Optimal Estimation
(Lecture notes)

Jun Xu
Harbin Institute of Technology (Shenzhen)

Contents

1 Introduction **1**
- 1.1 Estimation examples . . . 1
- 1.2 Estimation in history . . . 3
- 1.3 Mathematical description of estimation . . . 4
- 1.4 Outline . . . 7

2 Random variables and stochastic processes **9**
- 2.1 Probability theory . . . 9
- 2.2 Random variables . . . 11
- 2.3 Stochastic process theory . . . 19
- 2.4 Several stochastic processes . . . 23

3 Estimation Theory **31**
- 3.1 Estimation problem formulation . . . 31
- 3.2 Maximum likelihood estimation . . . 32
- 3.3 Maximum a posteriori estimation . . . 34
- 3.4 Naïve Bayes and Logistic Regression . . . 35
- 3.5 Minimum Mean-Square Error Estimation . . . 42

4 Least Squares **49**
- 4.1 Linear least squares . . . 49
- 4.2 Recursive least squares . . . 54
- 4.3 Curve fitting . . . 63

5 Kalman Filter **67**
- 5.1 Propagation of states and covariances . . . 67
- 5.2 Discrete-time Kalman filter . . . 74
- 5.3 Continuous-time Kalman filter . . . 85
- 5.4 Kalman filter generalizations . . . 92
- 5.5 Nonlinear Kalman filter . . . 97

Chapter 1

Introduction

In this introduction we will give an overview of optimal estimation, focusing on the principle of estimation. The concepts introduced informally here will be covered in later chapters, with more care and technical detail.

1.1 Estimation examples

One of the most common problems in science and engineering is the estimation of various quantities based on a collection of measurements. This includes the estimation of a signal based on measurements that relate to the signal, the estimation of the state of a system based on noisy measurements of the state, and the estimation of parameters in some functional relationship. The use of estimation techniques occurs in a very wide range of technology areas such as aerospace systems (the estimation of an aircraft's or spacecraft's position and velocity based on radar measurements of position), communications (the estimation of congestion in a computer communications network), biomedical engineering (the estimation of the health of a person's heart based on an electrocardiogram), and manufacturing, chemical engineering, robotics, economics, ecology and so on.

A Chinese idiom: Measure the boat for the sword

Once upon a time, there was a man from Chu who went out on a long journey. When he was crossing the river by ship, his sword fell into the water because of his carelessness. The man from Chu immediately made a mark with his knife on the side of the boat where his sword fell in. He then turn around and said to everybody: "This is where my sword fell in." The other people looked confusedly at the imprint made by the knife. Some people urged him by saying: "Hurry and find the sword!" The man from Chu said: "There's no need to hurry, I have made a mark." The ship continued to move forward, and another man said to him: "The ship is going farther and farther. If you don't look for the sword, you won't get it back." The man from Chu still said confidently: "Don't worry, don't worry. The mark is still there." When the ship stopped at the shore, the man from Chu jumped into the water at the place indicated by his mark to find the sword. He looked for

the sword in the water below the boat at the shore, taking a long time, but in vain. And he was laughed at by the others.

This idiom is used as a metaphor to refer to those who consider things from a static viewpoint without taking into account the ever changing situation. From the point of estimation, the man from Chu aimed to estimate the position where the sword fell into the river, which is dynamically changing with respect to the boat. However, that man used a static estimation, which then introduced large deviations when the ship stopped at the shore. In the context of the estimation, the man from Chu has made an error in determining system dynamics.

Autonomous vehicle

A self-driving car, also known as an autonomous vehicle, is a vehicle is capable of sensing its environment and moving safely with little or no human input. Self-driving cars combine a variety of sensors to perceive their surroundings, such as radar, lidar, sonar, GPS, odometry and inertial measurement units. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage, the process of which includes the core technology of state and signal estimation.

Navigation at sea and the magnetic compass

A compass is an instrument used for navigation and orientation that shows direction relative to the geographic cardinal directions (or points). Usually, a diagram called a compass rose shows the directions north, south, east, and west on the compass face as abbreviated initials. When the compass is used, the rose can be aligned with the corresponding geographic directions; for example, the "N" mark on the rose points northward. Compasses often display markings for angles in degrees in addition to (or sometimes instead of) the rose. North corresponds to 0, and the angles increase clockwise, so east is 90 degrees, south is 180, and west is 270.

The first Western civilization known to have developed the art of navigation at sea were the Phoenicians, about 4,000 years ago (c. 2000 B.C.E.). Phoenician sailors accomplished navigation by using primitive charts and observations of the Sun and stars to determine directions. In the 15th century, global navigation on the open sea became possible. Maps, compasses, astrolabes, and calipers are among the early tools used by ocean navigators. In the modern era, these tools have been largely replaced by electronic and technological equivalents. During the navigation at sea, it is of crucially important to estimate the latitude and longitude. Determining latitude can be accomplished relatively easily using celestial navigation. In the Northern Hemisphere, mariners could determine the latitude by measuring the altitude of the North Star above the horizon. The angle in degrees was the latitude of the ship. Throughout the history of navigation, latitude could be found relatively accurately using celestial navigation. However, longitude could only be estimated, at best. This was because the measurement of longitude is made by comparing the time-of-day difference between the mariner's starting location and new location. In 1764, British clockmaker John Harrison (1693-1776) invented the seagoing chronometer. This invention was the most important advance to marine navigation in the three millenia that open-ocean mariners had been going to sea.

The twentieth century brought important advances to marine navigation, with radio beacons, radar, the gyroscopic compass, and the global positioning system

(GPS).

It is interesting for one kind of bird, called the arctic tern, who travelled 90,000 kilometers per year for migration, from Antarctic to Arctic. Arctic terns travel an estimated 2.4 million km (1.491 million miles) in their lifetimes. Thats three round-trip flights to the Moon. So it is mystery that how they achieved optimal estimation.

1.2 Estimation in history

Estimation in Astornomy The method of least squares was pioneered by Gauss, who has done these 3 kinds of things. First, he has developed the technique to minimize the impact of measurement error in the prediction of orbits. Second, he was the first to use least squares to predict the position of dwarf planet, and at that time, he was just 23 years old. Besides, he has proved that the least-squares method is optimal under the assumption of normally distributed errors when he was 31.

Kalman filter Another important person in the history of estimation is Rudolf E. Kálmán, and the method that he used to minimize the impact of measurement error is Kalman filter. Kalman has published 2 landmark papers in 1960 about this kind of filter, in which he introduced the notion of observability (i.e., a state can be inferred from a set of measurements in a dynamic system), an optimal framework for estimating a system's state in the presence of measurement noise, and developed the famous Kalman filter.

Kalman filter is basically an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe. The Kalman filter has numerous applications in technology. A common application is for guidance, navigation, and control of vehicles, particularly aircraft, spacecraft and dynamically positioned ships. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometrics. Kalman filters also are one of the main topics in the field of robotic motion planning and control and can be used in trajectory optimization. The Kalman filter also works for modeling the central nervous system's control of movement. Due to the time delay between issuing motor commands and receiving sensory feedback, use of the Kalman filter supports a realistic model for making estimates of the current state of the motor system and issuing updated commands. It is also worthy mentioned that Kalman filter has been successfully applied in Apollo 11 lunar module. In particular, the on-board computer on the Apollo 11 lunar module, the first manned spacecraft to land on the surface of the Moon, employed a Kalman filter to estimate the module's position above the lunar surface based on noisy radar measurements.

