Fundamentals and Applications of Recursive Estimation Theory : Excerpts of Chapter 1 and Chapter 2

Hao LI

This is an excerpt version of Chapter 1 and Chapter 2 of the book "*Fundamentals and Applications of Recursive Estimation Theory*" (in English) written by *Hao LI* and published by *Shanghai Jiao Tong University Press* in 2022.

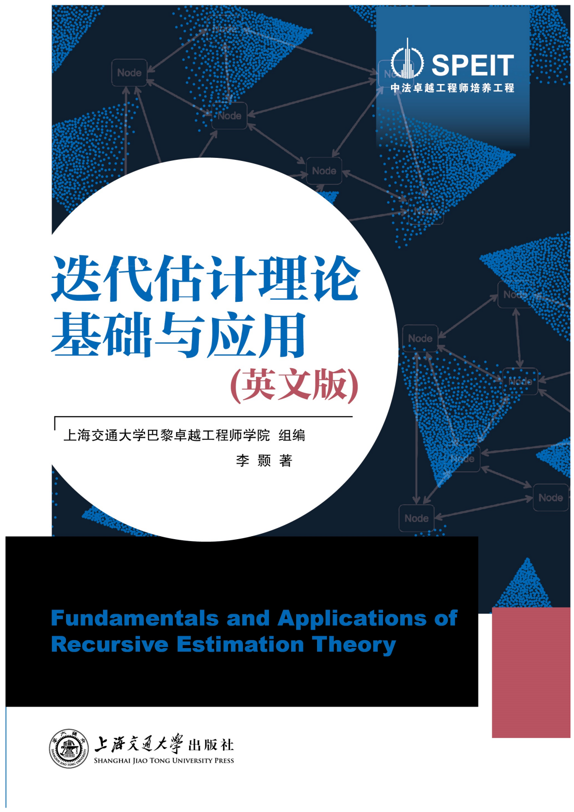

Figure 0: Cover of "*Fundamentals and Applications of Recursive Estimation Theory*" (in English)

For official citation of materials in this excerpt version [1], please adopt the following format

H. Li, "Fundamentals and Applications of Recursive Estimation Theory", Shanghai Jiao Tong University Press, 2022

[1]This excerpt differs from the officially published book by very few points. For example, this excerpt has no format editing whereas the officially published book has, so the format of materials in the former seems not as "beautiful" as that in the latter. Reference indexing in this excerpt is also different due to partial presence of the references of the officially published book. Very few contents of the officially published book may be updated in this excerpt. However, this excerpt has absolutely no essential altering of the original materials presented in the officially published book.

Chapter 1 Introduction

Estimation is a ubiquitous process in our daily life and exists even without our attention in every aspect of human activities. No matter in which activity, we normally do not act blindly but act reasonably according to our conscious estimation of things and situations involved in the activity. Take some routine and ordinary behaviours as examples. Walking is a common activity, during which we unceasingly observe environment objects around us and estimate the spatial relationship between our body and them so that we can dynamically determine a navigable (and often somehow optimal) path for our potential walking movements. Eating is also an indispensable activity, during which we dynamically estimate the spatial relationship between the food and our mouth and adjust our arm and hand movement accordingly, until the food is desirably put into our mouth.

By analogue to humans, any autonomous or partially-autonomous system that interacts with the environment should possess estimation ability according to its operation requirements. Intuitively speaking, **estimation** aims at *revealing* or *deriving* the truth of something essential to a system's operation.

1.1 State

The intuitive words "something essential" in above description can be called more formally as **state** which can potentially be of unlimited kinds. A state can be the temperature, heart rate, pulse strength, and even the overall healthy status of a patient. A state can be the trading volume of the capital market of a town, a city, a country, or the world. A state can be the structure of relationships among people in an enterprise or a society. A state can be some aspect of human feeling towards an object or an event.

In the context of intelligent systems, for an indoor mobile robot system or an outdoor intelligent vehicle system, a state can be the system's pose in a two-dimensional or three-dimensional global reference [1, 2, 3, 4, 5, 6]. A state can be the locations of the system's surrounding objects such as lane marks [7, 8] and pedestrians [9, 10]. A state can be the spatial representation of both stationary and dynamic objects in the system's local environment [11, 12, 13, 14]. From the distributed perspective of an individual system among multiple cooperative systems, a state can be the system's pose concatenated with other systems' poses in a global reference [15, 16, 17].

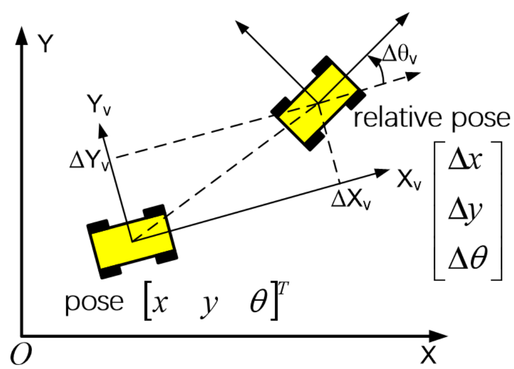

Figure 1.1: Vehicle pose in a two-dimensional global reference

Besides above examples, a state can be something essential to systems operating in a wide range of engineering activities, such as marine and submarine vehicles [18] [19], special multiped robots [20] [21], and drones and unmanned aerial vehicles [22].

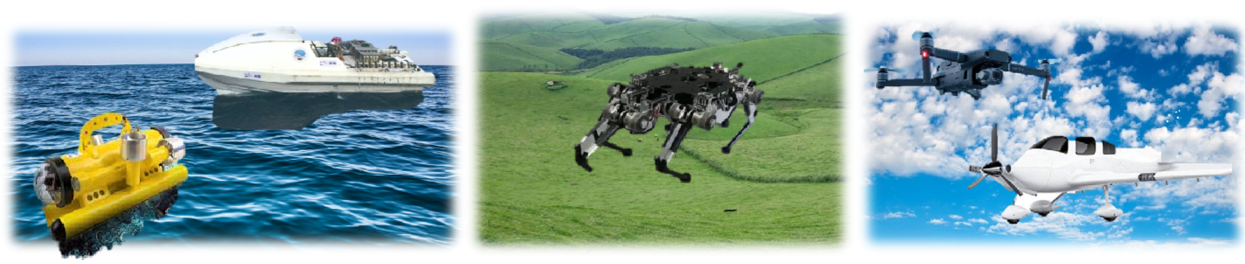

Figure 1.2: Engineering activities involving state estimation: (a) marine; (b) land; (c) aerial

What a state can be depends on concrete practices. A state can potentially be of unlimited kinds and serve as a general concept in extremely broad sense. However, we refrain from abusing the potential generality of the concept state and hence refrain from entangling this book with too many topics. For example, we refrain from entangling this book with machine learning based pattern recognition, though a pattern such as rain distribution [23] may somehow be regarded as a state and pattern recognition may somehow be regarded as state estimation.

1.4 Recursive estimation

After introduction of the important concepts **state**, **system model**, **measurement** and **measurement model**, we can clarify the concept estimation: For a system, **estimation** is the process of inferring the state from measurements, with the help of a given system model that describes state evolution and a given measurement model that describes state-measurement causal relationship.

