

$$L_t(x_{t-1}, x_t) \\ = \tfrac{1}{2}\,(x_{t-1} - \Phi_t\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t,\mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}])^T\,\Phi^{-1} \\ (x_{t-1} - \Phi_t\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t,\mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}]) \tag{3.57}$$

with

$$\Phi_t = (G_t^T\,R_t^{-1}\,G_t + \Sigma_{t-1}^{-1})^{-1} \tag{3.58}$$

and hence

$$L_t(x_t) = \tfrac{1}{2}\,(x_t - g(u_t,\mu_{t-1}) + G_t\mu_{t-1})^T\,R_t^{-1}\,(x_t - g(u_t,\mu_{t-1}) + G_t\mu_{t-1}) \\ +\tfrac{1}{2}\,(x_{t-1} - \mu_{t-1})^T\,\Sigma_{t-1}^{-1}\,(x_{t-1} - \mu_{t-1}) \\ -\tfrac{1}{2}\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t,\mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}]^T \\ \Phi_t\,[G_t^T\,R_t^{-1}\,(x_t - g(u_t,\mu_{t-1}) + G_t\mu_{t-1}) + \Sigma_{t-1}^{-1}\mu_{t-1}] \tag{3.59}$$

As the reader can easily verify, 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 (3.27) through (3.31). The second derivative is given by $(R_t + G_t\,\Sigma_{t-1}\,G_t^T)^{-1}$ (see (3.32)).

The measurement update is also derived analogously to the Kalman filter in Chapter 3.2.4. In analogy to (3.33), we have for the EKF

$$bel(x_t) = \eta \quad \underbrace{p(z_t \mid x_t)}_{\sim\mathcal{N}(z_t;h(\bar\mu_t)+H_t\,(x_t-\bar\mu_t),Q_t)} \quad \underbrace{\overline{bel}(x_t)}_{\sim\mathcal{N}(x_t;\bar\mu_t,\bar\Sigma_t)} \tag{3.60}$$

using the linearized state transition function from (3.53). This leads to the exponent (see (3.35)):

$$J_t = \tfrac{1}{2}\,(z_t - h(\bar\mu_t) - H_t\,(x_t - \bar\mu_t))^T\,Q_t^{-1}\,(z_t - h(\bar\mu_t) - H_t\,(x_t - \bar\mu_t)) \\ +\tfrac{1}{2}(x_t - \bar\mu_t)^T\bar\Sigma_t^{-1}(x_t - \bar\mu_t) \tag{3.61}$$

The resulting mean and covariance is given by

$$\mu_t = \bar\mu_t + K_t(z_t - h(\bar\mu_t)) \tag{3.62}$$

$$\Sigma_t = (I - K_t\,H_t)\,\bar\Sigma_t \tag{3.63}$$

with the Kalman gain

$$K_t = \bar\Sigma_t\,H_t^T\,(H_t\,\bar\Sigma_{t-1}\,H_t^T + Q_t)^{-1} \tag{3.64}$$

The derivation of these equations is analogous to Equations (3.36) through (3.47).

3.3.5 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 $O(k^{2.4} + n^2)$, 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.

An important limitation of the EKF arises from the fact that it approximates state transitions and measurements using linear Taylor expansions. In most robotics problems, state transitions and measurements are nonlinear. The goodness of the linear approximation applied by the EKF depends on two main factors: The degree of uncertainty and the degree of local nonlinearity of the functions that are being approximated. The two graphs in Figure 3.5 illustrate the dependency on the uncertainty. Here, two Gaussian random variables are passed through the same nonlinear function (c.f. also Figure 3.4). While both Gaussians have the same mean, the variable shown in (a) has a higher uncertainty than the one in (b). Since the Taylor expansion only depends on the mean, both Gaussians are passed through the same linear approximation. The gray areas in the upper left plots of the two figures show the densities of the resulting random variable, computed by Monte-Carlo estimation. The density resulting from the wider Gaussian is far more distorted than the density resulting from the narrow, less uncertain Gaussian. The Gaussian approximations of these densities are given by the solid lines in the figures. The dashed graphs show the Gaussians estimated by the