Early estimation milestones

Many incremental improves have been made to the field of state estimation since these early milestones.

Table 1.1 Early estimation milestones

1654	Pascal and Fermat lay foundations of probability theory
1764	Bayes' rule
1801	Gauss uses least-squares to estimate the orbit of the planetoid Cares
1805	Legendre publishes "least squares"
1913	Markov chains
1933	(Chapman)-Kolmogorov equations
1949	Wiener filter
1960	Kalman(Bucy) filter
1965	Rauch-Tung-Striebel smoother
1970	Jazwinski coins "Bayes filter"

However, until about 15 years ago, it seemed that estimation was possibly waning as an active research area. But, something has happened to change that; exciting new sensing technologies are coming along (e.g., digital cameras, laser imaging, the Global Positioning System) and pose new challenges to this old field.

1.3 Mathematical description of estimation

Given that a collection of measurements is provided, assume the statistical properties of the noise is known, the problem of estimation refers to reconstructing the underlying state (signal, parameter) of a system given a sequence of measurements as well as a priori model (information) of the systems. Generally speaking, there are 3 kinds of estimation, i.e., signal estimation, state estimation and parameter estimation.

1.3.1 Signal estimation

Assume $s(t)$ is a real-valued function of the continuous-time t, $z(t)$ is generated from $s(t)$, $v(t)$ is a noise or disturbance term. There are 2 explanations for these signals, the first is that in a communications system $s(t)$ may be a transmitted signal and $z(t)$ is the received signal (a distorted version of $s(t)$) and the second is that $z(t)$ may be a measurement of the signal $s(t)$ obtained from a sensor ($s(t)$ may be the output of a process or system), and the task of signal estimation is to estimate the true signal $s(t)$ from $z(t)$, provided that the distribution of the noise $v(t)$ is known. Fig. 1.1 explains this.

Take a target tracking problem for example, assume $z(t)$ is a noisy measurement of a target's position provided by a radar, i.e.,

$$z(t) = s(t) + v(t)$$

we can use the method of filtering or estimation to reconstruct $s(t)$ from $z(t)$. In fact, the estimator (filter) can be described as a dynamical system, which can be

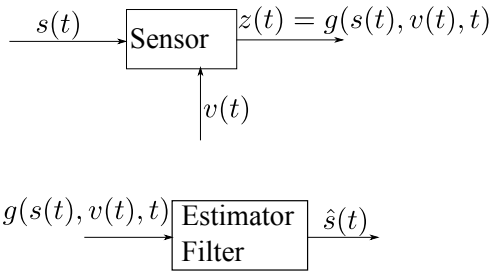


Figure 1.1 Illustration of signal estimation.

expressed a function f, and the estimate $\hat{s}(t)$ at time t can be written as

$$\hat{s}(t) = f(z(\tau : -\infty < \tau \leq t), t). \tag{1.1}$$

1.3.2 State estimation

Estimation is often carried in a state model framework defined in terms of state variables $x_1(t), x_2(t), \ldots, x_N(t)$, where N is the number of state variables. Assume $x(t) = [x_1(t), x_2(t), \ldots, x_N(t)^T]$, the state estimation problem is to generate an estimate $\hat{x}(t)$ of the state $x(t)$ at time t using the measurements $z(\tau)$ for $0 \leq \tau \leq t$, assuming measurements begin at time $t = 0$. From the description, we can know that the estimator is a casual system.

A real life example for state estimation is to estimate the state of a car when driving, in which the state denotes the position and speed of the car. The information available is a car with (approximately known) dynamics, noisy sensor data (including position and velocity), and the control commands from the driver, natural questions includes how can we predict the car's next state ? and how can we control the car? The estimation of the state can be fulfilled using the state estimation technology, and the control problem can be formulated as an optimal control problem.

In early years, state observer is used to constitute the state from the output of the system, see Fig. 1.2 for illustration.

From Fig. 1.2, the linear time-invariant continuous-time system can be described as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

and the corresponding state observer is then

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + H(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t). \end{aligned}$$

It has been proved that if the state observer is asymptotically stable, we have

$$\lim_{t \to \infty} (\hat{x}(t) - x(t)) = 0.$$

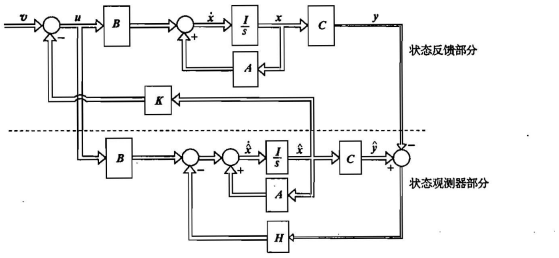


Figure 1.2 Illustration of signal estimation.

1.3.3 Parameter estimation

Parameter Estimation is a branch of statistics that involves using sample data to estimate the parameters of a distribution. A common used method for parameter estimation is Least squares (LS). LS estimation of parameters can be described as follows. Assume

$$s(n) = \sum_{j=1}^{q} \theta_j \gamma_j(n)$$
$$z(n) = s(n) + v(n)$$

where $\theta_1, \theta_2, \ldots, \theta_q$ are unknown parameters and $\gamma_1(n), \gamma_2(n), \ldots, \gamma_q(n)$ are known functions of n. The data $s(n)$ and $z(n)$ refer to the nominal and actual training output. The goal of the estimation is to find the best estimate of the parameter

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_q \end{bmatrix}$$

such that the deviation between the nominal and actual training output is minimized.

This can be formulated as an optimization problem,

$$\min[z(1) - \hat{s}(1)]^2 + [z(2) - \hat{s}(2)]^2 + \ldots + [z(n) - \hat{s}(n)]^2 \tag{1.2}$$

where $\hat{s}(n) = \sum_{j=1}^{q} \hat{\theta}_j \gamma_j(n)$. The optimization problem (1.2) can then be expressed the vector form,

$$\min(Z_n - \Gamma_n \hat{\theta})^T (Z_n - \Gamma_n \hat{\theta})$$

where $Z_n = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(n) \end{bmatrix}$, $\Gamma_n = \begin{bmatrix} \gamma(1) \\ \gamma(2) \\ \vdots \\ \gamma(n) \end{bmatrix}$, $\gamma(n) = [\gamma_1(n), \gamma_2(n), \ldots, \gamma_q(n)]$. The LS estimate of the parameter θ can then be given by,

$$\hat{\theta} = \left[\Gamma_n^T \Gamma_n\right]^{-1} \Gamma_n^T Z_n$$

provided that Γ_n is column full rank.

1.4 Outline

What we are going to do in this course is as follows:

- Discuss mathematical approaches to the best possible way of estimating signal, state or parameters
  - more in the field of engineering, or applied mathematics
- The approaches that we present for estimation are given with the goal of eventual implementation in software
  - mostly geared toward discrete-time systems

Chapter 2

Random variables and stochastic processes

2.1 Probability theory

In our attempts to filter a signal, we well be trying to extract meaningful information from a noisy signal. In order to accomplish this, we need to know the information concerning what the noise is, the characteristic of the noise, and how it works.