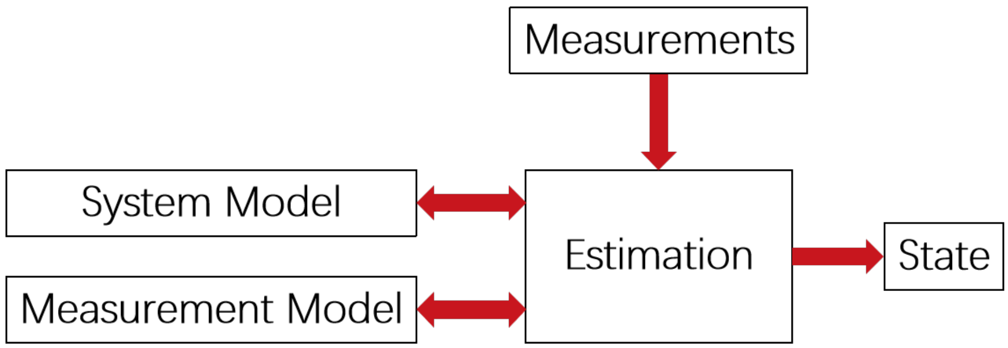

Figure 1.5: Methodology of estimation

In many applications especially real-time applications, the state evolves dynamically over time and its corresponding measurements are also obtained dynamically. Whenever a new measurement is available, instead of using all historical measurements to estimate current state from scratch, we can fairly take advantage of last state estimate which contains historical information implicitly and fuse it only with the new measurement to obtain current state estimate. This methodology, called **recursive estimation**, can render the estimation process much more efficient but essentially no less effective than estimating the state from scratch will all historical measurements. The basic spirit of recursive estimation can be illustrated by a **dynamic Bayesian network** (DBN) [24] as in Figure 1.6. The general mathematical formalism of recursive estimation from the dynamic Bayesian network perspective is postponed to later chapters.

In this book, we gradually clarify some fundamental (and hence essential) knowledge of recursive estimation: In Chapter 2, we introduce the Kalman filter which is probably the most popular method of recursive estimation and explain the basic spirit and utilities of recursive estimation. In Chapter 3, we demonstrate the importance of consistent system modelling to recursive estimation and introduce how to handle non-deterministic systems via the interacting multiple model method. In Chapter 4, we explain the motivation of

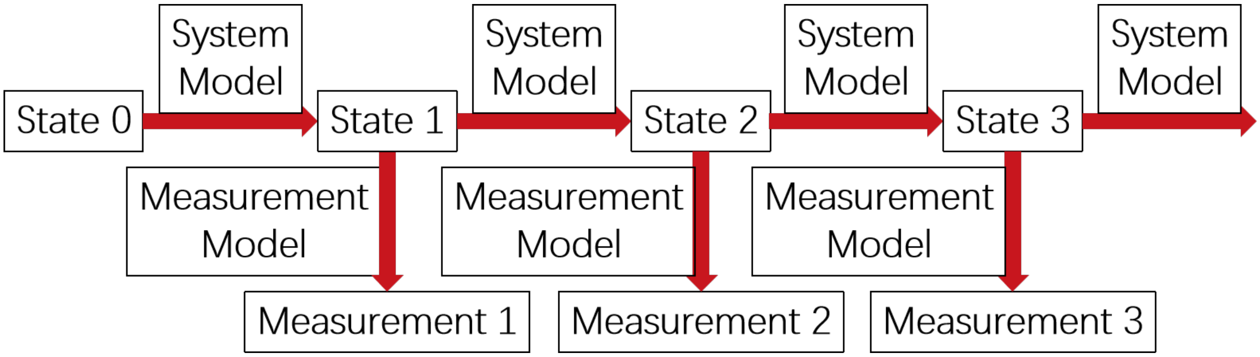

Figure 1.6: Recursive estimation from the dynamic Bayesian network (DBN) perspective

handling nonlinear system and measurement models in recursive estimation and present two well-known methods for such purpose, namely the extended Kalman filter and the unscented Kalman filter. In Chapter 5, we describe the general mathematical formalism of Bayesian inference for recursive estimation and introduce the particle filter which is a representative realization of such inference via the sampling (or Monte Carlo) strategy. In Chapter 6, we demonstrate the influence of data correlation to recursive estimation and describe how to handle data correlation especially unknown data correlation in recursive estimation.

Chapter 2 Basic Spirit And Utilities

2.2 Kalman filter

The Kalman filter, which is put forward in a milestone article [25] more than half a century ago, is probably the most popular way of instantiating the methodology of recursive estimation. It has been broadly applied in engineering activities [26, 27].

The Kalman filter consists of two essential steps: **prediction** and **update**. The prediction step propagates the estimate from time $t-1$ to time t, namely, to predict $\{\mathbf{x}_t,\ \mathbf{\Sigma}_t\}$ from $\{\hat{\mathbf{x}}_{t-1},\ \hat{\mathbf{\Sigma}}_{t-1}\}$ according to the system model. To avoid notation confusion, we denote the predicted estimate with a *bar* on variable, as $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$. The predicted estimate is also called the *a priori* estimate.

The update step updates the *a priori* estimate $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$ with the new measurement $\mathbf{z}_t$ to obtain the *a posteriori* estimate $\{\hat{\mathbf{x}}_t,\ \hat{\mathbf{\Sigma}}_t\}$ which is the final estimate at current time t. The update step embodies an important spirit of data fusion, which will be explained later.

2.2.1 Linear-Gaussian modelling

The original Kalman filter relies on the linear-Gaussian assumption, namely, the system model and the measurement model are formalized as *linear* relationships (2.3) and (2.4) respectively:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \epsilon_t \tag{2.3}$$

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \gamma_t \tag{2.4}$$

where all involved random variables follow the *Gaussian* or *normal* distribution assumption. In other words, $\mathbf{x}_t \sim N(\hat{\mathbf{x}}_t, \hat{\mathbf{\Sigma}}_t)$, $\mathbf{x}_{t-1} \sim N(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{\Sigma}}_{t-1})$, $\mathbf{u}_t \sim N(\hat{\mathbf{u}}_t, \mathbf{\Sigma}_{\mathbf{u}})$, $\epsilon_t \sim N(0, \mathbf{\Sigma}_\epsilon)$, and $\gamma_t \sim N(0, \mathbf{\Sigma}_\gamma)$.

Here, $\hat{\mathbf{u}}$ with *hat* denotes monitored control input, whereas $\mathbf{u}$ without *hat* denotes real control input which is used only in theoretical sense in system modelling. In field applications, it is always the monitored control input $\hat{\mathbf{u}}$ that is actually used in concrete procedures of recursive estimation and is used somehow as time-variant parameters of the system model. Therefore throughout this book, without causing confusion, we abuse the notation $\mathbf{u}$ to denote both real control input that is used theoretically in system modelling and monitored control input that is used somehow as system model parameters in concrete procedures of recursive estimation.