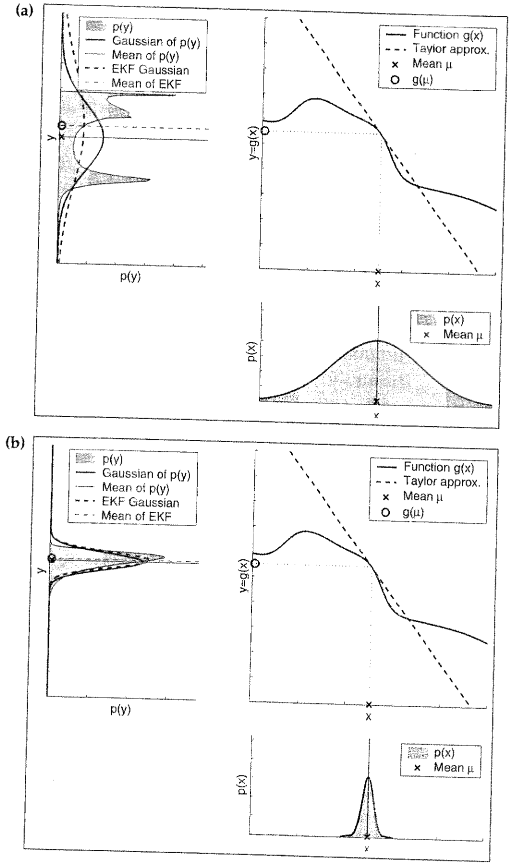

Figure 3.5 Dependency of approximation quality on uncertainty. Both Gaussians (lower right) have the same mean and are passed through the same nonlinear function (upper right). The higher uncertainty of the left Gaussian produces a more distorted density of the resulting random variable (gray area in upper left graph). The solid lines in the upper left graphs show the Gaussians extracted from these densities. The dashed lines represent the Gaussians generated by the linearization applied by the EKF.

linearization. A comparison to the Gaussians resulting from the Monte-Carlo approximations illustrates the fact that higher uncertainty typically results in less accurate estimates of the mean and covariance of the resulting random variable.

The second factor for the quality of the linear Gaussian approximation is the local nonlinearity of the function g, as illustrated in Figure 3.6. Shown there are two Gaussians with the same variance passed through the same nonlinear function. In Panel (a), the mean of the Gaussian falls into a more nonlinear region of the function g than in Panel (b). The mismatch between the accurate Monte-Carlo estimate of the Gaussian (solid line, upper left) and the Gaussian resulting from linear approximation (dashed line) shows that higher nonlinearities result in larger approximation errors. The EKF Gaussian clearly underestimates the spread of the resulting density.

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 Gaussians may be of the form

MIXTURE OF GAUSSIANS

(3.65)
$$bel(x_t) \;=\; \frac{1}{\sum_l \psi_{t,l}} \sum_l \psi_{t,l} \det\left(2\pi\Sigma_{t,l}\right)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(x_t-\mu_{t,l})^T\Sigma_{t,l}^{-1}(x_t-\mu_{t,l})\right\}$$

Here $\psi_{t,l}$ are mixture parameters with $\psi_{t,l} \geq 0$. These parameters serve as weights of the mixture components. They are estimated from the likelihoods of the observations conditioned on the corresponding Gaussians. EKFs that utilize such mixture representations are called *multi-hypothesis (extended) Kalman filters*, or *MHEKF*.

MULTI-HYPOTHESIS EKF

To summarize, if the nonlinear functions are approximately linear at the mean of the estimate, then the EKF approximation may generally be a good one, and EKFs may approximate the posterior belief with sufficient accuracy. Furthermore, 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.

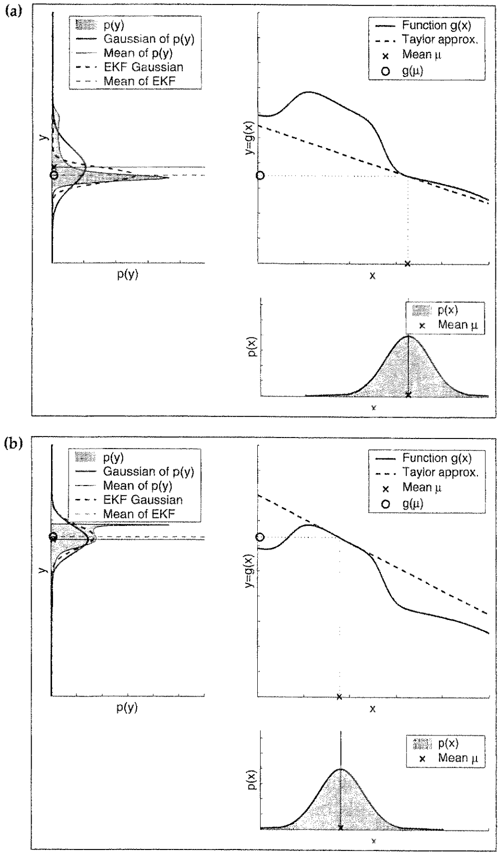

Figure 3.6 Dependence of the approximation quality on local nonlinearity of the function g. Both Gaussians (lower right in each of the two panels) have the same covariance and are passed through the same function (upper right). The linear approximation applied by the EKF is shown as the dashed lines in the upper right graphs. The solid lines in the upper left graphs show the Gaussians extracted from the highly accurate Monte-Carlo estimates. The dashed lines represent the Gaussians generated by the EKF linearization.

3.4 The Unscented Kalman Filter

The Taylor series expansion applied by the EKF is only one way to linearize the transformation of a Gaussian. Two other approaches have often been found to yield superior results. One is known as *moments matching* (and the resulting filter is known as *assumed density filter*, or *ADF*), 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). Another linearization method is applied by the *unscented Kalman filter*, or *UKF*, which performs a stochastic linearization through the use of a weighted statistical linear regression process. We now discuss the UKF algorithm without mathematical derivation. The reader is encouraged to read more details in the literature referenced in the bibliographical remarks.

