Chapter 5

Filtering

In Chapter 4, we developed state-space models that describe the dynamic behavior of time-varying processes and how the sensor measurements relate to the state. Unfortunately, the estimation methods developed earlier do not work for such dynamic problems: These solve the estimation problem in the static case, but not when the parameters (i.e., the states) vary between samples. Instead, new methods are required which are able to combine the prior information from the dynamic model and the measurements at different points in time. This methodology is called filtering and the general approach is discussed in Section 5.1, which is followed by the exact solution of the filtering problem for linear state-space models in Section 5.2. Approximate filtering methods for nonlinear systems are discussed in Section 5.3–5.5.

5.1 The Filtering Approach

The basic discussion of the filtering approach is based on the general discrete-time state-space model of the form

$$\boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}) + \boldsymbol{q}_n, \tag{5.1a}$$

$$\boldsymbol{y}_n = g(\boldsymbol{x}_n) + \boldsymbol{r}_n, \tag{5.1b}$$

where the process and measurement noises are zero-mean ($\mathrm{E}\{\boldsymbol{q}_n\} = \mathrm{E}\{\boldsymbol{r}_n\} = 0$) with covariances $\mathrm{Cov}\{\boldsymbol{q}_n\} = \boldsymbol{Q}_n$ and $\mathrm{Cov}\{\boldsymbol{r}_n\} = \boldsymbol{R}_n$, respectively. Note that the dynamic model (5.1a) may be an inherently discrete-time model (Section 4.2) or the result of discretizing a continuous-time dynamic model.

An aspect of the state-space model that has been neglected so far are the initial conditions. Since the dynamic model (5.1a) is based on a differential (difference) equation, it also requires knowledge of the initial conditions of the system. Naturally, in the sensor fusion and estimation context, the initial conditions are unknown (e.g., it is not known where exactly a target is located in the beginning). Instead, it is suitable to specify the initial conditions in a probabilistic way. In other words, we assume that the state at t_0, denoted as $\boldsymbol{x}_0 \triangleq \boldsymbol{x}(t_0)$, is a random variable which follows a probability density function

$p(\boldsymbol{x}_0)$, that is,

$$\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0).$$

This allows us to specify the prior knowledge about where the initial state may be but also take into account that this is an uncertain guess. In practice, the distribution of the initial state is often assumed to follow a Gaussian distribution, but this does not need to be the case. Here, we only make the assumptions that the mean and covariance of the initial state are given by

$$\mathrm{E}\{\boldsymbol{x}_0\} = \boldsymbol{m}_0, \tag{5.2a}$$
$$\mathrm{Cov}\{\boldsymbol{x}_0\} = \boldsymbol{P}_0. \tag{5.2b}$$

Given the state-space model defined by (5.1) and (5.2), the filtering approach can now be formulated. In particular, filtering iterates between 1) a *prediction* step, which predicts the current state using the dynamic model and the previous estimate of the state (also called *time update*), and 2) a *measurement update* step that estimates the current state using the prediction and the new measurement. This prediction–update strategy is actually very general, and as it will be shown, all the algorithms discussed in this chapter make use of these two steps.

Prediction. The aim of the prediction step is to estimate the current state $\boldsymbol{x}_n$ at t_n given the set of all the previous measurement data $\boldsymbol{y}_{1:n-1} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n-1}\}$, and hence, given the estimate of the state $\boldsymbol{x}_{n-1}$ at t_{n-1}. Here, the mean of the prediction is denoted as $\hat{\boldsymbol{x}}_{n|n-1}$, which is read as "the estimate of the state at t_n given the data up to t_{n-1}". The hat indicates that it is an estimation, whereas the subscript is closely related to the notation for conditional expectations and densities, that is, it is read as "n given $n-1$". Similarly, the covariance of the predicted $\boldsymbol{x}_n$ is denoted as $\boldsymbol{P}_{n|n-1}$.

Measurement Update. In the measurement update, the newly obtained measurement $\boldsymbol{y}_n$ is used together with the prediction (and its uncertainty) to estimate the current value of the state $\boldsymbol{x}_n$. Similar to the prediction, the mean of the updated state estimate at time t_n is denoted as $\hat{\boldsymbol{x}}_{n|n}$, which is read as "the estimate of the state at t_n given the data up to t_n". In other words, this estimate is now based on the set of all measurements including the latest measurement at t_n given by $\boldsymbol{y}_{1:n} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$. Again, the covariance of the updated estimate is denoted as $\boldsymbol{P}_{n|n}$.

5.2 Kalman Filter

In this section, filtering for linear state-space models is considered. Recall that the linear state-space model with continuous-time dynamic model is (see Section 4.1)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{w}(t), \tag{5.3a}$$
$$\boldsymbol{y}_n = \boldsymbol{G}_n\boldsymbol{x}(t_n) + \boldsymbol{r}_n. \tag{5.3b}$$

Together with the initial distribution (5.2), the model (5.3) has the discrete-time equivalent

$$\boldsymbol{x}_n = \boldsymbol{F}_n \boldsymbol{x}_{n-1} + \boldsymbol{q}_n \tag{5.4a}$$
$$\boldsymbol{y}_n = \boldsymbol{G}_n \boldsymbol{x}_n + \boldsymbol{r}_n \tag{5.4b}$$

with

$$\mathrm{E}\{\boldsymbol{x}_0\} = \boldsymbol{m}_0, \ \mathrm{Cov}\{\boldsymbol{x}_0\} = \boldsymbol{P}_0, \tag{5.5a}$$
$$\mathrm{E}\{\boldsymbol{q}_n\} = 0, \ \mathrm{Cov}\{\boldsymbol{q}_n\} = \boldsymbol{Q}_n, \tag{5.5b}$$
$$\mathrm{E}\{\boldsymbol{r}_n\} = 0, \ \mathrm{Cov}\{\boldsymbol{r}_n\} = \boldsymbol{R}_n. \tag{5.5c}$$

Based on (5.4)–(5.5), the prediction and update steps can now be derived. For ease of presentation, the measurement update step is presented first, followed by the prediction.

Measurement Update. For the measurement update at time t_n, assume that there exists a prediction of the mean of the state $\hat{\boldsymbol{x}}_{n|n-1}$ and its covariance $\boldsymbol{P}_{n|n-1}$. This can be seen as the *prior knowledge* of the state at t_n. Then, the new measurement $\boldsymbol{y}_n$ provides the actual (noisy) information about the true state. Hence, the objective is to minimize the error with respect to the measurement, taking the prior information (prediction) into account.

Recall that we have based our estimators mainly on cost functions in general and quadratic cost functions (least squares) in particular. A cost function that is particularly well suited for this kind of problem is the *regularized least squares* cost function. Hence, for the filtering problem with the linear measurement model, the cost function becomes

$$\begin{aligned} J_{\mathrm{ReLS}}(\boldsymbol{x}_n) = (\boldsymbol{y}_n - \boldsymbol{G}_n \boldsymbol{x}_n)^{\mathsf{T}} \boldsymbol{R}_n^{-1} (\boldsymbol{y}_n - \boldsymbol{G}_n \boldsymbol{x}_n) \\ + (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})^{\mathsf{T}} \boldsymbol{P}_{n|n-1}^{-1} (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1}), \end{aligned} \tag{5.6}$$

where the first term accounts for the measurement $\boldsymbol{y}_n$ and the second term, the regularization term, accounts for the prior information from the prediction.