2.2.2 Prediction-update formalism

The prediction and update steps of the Kalman filter are realized as (2.5) and (2.6) respectively:

Prediction: $\mathbf{x}_t$ (*a priori*) $\sim N(\bar{\mathbf{x}}_t, \bar{\mathbf{\Sigma}}_t)$

$$\begin{aligned} \bar{\mathbf{x}}_t &= \mathbf{A}\hat{\mathbf{x}}_{t-1} + \mathbf{B}\mathbf{u}_t \\ \bar{\mathbf{\Sigma}}_t &= \mathbf{A}\hat{\mathbf{\Sigma}}_{t-1}\mathbf{A}^T + \mathbf{B}\mathbf{\Sigma}_{\mathbf{u}}\mathbf{B}^T + \mathbf{\Sigma}_\epsilon \end{aligned} \tag{2.5}$$

Update: $\mathbf{x}_t$ (*a posteriori*) $\sim N(\hat{\mathbf{x}}_t, \hat{\mathbf{\Sigma}}_t)$

$$\begin{aligned} \hat{\mathbf{x}}_t &= \bar{\mathbf{x}}_t + \mathbf{K}(\mathbf{z}_t - \mathbf{H}\bar{\mathbf{x}}_t) \\ \hat{\mathbf{\Sigma}}_t &= (\mathbf{I} - \mathbf{K}\mathbf{H})\bar{\mathbf{\Sigma}}_t \\ \text{where } \mathbf{K} &= \bar{\mathbf{\Sigma}}_t\mathbf{H}^T(\mathbf{H}\bar{\mathbf{\Sigma}}_t\mathbf{H}^T + \mathbf{\Sigma}_{\mathbf{z}})^{-1} \end{aligned} \tag{2.6}$$

In (2.5), $\mathbf{u}_t$ actually denotes monitored control input in practices. The formalism (2.5) and (2.6) is a commonly-adopted version of the original Kalman filter, whereas other formalism versions also exist. Reasoning of (2.5) is straightforward, whereas a complete understanding of (2.6) necessitates a bit more complicated derivation. Based on the Gaussian distribution assumption, (2.6) can be derived strictly via Bayesian inference. However, we would rather explain (2.6) from data fusion perspective in next section to intuitively highlight the essence of this famous recursive estimation method.

2.3 Data fusion perspective

On data fusion, we would like to begin with a daily-life example. Suppose we have two thermometers of the same quality to measure our room temperature. One thermometer indicates a value of 23 degrees centigrade, whereas the other indicates 27 degrees centigrade.

Our question is: what is the room temperature? or more specifically, what would be the most-likely room temperature?

If we had only one thermometer, then we would simply take its indicated value as the room temperature value. However, we have two thermometers which indicate different temperature values. What would be a more reasonable answer than each of the two indicated temperature values? Apparently, we should neither trust the first thermometer only nor trust the second thermometer only; instead, we may form our answer to the question by incorporating information conveyed by both thermometers. Since the two thermometers are of the same quality, a natural intuition is to take an average of the two indicated temperature values and we have our answer $(23 + 27)/2 = 25$ degrees centigrade. In other words, we *fuse* the two indicated temperature values by averaging them.

Now suppose the thermometers are of different qualities; the first thermometer is better and its error level is only a third of the second thermometer's error level. In this case, what would be the most-likely room temperature?

Since the two thermometers are of different qualities, we had better give the better thermometer a larger confidence weight and give the worse thermometer a smaller confidence weight while fusing their indicated temperature values. The first thermometer's error level is only a third of the second thermometer's error level; in other words, the first thermometer's quality level is three times the second thermometer's quality level. By intuition, we may give a confidence weight of three to the first thermometer and a confidence weight of one to the second thermometer. Thereafter we take a weighted average of the two indicated temperature values and our answer to the question is $(23 * 3 + 27 * 1)/(3 + 1) = 24$ degrees centigrade.

Above practice of taking a weighted average has its probability theory foundation. We can treat the two thermometers' indicated temperature values as two independent random variables $\{x_1, \sigma_1^2\}$ and $\{x_2, \sigma_2^2\}$ for the same event (namely the room temperature in this example) and $\sigma_1^2 = \sigma_2^2/3$. Let the fusion weights assigned to the two thermometers are k_1 and k_2 respectively and $k_1 + k_2 = 1$. We follow the Gaussian distribution assumption and aims at finding the optimal weights in the maximum likelihood sense [28]:

$$\begin{aligned}
&p(x_1, x_2|k_1x_1 + k_2x_2) \\
&= p(x_1|k_1x_1 + k_2x_2)p(x_2|k_1x_1 + k_2x_2) \\
&\propto exp(-\frac{(k_1x_1 + k_2x_2 - x_1)^2}{\sigma_1^2})exp(-\frac{(k_1x_1 + k_2x_2 - x_2)^2}{\sigma_2^2}) \\
&= exp(-\frac{[k_2(x_2 - x_1)]^2}{\sigma_1^2} - \frac{[k_1(x_1 - x_2)]^2}{\sigma_2^2}) \\
&= exp(-(\frac{k_2^2}{\sigma_1^2} + \frac{k_1^2}{\sigma_2^2})(x_1 - x_2)^2)
\end{aligned}$$

Note that

$$(\sigma_1^2 + \sigma_2^2)(\frac{k_2^2}{\sigma_1^2} + \frac{k_1^2}{\sigma_2^2}) \geq (\frac{k_2}{\sigma_1}\sigma_1 + \frac{k_1}{\sigma_2}\sigma_2)^2 = (k_2 + k_1)^2 = 1$$

so

$$\frac{k_2^2}{\sigma_1^2} + \frac{k_1^2}{\sigma_2^2} \geq \frac{1}{\sigma_1^2 + \sigma_2^2}$$

In above derivation, Cauchy's inequality [29] is used and the equality condition holds if and only if

$$\frac{k_2}{\sigma_1}/\sigma_1 = \frac{k_1}{\sigma_2}/\sigma_2 \Rightarrow \frac{k_1}{k_2} = \frac{\sigma_2^2}{\sigma_1^2} = 3$$

Therefore, when the weight assigned to the first thermometer is three times the weight assigned to the second thermometer, the weighted average of the two thermometers' indicated temperature values is of the maximum likelihood.

2.3.1 Optimal weighted average

From the simple example of fusing two thermometers' indicated temperature values, we extend to fusion of two generic source estimates $\{\mathbf{x}_1, \mathbf{\Sigma}_1\}$ and $\{\mathbf{x}_2, \mathbf{\Sigma}_2\}$ of a state $\mathbf{x}$. Their covariance matrices $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ reflect their uncertainty respectively, and accordingly their covariance inverses $\mathbf{\Sigma}_1^{-1}$ and $\mathbf{\Sigma}_2^{-1}$ can be used to reflect their *quality* respectively. The smaller the estimate covariance is, the higher the estimate quality is. Such *quality* is also known as *information* and such covariance inverses are called *information matrices*.

The *weighted average* of the two estimates $\{\mathbf{x}_1, \mathbf{\Sigma}_1\}$ and $\{\mathbf{x}_2, \mathbf{\Sigma}_2\}$ can be formulated as:

$$\begin{aligned} \hat{\mathbf{\Sigma}}^{-1} &= \mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1} \\ \hat{\mathbf{x}} &= \hat{\mathbf{\Sigma}}(\mathbf{\Sigma}_1^{-1}\mathbf{x}_1 + \mathbf{\Sigma}_2^{-1}\mathbf{x}_2) \end{aligned} \tag{2.7}$$

The fusion formula (2.7) is intuitively reasonable. In fact, its **optimality** among all possible weighted averages can also be proved — The weighted averages mentioned here actually mean linear weighted averages, whereas the concept *weighted average* itself can mean a nonlinear weighted average as well from pure mathematical perspective. However, in field applications, a nonlinear weighted average of two source estimates is usually of no practical meaning. For example, what would be the meaning of the square or root of a generic state estimate vector? So we neglect consideration of nonlinear weighted averages in following analysis.