UNSCENTED KALMAN FILTER

3.4.1 Linearization Via the Unscented Transform

Figure 3.7 illustrates the linearization applied by the UKF, called the *unscented transform*. Instead of approximating the function g by a Taylor series expansion, the UKF deterministically extracts so-called *sigma points* from the Gaussian and passes these through g. In the general case, these sigma points are located at the mean and symmetrically along the main axes of the covariance (two per dimension). For an n-dimensional Gaussian with mean μ and covariance Σ, the resulting $2n+1$ sigma points $\mathcal{X}^{[i]}$ are chosen according to the following rule:

SIGMA POINT

$$\begin{aligned}
\mathcal{X}^{[0]} &= \mu \\
\mathcal{X}^{[i]} &= \mu + \left(\sqrt{(n+\lambda)\,\Sigma}\right)_i \qquad \text{for } i = 1,\ldots,n \\
\mathcal{X}^{[i]} &= \mu - \left(\sqrt{(n+\lambda)\,\Sigma}\right)_{i-n} \qquad \text{for } i = n+1,\ldots,2n
\end{aligned} \tag{3.66}$$

Here $\lambda = \alpha^2(n+\kappa) - n$, with α and κ being scaling parameters that determine how far the sigma points are spread from the mean. Each sigma point $\mathcal{X}^{[i]}$ has two weights associated with it. One weight, $w_m^{[i]}$, is used when computing the mean, the other weight, $w_c^{[i]}$, is used when recovering the covariance of the Gaussian.

$$\begin{aligned}
w_m^{[0]} &= \frac{\lambda}{n+\lambda} \\
w_c^{[0]} &= \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)
\end{aligned} \tag{3.67}$$

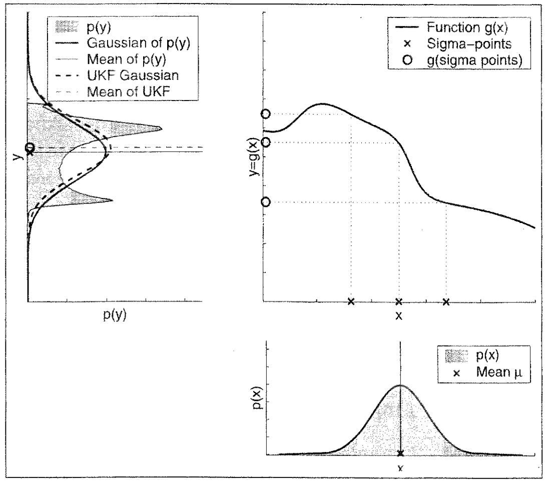

Figure 3.7 Illustration of linearization applied by the UKF. The filter first extracts $2n+1$ weighted sigma points from the n-dimensional Gaussian ($n = 1$ in this example). These sigma points are passed through the nonlinear function g. The linearized Gaussian is then extracted from the mapped sigma points (small circles in the upper right plot). As for the EKF, the linearization incurs an approximation error, indicated by the mismatch between the linearized Gaussian (dashed) and the Gaussian computed from the highly accurate Monte-Carlo estimate (solid).

$$w_m^{[i]} \;=\; w_c^{[i]} \;=\; \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1, \ldots, 2n.$$

The parameter β can be chosen to encode additional (higher order) knowledge about the distribution underlying the Gaussian representation. If the distribution is an exact Gaussian, then $\beta = 2$ is the optimal choice.

The sigma points are then passed through the function g, thereby probing how g changes the shape of the Gaussian.

(3.68) $$\mathcal{Y}^{[i]} \;=\; g(\mathcal{X}^{[i]})$$

The parameters (μ' Σ') of the resulting Gaussian are extracted from the mapped sigma points $\mathcal{Y}^{[i]}$ according to

(3.69) $$\mu' \;=\; \sum_{i=0}^{2n} w_m^{[i]}\, \mathcal{Y}^{[i]}$$

$$\Sigma' \;=\; \sum_{i=0}^{2n} w_c^{[i]}\, (\mathcal{Y}^{[i]} - \mu')(\mathcal{Y}^{[i]} - \mu')^T.$$

Figure 3.8 illustrates the dependency of the unscented transform on the uncertainty of the original Gaussian. For comparison, the results using the EKF Taylor series expansion are plotted alongside the UKF results.

Figure 3.9 shows an additional comparison between UKF and EKF approximation, here in dependency of the local nonlinearity of the function g. As can be seen, the unscented transform is more accurate than the first order Taylor series expansion applied by the EKF. In fact, it can be shown that the unscented transform is accurate in the first two terms of the Taylor expansion, while the EKF captures only the first order term. (It should be noted, however, that both the EKF and the UKF can be modified to capture higher order terms.)

3.4.2 The UKF Algorithm

The UKF algorithm utilizing the unscented transform is presented in Table 3.4. The input and output are identical to the EKF algorithm. Line 2 determines the sigma points of the previous belief using Equation (3.66), with γ short for $\sqrt{n+\lambda}$. These points are propagated through the noise-free state prediction in line 3. The predicted mean and variance are then computed from the resulting sigma points (lines 4 and 5). R_t in line 5 is added to the sigma point covariance in order to model the additional prediction noise uncertainty (compare line 3 of the EKF algorithm in Table 3.3). The prediction noise R_t is assumed to be additive. Later, in Chapter 7, we present a version of the UKF algorithm that performs more accurate estimation of the prediction and measurement noise terms.