First, we give the definition of probability.

The probability.....

In particular, the probability of an event A is defined as

$$\mathbf{prob}(A) = \frac{\text{Number of times } A \text{ occurs}}{\text{Total number of outcomes}}$$

For example, when rolling a six-sided die 6 times, we can compute the probability of getting the number 1 for four times, which can be expressed as,

$$\mathbf{prob}(A) = \frac{C_6^4 \cdot 5 \cdot 5}{6^6} = 0.0080$$

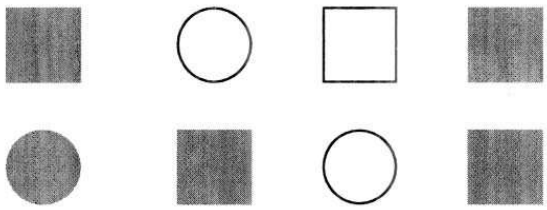
An event is an outcome or a union of outcomes, when the outcomes are the occurrences over which you can assign probabilities (or measures). A random variable is a variable whose domain is the set of basic events, and whose range (outcome) could be numerical or categorical.

Besides, we can also define the conditional probability....

In particular, the conditional probability of event A given event B ($\mathbf{prob}(B) \neq 0$) can be described as

$$\mathbf{prob}(A|B) = \frac{\mathbf{prob}(A, B)}{\mathbf{prob}(B)},$$

which equals the probability that A occurs given the fact that B has occurred. The term $\mathbf{prob}(A, B)$ is the joint probability of A and B, i.e., the probability that the event A and B both occur. The terms $\mathbf{prob}(A)$ or $\mathbf{prob}(B)$ are called an *a priori*



probability as it applies to the probability of an event apart from any previously known information. Conversely, the conditional probability is called an *a posteriori* probability as it applies to a probability given the fact that some information about a possibly related event is already known.

An example is given as follows to illustrate these definitions. There are 8 shapes in Fig. 2.1, including 3 circles and 5 squares. Of the 3 circles, one is gray, and there is one of the 5 squares that is white. We can calculate the probability and conditional probability as follows.

$$\begin{aligned} &\mathbf{prob}(\text{circle}) = 3/8, \mathbf{prob}(\text{square}) = 5/8; \\ &\mathbf{prob}(\text{gray, circle}) = 1/8, \mathbf{prob}(\text{gray}|\text{circle}) = 1/3; \\ &\mathbf{prob}(\text{white}|\text{square}) = \tfrac{1/8}{5/8} = 1/5. \end{aligned} \tag{2.1}$$

Next we'll introduce Bayers' Rule, which is critical in the calculation of the conditional probability. Bayes' Rule is often written as follows,

$$\mathbf{prob}(A|B) = \frac{\mathbf{prob}(B|A)\,\mathbf{prob}(A)}{\mathbf{prob}(B)}. \tag{2.2}$$

Alternatively, we can also write it as

$$\mathbf{prob}(A, B) = \mathbf{prob}(A|B)\,\mathbf{prob}(B) = \mathbf{prob}(B|A)\,\mathbf{prob}(A).$$

Using Bayers' Rule, the probability in the example can be computed, i.e.,

$$\mathbf{prob}(\text{gray}|\text{circle}) = \frac{\mathbf{prob}(\text{circle}|\text{gray})\,\mathbf{prob}(\text{gray})}{\mathbf{prob}(\text{circle})} = \frac{(1/5)(5/8)}{3/8} = 1/3.$$

The definition of independence is closely related to Bayes' Rule. We say that two events are independent if the occurrence of one event has no effect on the probability of the occurrence of the other event, i.e.,

$$\mathbf{prob}(A, B) = \mathbf{prob}(A)\,\mathbf{prob}(B),$$

which is then equivalent to the expressions as follows,

$$\mathbf{prob}(A|B) = \mathbf{prob}(A), \mathbf{prob}(B|A) = \mathbf{prob}(B).$$

2.2 Random variables

2.2.1 Single random variables

We define a random variable (RV) as a functional mapping from a set of experimental outcomes (the domain) to a set of real numbers (the range). A RV can be either continuous or discrete (realizations belong to a discrete or continuous set of values). The probability distribution function (PDF) of a RV is defined as,

$$F_X(x) = \mathbf{prob}(X \leq x),$$

where $F_X(x)$ is the PDF of the RV X and x is a nonrandom independent variable or constant. There are several properties concerning the PDF of a RV, i.e.,

- $F_X(x) \in [0,1], F_X(-\infty) = 0, F_X(\infty) = 1$
- $F_X(a) \leq F_X(b)$ if $a \leq b$
- $\mathbf{prob}(a < X \leq b) = F_X(b) - F_X(a)$

The probability density function (pdf) of a RV X is then defined as,

$$f_X(x) = \frac{dF_X(x)}{dx},$$

which is the derivative of the PDF. Some properties of the pdf is listed as follows,

- $F_X(x) = \int_{-\infty}^{x} f_X(z)dz$
- $f_X(x) \geq 0$
- $\int_{-\infty}^{\infty} f_X(x)dx = 1$
- $\mathbf{prob}(a < x \leq b) = \int_a^b f_X(x)dx$

Following gives 2 examples concerning different kind of RVs.

Example 2.1 Uniformly-distributed RV. Suppose we take a measurement with the set S of outcomes equal to any number between -1 and 1, define the RV Z by $Z(\alpha) = \alpha$, i.e., when a measurement is taken (the experiment is performed), the value of z is equal to the value of the measurement. We assume the distribution function is given by

$$F_Z(z) = \begin{cases} 0, & z < -1 \\ 0.5(z+1), & -1 < z < 1; \\ 1, & z > 1; \end{cases}$$

The density function is

$$f_Z(z) = \begin{cases} 0.5, & -1 < z < 1 \\ 0, & \text{otherwise.} \end{cases}$$

Then the RV Z is a uniformly distributed continuous RV.

Example 2.2 Gaussian RV. Suppose we take a measurement with the set S of outcomes equal to any number between -1 and 1, and define the RV Z by $Z(\alpha) = \alpha$. Assume the density function is given by

$$f_Z(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\eta)^2}{2\sigma^2}},$$

where η is a real number and σ is a positive number. Then the RV Z is a Gaussian or normal RV, denoted as $Z \sim \mathcal{N}(\eta, \sigma^2)$.

The expected value (expectation, mean, average) of a RV X is defined as its average value over a large number of experiments, i.e.,

$$E(X) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{m} A_i n_i.$$

in which the outcome A_i occurs n_i times and $N \to \infty$. The expected value of any function $g(X)$ is then given by

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx.$$

The variance of a RV is a measure of how much we expect the RV to vary from its mean, and can be seen as a measure of how much variability there is in a RV. The variance of a RV X is described as,

$$\sigma_X^2 = E[(X - \bar{x})^2] = \int_{-\infty}^{\infty} (x - \bar{x})^2 f_X(x) dx,$$

i.e.,

$$\sigma_X^2 = E(X^2) - \bar{x}^2.$$

The standard deviation of an RV is σ, which is the square root of the variance.