To estimate the state, we can now solve the regularized least squares problem

$$\hat{\boldsymbol{x}}_{n|n} = \operatorname*{argmin}_{\boldsymbol{x}_n} J_{\mathrm{ReLS}}(\boldsymbol{x}_n). \tag{5.7}$$

Since the cost function is completely linear in the unknown $\boldsymbol{x}_n$, solving (5.7) follows the same steps as solving the linear least squares problems in Chapter 2 and we can find a closed-form solution.

Fortunately, we have already solved this problem for the static case. The solution is given by (see Section 2.4)

$$\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - \boldsymbol{G}_n \hat{\boldsymbol{x}}_{n|n-1}) \tag{5.8}$$

where $\boldsymbol{K}_n$ is called the *Kalman gain* that is given by

$$\boldsymbol{K}_n = \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n^{\mathsf{T}} (\boldsymbol{G}_n \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n^{\mathsf{T}} + \boldsymbol{R}_n)^{-1}. \tag{5.9}$$

Furthermore, we have also shown that the covariance of $\hat{\boldsymbol{x}}_{n|n}$ is

$$\boldsymbol{P}_{n|n} = \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n(\boldsymbol{G}_n\boldsymbol{P}_{n|n-1}\boldsymbol{G}_n^\mathsf{T} + \boldsymbol{R}_n)\boldsymbol{K}_n^\mathsf{T}. \tag{5.10}$$

An important property of the measurement update step is that the updated estimate and its covariance actually are the mean and covariance of the state given the set of all measurements $\boldsymbol{y}_{1:n} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$ (Särkkä, 2013). In other words, it holds that

$$\mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n}\} = \hat{\boldsymbol{x}}_{n|n}, \tag{5.11a}$$

$$\mathrm{Cov}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n}\} = \boldsymbol{P}_{n|n}, \tag{5.11b}$$

where $\mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n}\}$ denotes the conditional expectation of $\boldsymbol{y}_n$ given the data $\boldsymbol{y}_{1:n}$ (and similar for the covariance).

Prediction. Given the estimated mean $\mathrm{E}\{\boldsymbol{x}_{n-1} \mid \boldsymbol{y}_{1:n-1}\} = \hat{\boldsymbol{x}}_{n-1|n-1}$ and its covariance $\mathrm{Cov}\{\boldsymbol{x}_{n-1} \mid \boldsymbol{y}_{1:n-1}\} = \boldsymbol{P}_{n-1|n-1}$ from the previous time step t_{n-1}, the predicted mean and covariance can now be calculated. The mean is given by

$$\hat{\boldsymbol{x}}_{n|n-1} = \mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\}.$$

Substituting $\boldsymbol{x}_n$ in the expectation using the dynamic model given in (5.4), the mean can be rewritten as

$$\mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} = \mathrm{E}\{\boldsymbol{F}_n\boldsymbol{x}_{n-1} + \boldsymbol{q}_n \mid \boldsymbol{y}_{1:n-1}\}.$$

Distributing the expectation over the sum and noting that $\mathrm{E}\{\boldsymbol{q}_n \mid \boldsymbol{y}_{1:n-1}\} = 0$ yields

$$\begin{aligned}\mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} &= \boldsymbol{F}_n \mathrm{E}\{\boldsymbol{x}_{n-1} \mid \boldsymbol{y}_{1:n-1}\} \\ &= \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1|n-1},\end{aligned} \tag{5.12}$$

where the last equality makes use of (5.11).

Similarly, the covariance is found from

$$\begin{aligned}\boldsymbol{P}_{n|n-1} &= \mathrm{Cov}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} \\ &= \mathrm{E}\{(\boldsymbol{x}_n - \mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\})(\boldsymbol{x}_n - \mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\}.\end{aligned}$$

Using the expression of the dynamic model (5.4) for $\boldsymbol{x}_n$ and (5.12) for $\mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\}$ yields

$$\begin{aligned}&\mathrm{Cov}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} \\ &\quad= \mathrm{E}\{(\boldsymbol{F}_n\boldsymbol{x}_{n-1} + \boldsymbol{q}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1|n-1})(\boldsymbol{F}_n\boldsymbol{x}_{n-1} + \boldsymbol{q}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\}.\end{aligned}$$

By rewriting and expanding the quadratic term and distributing the expectation we obtain

$$\begin{aligned}\mathrm{Cov}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} = \mathrm{E}\{&(\boldsymbol{F}_n\boldsymbol{x}_{n-1} - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1|n-1})(\boldsymbol{F}_n\boldsymbol{x}_{n-1} - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T}\} \\ &+ \mathrm{E}\{\boldsymbol{q}_n(\boldsymbol{F}_n\boldsymbol{x}_{n-1} - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\ &+ \mathrm{E}\{(\boldsymbol{F}_n\boldsymbol{x}_{n-1} - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1|n-1})\boldsymbol{q}_n^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} + \mathrm{E}\{\boldsymbol{q}_n\boldsymbol{q}_n^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\}.\end{aligned}$$

Algorithm 5.1 Kalman Filter

1: Initialize $\hat{\boldsymbol{x}}_{0|0} = \boldsymbol{m}_0$, $\boldsymbol{P}_{0|0} = \boldsymbol{P}_0$
2: **for** $n = 1, 2, \ldots$ **do**
3: Prediction (time update):

$$\begin{aligned}\hat{\boldsymbol{x}}_{n|n-1} &= \boldsymbol{F}_n \hat{\boldsymbol{x}}_{n-1|n-1} \\ \boldsymbol{P}_{n|n-1} &= \boldsymbol{F}_n \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_n^\mathsf{T} + \boldsymbol{Q}_n\end{aligned}$$

4: Measurement update:

$$\begin{aligned}\boldsymbol{K}_n &= \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n^\mathsf{T} (\boldsymbol{G}_n \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n + \boldsymbol{R}_n)^{-1} \\ \hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - \boldsymbol{G}_n \hat{\boldsymbol{x}}_{n|n-1}) \\ \boldsymbol{P}_{n|n} &= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n (\boldsymbol{G}_n \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n + \boldsymbol{R}_n) \boldsymbol{K}_n^\mathsf{T}\end{aligned}$$

5: **end for**

Noting that the middle terms are zero due to the factors being independent and $\boldsymbol{q}_n$ being zero-mean, what remains is

$$\begin{aligned}\operatorname{Cov}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} = \operatorname{E}\{&(\boldsymbol{F}_n \boldsymbol{x}_{n-1} - \boldsymbol{F}_n \hat{\boldsymbol{x}}_{n-1|n-1})(\boldsymbol{F}_n \boldsymbol{x}_{n-1} - \boldsymbol{F}_n \hat{\boldsymbol{x}}_{n-1|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\ &+ \operatorname{E}\{\boldsymbol{q}_n \boldsymbol{q}_n^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\},\end{aligned}$$

and finally,

$$\operatorname{Cov}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} = \boldsymbol{F}_n \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_n^\mathsf{T} + \boldsymbol{Q}_n. \tag{5.13}$$

Kalman Filter. Gathering the results (5.8)–(5.10) and (5.12)–(5.13), the filtering algorithm for linear state-space models can now be formulated. First, during the prediction (time update) step the predicted mean and covariance are calculated according to

$$\hat{\boldsymbol{x}}_{n|n-1} = \boldsymbol{F}_n \hat{\boldsymbol{x}}_{n-1|n-1}, \tag{5.14a}$$

$$\boldsymbol{P}_{n|n-1} = \boldsymbol{F}_n \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_n^\mathsf{T} + \boldsymbol{Q}_n. \tag{5.14b}$$