A new set of sigma points is extracted from the predicted Gaussian in line 6. This sigma point set $\bar{\mathcal{X}}_t$ now captures the overall uncertainty after the prediction step. In line 7, a predicted observation is computed for each sigma point. The resulting observation sigma points $\bar{\mathcal{Z}}_t$ are used to compute the predicted observation $\hat{z}_t$ and its uncertainty, S_t. The matrix Q_t is the covariance matrix of the additive measurement noise. Note that S_t represents the same uncertainty as $H_t\ \bar{\Sigma}_t\ H_t^T + Q_t$ in line 4 of the EKF algorithm in

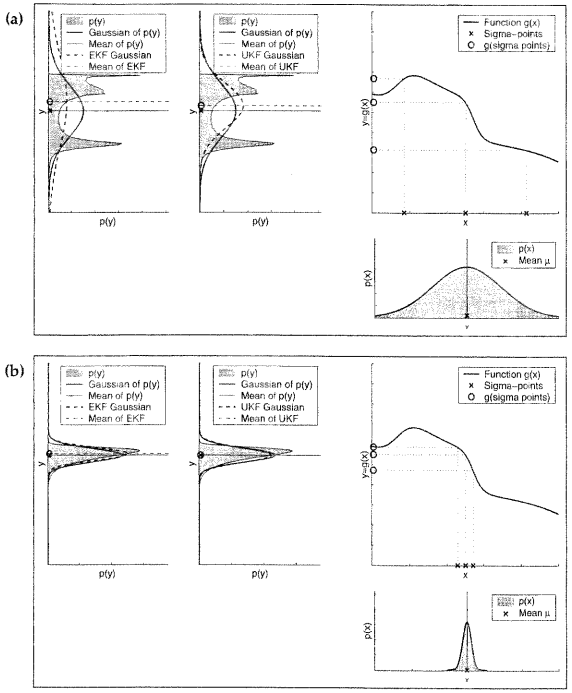

Figure 3.8 Linearization results for the UKF depending on the uncertainty of the original Gaussian. The results of the EKF linearization are also shown for comparison (c.f. Figure3.5). The unscented transform incurs smaller approximation errors, as can be seen by the stronger similarity between the dashed and the solid Gaussians.

Table 3.3. Line 10 determines the cross-covariance between state and observation, which is then used in line 11 to compute the Kalman gain K_t. The cross-covariance $\bar{\Sigma}_t^{x,z}$ corresponds to the term $\bar{\Sigma}_t\ H_t^T$ in line 4 of the EKF algorithm. With this in mind it is straightforward to show that the estimation update performed in lines 12 and 13 is of equivalent form to the update performed by the EKF algorithm.

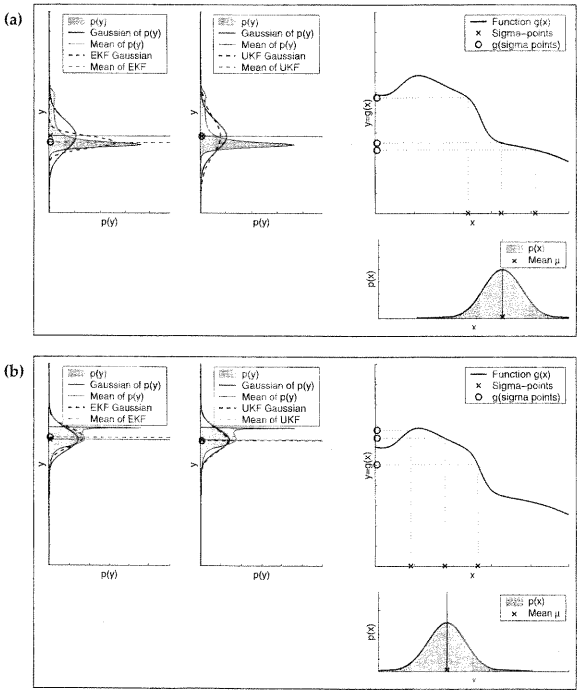

Figure 3.9 Linearization results for the UKF depending on the mean of the original Gaussian. The results of the EKF linearization are also shown for comparison (c.f. Figure3.6). The sigma point linearization incurs smaller approximation errors, as can be seen by the stronger similarity between the dashed and the solid Gaussians.

The asymptotic complexity of the UKF algorithm is the same as for the EKF. In practice, the EKF is often slightly faster than the UKF. The UKF is still highly efficient, even with this slowdown by a constant factor. Furthermore, the UKF inherits the benefits of the unscented transform for linearization. For purely linear systems, it can be shown that the estimates generated by the UKF are identical to those generated by the Kalman filter. For nonlin-

```
Algorithm Unscented_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
    $\mathcal{X}_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + \gamma\sqrt{\Sigma_{t-1}} \quad \mu_{t-1} - \gamma\sqrt{\Sigma_{t-1}})$
    $\bar{\mathcal{X}}_t^* = g(u_t, \mathcal{X}_{t-1})$
    $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$
    $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + R_t$
    $\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} \quad \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t})$
    $\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$
    $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$
    $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$
    $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$
    $K_t = \bar{\Sigma}_t^{x,z} \, S_t^{-1}$
    $\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$
    $\Sigma_t = \bar{\Sigma}_t - K_t \, S_t \, K_t^T$
    return $\mu_t, \Sigma_t$
```

Table 3.4 The unscented Kalman filter algorithm. The variable n denotes the dimensionality of the state vector.

ear systems the UKF produces equal or better results than the EKF, where the improvement over the EKF depends on the nonlinearities and spread of the prior state uncertainty. In many practical applications, the difference between EKF and UKF is negligible.