Transformations of random variables can be used derive the pdf of some RV from that of another RV. Suppose that we have two RVs, say X and Y, related to one another by the monotonic functions $g(\cdot)$ and $h(\cdot)$:

$$\begin{aligned} Y &= g(X) \\ X &= g^{-1}(Y) = h(Y) \end{aligned}$$

If we know the pdf of X, then we can compute the pdf of Y as follows:

$$\begin{aligned} \mathbf{prob}(X \in [x, x+dx]) &= \mathbf{prob}(Y \in [y, y+dy])(dx > 0) \\ \int_x^{x+dx} f_X(z) dz &= \begin{cases} \int_y^{y+dy} f_Y(z) dz & \text{if dy¿0} \\ -\int_y^{y+dy} f_Y(z) dz & \text{if dy¡0} \end{cases} \\ f_X(x) dx &= f_Y(y) |dy| \\ f_Y(y) &= |\tfrac{dx}{dy}| f_X[h(y)] = |h'(y)| f_X[h(y)] \end{aligned}$$

Example 2.3 Find the pdf of a linear function of a Gaussian RV. Suppose $X \sim N(\bar{x}, \sigma_x^2)$ and $Y = g(X) = aX + b, a, b \in \mathbf{R}$, the pdf $f_Y(y)$ can be solved,

$$\begin{aligned} X &= h(Y) \\ &= (Y-b)/a \\ h'(y) &= 1/a \\ f_Y(y) &= |h'(y)| f_X[h(y)] \\ &= |\tfrac{1}{a}| \tfrac{1}{\sigma_X \sqrt{2\pi}} \exp\left\{ \tfrac{-[(y-b)/a - \bar{x}]^2}{2\sigma_X^2} \right\} \\ &= \tfrac{1}{|a|\sigma_X \sqrt{2\pi}} \exp\left\{ \tfrac{-[y-(a\bar{x}+b)]^2}{2a^2\sigma_X^2} \right\} \end{aligned}$$

i.e., $Y \sim N(a\bar{x} + b, a^2\sigma_x^2)$.

In the more general case, if the RVs are related by the function $Y = g(X)$, where $g(\cdot)$ is a non monotonic function, the pdf of Y (evaluated at y) can be computed from the pdf of X as

$$f_Y(y) = \sum_i f_X(x_i)/|g'(x_i)|$$

where the x_i values are the solutions of the equation $y = g(x)$. The proof is left to the readers.

2.2.2 Multiple random variables

For RVs X and Y, define the joint distribution function as

$$F_{XY}(x, y) = \mathbf{prob}(X \le x, Y \le y),$$

which is usually written as $F(x, y)$ for short. The properties of $F(x, y)$ is listed as follows,

- $F(x, y) \in [0, 1]$, $F(x, -\infty) = F(-\infty, y) = 0$, $F(\infty, \infty) = 1$
- $F(a, c) \le F(b, d)$ if $a \le b$ and $c \le d$
- $\mathbf{prob}(a < X \le b, c < Y \le d) = F(b, d) + F(a, c) - F(a, d) - F(b, c)$
- $F(x, \infty) = F(x)$, $F(\infty, y) = F(y)$ (marginal distribution function)

The joint probability density function of RVs X and Y is defined as

$$f_{XY}(x, y) = \frac{\partial^2 F_{XY}(x, y)}{\partial x \partial y},$$

and we use $f(x, y)$ for short. The properties of $f(x, y)$ is listed as follows,

- $F(x, y) = \int_{-\infty}^{x} \int_{\infty}^{y} f(z_1, z_2) dz_1 dz_2$
- $f(x, y) \ge 0$, $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = 1$

- $\mathbf{prob}(a < X \leq b, c < Y \leq d) = \int_c^d \int_a^b f(x,y)dxdy$
- $f(x) = \int_{-\infty}^{\infty} f(x,y)dy$, $f(y) = \int_{-\infty}^{\infty} f(x,y)dx$ (marginal density function)

We can also define the expectation of functions of X and Y, i.e.,

$$E[g(X,Y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)f(x,y)dxdy$$

In particular, the covariance of two scalar RVs X and Y is defined as,

$$C_{XY} = E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$$

The RVs X and Y are independent if they satisfy the following equality

$$\mathbf{prob}(X \leq x, Y \leq y) = \mathbf{prob}(X \leq x)\,\mathbf{prob}(Y \leq y),\ \forall x, y,$$

which can be equivalently written as,

$$F_{XY}(x,y) = F_X(x)F_Y(y), f_{XY}(x,y) = f_X(x)f_Y(y).$$

We can also define the statistical uncorrelatedness of 2 scalar RVs. Before this, the correlation coefficient of two scalar RVs X and Y is introduced, i.e.,

$$\rho = \frac{C_{XY}}{\sigma_x \sigma_y}.$$

We say that the RVs X and Y are uncorrelated if

$$\rho = 0 \text{ or } R_{XY} = E(X)E(Y).$$

It is noted that the independence of 2 RVs does not equal the uncorrelatedness of 2 RVs, which is because that the uncorrelatedness is basically linear uncorrelatedness. Hence, if 2 RVs are independent, then they are uncorrelated, but the reverse does not hold. Following gives a detailed explanation of this. Independent means that two RVs X and Y don't have a relationship at all. In case that the 2 RVs X and Y have a relationship, the relationship is either linear or nonlinear and we don't have a third option. Assume $Y = aX + b$ (they have a linear relationship), we have

$$E(XY) = aEX^2 + bEX$$

On the other hand,

$$EXEY = a(EX)^2 + bEX$$

Except for the case $EX^2 = (EX)^2$, i.e., $DX = 0$, we have

$$E(XY) \neq EXEY$$

When $DX = 0$, the 2 RVs X and Y are uncorrelated, and in this case, the RV X is constant. The following example shows another case when there are no linear relationship between 2 RVs.

Example 2.4 Assume (X, Y) conforms to the uniform distribution, and they satisfy,

$$X^2 + Y^2 = 1,$$

then we have $f(x, y) = \frac{1}{\pi}, \forall x \in [-1, 1], y = \sqrt{1 - x^2}$.

We further have

$$f_X(x) = \frac{2\sqrt{1-x^2}}{\pi}, \forall x \in [-1, 1]$$
$$f_Y(y) = \frac{2\sqrt{1-y^2}}{\pi}, \forall y \in [-1, 1]$$

Thus

$$E(XY) = 0, EX = 0, EY = 0$$

In fact, the 2 RVs X and Y are dependent, but there are uncorrelated.

There are cases in which uncorrelatedness does imply independence. One of these cases is the one in which both random variables are two-valued (so each can be linearly transformed to have a Bernoulli distribution). Further, two jointly normally distributed random variables are independent if they are uncorrelated, although this does not hold for variables whose marginal distributions are normal and uncorrelated but whose joint distribution is not joint normal.

For 2 RVs, we say they are statistical orthogonal if

$$R_{XY} = 0.$$

For 2 uncorrelated RVs, they are orthogonal only if at least one of them is zero-mean.

Following gives an example to illustrate these definitions.

Example 2.5 A slot machine is rigged so you get -1, 0, or 1 with equal probability the first spin X. On the second spin Y you get 1 if $X = 0$, and 0 if $X \neq 0$.

$$\begin{aligned} E(X) &= \tfrac{-1+0+1}{3} = 0 \\ E(Y) &= \tfrac{0+1+0}{3} = 1/3 \\ E(XY) &= \tfrac{(-1)(0)+(0)(1)+(1)(0)}{3} = 0 \end{aligned}$$

We can get the following conclusions.