Second, in the measurement update step, the new measurement is incorporated and the new state is estimated using

$$\boldsymbol{K}_n = \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n^\mathsf{T} (\boldsymbol{G}_n \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n^\mathsf{T} + \boldsymbol{R}_n)^{-1}, \tag{5.15a}$$

$$\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - \boldsymbol{G}_n \hat{\boldsymbol{x}}_{n|n-1}), \tag{5.15b}$$

$$\boldsymbol{P}_{n|n} = \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n (\boldsymbol{G}_n \boldsymbol{P}_{n|n-1} \boldsymbol{G}_n^\mathsf{T} + \boldsymbol{R}_n) \boldsymbol{K}_n^\mathsf{T}. \tag{5.15c}$$

Furthermore, the recursion is initialized by letting $\hat{\boldsymbol{x}}_{0|0} = \boldsymbol{m}_0$ and $\boldsymbol{P}_{0|0} = \boldsymbol{P}_0$. This leads to the algorithm shown in Algorithm 5.1, which is called the *Kalman filter* (KF) (Kalman, 1960).

It is worth pointing out several aspects of the prediction and measurement update (5.14)–(5.15). First, note that in the prediction, the covariance increases due to the scaling factor $\boldsymbol{F}_n$ and the added uncertainty by the process noise. On the other hand, during the measurement update, the covariance is decreased by a factor that scales quadratically with the gain $\boldsymbol{K}_n$. This follows the intuition that a prediction should increase the uncertainty, while incorporating new information should decrease it.

Second, the measurement update is a sum of the prediction $\hat{\boldsymbol{x}}_{n|n-1}$ and the term $\boldsymbol{y}_n - \boldsymbol{G}_n\hat{\boldsymbol{x}}_{n|n-1}$ scaled by the gain $\boldsymbol{K}_n$. Noting that $\boldsymbol{G}_n$ actually is the matrix relating the state $\boldsymbol{x}_n$ to the measurement $\boldsymbol{y}_n$, the term $\boldsymbol{G}_n\hat{\boldsymbol{x}}_{n|n-1}$ can be interpreted as a prediction of the output, which it in fact is. To show this, consider the expected value of the output $\boldsymbol{y}_n$ given all the data up to t_{n-1}, that is,

$$\begin{aligned}\mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} &= \mathrm{E}\{\boldsymbol{G}_n\boldsymbol{x}_n + \boldsymbol{r}_n \mid \boldsymbol{y}_{1:n-1}\} \\ &= \boldsymbol{G}_n\,\mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} + \mathrm{E}\{\boldsymbol{r}_n \mid \boldsymbol{y}_{1:n-1}\}\end{aligned}$$

and thus

$$\mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} = \boldsymbol{G}_n\hat{\boldsymbol{x}}_{n|n-1}. \tag{5.16}$$

Hence, the difference between the measurement and its prediction gives an indication about how far the predicted state is from the true state: If the difference is large, the predicted state is far and should be corrected a lot, if it is small, it is close and does not need to be corrected much. Hence, this difference is also called the *innovation*.

Third, the covariance of the predicted output $\boldsymbol{y}_n$ is given by

$$\begin{aligned}\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} &= \mathrm{E}\{(\boldsymbol{y}_n - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\})(\boldsymbol{y}_n - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\ &= \mathrm{E}\{(\boldsymbol{G}_n\boldsymbol{x}_n + \boldsymbol{r}_n - \boldsymbol{G}_n\hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{G}_n\boldsymbol{x}_n + \boldsymbol{r}_n - \boldsymbol{G}_n\hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\ &= \mathrm{E}\{(\boldsymbol{G}_n(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T}\boldsymbol{G}_n^\mathsf{T} + \boldsymbol{r}_n\boldsymbol{r}_n^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\}\end{aligned}$$

and finally,

$$\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} = \boldsymbol{G}_n\boldsymbol{P}_{n|n-1}\boldsymbol{G}_n^\mathsf{T} + \boldsymbol{R}_n, \tag{5.17}$$

which is the denominator of the Kalman gain $\boldsymbol{K}_n$. Similarly, the cross-covariance between the predicted state $\boldsymbol{x}_n$ and predicted output $\boldsymbol{y}_n$ is

$$\begin{aligned}\mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} &= \mathrm{E}\{(\boldsymbol{x}_n - \mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\})(\boldsymbol{y}_n - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\ &= \mathrm{E}\{(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{G}_n\boldsymbol{x}_n + \boldsymbol{r}_n - \boldsymbol{G}_n\hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\ &= \mathrm{E}\{(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T}\boldsymbol{G}_n^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\},\end{aligned}$$

which yields

$$\mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} = \boldsymbol{P}_{n|n-1}\boldsymbol{G}_n^\mathsf{T}. \tag{5.18}$$

Thus, using (5.16)–(5.18), the measurement update can be written in the alternative form as

$$\boldsymbol{K}_n = \mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}\,\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}^{-1}, \tag{5.19a}$$

$$\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n(\boldsymbol{y}_n - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}), \tag{5.19b}$$

$$\boldsymbol{P}_{n|n} = \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n\,\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}\boldsymbol{K}_n^\mathsf{T}. \tag{5.19c}$$

From (5.19), we can see that the denominator of the Kalman gain $\boldsymbol{K}_n$ is the covariance of the predicted measurement. Hence, if the uncertainty of the prediction is large (either due to a large uncertainty in the predicted state or due to large measurement noise covariance $\boldsymbol{R}_n$), the gain becomes small. This in turn means that the innovation only contributes little information to the updated state. Conversely, if the uncertainty of the prediction is small, the gain becomes large and the innovation contributes a lot.

5.3 Extended Kalman Filter

Similar to the static, linear case discussed in Chapter 2, the Kalman filter is the closed-form solution for the state estimation problem in linear state-space models and thus an exact solution. However, if we consider general nonlinear state-space models of the form (5.1)–(5.2), closed-form solutions are normally not available. Similar to the nonlinear static case, we need approximative solutions instead. A first approximation is found in the *extended Kalman filter* (EKF), which is based on a linearization using a Taylor series approximation of the nonlinear dynamic and measurement models. Using this linearization, prediction and measurement update steps similar to the Kalman filter are found.

Prediction. For the prediction, assume that the state estimate and its covariance for t_{n-1} are given by $\hat{\boldsymbol{x}}_{n-1|n-1}$ and $\boldsymbol{P}_{n-1|n-1}$, respectively. Then, the linear (first order) Taylor series approximation of the dynamic model around the linearization point $\boldsymbol{x}_{n-1} = \hat{\boldsymbol{x}}_{n-1|n-1}$ is given by

$$\begin{aligned} \boldsymbol{x}_n &= f(\boldsymbol{x}_{n-1}) + \boldsymbol{q}_n \\ &\approx f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n, \end{aligned} \tag{5.20}$$

where $\boldsymbol{F}_x$ is the Jacobian matrix of the vector-value function $f(\boldsymbol{x}_{n-1})$. Then, based on the linearized dynamic model (5.20), we can predict the mean and covariance of the state $\boldsymbol{x}_n$ at t_n given the set of all the measurements $\boldsymbol{y}_{1:n-1} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n-1}\}$.