Proof. Suppose the fused estimate $\hat{\mathbf{x}}$ is a weighted average of $\mathbf{x}_1$ and $\mathbf{x}_2$ as

$$\hat{\mathbf{x}} = \mathbf{C}\mathbf{x}_1 + (\mathbf{I} - \mathbf{C})\mathbf{x}_2$$

and hence the covariance of $\hat{\mathbf{x}}$ is

$$\hat{\mathbf{\Sigma}} = \mathbf{C}\mathbf{\Sigma}_1\mathbf{C}^T + (\mathbf{I} - \mathbf{C})\mathbf{\Sigma}_2(\mathbf{I} - \mathbf{C})^T$$

The covariance $\hat{\mathbf{\Sigma}}$ had better be as less as possible, which implies the minimality of estimate uncertainty. So the optimal weight $\mathbf{C}_{opt}$ can be determined by solving the optimization problem (2.8):

$$\mathbf{C}_{opt} = \arg\min_{\mathbf{C}} \hat{\mathbf{\Sigma}}(\mathbf{C}) = \arg\min_{\mathbf{C}}(\mathbf{C}\mathbf{\Sigma}_1\mathbf{C}^T + (\mathbf{I} - \mathbf{C})\mathbf{\Sigma}_2(\mathbf{I} - \mathbf{C})^T) \tag{2.8}$$

We may neglect mathematical ambiguity in above definition of the objective function, because it has no essential influence on deriving the optimal weight. Consider differentiation of $\hat{\mathbf{\Sigma}}(\mathbf{C})$ with respect to $\mathbf{C}$:

$$\begin{aligned} \Delta\hat{\mathbf{\Sigma}}(\mathbf{C}) &= 2\mathbf{C}\mathbf{\Sigma}_1\Delta\mathbf{C}^T + 2(\mathbf{I} - \mathbf{C})\mathbf{\Sigma}_2(-\Delta\mathbf{C})^T \\ &= 2[\mathbf{C}(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2) - \mathbf{\Sigma}_2]\Delta\mathbf{C}^T \end{aligned}$$

The variation $\Delta\mathbf{C}$ can be arbitrary; by the first optimality condition, we have

$$\mathbf{C}_{opt}(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2) - \mathbf{\Sigma}_2 = 0$$

so

$$\begin{aligned}
\mathbf{C}_{opt} &= \mathbf{\Sigma}_2(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1} = (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}\mathbf{\Sigma}_1^{-1} \\
\mathbf{I} - \mathbf{C}_{opt} &= \mathbf{\Sigma}_1(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1} = (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}\mathbf{\Sigma}_2^{-1} \\
\hat{\mathbf{\Sigma}} &= \mathbf{C}_{opt}\mathbf{\Sigma}_1\mathbf{C}_{opt}^T + (\mathbf{I} - \mathbf{C}_{opt})\mathbf{\Sigma}_2(\mathbf{I} - \mathbf{C}_{opt})^T = (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1} \\
\hat{\mathbf{x}} &= \mathbf{C}_{opt}\mathbf{x}_1 + (\mathbf{I} - \mathbf{C}_{opt})\mathbf{x}_2 \\
&= (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}\mathbf{\Sigma}_1^{-1}\mathbf{x}_1 + (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}\mathbf{\Sigma}_2^{-1}\mathbf{x}_2 \\
&= \hat{\mathbf{\Sigma}}(\mathbf{\Sigma}_1^{-1}\mathbf{x}_1 + \mathbf{\Sigma}_2^{-1}\mathbf{x}_2)
\end{aligned}$$

which is exactly the fusion formula (2.7). □

In the update step of the Kalman filter, we can treat the predicted state estimate $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$ as one source estimate and the new measurement $\{\mathbf{z}_t,\ \mathbf{\Sigma}_\mathbf{z}\}$ as another source estimate. What the update step of the Kalman filter does essentially is to fuse these two source estimates according to above weighted averaging strategy, as illustrated in Figure 2.2.

In field applications, the predicted state estimate $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$ is always a complete state estimate, yet the measurement $\{\mathbf{z}_t,\ \mathbf{\Sigma}_\mathbf{z}\}$ may be a partial measurement, or in other words, the measurement matrix $\mathbf{H}$ may be rank deficient. Consequently, we cannot always fuse $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$ and $\{\mathbf{z}_t,\ \mathbf{\Sigma}_\mathbf{z}\}$ via (2.7) directly; instead, we can resort to (2.6) to fuse $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$ and $\{\mathbf{z}_t,\ \mathbf{\Sigma}_\mathbf{z}\}$, regardless of whether $\{\mathbf{z}_t,\ \mathbf{\Sigma}_\mathbf{z}\}$ is a complete measurement or partial measurement.

Although (2.6) seems to be more general than (2.7) in practices, (2.6) can actually be derived from (2.7). This is why we treat the update step of the Kalman filter essentially as weighted averaging of the predicted state estimate $\{\bar{\mathbf{x}}_t,\ \bar{\mathbf{\Sigma}}_t\}$ and the new measurement $\{\mathbf{z}_t,\ \mathbf{\Sigma}_\mathbf{z}\}$. The derivation of (2.6) from (2.7) will be detailed in the following section.

2.3.2 Derivation of the update formalism of the Kalman filter

Suppose there exist a complete source estimate $\{\mathbf{x}_1,\ \mathbf{\Sigma}_1\}$ and a partial source estimate $\{\mathbf{z} = \mathbf{H}\mathbf{x}_2,\ \mathbf{\Sigma}_\mathbf{z}\}$. We augment $\mathbf{z}$ to a complete source estimate $\begin{bmatrix}\mathbf{z} & \mathbf{z}_0\end{bmatrix}^T$ where $\mathbf{z}_0$ is set arbitrary

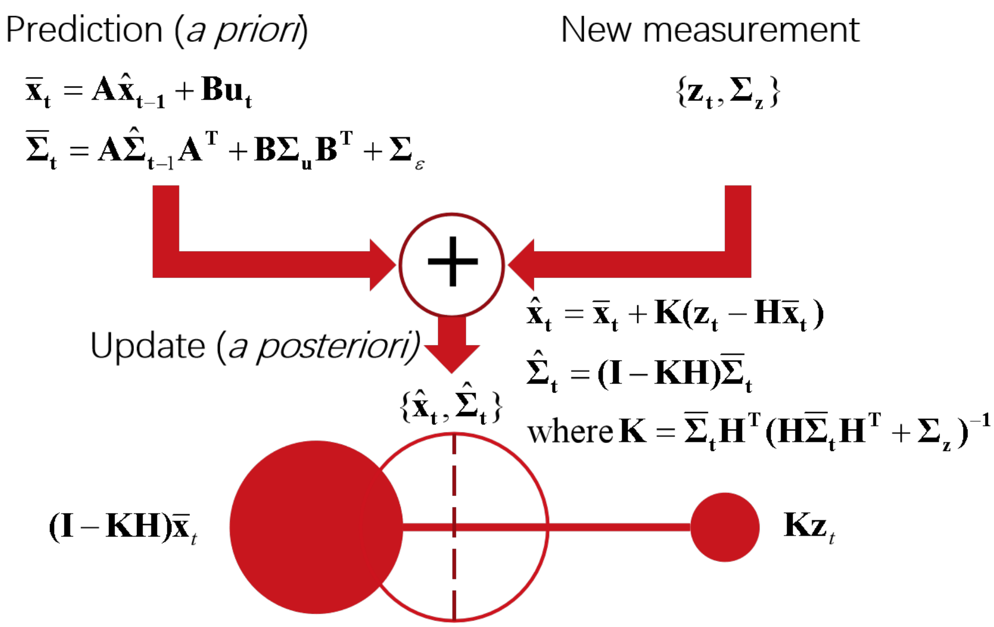

Figure 2.2: Kalman filter update: weighted average of the predicted estimate and the measurement

and its covariance is set as $\mathbf{\Sigma}_{\mathbf{z}_0} = \infty$.