- X and Y are uncorrelated because $E(XY) = E(X)E(Y)$
- X and Y are orthogonal because $E(XY) = 0$
- The two RVs are dependent because the realization of Y depends on the realization of X.

For 2 RVs, we can also define the conditional density functions. Let X and Y be jointly distributed RVs, we define the conditional distribution function $F_Y(y|x_1 < X \leq x_2)$ as the conditional probability of the event $\{Y \leq y\}$ given that the event $\{x_1 < X \leq x_2\}$ has occurred, i.e.,

$$F_Y(y|x_1 < X \leq x_2) = \mathbf{prob}(Y \leq y|x_1 < X \leq x_2).$$

Similarly, the conditional density function $f_Y(y|X = x)$ is defined as

$$f_Y(y|X = x) = \lim_{\Delta x \to 0} f_Y(y|x < X \leq x + \Delta x). \tag{2.3}$$

According to Bayes' rule, we have

$$f_Y(y|X = x) = \frac{f_{X,Y}(x, y)}{f_X(x)} \tag{2.4}$$

and

$$f_Y(y|X = x) = \frac{f_X(x|Y = y)f_Y(y)}{f_X(x)}.$$

Detailed proof for this. According to the definition (2.3), we have

$$f_Y(y|X = x) = \lim_{\Delta x \to 0} \frac{\partial F_Y(y|x < X \leq x + \Delta x)}{\partial y}$$

The PDF $F_Y(y|x < X \leq x + \Delta x)$ can be expressed as,

$$F_Y(y|x < X \leq x+\Delta x) = \mathbf{prob}(Y \leq y|x < X \leq x+\Delta x) = \frac{\mathbf{prob}(Y \leq y, x < X \leq x + \Delta x)}{\mathbf{prob}(x < X \leq x + \Delta x)},$$

in which

$$\mathbf{prob}(Y \leq y, x < X \leq x + \Delta x) = \int_{-\infty}^{y} \int_{x}^{x+\Delta x} f(x, y)dxdy$$

and

$$\mathbf{prob}(x < X \leq x + \Delta x) = \int_{x}^{x+\Delta x} f(x)dx.$$

Taking the derivative of $F_Y(y|x < X \leq x + \Delta x)$ with respect to y gives rise to $f_Y(y|x < X \leq x + \Delta x)$, i.e.,

$$f_Y(y|x < X \leq x + \Delta x) = \frac{\int_x^{x+\Delta x} f(x, y)dx}{\int_x^{x+\Delta x} f(x)dx}, \tag{2.5}$$

when $\Delta x \to 0$, (2.5) approximates

$$\frac{f(x, y)\Delta x}{f(x)\Delta x},$$

thus we have (2.4).

Starting from 2 RVs, we can also consider multivariate statistics. Given an n-element RV X and an m-element RV Y (assuming that both X and Y are column vectors), their correlation is defined as

$$R_{XY} = E(XY^T) = \begin{bmatrix} E(X_1Y_1) & \cdots & E(X_1Y_m) \\ \vdots & & \vdots \\ E(X_nY_1) & \cdots & E(X_nY_m) \end{bmatrix}$$

Their covariance is defined as

$$C_{XY} = E[(X - E(X))(Y - E(Y))^T] = E(XY^T) - E(X)E(Y)^T$$

The autocorrelation of the n-element RV X is defined as

$$R_X = E[XX^T] = \begin{bmatrix} E(X_1^2) & \cdots & E(X_1X_n) \\ \vdots & & \vdots \\ E(X_nX_1) & \cdots & E(X_n^2) \end{bmatrix}$$

We have $R_X = R_X^T$, i.e., an autocorrelation matrix is always symmetric. Besides, an autocorrelation matrix is always positive semidefinite.

$$z^T R_X z = z^T E[XX^T]z = E[z^T XX^T z] = E[(z^T X)^2] \geq 0$$

The autocovariance of n-element RV X is defined as

$$\begin{aligned} C_X &= E[(X - E(X))(X - E(X)^T)] \\ &= \begin{bmatrix} E(X_1 - E(X_1))^2 & \cdots & E[(X_1 - E(X_1))(X_n - E(X_n))] \\ \vdots & & \vdots \\ E[(X_n - E(X_n))(X_1 - E(X_1))] & \cdots & E[(X_n - E(X_n))^2] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n} \\ \vdots & & \vdots \\ \sigma_{n1} & \cdots & \sigma_n^2 \end{bmatrix} \end{aligned}$$

An auto covariance matrix is always symmetric and positive semidefinite, i.e.,

$$z^T C_X z = z^T E[(X - \bar{X})(X - \bar{X})^T]_z = E[(z^T(X - \bar{X}))^2] \geq 0$$

Corresponding to the linear transformation of scalar RV, here we can also investigate the linear transformation of vector RV. Take vector Gaussian RV for example. An n-element RV X is Gaussian (normal) if

$$\text{pdf}(X) = \frac{1}{(2\pi)^{n/2}|\det(C_X)|^{1/2}} \exp\left[-\frac{1}{2}(X - E(X))^T C_X^{-1}(X - E(X))\right]$$

Consider a Gaussian RV X that undergoes a linear transformation

$$Y = g(X) = AX + b,$$

where $A \in \mathbf{R}^{n\times n}$, $b \in \mathbf{R}^n$. If A is invertible, we have

$$\begin{aligned} &f_Y(y) = |h'(y)| f_X[h(y)] \\ &= \frac{1}{(2\pi)^{n/2}|\det(AC_XA^T)|^{1/2}} \exp\left[-\tfrac{1}{2}(y - E(Y))^T (AC_XA^T)^{-1}(y - E(Y))\right], \end{aligned}$$

i.e., $Y \sim N(AE(X) + b, AC_XA^T)$. The normality is preserved in linear transformations of random vectors (just as in scalar case).

The detailed proof is.

$$\begin{aligned}
f_Y(y) &= |h'(y)| f_X[h(y)] \\
&= |\det(A^{-1})| f_X[h(y)] \\
&= |\det(A^{-1})| \frac{1}{(2\pi)^{n/2}|\det(C_X)|^{1/2}} \cdot \\
&\quad \exp\left\{-\tfrac{1}{2}\left[A^{-1}(y-b)-E(X)\right]^T C_X^{-1}[*]\right\} \\
&= |\det(A^{-1})| \frac{1}{(2\pi)^{n/2}|\det(C_X)|^{1/2}} \cdot \\
&\quad \exp\left\{-\tfrac{1}{2}\left[A^{-1}y-A^{-1}b-\bar{x}\right]^T C_X^{-1}[*]\right\} \\
&= \frac{1}{(2\pi)^{n/2}|\det(A)||\det(C_X)|^{1/2}} \cdot \\
&\quad \exp\left\{-\tfrac{1}{2}\left[A^{-1}y-A^{-1}b-A^{-1}\bar{y}+A^{-1}b\right]^T C_X^{-1}[*]\right\} \\
&= \frac{1}{(2\pi)^{n/2}|\det(A)|^{1/2}|\det(C_X)|^{1/2}|\det(A^T)|^{1/2}} exp\left[-\tfrac{1}{2}(y-\bar{y})^T (A^{-1})^T C_X^{-1} A^{-1}(y-\bar{y})\right] \\
&= \frac{1}{(2\pi)^{n/2}|\det(AC_XA^T)|^{1/2}} \exp\left[-\tfrac{1}{2}(y-\bar{y})^T (AC_XA^T)^{-1}(y-\bar{y})\right]
\end{aligned}$$