The mean is given by

$$\begin{aligned} \hat{\boldsymbol{x}}_{n|n-1} &= \mathrm{E}\{\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n-1}\} \\ &\approx \mathrm{E}\{f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n \mid \boldsymbol{y}_{1:n-1}\} \\ &= f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x \,\mathrm{E}\{\boldsymbol{x}_{n-1} \mid \boldsymbol{y}_{1:n-1}\} - \boldsymbol{F}_x\hat{\boldsymbol{x}}_{n-1|n-1} \\ &= f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x\hat{\boldsymbol{x}}_{n-1|n-1} - \boldsymbol{F}_x\hat{\boldsymbol{x}}_{n-1|n-1}, \end{aligned}$$

and thus

$$\hat{\boldsymbol{x}}_{n|n-1} = f(\hat{\boldsymbol{x}}_{n-1|n-1}). \tag{5.21}$$

Similarly, the covariance is

$$\begin{aligned}
\boldsymbol{P}_{n|n-1} &= \mathrm{E}\{(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\
&\approx \mathrm{E}\{(f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n - f(\hat{\boldsymbol{x}}_{n-1|n-1})) \\
&\qquad \times (f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n - f(\hat{\boldsymbol{x}}_{n-1|n-1}))^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\
&= \mathrm{E}\{(\boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n)(\boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n)^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\} \\
&= \boldsymbol{F}_x \,\mathrm{E}\{(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1})(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1})^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\}\boldsymbol{F}_x^\mathsf{T} \\
&\qquad + \mathrm{E}\{\boldsymbol{q}_n \boldsymbol{q}_n^\mathsf{T} \mid \boldsymbol{y}_{1:n-1}\}
\end{aligned}$$

and finally,

$$\boldsymbol{P}_{n|n-1} = \boldsymbol{F}_x \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_x^\mathsf{T} + \boldsymbol{Q}_n. \tag{5.22}$$

Measurement Update. The derivation of the measurement update follows very similar steps as the prediction in the EKF and the measurement update in the KF. Given the predicted mean $\boldsymbol{x}_{n|n-1}$ and covariance $\boldsymbol{P}_{n|n-1}$, the nonlinear measurement model is linearized using a first order Taylor series expansion around the prediction $\hat{\boldsymbol{x}}_{n|n-1}$. This yields

$$\begin{aligned}
\boldsymbol{y}_n &= g(\boldsymbol{x}_n) + \boldsymbol{r}_n \\
&= g(\hat{\boldsymbol{x}}_{n|n-1}) + \boldsymbol{G}_x(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1}) + \boldsymbol{r}_n,
\end{aligned} \tag{5.23}$$

where $\boldsymbol{G}_x$ denotes the Jacobian matrix of $g(\boldsymbol{x}_n)$ evaluated at $\boldsymbol{x}_n = \hat{\boldsymbol{x}}_{n|n-1}$.

Next, introducing the regularized least squares criterion for the linearized model (5.23) yields

$$\begin{aligned}
J_{\mathrm{ReLS}}(\boldsymbol{x}_n) = (\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1}) - \boldsymbol{G}_x(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1}))^\mathsf{T} \boldsymbol{R}_n^{-1} \\
\times (\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1}) - \boldsymbol{G}_x(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})) \\
+ (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T} \boldsymbol{P}_{n|n-1}^{-1} (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1}).
\end{aligned} \tag{5.24}$$

By introducing a change of variables with $\boldsymbol{z}_n = \boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1}) + \boldsymbol{G}_x \hat{\boldsymbol{x}}_{n|n-1}$, (5.24) can be rewritten as

$$\begin{aligned}
J_{\mathrm{ReLS}}(\boldsymbol{x}_n) = (\boldsymbol{z}_n - \boldsymbol{G}_x \boldsymbol{x}_n)^\mathsf{T} \boldsymbol{R}_n^{-1} (\boldsymbol{z}_n - \boldsymbol{G}_x \boldsymbol{x}_n) \\
+ (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1})^\mathsf{T} \boldsymbol{P}_{n|n-1}^{-1} (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1}),
\end{aligned} \tag{5.25}$$

which is linear in $\boldsymbol{x}_n$ and has the same form as (5.6) but with the Jacobian matrix $\boldsymbol{G}_x$ rather than the measurement matrix $\boldsymbol{G}_n$. Hence, solving

$$\hat{\boldsymbol{x}}_{n|n} = \operatorname*{argmin}_{\boldsymbol{x}_n} J_{\mathrm{ReLS}}(\boldsymbol{x}_n) \tag{5.26}$$

yields

$$\begin{aligned}
\hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n(\boldsymbol{z}_n - \boldsymbol{G}_x \hat{\boldsymbol{x}}_{n|n-1}), \\
\boldsymbol{K}_n &= \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} + \boldsymbol{R}_n)^{-1},
\end{aligned}$$

Algorithm 5.2 Extended Kalman Filter

1: Initialize $\hat{\boldsymbol{x}}_{0|0} = \boldsymbol{m}_0$, $\boldsymbol{P}_{0|0} = \boldsymbol{P}_0$
2: **for** $n = 1, 2, \ldots$ **do**
3: Prediction (time update):

$$\begin{aligned}\hat{\boldsymbol{x}}_{n|n-1} &= f(\hat{\boldsymbol{x}}_{n-1|n-1}) \\ \boldsymbol{P}_{n|n-1} &= \boldsymbol{F}_x \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_x^\mathsf{T} + \boldsymbol{Q}_n\end{aligned}$$

4: Measurement update:

$$\begin{aligned}\boldsymbol{K}_n &= \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} + \boldsymbol{R}_n)^{-1} \\ \hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1})) \\ \boldsymbol{P}_{n|n} &= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} + \boldsymbol{R}_n) \boldsymbol{K}_n^\mathsf{T}\end{aligned}$$

5: **end for**

with covariance

$$\boldsymbol{P}_{n|n} \approx \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} + \boldsymbol{R}_n) \boldsymbol{K}_n^\mathsf{T}.$$

Finally, changing $\boldsymbol{z}_n$ back to its definition yields

$$\begin{aligned}\hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1}) + \boldsymbol{G}_x \hat{\boldsymbol{x}}_{n|n-1} - \boldsymbol{G}_x \hat{\boldsymbol{x}}_{n|n-1}) \\ &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1}))\end{aligned}$$

for the measurement update.

Extended Kalman Filter. We can now summarize the prediction and measurement update steps for the EKF. First, during the predictions step, the predicted mean and covariance are calculated according to

$$\hat{\boldsymbol{x}}_{n|n-1} = f(\hat{\boldsymbol{x}}_{n-1|n-1}), \tag{5.27a}$$

$$\boldsymbol{P}_{n|n-1} = \boldsymbol{F}_x \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_x^\mathsf{T} + \boldsymbol{Q}_n, \tag{5.27b}$$

where $\boldsymbol{F}_x$ is the Jacobian matrix of the dynamic model. Second, the measurement update is

$$\boldsymbol{K}_n = \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} + \boldsymbol{R}_n)^{-1}, \tag{5.28a}$$

$$\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1})), \tag{5.28b}$$

$$\boldsymbol{P}_{n|n} = \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^\mathsf{T} + \boldsymbol{R}_n) \boldsymbol{K}_n^\mathsf{T}. \tag{5.28c}$$

These two steps are then performed iteratively, and the algorithm is initialized with the initial conditions. This yields the complete EKF algorithm shown in Algorithm 5.2.