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{z}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix} \mathbf{x}_2 \text{ where } \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix} \text{ is an invertible matrix}$$

Therefore

$$\mathbf{x}_2 = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z} \\ \mathbf{z}_0 \end{bmatrix}$$
$$\mathbf{\Sigma}_2 = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Sigma}_\mathbf{z} & \mathbf{0} \\ \mathbf{0} & \infty \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix}^{-T}$$

and

$$\mathbf{\Sigma}_2^{-1} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix}^{T} \begin{bmatrix} \mathbf{\Sigma}_\mathbf{z}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix} = \mathbf{H}^T\mathbf{\Sigma}_\mathbf{z}^{-1}\mathbf{H}$$

Fuse $\{\mathbf{x}_1, \boldsymbol{\Sigma}_1\}$ and $\{\mathbf{x}_2, \boldsymbol{\Sigma}_2\}$ via (2.7) and we have

$$\begin{aligned}
\boldsymbol{\Sigma} &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} = (\boldsymbol{\Sigma}_1^{-1} + \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1} \mathbf{H})^{-1} \\
&= (\mathbf{I} + \boldsymbol{\Sigma}_1 \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1} \mathbf{H})^{-1} \boldsymbol{\Sigma}_1 = [\sum_0^\infty (-\boldsymbol{\Sigma}_1 \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1} \mathbf{H})^i] \boldsymbol{\Sigma}_1 \\
&= \{\mathbf{I} - \boldsymbol{\Sigma}_1 \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1} [\sum_0^\infty (-\mathbf{H} \boldsymbol{\Sigma}_1 \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1})^i] \mathbf{H}\} \boldsymbol{\Sigma}_1 \\
&= \{\mathbf{I} - \boldsymbol{\Sigma}_1 \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1} (\mathbf{I} + \mathbf{H} \boldsymbol{\Sigma}_1 \mathbf{H}^T \boldsymbol{\Sigma}_{\mathbf{z}}^{-1})^{-1} \mathbf{H}\} \boldsymbol{\Sigma}_1 \\
&= \{\mathbf{I} - \boldsymbol{\Sigma}_1 \mathbf{H}^T (\boldsymbol{\Sigma}_{\mathbf{z}} + \mathbf{H} \boldsymbol{\Sigma}_1 \mathbf{H}^T)^{-1} \mathbf{H}\} \boldsymbol{\Sigma}_1 \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H}) \boldsymbol{\Sigma}_1 \text{ where } \mathbf{K} = \boldsymbol{\Sigma}_1 \mathbf{H}^T (\boldsymbol{\Sigma}_{\mathbf{z}} + \mathbf{H} \boldsymbol{\Sigma}_1 \mathbf{H}^T)^{-1}
\end{aligned}$$

It is worth noting that the infinite matrix series expansion in above derivation holds true only when the eigenvalues of relevant matrices are within the unit circle in the complex plane. But this does not influence the equality $\boldsymbol{\Sigma} = (\mathbf{I} - \mathbf{K}\mathbf{H})\boldsymbol{\Sigma}_1$ which is equivalent to the equality of two finite-order polynomials. Since the equality holds true for infinite choices of matrix elements involved, the equality must always hold true. Thus above derivation result $\boldsymbol{\Sigma} = (\mathbf{I} - \mathbf{K}\mathbf{H})\boldsymbol{\Sigma}_1$ always holds true. By (2.7) we also have

$$\begin{aligned}
\mathbf{x} &= \boldsymbol{\Sigma}(\boldsymbol{\Sigma}_1^{-1}\mathbf{x}_1 + \boldsymbol{\Sigma}_2^{-1}\mathbf{x}_2) = \boldsymbol{\Sigma}\boldsymbol{\Sigma}_1^{-1}\mathbf{x}_1 + \boldsymbol{\Sigma}\boldsymbol{\Sigma}_2^{-1}\mathbf{x}_2 \\
&= \boldsymbol{\Sigma}\boldsymbol{\Sigma}_1^{-1}\mathbf{x}_1 + (\mathbf{I} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}_1^{-1})\mathbf{x}_2 \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{x}_1 + \mathbf{K}\mathbf{H}\mathbf{x}_2 \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{x}_1 + \mathbf{K}\mathbf{H} \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z} \\ \mathbf{z}_0 \end{bmatrix} \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{x}_1 + \mathbf{K} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{z}_0 \end{bmatrix} \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{x}_1 + \mathbf{K}\mathbf{z} \\
&= \mathbf{x}_1 + \mathbf{K}(\mathbf{z} - \mathbf{H}\mathbf{x}_1)
\end{aligned}$$

If we substitute $\{\bar{\mathbf{x}}_t, \bar{\boldsymbol{\Sigma}}_t\}$ for $\{\mathbf{x}_1, \boldsymbol{\Sigma}_1\}$ and $\{\mathbf{z}_t, \boldsymbol{\Sigma}_{\mathbf{z}}\}$ for $\{\mathbf{z}, \boldsymbol{\Sigma}_{\mathbf{z}}\}$ in above derivation, we will have the update formalism (2.6).

2.4 Application

The principles of how the Kalman filter instantiates the general recursive estimation methodology (Figure 1.6 and Figure 2.1) based on the linear-Gaussian assumption have been introduced in previous sections. In this section, we demonstrate a concrete application of the Kalman filter in the context of intelligent vehicles, namely realization of vehicle localization by fusing commonly-available on-vehicle motion data and Bei-Dou GPS measurements via the Kalman filter. Based on this application example, we clarify basic utilities of recursive estimation in engineering activities.

2.4.1 Application description

Suppose a vehicle is equipped with motion sensors that monitor the vehicle yawrate and velocity regularly. These motion data are used in the system model that describes how the vehicle state i.e. its pose $\mathbf{x} = (x, y, \theta)$ evolves. Here, the vehicle system model is formalized as a two-dimensional bicycle kinematics model — In field applications, a vehicle can hardly navigate on an ideal two-dimensional plane due to inherent unevenness of the earth surface at both large and small scales. However, we can fairly approximate the earth surface by a collection of two-dimensional plane patches in region-wise way so that during a moderate time interval the vehicle can be fairly assumed to navigate on a two-dimensional plane. Besides, assuming that the vehicle navigates on a two-dimensional plane has no influence on demonstration of basic utilities of recursive estimation — The two-dimensional bicycle kinematics model can be approximated in discrete form as (2.9) .

$$\begin{cases} x_t &= x_{t-1} + v_t \Delta T cos(\theta_{t-1} + w_t \Delta T/2) \\ y_t &= y_{t-1} + v_t \Delta T sin(\theta_{t-1} + w_t \Delta T/2) \\ \theta_t &= \theta_{t-1} + w_t \Delta T \end{cases} \tag{2.9}$$

where ΔT denotes the system period; v and w denote the vehicle velocity and yawrate respectively.