Supplementary material

Matrix derivative

Usually, for vector derivative, the vector is defined as a column vector. For $f(x): \mathbf{R}^n \to \mathbf{R}$, the Jacobian of $f(x)$ is an $n \times 1$ vector and the Hessian of $f(x)$ is an $n \times n$ matrix. Here we use the denominator layout.

$$\nabla_x f = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}, \nabla_x^2 f = \frac{\partial^2 f}{\partial x \partial x^T} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Concerning vector by vector derivative, $f(x): \mathbf{R}^n \to \mathbf{R}^m (m > 1)$, where $f = [f_1, \ldots, f_m]^T$, $x = [x_1, \ldots, x_n]^T$, the Jacobian matrix,

$$\nabla_x f = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Consider scalar by matrix derivative, $f(X): \mathbf{R}^{n \times m} \to \mathbf{R}$, where $n, m > 1$ and

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

we have

$$\nabla_x f = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \cdots & \frac{\partial f}{\partial x_{1m}} \\ \frac{\partial f}{\partial x_{21}} & \cdots & \frac{\partial f}{\partial x_{2m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_{n1}} & \cdots & \frac{\partial f}{\partial x_{nm}} \end{bmatrix}$$

Last, matrix by scalar derivative, $F(x) : \mathbf{R} \to \mathbf{R}^{n \times m}$, where $n, m > 1$ and

$$F = \begin{bmatrix} f_{11} & \cdots & f_{1m} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nm} \end{bmatrix}$$

we have

$$\nabla_x F = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_{11}}{\partial x} & \cdots & \frac{\partial f_{n1}}{\partial x} \\ \frac{\partial f_{12}}{\partial x} & \cdots & \frac{\partial f_{n2}}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{1m}}{\partial x} & \cdots & \frac{\partial f_{nm}}{\partial x} \end{bmatrix}$$

Properties of the determinant

- $\det(I_n) = 1$ where I_n is the $n \times n$ identity matrix.
- $\det\left(A^T\right) = \det(A)$, $\det\left(A^T\right) = \det(A)$,
- $\det\left(A^{-1}\right) = \dfrac{1}{\det(A)} = \det(A)^{-1}$.
- For square matrices A and B of equal size, $\det(AB) = \det(A)\det(B)$.
- $\det(cA) = c^n \det(A)$ for an $n \times n$ matrix A.

2.3 Stochastic process theory

A stochastic process, also called a random process, is a very simple generalization of the concept of a RV. A stochastic process $X(t)$ is a RV X that changes with time. There are 4 kinds of stochastic processes.

- continuous random process, which is basically the RV at each time is continuous and time is continuous. Typical example is the temperature at each moment of the day.
- discrete random process, which indicates that the RV at each time is discrete and time is continuous. For example, the number of people in a given building at each moment of the day is a discrete random process.
- continuous random sequence, which refers to the RV at each time is continuous and time is discrete. The high temperature each day is one example of continuous random sequence.
- discrete random sequence, which is the RV at each time is discrete and time is discrete. Similar to the previous example, the number of people in a given building each day is a discrete random sequence.

For stochastic processes, we can also define the distribution and density. Since a stochastic process is a RV that changes with time, it has a distribution and density function that are functions of time. The PDF of $X(t)$ is defined as

$$F_X(x,t) = \mathbf{prob}(X(t) \leq x),$$

which changes with respect to time. If $X(t)$ is a random vector, then the inequality above is an element-by-element inequality, i.e.,

$$F_X(x,t) = \mathbf{prob}[X_1(t) \leq x_1, \cdots, X_n(t) \leq x_n].$$

Similarly, the pdf of $X(t)$ is defined as

$$f_X(x,t) = \frac{dF_X(x,t)}{dx}.$$

Again, if $X(t)$ is a random vector, then the derivative is taken once with respect to each element of x, i.e.,

$$f_X(x,t) = \frac{d^n F_X(x,t)}{dx_1 \cdots dx_n}.$$

The mean and covariance of $X(t)$ are also functions of time, i.e.,

$$\bar{x}(t) = \int_{-\infty}^{\infty} x f(x,t) dx,$$

$$C_X(t) = E\{[X(t) - \bar{x}(t)][X(t) - \bar{x}(t)]^T\} = \int_{-\infty}^{\infty} [x - \bar{x}(t)][x - \bar{x}(t)]^T f(x,t) dx,$$

in which $\bar{x}(t)$ and $C_X(t)$denote the mean and covariance, respectively.

Stochastic process at two different times t_1 and t_2 are 2 different RVs $X(t_1)$ and $X(t_2)$. For these 2 RVs, we can define the joint distribution and density functions. The joint distribution (second-order distribution) function is defined as,

$$F(x_1, x_2, t_1, t_2) = \mathbf{prob}(X(t_1) \leq x_1, X(t_2) \leq x_2)$$

and the joint density (second-order density) function is defined as,

$$f(x_1, x_2, t_1, t_2) = \frac{\partial^2 F(x_1, x_2, t_1, t_2)}{\partial x_1 \partial x_2}$$

It is noted that if $X(t)$ is an n-element random vector, then the inequality that defines $F(x_1, x_2, t_1, t_2)$ actually consists of $2n$ inequalities, and the derivative that defines $f(x_1, x_2, t_1, t_2)$ actually consists of $2n$ derivatives.

For a stochastic process, we can also define the autocorrelation and autocovariance. Given a stochastic process $X(t)$, the autocorrelation is defined as

$$R_X(t_1, t_2) = E[X(t_1)X^T(t_2)],$$

i.e., the correlation between the 2 RVs $X(t_1)$ and $X(t_2)$. Similarly, the autocovariance of $X(t)$ is defined as,

$$C_X(t_1, t_2) = E\{[X(t_1) - \bar{x}(t_1)][X(t_2) - \bar{x}(t_2)]^T\}.$$

Next we'll introduce an important kind of stochastic process, which is called stationary stochastic process. In fact, in order to study the stochastic process, we usually assume that the stochastic process is stationary. The are 2 kinds of stationarity which is strict-sense stationary and wide-sense stationary. The stochastic process $\{X(t)\}$ is said to be strictly stationary, strongly stationary or strict-sense stationary if

$$F_X(x(t_1+\tau),\ldots,x(t_n+\tau)) = F_X(x(t_1),\ldots,x(t_n))$$

for all $\tau, t_1, \ldots, t_n \in \mathbf{R}$ and for all $n \in \mathbf{Z}_+$. e.g., flipping a coin ten times. A stochastic process $X(t)$ is called weak stationary stochastic process if the mean of the stochastic process is constant with respect to time, and the autocorrelation is a function of the time difference $t_2 - t_1$ (not a function of the absolute times):

$$E[X(t)] = \bar{x}, \quad E[X(t_1)X^T(t_2)] = R_X(t_2 - t_1).$$

It is clear that stationary implies wide-sense stationary but wide-sense stationary does not implies stationary. From the definition of strict-sense and weak-sense stationary, we can know that it is easier to check whether a stochastic process is weak-sense stationary, hence the stochastic process we investigated are usually assumed to be wide-sense stationary.