DERIVATIVE-FREE FILTER

Another advantage of the UKF is the fact that it does not require the computation of Jacobians, which are difficult to determine in some domains. The UKF is thus often referred to as a *derivative-free filter*.

Finally, the unscented transform has some resemblance to the sample-based representation used by particle filters, which will be discussed in the next chapter. A key difference however, is that the sigma points of the unscented transform are determined deterministically, while particle filters draw samples randomly. This has important implications. If the underlying distribution is approximately Gaussian, then the UKF representation is far more efficient than the particle filter representation. If, on the other hand, the belief is highly non-Gaussian, then the UKF representation is too restrictive and the filter can perform arbitrarily poorly.

3.5 The Information Filter

The dual of the Kalman filter is the *information filter*, or *IF*. Just like the KF and its nonlinear versions, the EKF and the UKF, the information filter 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 parameterization, which is comprised of an information matrix and an information vector. The difference in parameterization leads to different update equations. In particular, what is computationally complex in one parameterization happens to be simple in the other (and vice versa). The canonical and the moments parameterizations are often considered *dual* to each other, and thus are the IF and the KF.

3.5.1 Canonical Parameterization

CANONICAL PARAMETERIZATION

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

$$\Omega = \Sigma^{-1} \tag{3.70}$$

INFORMATION MATRIX

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

$$\xi = \Sigma^{-1}\,\mu \tag{3.71}$$

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 parameterization by the inverse of (3.70) and (3.71):

$$\Sigma = \Omega^{-1} \tag{3.72}$$

$$\mu = \Omega^{-1}\,\xi \tag{3.73}$$

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

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{3.74}$$

A straightforward sequence of transformations leads to the following parameterization:

$$\begin{aligned} p(x) &= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \tfrac{1}{2}\mu^T\Sigma^{-1}\mu\right\} \\ &= \underbrace{\det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}\mu^T\Sigma^{-1}\mu\right\}}_{\text{const.}} \exp\left\{-\tfrac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\} \end{aligned} \tag{3.75}$$

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

$$p(x) = \eta \exp\left\{-\tfrac{1}{2}x^T\,\Sigma^{-1}\,x + x^T\,\Sigma^{-1}\,\mu\right\} \tag{3.76}$$

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

$$p(x) = \eta \exp\left\{-\tfrac{1}{2}x^T\,\Omega\,x + x^T\,\xi\right\} \tag{3.77}$$

In many ways, the canonical parameterization is more elegant than the moments parameterization. In particular, the negative logarithm of the Gaussian is a quadratic function in x, with the canonical parameters Ω and ξ:

$$-\log p(x) = \text{const.} + \tfrac{1}{2}x^T\,\Omega\,x - x^T\,\xi \tag{3.78}$$

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 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 (3.78) to zero:

$$\frac{\partial[-\log p(x)]}{\partial x} = 0 \iff \Omega x - \xi = 0 \iff x = \Omega^{-1}\xi \tag{3.79}$$

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 a *Mahalanobis distance*. MAHALANOBIS DISTANCE

```
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 $\xi_t, \Omega_t$
```

Table 3.5 The information filter algorithm.

3.5.2 The Information Filter Algorithm

Table 3.5 states the update algorithm known as *information filter*. Its input is a Gaussian in its canonical parameterization 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 Chapter 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):

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{3.80}$$

$$z_t = C_t x_t + \delta_t \tag{3.81}$$

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 3.5. 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 3.5, 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

$O(n^{2.4})$ 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 $O(n^{2.4})$. This illustrates the dual character of Kalman and information filters.

3.5.3 Mathematical Derivation of the Information Filter

The derivation of the information filter is analogous to that of the Kalman filter.

PREDICTION STEP

To derive the *prediction step* (lines 2 and 3 in Table 3.5), 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:

$$\bar{\mu}_t = A_t\, \mu_{t-1} + B_t\, u_t \tag{3.82}$$

$$\bar{\Sigma}_t = A_t\, \Sigma_{t-1}\, A_t^T + R_t \tag{3.83}$$

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 (3.72) and (3.73):

$$\mu_{t-1} = \Omega_{t-1}^{-1}\, \xi_{t-1} \tag{3.84}$$

$$\Sigma_{t-1} = \Omega_{t-1}^{-1} \tag{3.85}$$

Substituting these expressions in (3.82) and (3.83) gives us the set of prediction equations

$$\bar{\Omega}_t = (A_t\, \Omega_{t-1}^{-1}\, A_t^T + R_t)^{-1} \tag{3.86}$$

$$\bar{\xi}_t = \bar{\Omega}_t(A_t\, \Omega_{t-1}^{-1}\, \xi_{t-1} + B_t\, u_t) \tag{3.87}$$

These equations are identical to those in Table 3.5. 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 that will be discussed later in this book.