Note that this is essentially the same algorithm as the original KF where the nonlinear function takes the place of the prediction (both in the prediction step and the prediction of the output) and in the covariance updates, the Jacobian matrices take the place of the matrices in the dynamic model and measurement model. Furthermore, if either of the models is linear, the corresponding step (prediction or measurement update) may be replaced by an exact update step from the Kalman filter in Section 5.2. Finally, note that Algorithm 5.2 is an approximation of the filtering problem since the nonlinear function is approximated around the filtered and predicted state, meaning that both the state estimate and its covariance may or may not converge to the true state (similar as for static nonlinear problems).

5.4 Unscented Kalman Filtering

One of the major problems of the EKF is that the linearization is local, which often leads to problems such as the covariance of the state being underestimated (which in turn affects the Kalman gain in the next iteration). Another approach to solve the filtering problem within the Kalman filtering framework is the *unscented Kalman filter* (UKF) (Wan and Van Der Merwe, 2000; Julier and Uhlmann, 2004), which makes use of a so-called *unscented transform*. Hence, this transform will be discussed first, and then the prediction and measurement update steps for the UKF are derived.

Unscented Transform. The unscented transform is based on calculating the moments of the nonlinear transformation of a finite set of deterministic sampling points as follows. Given a random variable $\boldsymbol{x}$ with mean $\boldsymbol{m}$ and covariance $\boldsymbol{P}$, we can find a set of J points $\{\boldsymbol{x}^0, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{J-1}\}$, so-called *sigma-points*, such that their weighted sum is equal to the mean and covariance of $\boldsymbol{x}$, that is, such that

$$\boldsymbol{m} = \sum_{j=0}^{J-1} w_m^j \boldsymbol{x}^j, \tag{5.29a}$$

$$\boldsymbol{P} = \sum_{j=0}^{J-1} w_P^j (\boldsymbol{x}^j - \boldsymbol{m})(\boldsymbol{x}^j - \boldsymbol{m})^\mathsf{T}, \tag{5.29b}$$

where it must hold that $\sum_{j=0}^{J-1} w_m^j = 1$ since the expected value of the sum should be unbiased.

Then, given the nonlinear function $\boldsymbol{z} = h(\boldsymbol{x})$, we can calculate the transformed sigma-points according to

$$\boldsymbol{z}^j = h(\boldsymbol{x}^j).$$

Based on the transformed sigma-points, the mean and covariance of the transformed variable $\boldsymbol{z}$ as well as the cross-covariance between $\boldsymbol{x}$ and $\boldsymbol{z}$ can then be calculated

according to

$$\mathrm{E}\{\boldsymbol{z}\} \approx \sum_{j=1}^{J} w_m^j \boldsymbol{z}^j, \tag{5.30a}$$

$$\mathrm{Cov}\{\boldsymbol{z}\} \approx \sum_{j=1}^{J} w_P^j (\boldsymbol{z}^j - \mathrm{E}\{\boldsymbol{z}\})(\boldsymbol{z}^j - \mathrm{E}\{\boldsymbol{z}\})^\mathsf{T}, \tag{5.30b}$$

$$\mathrm{Cov}\{\boldsymbol{x}, \boldsymbol{z}\} \approx \sum_{j=1}^{J} w_P^j (\boldsymbol{x}^j - \boldsymbol{m})(\boldsymbol{z}^j - \mathrm{E}\{\boldsymbol{z}\})^\mathsf{T}. \tag{5.30c}$$

The question then is how to chose the sigma-points $\boldsymbol{x}^j$ and their weights for the mean w_m^j and covariance w_P^j. The unscented transform is one such approach for choosing sigma-points that fulfill the conditions (5.29). Here, $J = 2L + 1$ (with L being the dimension of the vector $\boldsymbol{x}$) sigma-points are chosen according to

$$\boldsymbol{x}^0 = \boldsymbol{m}, \tag{5.31a}$$

$$\boldsymbol{x}^j = \boldsymbol{m} + \sqrt{L + \lambda}[\sqrt{\boldsymbol{P}}]_j, \qquad j = 1, \ldots, L, \tag{5.31b}$$

$$\boldsymbol{x}^j = \boldsymbol{m} - \sqrt{L + \lambda}[\sqrt{\boldsymbol{P}}]_{(j-L)}, \qquad j = L + 1, \ldots, 2L. \tag{5.31c}$$

In (5.31), $\sqrt{\boldsymbol{P}}$ denotes a matrix square root such that $\boldsymbol{P} = \sqrt{\boldsymbol{P}}\sqrt{\boldsymbol{P}}^\mathsf{T}$ (in practice, this can be implemented using the Cholesky factorization) and $[\sqrt{\boldsymbol{P}}]_j$ denotes the jth column of the matrix $\sqrt{\boldsymbol{P}}$. The weights corresponding to the above sigma-points are

$$w_m^0 = \frac{\lambda}{L + \lambda}, \tag{5.32a}$$

$$w_P^0 = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta), \tag{5.32b}$$

$$w_m^j = w_P^j = \frac{1}{2(L + \lambda)}, \qquad j = 1, \ldots, 2L. \tag{5.32c}$$

In both (5.31) and (5.32), λ is

$$\lambda = \alpha^2(L + \kappa) - L, \tag{5.33}$$

and α, β, and κ are tuning parameters. The parameters α and κ determine the scaling of the spread of the sigma-points in the direction of the covariance $\boldsymbol{P}$ whereas β is related the higher order moments of $\boldsymbol{x}$ and only affects the weight for the central sigma-point in the covariance calculations.

The choice of the tuning parameters may greatly affect the performance of the filter and several default choices have been suggested. The value for κ is most often chosen to be zero (i.e., $\kappa = 0$), which leaves the choices for α and β. First, note that with $\kappa = 0$, the weight of the central sigma-point becomes

$$w_m^0 = \frac{\lambda}{L + \lambda} = \frac{L\alpha^2 - L}{L + L\alpha^2 - L} = \frac{\alpha^2 - 1}{\alpha^2}, \tag{5.34}$$

and the scaling factor of the root of the covariance matrix becomes

$$\sqrt{L+\lambda} = \sqrt{L+\alpha^2 L - L} = \alpha\sqrt{L}. \tag{5.35}$$

One suggestion is to choose $\alpha = 1 \times 10^{-3}$ (Wan and Van Der Merwe, 2000). This gives a quite large and *negative* weight w_m^0 [see (5.34)] and a small scaling factor [see (5.35)], which makes the sigma-points lie close to the mean. To avoid this, it is sometimes more intuitive to instead start from the weight of the central point w_m^0 and then determine α based on reformulating (5.34), which yields

$$\alpha = \sqrt{\frac{1}{1-w_m^0}}. \tag{5.36}$$

Furthermore, since all the weights for $j > 0$ are the same [see (5.32)], it must also hold that

$$w_m^j = \frac{1-w_m^0}{2L}.$$

Finally, β only affects the central weight of the covariance and a good starting point is normally to chose $\beta = 2$ (see, for example, Wan and Van Der Merwe (2000)).