There are several properties concerning wide-sense stationary stochastic process.

- $R_X(0) = E[X(t)X^T(t)]$
- $R_X(-\tau) = R_X^T(\tau)$
- For scalar stochastic processes, we have $|R_X(\tau)| \leq R_X(0)$

Example 2.6 Examples of stationary and non stationary stochastic process.

Non stationary process: the high temperature each day; tomorrow's closing price of the Dow Jones Industrial Average.

Stationary process: electrical noise. If the statistics of the noise are the same every day, then the electrical noise is a stationary process. For practical purposes, if the statistics of a random process do not change over the time interval of interest, then we consider the process to be stationary.

For a stochastic process, we also define the time average (*sample average*) and autocorrelation. For continuous-time random processes, suppose it has a realization $x(t)$, then the time average of $X(t)$ is defined as,

$$A[X(t)] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)dt.$$

The time autocorrelation is defined as,

$$R[X(t), \tau] = A[X(t)X^T(t+\tau)].$$

Another important concept in stochastic process is the ergodic process. An ergodic process is a stationary random process for which

$$A[X(t)] = E(X), R[X(t), \tau] = R_X(\tau)$$

Basically, ergodicity means that every realization go through every state of the random process. In the real world, we are often limited to only a few realizations of a stochastic process. We can compute the time average, time autocorrelation, and other time-based statistics of the realization. If the random process is ergodic, then we can use those time averages to estimate the statistics of the stochastic process.

Example 2.7 Ergodic process. Take the waves coming up on a beach for example. If you look from side-to-side, you get an idea of the distribution of heights at different spots at any one time. And if you measure at one spot, you get an idea of the distribution of heights at one spot over time. Assume the process is ergodic, you would look up and down at a specific spot of the beach and infer the time series behavior of waves. You will fail if the waves are not ergodic over the relevant time scale (we can assume a time scale for the ergodicity to be valid)

Can you find more examples for ergodic processes?

Example 2.8 Suppose X is a random variable, and $Y(t) = X \cos t$ is a stochastic process, find the expected value of $Y(t)$, find $A[Y(t)]$, the time average of $Y(t)$. Under what condition is $E[Y(t)] = A[Y(t)]$?

The expectation of $Y(t)$ can be expressed as,

$$E[Y(t)] = \cos t EX$$

The time average $A[Y(t)]$ is,

$$A[Y(t)] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x \cos t dt = 0$$

Hence, if $EX = 0$, we have $E[Y(t)] = A[Y(t)]$. Figure 2.1 illustrates this.

We can also consider 2 stochastic processes. For 2 stochastic processes $X(t)$ and $Y(t)$, the cross correlation of $X(t)$ and $Y(t)$ is defined as

$$R_{XY}(t_1, t_2) = E[X(t_1)Y^T(t_2)].$$

Two random processes $X(t)$ and $Y(t)$ are said to be uncorrelated if

$$R_{XY}(t_1, t_2) = E[X(t_1)]E[Y(t_2)]^T$$

for all t_1 and t_2.

The cross covariance of $X(t)$ and $Y(t)$ is defined as

$$C_{XY}(t_1, t_2) = E\{[X(t_1) - \bar{X}(t_1)][Y(t_2) - \bar{Y}(t_2)]^T\}.$$

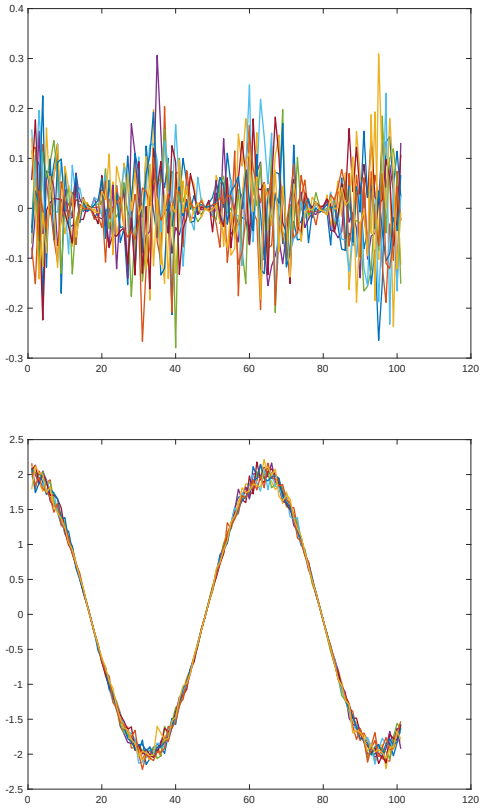


Figure 2.1 Simulation of ergodic process and non-ergodic process. Top: ergodic process. Bottom: non-ergodic process.

2.4 Several stochastic processes

2.4.1 Markov Chain

In probability theory, a Markov model is a stochastic model used to model randomly changing systems. It is assumed that future states depend only on the current state, not on the events that occurred before it (that is, it assumes the *Markov property*). Generally speaking, there are 4 kinds of Markov model, which is listed in Table 2.1.

Markov Chain is considered as a discrete stochastic sequence. For a discrete random sequence, the outcome of the n-th trial is the random variable X_n, X_0 is the initial position of the process.

Table 2.1 Different kinds of Markov model

	System state if fully observable	System state is partially observable
System is autonomous	Markov chain	Hidden markov model
System is controled	Markov decision process	Partially observable Markov decision process

The discrete random sequence is called a **Markov Chain**, if we have

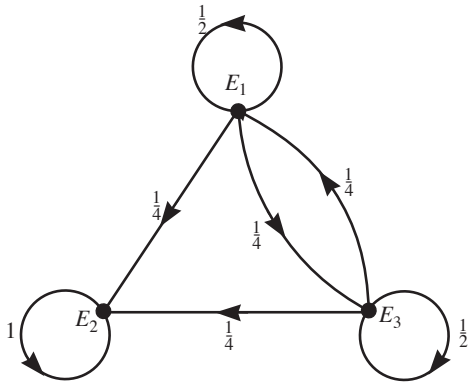
$$\begin{aligned}&\mathbf{prob}\{X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n\} \\ &= \mathbf{prob}\{X_{n+1} = i_{n+1} | X_n = i_n\}\end{aligned}$$

for all $n \in \mathbf{Z}_+$, $i_0, \ldots, i_n, i_{n+1} \in S$ (S is the state set).

The **Markov property** is used to describe the memoryless property of a stochastic process.

The following example illustrates a Markov process.