MEASUREMENT UPDATE

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 (3.35) and is restated here once again:

$$\begin{aligned} bel(x_t) & \\ = \ & \eta\, \exp\left\{-\tfrac{1}{2}\,(z_t - C_t x_t)^T\, Q_t^{-1}\,(z_t - C_t x_t) - \tfrac{1}{2}\,(x_t - \bar{\mu}_t)^T\, \bar{\Sigma}_t^{-1}\,(x_t - \bar{\mu}_t)\right\}\end{aligned} \tag{3.88}$$

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

$$\begin{aligned} bel(x_t) & \\ = \ & \eta\, \exp\left\{-\tfrac{1}{2}\, x_t^T\, C_t^T\, Q_t^{-1}\, C_t\, x_t + x_t^T\, C_t^T\, Q_t^{-1}\, z_t - \tfrac{1}{2}\, x_t^T\, \bar{\Omega}_t x_t + x_t^T \bar{\xi}_t\right\}\end{aligned} \tag{3.89}$$

which, by reordering the terms in the exponent, resolves to

$$bel(x_t) = \eta\, \exp\left\{-\tfrac{1}{2}\, x_t^T\, [C_t^T\, Q_t^{-1}\, C_t + \bar{\Omega}_t]\, x_t + x_t^T\, [C_t^T\, Q_t^{-1}\, z_t + \bar{\xi}_t]\right\} \tag{3.90}$$

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

$$\xi_t = C_t^T\, Q_t^{-1}\, z_t + \bar{\xi}_t \tag{3.91}$$

$$\Omega_t = C_t^T\, Q_t^{-1}\, C_t + \bar{\Omega}_t \tag{3.92}$$

These equations are identical to the measurement update equations in lines 4 and 5 of Table 3.5.

3.5.4 The Extended Information Filter Algorithm

The *extended information filter*, or *EIF*, extends the information filter to the nonlinear case, very much in the same way the EKF is the non-linear extension of the Kalman filter. Table 3.6 depicts the EIF 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 state transition function and measurement function, respectively. Those were defined in (3.48) and (3.49) and are restated here:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{3.93}$$

$$z_t = h(x_t) + \delta_t \tag{3.94}$$

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

```
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 $\xi_t, \Omega_t$
```

Table 3.6 The extended information filter (EIF) algorithm.

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.

3.5.5 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 (3.51) and (3.53), the extended information filter approximates g and h by a Taylor expansion:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t\ (x_{t-1} - \mu_{t-1}) \tag{3.95}$$

$$h(x_t) \approx h(\bar{\mu}_t) + H_t\ (x_t - \bar{\mu}_t) \tag{3.96}$$

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

$$G_t = g'(u_t, \mu_{t-1}) \tag{3.97}$$

$$H_t = h'(\bar{\mu}_t) \tag{3.98}$$

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:

$$\bar{\Sigma}_t = G_t\ \Sigma_{t-1}\ G_t^T + R_t \tag{3.99}$$

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) \tag{3.100}$$

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:

$$\bar{\Omega}_t = (G_t\ \Omega_{t-1}^{-1}\ G_t^T + R_t)^{-1} \tag{3.101}$$

$$\bar{\xi}_t = \bar{\Omega}_t\ g(u_t, \Omega_{t-1}^{-1}\ \xi_{t-1}) \tag{3.102}$$

The measurement update is derived from Equations (3.60) and (3.61). In particular, (3.61) defines the following Gaussian posterior:

$$\begin{aligned} bel(x_t) = \eta\ \exp\{ & -\tfrac{1}{2}\ (z_t - h(\bar{\mu}_t) - H_t\ (x_t - \bar{\mu}_t))^T\ Q_t^{-1} \\ & (z_t - h(\bar{\mu}_t) - H_t\ (x_t - \bar{\mu}_t)) - \tfrac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\} \end{aligned} \tag{3.103}$$

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

$$\begin{aligned} bel(x_t) &= \eta\ \exp\{-\tfrac{1}{2}\ x_t^T\ H_t^T\ Q_t^{-1}\ H_t\ x_t + x_t^T\ H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) + H_t\ \bar{\mu}_t] \\ &\quad -\tfrac{1}{2} x_t^T\ \bar{\Sigma}_t^{-1}\ x_t + x_t^T\ \bar{\Sigma}_t^{-1}\ \bar{\mu}_t\} \\ &= \eta\ \exp\{-\tfrac{1}{2}\ x_t^T\ [H_t^T\ Q_t^{-1}\ H_t + \bar{\Sigma}_t^{-1}]\ x_t \\ &\quad + x_t^T\ [H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) + H_t\ \bar{\mu}_t] + \bar{\Sigma}_t^{-1}\ \bar{\mu}_t] \end{aligned} \tag{3.104}$$

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

$$\begin{aligned} bel(x_t) &= \eta\ \exp\{-\tfrac{1}{2}\ x_t^T\ [H_t^T\ Q_t^{-1}\ H_t + \bar{\Omega}_t]\ x_t \\ &\quad + x_t^T\ [H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) + H_t\ \bar{\mu}_t] + \bar{\xi}_t] \end{aligned} \tag{3.105}$$

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

$$\Omega_t = \bar{\Omega}_t + H_t^T\ Q_t^{-1}\ H_t \tag{3.106}$$

$$\xi_t = \bar{\xi}_t + H_t^T\ Q_t^{-1}\ [z_t - h(\bar{\mu}_t) + H_t\ \bar{\mu}_t] \tag{3.107}$$

3.5.6 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. The information filter tends to be numerically more stable than the Kalman filter in many of the applications discussed later in this book.