Prediction. The unscented transform as introduced above can directly be used in the prediction step to calculate the predicted mean and the covariance. In this case, the nonlinear transformation is the function of the dynamic model, that is, $f(\boldsymbol{x}_{n-1})$. The sigma-points are calculated using the estimated mean and covariance at t_n, that is,

$$\boldsymbol{x}_{n-1}^0 = \hat{\boldsymbol{x}}_{n-1|n-1} \tag{5.37a}$$

$$\boldsymbol{x}_{n-1}^j = \hat{\boldsymbol{x}}_{n-1|n-1} + \sqrt{L+\lambda}\left[\sqrt{\boldsymbol{P}_{n-1|n-1}}\right]_j, \qquad j = 1,\ldots,L, \tag{5.37b}$$

$$\boldsymbol{x}_{n-1}^j = \hat{\boldsymbol{x}}_{n-1|n-1} - \sqrt{L+\lambda}\left[\sqrt{\boldsymbol{P}_{n-1|n-1}}\right]_{(j-L)}, \qquad j = L+1,\ldots,2L. \tag{5.37c}$$

The weights of the sigma-points are given by (5.32) and λ is as in (5.33).

Then, the moments of the transformed variable, that is, the moments of the prediction become

$$\boldsymbol{x}_n^j = f(\boldsymbol{x}_{n-1}^j), \qquad j = 0,\ldots,2L, \tag{5.38a}$$

$$\hat{\boldsymbol{x}}_{n|n-1} = \sum_{j=0}^{2L} w_m^j \boldsymbol{x}_n^j, \tag{5.38b}$$

$$\boldsymbol{P}_{n|n-1} = \sum_{j=0}^{2L} w_c^j (\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n-1})^{\mathsf{T}} + \boldsymbol{Q}_n. \tag{5.38c}$$

Note that the additional term $\boldsymbol{Q}_n$ in the covariance is due to the fact that the dynamic model also includes the process noise term $\boldsymbol{q}_n$, which increases the uncertainty. In other words, the unscented transform only calculates the covariance of the $\boldsymbol{x}_{n-1}$ transformed by $f(\boldsymbol{x}_{n-1})$ and does not take the effect of the noise into account in this form.

Measurement Update. For the measurement update, first recall that it can be written in terms of the predicted output, its covariance, and the covariance between the predicted output and the state, see (5.19). Hence, we can also use the unscented transform to calculate these means and covariances.

In this case, the sigma points are calculated from the predicted mean $\hat{\boldsymbol{x}}_{n|n-1}$ and the covariance $\boldsymbol{P}_{n|n-1}$ according to

$$\boldsymbol{x}_n^0 = \hat{\boldsymbol{x}}_{n|n-1} \tag{5.39a}$$

$$\boldsymbol{x}_n^j = \hat{\boldsymbol{x}}_{n|n-1} + \sqrt{L+\lambda}\left[\sqrt{\boldsymbol{P}_{n|n-1}}\right]_j, \qquad j = 1, \ldots, L, \tag{5.39b}$$

$$\boldsymbol{x}_n^j = \hat{\boldsymbol{x}}_{n|n-1} - \sqrt{L+\lambda}\left[\sqrt{\boldsymbol{P}_{n|n-1}}\right]_{(j-L)}, \qquad j = L+1, \ldots, 2L, \tag{5.39c}$$

with the weights as in (5.32). Then, the necessary moments become

$$\boldsymbol{y}_n^j = g(\boldsymbol{x}_n^j), \qquad j = 0, \ldots, 2L, \tag{5.40a}$$

$$\mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} = \sum_{j=0}^{2L} w_m^j \boldsymbol{y}_n^j, \tag{5.40b}$$

$$\begin{aligned}\mathrm{Cov}\{\boldsymbol{y} \mid \boldsymbol{y}_{1:n-1}\} = &\sum_{j=0}^{2L} w_P^j (\boldsymbol{y}_n^j - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\})(\boldsymbol{y}_n^j - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\})^{\mathsf{T}} \\ &+ \boldsymbol{R}_n,\end{aligned} \tag{5.40c}$$

$$\mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} = \sum_{j=0}^{2L} w_P^j (\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n-1})(\boldsymbol{y}_n^j - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\})^{\mathsf{T}}, \tag{5.40d}$$

and the measurement update given in (5.19) can be performed.

Unscented Kalman Filter. Based on the prediction and measurement update discussed above, the unscented Kalman filter then becomes as shown in Algorithm 5.3. Note that the UKF still performs the two steps very similar to the original Kalman filter for linear systems, that is, it basically estimates the mean and covariance of the state at each time step and propagates this information between time steps, but without analytical linearization. The latter also implies that the UKF does not require any Jacobians to be calculated and is thus somewhat easier to implement.

5.5 Bootstrap Particle Filter

Another approach of solving the estimation problem in dynamic systems is particle filtering, which differs quite a lot from the Kalman filtering approaches discussed in the previous sections. Rather than being based on predicting and updating the mean and covariance from time step to time step, particle filtering propagates a set of weighted random samples (called *particles*) and hence there is some resemblance to the unscented

Algorithm 5.3 Unscented Kalman Filter

1: Initialize $\hat{\boldsymbol{x}}_{0|0} = \boldsymbol{m}_0$, $\boldsymbol{P}_{0|0} = \boldsymbol{P}_0$
2: **for** $n = 1, 2, \ldots$ **do**
3: Calculate $\boldsymbol{x}_{n-1}^j$, w_m^j, and w_P^j using (5.37) and (5.32)
4: Calculate $\hat{\boldsymbol{x}}_{n|n-1}$ and $\boldsymbol{P}_{n|n-1}$ using (5.38)
5: Calculate $\boldsymbol{x}_n^j$, w_m^j, and w_P^j using (5.39) and (5.32)
6: Calculate $\mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}$, $\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}$, and $\mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}$ using (5.40)
7: Measurement update:

$$\begin{aligned}
\boldsymbol{K}_n &= \mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}\,\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}^{-1} \\
\hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n(\boldsymbol{y}_n - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}) \\
\boldsymbol{P}_{n|n} &= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n\,\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}\boldsymbol{K}_n^{\mathsf{T}}
\end{aligned}$$

8: **end for**

Kalman filter. The particles can be seen as qualified but random guesses of the state whereas the weights provide an indication of how good the guess is. Particle filtering is a very general methodology and comprises a large class of filtering algorithms. Here, we focus on the *bootstrap particle filter*, which has an intuitive explanation but also a rigorous mathematical background (Gordon et al., 1993; Doucet and Johansen, 2011).