Proof. Following physical laws we have

$$\begin{cases} dx/dt &= v cos\theta \\ dy/dt &= v sin\theta \\ d\theta/dt &= w \end{cases} \tag{2.10}$$

From the third equation of (2.10) we have

$$\theta_t = \theta_{t-1} + \int_{(t-1)\Delta T}^{t\Delta T} w dt \approx \theta_{t-1} + w_t \Delta T$$

From the first equation of (2.10) we have

$$\begin{aligned}
x_t &= x_{t-1} + \int_{(t-1)\Delta T}^{t\Delta T} v cos\theta dt \approx x_{t-1} + v_t \int_{(t-1)\Delta T}^{t\Delta T} cos\theta d\theta \frac{dt}{d\theta} \\
&= x_{t-1} + v_t \int_{(t-1)\Delta T}^{t\Delta T} \frac{d(sin\theta)}{w} \approx x_{t-1} + \frac{v_t}{w_t}(sin\theta_t - sin\theta_{t-1}) \\
&= x_{t-1} + 2\frac{v_t}{w_t} sin(\frac{\theta_t - \theta_{t-1}}{2}) cos(\frac{\theta_t + \theta_{t-1}}{2}) \\
&\approx x_{t-1} + v_t(\frac{\theta_t - \theta_{t-1}}{w_t}) cos(\frac{\theta_t + \theta_{t-1}}{2}) \approx x_{t-1} + v_t \Delta T cos(\frac{\theta_t + \theta_{t-1}}{2}) \\
&\approx x_{t-1} + v_t \Delta T cos(\theta_{t-1} + w_t \Delta T/2)
\end{aligned}$$

From the second equation of (2.10) we have

$$\begin{aligned}
y_t &= y_{t-1} + \int_{(t-1)\Delta T}^{t\Delta T} v sin\theta dt \approx y_{t-1} + v_t \int_{(t-1)\Delta T}^{t\Delta T} sin\theta d\theta \frac{dt}{d\theta} \\
&= y_{t-1} + v_t \int_{(t-1)\Delta T}^{t\Delta T} \frac{d(-cos\theta)}{w} \approx y_{t-1} + \frac{v_t}{w_t}(cos\theta_{t-1} - cos\theta_t) \\
&= y_{t-1} + 2\frac{v_t}{w_t} sin(\frac{\theta_t - \theta_{t-1}}{2}) sin(\frac{\theta_t + \theta_{t-1}}{2}) \\
&\approx y_{t-1} + v_t(\frac{\theta_t - \theta_{t-1}}{w_t}) sin(\frac{\theta_t + \theta_{t-1}}{2}) \approx y_{t-1} + v_t \Delta T sin(\frac{\theta_t + \theta_{t-1}}{2}) \\
&\approx y_{t-1} + v_t \Delta T sin(\theta_{t-1} + w_t \Delta T/2)
\end{aligned}$$

Derivation of the two-dimensional bicycle kinematics model (2.9) is done. □

The vehicle velocity v and yawrate w are monitored by on-vehicle motion sensors. The monitored velocity and yawrate values are denoted as $\hat{v}$ and $\hat{w}$ respectively. Their errors Δv and Δw follow the Gaussian distribution assumption, namely $\Delta v \sim N(0, \Sigma_v)$ and $\Delta w \sim N(0, \Sigma_w)$.

Suppose the vehicle is also equipped with a Bei-Dou GPS module that measures the vehicle position (x, y) — The Bei-Dou GPS system is familiar to public nowadays and here we assume the availability of a Bei-Dou GPS module simply to facilitate understanding of this application example. On the other hand, the basic principles reflected by this application example are general to applications with other configurations. We can fairly replace the Bei-Dou GPS module by a USA GPS module or by some *ad hoc* landmark based positioning module [1] — The Bei-Dou GPS measurement is denoted as $\mathbf{z}$ and the measurement model is formalized as (2.11):

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \gamma_t \tag{2.11}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

where γ denotes the measurement error which follows the Gaussian distribution assumption $\gamma \sim N(\mathbf{0}, \mathbf{\Sigma}_\gamma)$, namely a Gaussian distribution with zero mean and covariance $\mathbf{\Sigma}_\gamma$. It is worth noting that the measurement model (2.11) is a partial measurement model: there is no direct measurement of the orientation θ which is to be revealed indirectly via proper estimation especially recursive estimation.

Vehicle localization is a recursive process of inferring the vehicle state $\mathbf{x} = (x, y, \theta)$ from Bei-Dou GPS measurements of the vehicle position (x, y), with the help of the system model (2.9) and the measurement model (2.11). This process of recursive estimation is illustrated in Figure 2.3.

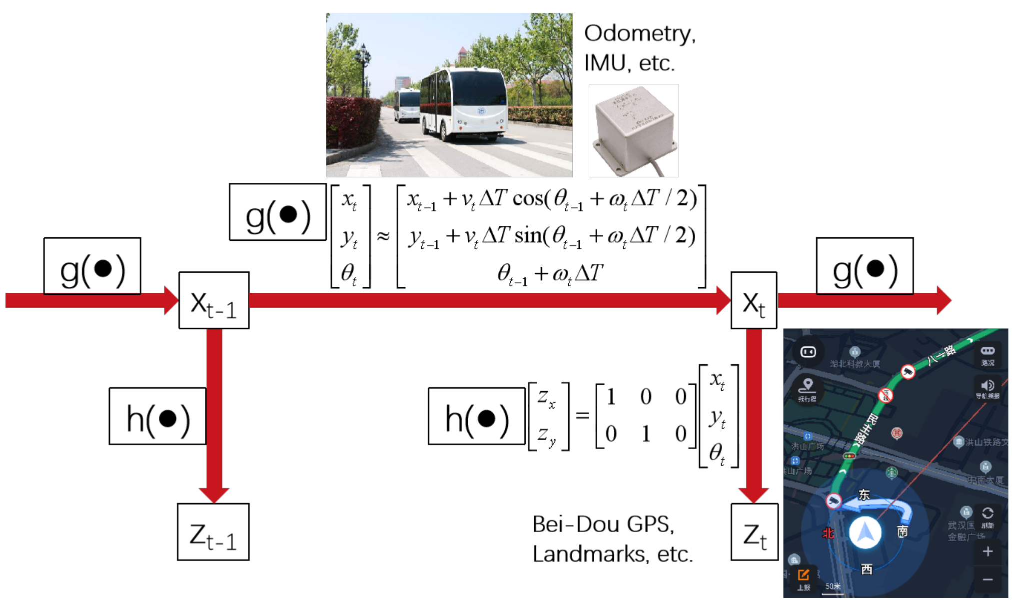

Figure 2.3: Kalman filter application example: vehicle localization

2.4.2 One-dimensional simplification of vehicle localization

2.4.3 Two-dimensional vehicle localization

The system model (2.9) is nonlinear with respect to the vehicle orientation θ and the vehicle yawrate w and does not satisfy the linear form of the system model (2.3). In order to apply the Kalman filter (2.5) and (2.6), we linearize the system model (2.9) locally about θ and w to transform it into an approximated-linear form. Such **local linearization** based variant of the Kalman filter is called **extended Kalman filter** (EKF) which will be detailed in Chapter 4. In the presented application example, the measurement model is already linear and hence exempt from local linearization. The system model after local linearization about θ and w is given as (2.16):