As we shall see in later chapters of this book, the information filter and several extensions enable a robot to integrate information without immediately resolving it into probabilities. This can be of great advantage in complex estimation problems, involving hundreds or even millions of variables. For such large problems, the integration á la Kalman filter induces severe computational problems, since any new piece of information requires propagation through a large system of variables. The information filter, with appropriate modification, can side-step this issue by simply adding the new information locally into the system. However, this is *not* a property yet of the simple information filter discussed here; we will extend this filter in Chapter 12.

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 parameterization—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 underexplored. We will revisit the multi-robot topic in Chapter 12.

These advantages of the information filter are offset by important limitations. A primary disadvantage of the extended information filter 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 extended information filter.

As we will see later in this book, these limitations do not necessarily apply to problems in which the information matrix possesses 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.

MARKOV RANDOM FIELD

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.

3.6 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 parameterization and the canonical parameterization. The moments parameterization consists of the mean (first moment) and the covariance (second moment) of the Gaussian. The canonical, or natural, parameterization consists of an information matrix and an information vector. Both parameterizations are duals of each other, and each can be recovered from the other via matrix inversion.

- Bayes filters can be implemented for both parameterizations. When using the moments parameterization, the resulting filter is called Kalman filter. The dual of the Kalman filter is the information filter, which represents the posterior in the canonical parameterization. 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. One 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 unscented Kalman filter uses a different linearization technique, called unscented transform. It probes the function to be linearized at selected points and calculates a linearized approximation based on the outcomes of these probes. This filter can be implemented without the need for any Jacobians, it is thus often referred to as derivative-free. The unscented Kalman filter is equivalent to the Kalman filter for linear systems but often provides improved estimates for nonlinear systems. The computational complexity of this filter is the same as for the extended Kalman filter.

- The accuracy of Taylor series expansions and unscented transforms 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.

- An extension of Gaussians to multimodal posteriors is known as multi-hypothesis Kalman filter. This filter represents the posterior by a mixture of Gaussians, which is nothing else but a weighted sum of Gaussians. The mechanics of updating this filter require mechanisms for splitting and fusing or pruning individual Gaussians. Multi-hypothesis Kalman filters are particularly well suited for problems with discrete data association, which commonly occur in robotics.

- 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 Gaussian filters, which address the various limitations and shortcomings of the individual filters.

A good number of algorithms in this book are based on Gaussian filters. Many practical robotics problems require extensions that exploit sparse structures or factorizations of the posterior.

3.7 Bibliographical Remarks

The Kalman filter was invented by Swerling (1958) and Kalman (1960). It is usually introduced as an optimal estimator under the least-squares assumption, and less frequently as a method for calculating posterior distributions—although under the appropriate assumptions both views are identical. There exists a number of excellent textbooks on Kalman filters and information filters, including the ones by Maybeck (1990) and Jazwinsky (1970). Contemporary treatments of Kalman filters with data association are provided by Bar-Shalom and Fortmann (1988); Bar-Shalom and Li (1998).

The inversion lemma can be found in Golub and Loan (1986). Matrix inversion can be carried out in $O(n^{2.376})$ time, according to Coppersmith and Winograd (1990). This result is the most recent one in a series of papers that provided improvements over the $O(n^3)$ complexity of the variable elimination algorithm. The series started with Strassen's (1969) seminal paper, in which he gave an algorithm requiring $O(n^{2.807})$. Cover and Thomas (1991) provides a survey of information theory, but with a focus on discrete systems. The unscented Kalman filter is due to Julier and Uhlmann (1997). A comparison of UKF to the EKF in the context of various state estimation problems can be found in van der Merwe (2004). Minka (2001) provided a recent treatment of moments matching and assumed density filtering for Gaussian mixtures.

3.8 Exercises

1. In this and the following exercise, you are asked to design a Kalman filter for a simple dynamical system: a car with linear dynamics moving in a linear environment. Assume $\Delta t = 1$ for simplicity. The position of the

car at time t is given by x_t. Its velocity is $\dot{x}_t$, and its acceleration is $\ddot{x}_t$. Suppose the acceleration is set randomly at each point in time, according to a Gaussian with zero mean and covariance $\sigma^2 = 1$.

(a) What is a minimal state vector for the Kalman filter (so that the resulting system is Markovian)?

(b) For your state vector, design the state transition probability $p(x_t \mid u_t, x_{t-1})$. Hint: this transition function will possess linear matrices A and B and a noise covariance R (c.f., Equation (3.4) and Table 3.1).