To derive the particle filter, we first take another look at the general discrete-time state-space model given by

$$\begin{aligned}
\boldsymbol{x}_n &= f(\boldsymbol{x}_{n-1}) + \boldsymbol{q}_n, \\
\boldsymbol{y}_n &= g(\boldsymbol{x}_{n-1}) + \boldsymbol{r}_n,
\end{aligned}$$

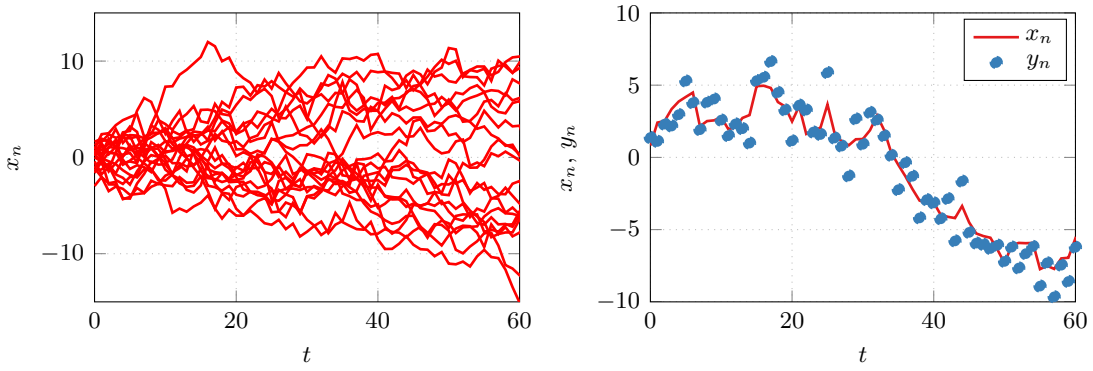
with $\boldsymbol{q}_n \sim p(\boldsymbol{q}_n)$ and $\boldsymbol{r}_n \sim p(\boldsymbol{r}_n)$. Recall that the dynamic model is a stochastic process, meaning that every time a system is observed, a new realization of that stochastic process is observed. For example, consider the scalar (one-dimensional) random walk dynamic model

$$x_n = x_{n-1} + q_n$$

with

$$\begin{aligned}
x_0 &\sim \mathcal{N}(0, 1), \\
q_n &\sim \mathcal{N}(0, 1).
\end{aligned}$$

This model essentially states that the initial state is a random variable distributed around $x_0 = 0$ with variance 1 and as time elapses, random steps from that initial point are taken. Then, at any time t_n, there are infinitely many random steps that can be taken through q_n (i.e., q_n can take on any value, but with some values more likely than others



(a) (b)

Figure 5.1: Random walk process example: (a) 20 realizations of the random walk process, and (b) one realization of the process together with its measurements.

as specified by the distribution of the random variable). This implies in turn that no two realizations of any random process are the same. As an example, Figure 5.1a illustrates 20 realization of the random walk process. As it can be seen, each realization takes a completely different path, and the realizations diverge as time elapses.

In practice, when estimating a system's state, all the observations (measurements) come from one single realization out of all the infinitely possible realizations of the process. The measurements then give us the information about the particular realization that is observed and the realization and the measurements are linked through the measurement model. For example, Figure 5.1b illustrates one particular realization of the random walk process together with the measurements from the linear measurement model

$$y_n = x_n + r_n$$

with $r_n \sim \mathcal{N}(0, 1)$.

This leads to a new strategy for filtering where we first simulate a set of trajectories and then evaluate how well each of these trajectories explain the measurements that we are observing. More formally, in the framework of the prediction and measurement update steps introduced in Section 5.1, this approach alternates between:

1. Prediction: Given a set of simulated states $\boldsymbol{x}_{n-1}^j$ (for $j = 1, \ldots, J$), simulate from t_{n-1} to t_n to obtain a set of simulated states $\boldsymbol{x}_n^j$ ($j = 1, \ldots, J$);

2. Measurement update: Evaluate how well the simulated states $\boldsymbol{x}_n^j$ explain the observed measurement $\boldsymbol{y}_n$.

For example, Figure 5.2 combines the different realizations of the state trajectories in Figure 5.1a (gray) with the measurements (blue dots) in Figure 5.1b (together with the true trajectory that generated the measurements in red). As it can be seen from

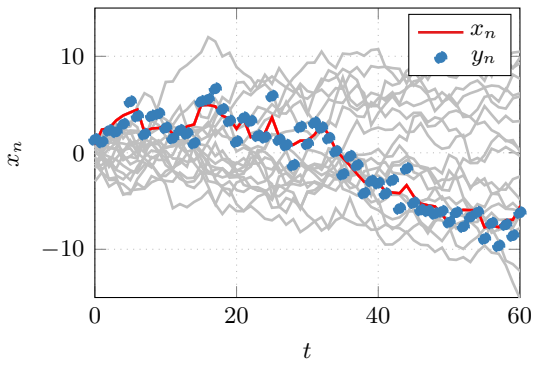


Figure 5.2: Random walk example with measurements y_n (blue) from one particular realization of the random walk process (red) and 20 other realizations.

the figure, several of the gray realizations could be the trajectories that generated the measurements in blue, at least at some points in time, whereas other realizations are very unlikely to be the trajectories that generated the measurements.

To develop a filtering algorithm based on this reasoning, we need to find a strategy to a) simulate samples $\boldsymbol{x}_n^j$ given the samples $\boldsymbol{x}_{n-1}^j$, and b) evaluate how well a particular sample $\boldsymbol{x}_n^j$ of the state explains the current measurement $\boldsymbol{y}_n$.

An intuitive way to generate new samples is to use the dynamic model to simulate one time step. In other words, we can generate a realization of the random variable $\boldsymbol{q}_n$ and use the nonlinear function $f(\boldsymbol{x}_{n-1})$ to pass the sample from t_{n-1} to t_n. This means performing the following two steps for each sample $j = 1, \ldots, J$:

1. Sample $\boldsymbol{q}_n^j \sim p(\boldsymbol{q}_n)$,
2. Calculate $\boldsymbol{x}_n^j = f(\boldsymbol{x}_{n-1}^j) + \boldsymbol{q}_n^j$.

In practice, the process noise $\boldsymbol{q}_n$ is often Gaussian, that is, $\boldsymbol{q}_n$ is a Gaussian random variable (e.g., when the discrete-time model comes from the discretization of a continuous-time model with a white noise process as the input). In that case, sampling $\boldsymbol{q}_n$ amounts to sampling from the normal distribution $\mathcal{N}(0, \boldsymbol{Q}_n)$ where $\boldsymbol{Q}_n$ is the covariance matrix of $\boldsymbol{q}_n$.

To evaluate the importance of each sample with respect to the current measurement $\boldsymbol{y}_n$, we assign a weight w_n^j (called the *importance weight*) to each sample $\boldsymbol{x}_n^j$. Intuitively, the weight should represent how well each sample explains the measurement, and thus, the closer the sample is to the true state, the higher the weight should be. If we ensure that the weights sum to one, that is, if

$$\sum_{j=1}^{J} w_n^j = 1,$$

we can also loosely interpret the weight w_n^j as the probability of the jth sample representing the correct state. The question then is how the weights should be calculated, given the

sample $\boldsymbol{x}_n^j$ and the measurement $\boldsymbol{y}_n$. So far, cost functions have been used to evaluate whether a state explains the measurements well. However, in this approach the cost should be low for a state that explains the measurements well, whereas the importance weight should be high and hence, we can not use cost functions[1].