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \approx \begin{bmatrix} \bar{x}_t \\ \bar{y}_t \\ \bar{\theta}_t \end{bmatrix} + \mathbf{A}(\mathbf{x}_{t-1}, \mathbf{u}_t) \begin{bmatrix} \Delta x_{t-1} \\ \Delta y_{t-1} \\ \Delta \theta_{t-1} \end{bmatrix} + \mathbf{B}(\mathbf{x}_{t-1}, \mathbf{u}_t) \begin{bmatrix} \Delta v_t \\ \Delta w_t \end{bmatrix} \tag{2.16}$$

where $\mathbf{u}_t = (v_t, w_t)$ and

$$\begin{cases} \bar{x}_t & = x_{t-1} + v_t \Delta T cos(\theta_{t-1} + w_t \Delta T/2) \\ \bar{y}_t & = y_{t-1} + v_t \Delta T sin(\theta_{t-1} + w_t \Delta T/2) \\ \bar{\theta}_t & = \theta_{t-1} + w_t \Delta T \end{cases}$$

$$\mathbf{A}(\mathbf{x}_{t-1}, \mathbf{u}_t) = \begin{bmatrix} 1 & 0 & -v_t \Delta T sin(\theta_{t-1} + w_t \Delta T/2) \\ 0 & 1 & v_t \Delta T cos(\theta_{t-1} + w_t \Delta T/2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}(\mathbf{x}_{t-1}, \mathbf{u}_t) = \begin{bmatrix} \Delta T cos(\theta_{t-1} + w_t \Delta T/2) & -v_t \Delta T^2 sin(\theta_{t-1} + w_t \Delta T/2)/2 \\ \Delta T sin(\theta_{t-1} + w_t \Delta T/2) & v_t \Delta T^2 cos(\theta_{t-1} + w_t \Delta T/2)/2 \\ 0 & \Delta T \end{bmatrix}$$

The matrices $\mathbf{A}(\mathbf{x}_{t-1}, \mathbf{u}_t)$ and $\mathbf{B}(\mathbf{x}_{t-1}, \mathbf{u}_t)$ are the Jacobian matrices of the vehicle state evolution function reflected by (2.9) with respect to the state $\mathbf{x}_{t-1}$ and the control input $\mathbf{u}_t$ respectively.

With the locally-linearized system model (2.16) and the linear measurement model (2.11), the prediction and update steps (2.17) and (2.18) of the extended Kalman filter can be performed. It is worth noting that model local linearization errors always exist and may be treated simply as model errors or in other ways. Since existence of model local linearization errors has no essential influence on demonstrating the performance of the extended Kalman filter, model local linearization errors are neglected here.

Prediction:

$$\begin{bmatrix}\bar{x}_t\\ \bar{y}_t\\ \bar{\theta}_t\end{bmatrix} = \begin{bmatrix}\hat{x}_{t-1} + \hat{v}_t \Delta T cos(\hat{\theta}_{t-1} + \hat{w}_t \Delta T/2)\\ \hat{y}_{t-1} + \hat{v}_t \Delta T sin(\hat{\theta}_{t-1} + \hat{w}_t \Delta T/2)\\ \hat{\theta}_{t-1} + \hat{w}_t \Delta T\end{bmatrix} \tag{2.17}$$

$$\bar{\Sigma}_t = \mathbf{A}(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{u}}_t)\hat{\Sigma}_{t-1}\mathbf{A}(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{u}}_t)^T + \mathbf{B}(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{u}}_t)\begin{bmatrix}\Sigma_v & 0\\ 0 & \Sigma_w\end{bmatrix}\mathbf{B}(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{u}}_t)^T$$

Update:

$$\mathbf{K} = \bar{\Sigma}_t \mathbf{H}^T(\mathbf{H}\bar{\Sigma}_t\mathbf{H}^T + \Sigma_\gamma)^{-1} \tag{2.18}$$

$$\hat{\mathbf{x}}_t = \begin{bmatrix}\bar{x}_t\\ \bar{y}_t\\ \bar{\theta}_t\end{bmatrix} + \mathbf{K}(z_t - \mathbf{H}\begin{bmatrix}\bar{x}_t\\ \bar{y}_t\\ \bar{\theta}_t\end{bmatrix})$$

$$\hat{\Sigma}_t = (\mathbf{I} - \mathbf{K}\mathbf{H})\bar{\Sigma}_t$$

Simulation

Synthetic data are generated according to the system model (2.9) and the measurement model (2.11). The performance of the extended Kalman filter based two-dimensional vehicle localization is evaluated based on the generated synthetic data.

In the simulation, set $\Delta T = 1(s)$, $\Sigma_v = 0.6^2(m^2/s^2)$, $\Sigma_w = 0.02^2(rad^2/s^2)$, and $\Sigma_\gamma = diag(15^2, 15^2)(m^2)$. Set the ground-truth $\mathbf{x}_0 = [0(m), 0(m), -\pi/2(rad)]^T$, $v_t = 15(m/s)$, and $w_t = 0.04(rad/s)$. Monitored speed and yawrate values are synthesized according to $\hat{v}_t \sim N(v_t, \Sigma_v)$ and $\hat{w}_t \sim N(w_t, \Sigma_w)$ respectively. Vehicle position measurements are synthesized according to $\mathbf{z}_t \sim N(\mathbf{H}\mathbf{x}_t, \Sigma_\gamma)$.

The extended Kalman filter (2.17) and (2.18) is applied to estimate the vehicle state i.e. the vehicle pose (x, y, θ) on the given two-dimensional plane. The result of one simulation trial is demonstrated in Figure 2.6, where the black dashed line represents the vehicle trajectory ground-truth, the the red line represents the estimated vehicle trajectory, and the blue crosses represent positioning module measurements. The estimated vehicle trajectory is noticeably better than raw position measurements in the sense that the former is noticeably smoother than the latter.

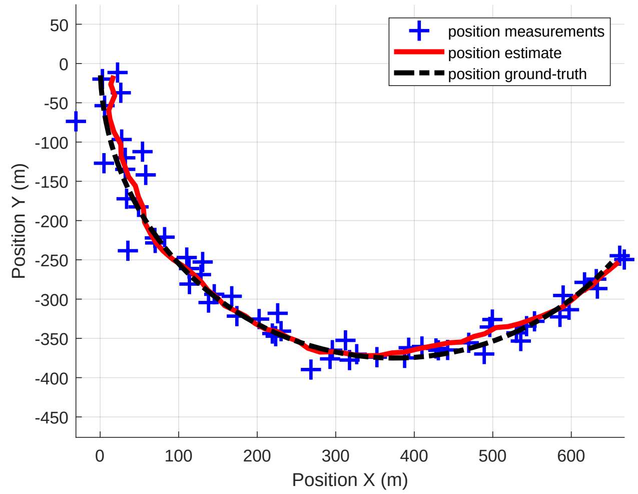

Figure 2.6: two-dimensional vehicle localization: estimate and measurement errors

The position estimate error $\sqrt{(\hat{x}_t - x_{truth})^2 + (\hat{y}_t - y_{truth})^2}$ and the position measurement error $\sqrt{(z_{x,t} - x_{truth})^2 + (z_{y,t} - y_{truth})^2}$ are computed and compared. The average result of 50 Monte Carlo simulation trials based on synthetic data is demonstrated in Figure 2.7, where the horizontal axis represents time and the vertical axis represents estimate and measurement errors. We can see that estimate errors are considerably smaller than measurement errors.