(c) Implement the state prediction step of the Kalman filter. Assuming we know at time $t = 0$, $x_0 = \dot{x}_0 = \ddot{x}_0 = 0$. Compute the state distributions for times $t = 1, 2, \ldots, 5$.

UNCERTAINTY ELLIPSE

(d) For each value of t, plot the joint posterior over x and $\dot{x}$ in a diagram, where x is the horizontal and $\dot{x}$ is the vertical axis. For each posterior, you are asked to plot an *uncertainty ellipse*, which is the ellipse of points that are one standard deviation away from the mean. Hint: If you do not have access to a mathematics library, you can create those ellipses by analyzing the eigenvalues of the covariance matrix.

(e) What will happen to the correlation between x_t and $\dot{x}_t$ as $t \uparrow \infty$?

2. We will now add measurements to our Kalman filter. Suppose at time t, we can receive a noisy observation of x. In expectation, our sensor measures the true location. However, this measurement is corrupted by Gaussian noise with covariance $\sigma^2 = 10$.

(a) Define the measurement model. Hint: You need to define a matrix C and another matrix Q (c.f., Equation (3.6) and Table 3.1).

(b) Implement the measurement update. Suppose at time $t = 5$, we observe a measurement $z = 5$. State the parameters of the Gaussian estimate before and after updating the KF. Plot the uncertainty ellipse before and after incorporating the measurement (see above for instructions as to how to plot an uncertainty ellipse).

3. In Chapter 3.2.4, we derived the prediction step of the KF. This step is often derived with Z transforms or Fourier transforms, using the Convolution Theorem. Re-derive the prediction step using transforms. *Notice: This exercise requires knowledge of transforms and convolution, which goes beyond the material in this book.*

4. We noted in the text that the EKF linearization is an approximation. To see how bad this approximation is, we ask you to work out an example. Suppose we have a mobile robot operating in a planar environment. Its state is its x-y-location and its global heading direction θ. Suppose we know x and y with high certainty, but the orientation θ is unknown. This is reflected by our initial estimate

$$\mu = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 10000 \end{pmatrix}$$

(a) Draw, graphically, your best model of the posterior over the robot pose after the robot moves $d = 1$ units forward. For this exercise, we assume the robot moves flawlessly without any noise. Thus, the expected location of the robot after motion will be

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + \cos\theta \\ y + \sin\theta \\ \theta \end{pmatrix}$$

For your drawing, you can ignore θ and only draw the posterior in x-y-coordinates.

(b) Now develop this motion into a prediction step for the EKF. For that, you have to define a state transition function and linearize it. You then have to generate a new Gaussian estimate of the robot pose using the linearized model. You should give the exact mathematical equations for each of these steps, and state the Gaussian that results.

(c) Draw the uncertainty ellipse of the Gaussian and compare it with your intuitive solution.

(d) Now incorporate a measurement. Our measurement shall be a noisy projection of the x-coordinate of the robot, with covariance $Q = 0.01$. Specify the measurement model. Now apply the measurement both to your intuitive posterior, and formally to the EKF estimate using the standard EKF machinery. Give the exact result of the EKF, and compare it with the result of your intuitive analysis.

(e) Discuss the difference between your estimate of the posterior, and the Gaussian produced by the EKF. How significant are those differences? What can be changed to make the approximation more accurate? What would have happened if the initial orientation had been known, but not the robot's y-coordinate?

5. The Kalman filter Table 3.1 lacked a constant additive term in the motion and the measurement models. Extend this algorithm to contain such terms.

6. Prove (via example) the existence of a sparse information matrix in multivariate Gaussians (of dimension d) that correlate all d variables with correlation coefficient that are ε-close to 1. We say an information matrix is sparse if all but a constant number of elements in each row and each column are zero.

4 Nonparametric Filters

A popular alternative to Gaussian techniques are *nonparametric filters*. Nonparametric filters do not rely on a fixed functional form of the posterior, such as Gaussians. Instead, they approximate posteriors by a finite number of values, each roughly corresponding to a region in state space. Some nonparametric Bayes filters rely on a decomposition of the state space, in which each such value corresponds to the cumulative probability of the posterior density in a compact subregion of the state space. Others approximate the state space by random samples drawn from the posterior distribution. In all cases, the number of parameters used to approximate the posterior can be varied. The quality of the approximation depends on the number of parameters used to represent the posterior. As the number of parameters goes to infinity, nonparametric techniques tend to converge uniformly to the correct posterior—under specific smoothness assumptions.

This chapter discusses two nonparametric approaches for approximating posteriors over continuous spaces with finitely many values. The first decomposes the state space into finitely many regions, and represents the posterior by a histogram. A histogram assigns to each region a single cumulative probability; they are best thought of as piecewise constant approximations to a continuous density. The second technique represents posteriors by finitely many samples. The resulting filter is known as *particle filter* and has become immensely popular in robotics.

Both types of techniques, histograms and particle filters, do not make strong parametric assumptions on the posterior density. In particular, they are well-suited to represent complex multimodal beliefs. For this reason, they are often the method of choice when a robot has to cope with phases of global uncertainty, and when it faces hard data association problems that yield separate, distinct hypotheses. However, the representational power of