Instead, we have another look at the measurement model. Recall that we have assumed that the measurement model is of the form

$$\boldsymbol{y}_n = g(\boldsymbol{x}_n) + \boldsymbol{r}_n,$$

where $\boldsymbol{r}_n \sim p(\boldsymbol{r}_n)$ is a random variable with probability density function $p(\boldsymbol{r}_n)$. Hence, $\boldsymbol{y}_n$ is a random variable too. Assuming that $\boldsymbol{x}_n$ is given, $\boldsymbol{y}_n$ follows the same distribution as $\boldsymbol{r}_n$ but offset by the constant term $g(\boldsymbol{x}_n)$. We can express this probability density function for $\boldsymbol{y}_n$ as $p(\boldsymbol{y}_n \mid \boldsymbol{x}_n)$ which is read as "the probability density function of $\boldsymbol{y}_n$ given that $\boldsymbol{x}_n$ is known". The probability density function $p(\boldsymbol{y}_n \mid \boldsymbol{x}_n)$ is commonly known as the *likelihood* because it indicates how likely a certain state $\boldsymbol{x}_n$ is when observing the measurement $\boldsymbol{y}_n$. Hence, the likelihood is a suitable measure for how well any given sample $\boldsymbol{x}_n^j$ explains the measurement $\boldsymbol{y}_n$, and we can define the non-normalized weight $\tilde{w}_n^j$ as

$$\tilde{w}_n^j = p(\boldsymbol{y}_n \mid \boldsymbol{x}_n^j).$$

$\tilde{w}_n^j$ is non-normalized because the sum over all weights does generally not fulfill the requirement that the weights should sum to one. This can be ensured by simply dividing each non-normalized weight by the sum of all the non-normalized weights, that is,

$$w_n^j = \frac{\tilde{w}_n^j}{\sum_{i=1}^J \tilde{w}_n^i}.$$

In practice, the measurement noise is often assumed to be a zero-mean Gaussian random variable with covariance matrix $\boldsymbol{R}_n$, that is, $p(\boldsymbol{r}_n) = \mathcal{N}(\boldsymbol{r}_n; 0, \boldsymbol{R}_n)$. In this case, the likelihood is also a Gaussian random variable with mean $g(\boldsymbol{x}_n)$ (due to the offset) and covariance $\boldsymbol{R}_n$ (due to the uncertainty introduced by $\boldsymbol{r}_n$). Hence, the likelihood becomes

$$p(\boldsymbol{y}_n \mid \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{y}_n; g(\boldsymbol{x}_n), \boldsymbol{R}_n).$$

Once the importance weights have been calculated, a point estimate of the current state and its covariance can be obtained. Similar to the unscented Kalman filter, these are given by the weighted sum of the individual samples $\boldsymbol{x}_n^j$, weighed by the importance weights w_n^j, that is,

$$\hat{\boldsymbol{x}}_{n|n} = \sum_{j=1}^J w_n^j \boldsymbol{x}_n^j, \tag{5.41a}$$

$$\boldsymbol{P}_{n|n} = \sum_{j=1}^J w_n^j (\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n})(\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n})^\mathsf{T}. \tag{5.41b}$$

[1] In practice, cost functions are closely related to the way the importance weights are calculated, but this is beyond the scope of this course.

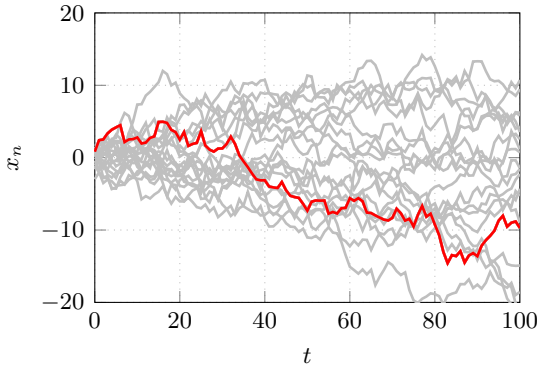


Figure 5.3: Illustration of the divergence problem when simulating trajectories.

Observe that the covariances for the process noise ($\boldsymbol{Q}_n$) and measurement noise ($\boldsymbol{R}_n$) do not enter these equations (contrary to the Kalman-filter-type algorithms). This is due to the fact that this uncertainty is accounted for when sampling and calculating the weights.

It would appear that this now yields a working algorithm. However, there is one important problem, illustrated for the random walk model in Figure 5.3: Assuming that the red trajectory represents the true realization of our state trajectory, the other simulated trajectories start to diverge from that trajectory as time elapses. Consequently, toward the end of the time scale, there are only very few trajectories in the neighborhood of the actual trajectory. In practice, this means that the weights for almost all but a few trajectories would become zero and eventually, after some more time has elapsed, there would be no trajectory close to the true realization anymore and the filter would break down completely.

To address this problem, one additional step called *resampling* has to be introduced. The basic idea of resampling is to make sure that samples with low weights, that is, samples that are unlikely to be close to the true state, are discarded and replaced with copies of the samples with high weight. Hence, resampling essentially regenerates the sample $\boldsymbol{x}_n^j$ such that there are a total of approximately $\lfloor w_n^j J \rceil$ copies of $\boldsymbol{x}_n^j$ in the resampled set of particles. Thus, if $\tilde{\boldsymbol{x}}_n^i$ (for $i = 1, \ldots, J$) denotes the resampled particle, that particle is equal to particle $\boldsymbol{x}_n^j$ with probability w_n^j, that is, we have that

$$\Pr\{\tilde{\boldsymbol{x}}_n^i = \boldsymbol{x}_n^j\} = w_n^j,$$

where $\Pr\{\cdot\}$ denotes the probability. This ensures that trajectories with low weight are discarded and that the samples remain close to the true state.

The final bootstrap particle filtering algorithm then consists of the following three steps:

1. Propagate the particles using the dynamic model,

2. calculate the importance weights using the measurement model, and

3. resample the particles according to their weights.

These three steps are then repeated for the complete dataset and the recursion is initialized by sampling from the initial distribution of the state $p(\boldsymbol{x}_0)$. Assuming Gaussian distributions for the initial state, process noise, and measurement noise, that is,

$$\begin{aligned}\boldsymbol{x}_0 &\sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{P}_0),\\ \boldsymbol{q}_n &\sim \mathcal{N}(0, \boldsymbol{Q}_n),\\ \boldsymbol{r}_n &\sim \mathcal{N}(0, \boldsymbol{R}_n),\end{aligned}$$

the resulting algorithm is as summarized in Algorithm 5.4.

Algorithm 5.4 Bootstrap Particle Filter (Gaussian Process and Measurement Noises)

1: Initialization: Sample $(j = 1, \ldots, J)$

$$\boldsymbol{x}_0^j \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{P}_0)$$

2: **for** $n = 1, 2, \ldots$ **do**

3: **for** $j = 1, 2, \ldots, J$ **do**

4: Sample

$$\boldsymbol{q}_n^j \sim \mathcal{N}(0, \boldsymbol{Q})$$

5: Propagate the state

$$\boldsymbol{x}_n^j = f(\boldsymbol{x}_{n-1}^j) + \boldsymbol{q}_n^j$$

6: Calculate the importance weights

$$\tilde{w}_n^j = \mathcal{N}(\boldsymbol{y}_n; g(\boldsymbol{x}_n^j), \boldsymbol{R}_n)$$

7: **end for**

8: Normalize the importance weights $(j = 1, \ldots, J)$

$$w_n^j = \frac{\tilde{w}_n^j}{\sum_{i=1}^J \tilde{w}_n^i}$$

9: Calculate the mean and covariance

$$\hat{\boldsymbol{x}}_{n|n} = \sum_{j=1}^J w_n^j \boldsymbol{x}_n^j$$

$$\boldsymbol{P}_{n|n} = \sum_{j=1}^J w_n^j (\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n})(\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n})^\mathsf{T}$$

10: Resample such that

$$\Pr\{\tilde{\boldsymbol{x}}_n^i = \boldsymbol{x}_n^j\} = w_n^j$$

11: **end for**