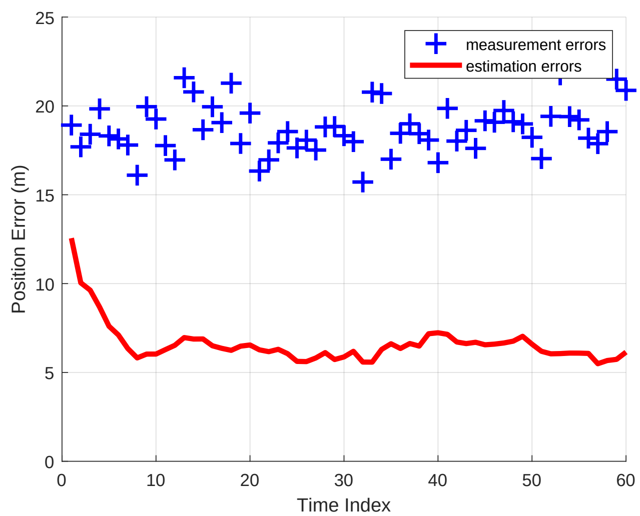

Figure 2.7: Two-dimensional vehicle position estimate and measurement errors for 50 Monte Carlo trials

The vehicle orientation estimate errors are also computed for the same 50 Monte Carlo tri-

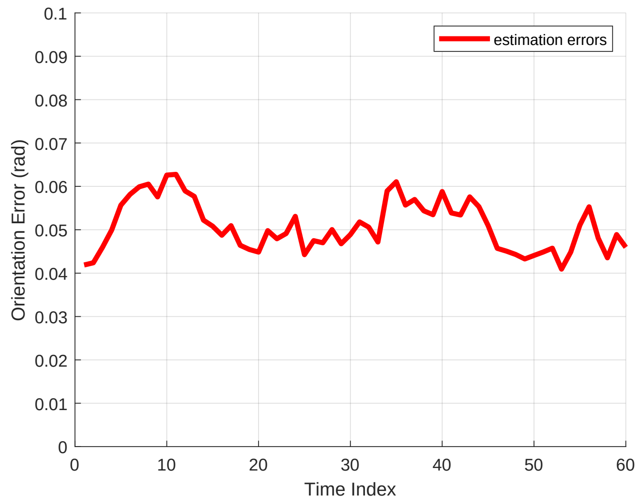

Figure 2.8: Two-dimensional vehicle orientation estimate errors for 50 Monte Carlo trials

als and the average result is demonstrated in Figure 2.8, where the horizontal axis represents time and the vertical axis represents orientation estimate errors. There are only orientation estimate errors without orientation measurement errors, because there is no direct measurement of the vehicle orientation.

As shown in Figure 2.8, the vehicle orientation estimate errors vary around 0.05 rad i.e. only a half of the tiny angle formed by two neighbouring tick marks on a clock. Here, we have no intention to analyse the quality of vehicle orientation estimates in an isolated way, because such estimate quality may change if vehicle configurations are different. The point to highlight here is another utility of recursive estimation, namely to reveal state information that is not directly measurable and hence obtain a complete state estimate.

It is worth noting that whether the complete state of a system is estimable depends on the system **observability** which is related to the system model as well as the measurement model. Readers can refer to control theory literature such as [30] for details on this issue.

2.5 Summary

In this chapter, we have introduced the general recursive estimation methodology with mathematical notations and have presented a popular recursive estimation method namely the Kalman filter. The spirit of the Kalman filter can be explained from data fusion perspective.

We have demonstrated a concrete application example of the Kalman filter, namely vehicle localization which belongs to mobile robotics [31] and is a core function for intelligent vehicles. The demonstration is based on simulation that is abstracted from field applications. Compared with demonstration directly by field applications, demonstration by simulation has two advantages: First, simulation enables us to focus on the field application part that is most related to recursive estimation and hence highlight the essential role of recursive estimation in field applications. Second, simulation is exempt from irrelevant *ad hoc* factors in field applications and enables different methods to be studied together under exactly the same conditions, which is especially desirable for a comparative study. Therefore, demonstration by simulation is commonly adopted throughout this book.

Recursive estimation has two basic utilities: First, to reveal state information that is not directly measurable and hence obtain a complete state estimate. Second, to provide more precise state estimates than raw measurements. In one word, these two basic utilities are *to know completely* and *to know better* respectively. In fact, they are also basic utilities of estimation in more general sense.

Bibliography

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, 2005.

[2] S. Thrun et al. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[3] I. Skog and P. Handel. In-car positioning and navigation technologies: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):4–21, 2009.

[4] H. Li, F. Nashashibi, and G. Toulminet. Localization for intelligent vehicle by fusing mono-camera, low-cost gps and map data. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1657–1662, 2010.

[5] H. Li, F. Nashashibi, and M. Yang. Split covariance intersection filter: Theory and its application to vehicle localization. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1860–1871, 2013.

[6] Y. Li and H. Li. Lidar-based initial global localization using two-dimensional (2d) submap projection image (spi). In *IEEE International Conference on Robotics and Automation*, pages 5063–5068, 2021.

[7] H. Li and F. Nashashibi. Robust real-time lane detection based on lane mark segment features and general a priori knowledge. In *IEEE International Conference on Robotics and Biomimetics*, pages 812–817, 2011.

[8] S. Narote, P. Bhujbal, A. Narote, and D. Dhane. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, 73:216–234, 2018.

[9] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457, 2016.

[10] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. Towards reaching human performance in pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):973–986, 2018.

[11] C.C. Wang, C. Thorpe, and S. Thrun. Simultaneous localization, mapping and moving object tracking. *International Journal of Robotics Research*, 26(9):889–916, 2007.

[12] T.D. Vu. *Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects*. Ph.D. Thesis, Institut National Polytechnique de Grenoble, 2009.

[13] H. Li, M. Tsukada, F. Nashashibi, and M. Parent. Multivehicle cooperative local mapping: a methodology based on occupancy grid map merging. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2089 – 2100, 2014.

[14] Z. Ying and H. Li. Imm-slammot: Tightly-coupled slam and imm-based multi-object tracking. *IEEE Transactions on Intelligent Vehicles*.

[15] H. Li and F. Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2):33–44, 2013.

[16] S. Saeedi, M. Trentini, M. Seto, and H. Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[17] S. Fang and H. Li. Multi-vehicle cooperative simultaneous lidar slam and object tracking in dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*.

[18] Y. Cheng and W. Zhang. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing*, 272:63–73, 2018.

[19] L. Qiao and W. Zhang. Double-loop integral terminal sliding mode tracking control for uuvs with adaptive dynamic compensation of uncertainties and disturbances. *IEEE Journal of Oceanic Engineering*, 44(1):29–53, 2019.

[20] X. Chen, F. Gao, C. Qi, X. Tian, and J. Zhang. Spring parameters design for the new hydraulic actuated quadruped robot. *Journal of Mechanisms and Robotics*, 6(021003):1–9, 2014.

[21] Y. Zhao, Y. Gao, Q. Sun, Y. Tian, L. Mao, and F. Gao. A real-time low-computation cost human-following framework in outdoor environment for legged robots. *Robotics and Autonomous Systems*, 146(103899):1–15, 2021.

[22] K. Valavanis and G. Vachtsevanos. *Handbook of unmanned aerial vehicles*. Springer, 2015.

[23] C. Chen and H. Li. Robust representation learning with feedback for single image deraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7742–7751, 2021.

[24] K.P. Murphy. *Dynamic Bayesian networks: Representation, inference and learning*. Ph.D. Dissertation, UC Berkeley, 2002.

[25] R.E. Kalman. A new approach to linear filtering and prediction problem. *ASME Trans, Ser. D, J. Basic Eng.*, 82:35–45, 1960.

[26] M.S. Grewal and A.P. Andrews. *Kalman filtering: Theory and practice*. New York, USA: Wiley, 2000.

[27] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura. Industrial applications of the kalman filter: a review. *IEEE Transactions on Industrial Electronics*, 60(12):5458–5471, 2013.

[28] R. Durrett. *Probability: theory and examples*. Cambridge university press, 2019.

[29] D. Mitrinovic and P. Vasic. *Analytic inequalities*. Springer-Verlag Berlin Heidelberg, 1970.

[30] E. Sontag. *Mathematical control theory: Deterministic finite dimensional systems*. Springer, 1998.

[31] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2016.