Example 2.9 We can define the transition matrix as,



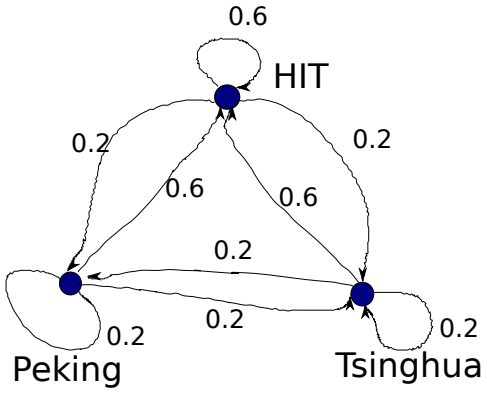
$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

Example 2.10 Where shall we go for lunch? Assume the preference for the restaurants can be described by the following graph and transition matrix.

$$T = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.6 & 0.2 & 0.2 \end{bmatrix}$$

Given

$$x_0 = [1\ 0\ 0],$$



we can predict the preference for the restaurant at time n, i.e., x_n. The steady state of preference for the restaurant can be obtained as

$$q = \lim_{n\to\infty} X_n$$

and it is interesting that this equal a constant value (under what conditions?)

For example, assume

$$T = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.5 & 0.2 & 0.3 \end{bmatrix}$$

then the steady state preference for the restaurant is [0.55560.22220.2222].

Hidden Markov Model (HMM) assumes that there is another process Y whose behavior "depends" on X. HMM stipulates that, for each time instance n_0, the conditional probability distribution of Y_{n_0} given the history $\{X_n = x_n\}_{n\leq n_0}$ must NOT depend on $\{x_n\}_{n<n_0}$ The goal of HMM model is to learn about X by observing Y. The definition of HMM can be listed as below.

Let X_n and Y_n be discrete-time stochastic processes and $n \geq 1$. The pair (X_n, Y_n) is a hidden markov model if

- X_n is a Markov process and is not directly observable ("hidden");
- $\mathbf{prob}\,(Y_n \in A|X_1 = x_1, \ldots, X_n = x_n) = \mathbf{prob}\,(Y_n \in A|X_n = x_n)$ for every $n \geq 1$, $x_1, \ldots, x_n$, and an arbitrary (measurable) set A.

The states of the process X_n are called hidden states, and $\mathbf{prob}\big(Y_n \in A \,\big|\, X_n = x_n\big)$ is called emission probability or output probability. The application of HMM can be found in many areas, including reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, and bioinformatics.

Example 2.11 A hypothetical dishonest casino

In general, the casino uses a fair die most of the time. Occasionally the casino secretly switches to a loaded die, and later the casino switches back to the fair die.

A probabilistic process determines the switching back-and-forth from loaded die to fair die and back again after each toss of the die, with the switch from fair-to-loaded occurring with probability 0.05 and from loaded-to-fair with probability 0.1. Assume that the loaded die will come up "six" with probability 0.5 and the remaining five numbers with probability 0.1 each, we have the following transition matrix and emission matrix.

The transition matrix is

$$A = \begin{bmatrix} & 0 & 1 \\ 0 & 0.95 & 0.05 \\ 1 & 0.1 & 0.9 \end{bmatrix}$$

and the emission probability matrix is

$$B = \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ F & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ L & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{2} \end{bmatrix}$$

If you can see only the sequence of rolls (the sequence of observations or signals) you do not know which rolls used a loaded die and which used a fair die, because the casino hides the state.

2.4.2 Random Walk

The term random walk was first introduced by Karl Pearson in 1905. A random walk is a stochastic or random process, that describes a path that consists of a succession of random steps on some mathematical space such as the integers.

Example 2.12 Random walk.

- The random walk on the integer number line, $\mathbf{Z}$, which starts at 0 and at each step moves +1 or -1 with equal probability.
- The path traced by a molecule as it travels in a liquid or a gas.
- The search path of a foraging animal.
- The price of a fluctuating stock.
- The financial status of a gambler.

In fact, the term random walk most often refers to a special category of Markov chains or Markov processes. Random walks can also take place on a variety of spaces, such as graphs, on the integers or the real line, in the plane or higher-dimensional vector spaces, on curved surfaces or higher-dimensional Riemannian manifolds, on groups finite, finitely generated or Lie.

Following gives the mathematical description of 1-dimensional Random Walk. Take independent random variables $Z_1, Z_2, \ldots$, where each variable is either 1 or -1, with a probability of p and $1 - p$, respectively. Set $S_0 = 0$ and $S_n = \sum_{j=1}^{n} Z_j$. The series $\{S_n\}$ is called the simple random walk on $\mathbf{Z}$, and if $p = 0.5$, we have

$$E(S_n) = \sum_{j=1}^{n} E(Z_j) = 0$$

$$E(S_n^2) = \sum_{i=1}^{n} E(Z_i^2) + 2 \sum_{1 \le i < j \le n} E(Z_i Z_j) = n.$$

A one-dimensional random walk can also be looked at as a **Markov chain**, whose state space is given by the integers $i = 0, \pm 1, \pm 2, \ldots$, the transition probablity

$$\underset{i,i+1}{\mathbf{prob}} = p = 1 - \underset{i,i-1}{\mathbf{prob}}\,.$$

2.4.3 Wiener Processes

A standard (one-dimensional) Wiener process (depicts Brownian motion) is a stochastic process $\{W_t\}_{t \ge 0_+}$ indexed by nonnegative real numbers t with the following properties:

- $W_0 = 0$
- W has independent increments, i.e., for every $t > 0$, the future increments $W_{t+u} - W_t, u \ge 0$, are independent of the past values $W_s, s \le t$.
- W has Gaussian increments: $W_{t+u} - W_t$ is normally distributed with mean 0 and variance u, $W_{t+u} - W_t \sim \mathcal{N}(0, u)$.
- W has continuous paths: W_t is continuous in t.

One of the many reasons that Brownian motion is important in probability theory is that it is, in a certain sense, a limit of rescaled simple random walks. Let $\xi_1, \xi_2, \ldots$ be i.i.d. random variables with mean 0 and variance 1. For each n, define a continuous time stochastic process

$$W_n(t) = \frac{1}{\sqrt{n}} \sum_{1 \le k \le \lfloor nt \rfloor} \xi_k, \qquad t \in [0, 1]$$

it is clear that the increments of W_n are independent because the ξ_k are independent. It is noted that for large n, $W_n(t) - W_n(s)$ is close to $\mathcal{N}(0, t - s)$ by the central limit theorem.

The Wiener process plays an important role in both pure and applied mathematics. In pure mathematics, the Wiener process gave rise to the study of continuous time martingales, it plays a vital role in stochastic calculus, diffusion processes and even potential theory. In applied mathematics, the Wiener process is used to represent the integral of a white noise Gaussian process. Besides, it is useful as a model of noise in electronics engineering (see **Brownian noise**), instrument errors in filtering theory. In control theory, Wiener Process is used to describe unknown forces.

2.4.4 Poisson Processes

Let $N(t)$ be a stochastic process. It is called a homogeneous Poisson counting process with rate $\lambda > 0$ if

- $\mathbf{prob}\{N(0) = 0\} = 1$
- $\forall n \in N, 0 < t_0 < t_1 < ... < t_n$: The increments $N(t_0), N(t_1) - N(t_0), \ldots, N(t_n) - N(t_{n-1})$ are independent
- $\forall 0 < s < t : N(t) - N(s) \sim \text{Pois}(\lambda(t-s))$

It is clear that

$$\mathbf{prob}(N(t) = n) = \mathbf{prob}(N(t) - N(0) = n | N(0) = 0) = \mathbf{prob}(N(t) - N(0) = n)$$
$$= \frac{(\lambda t)^n e^{-\lambda t}}{n!}$$
$$\sum_{n=0}^{\infty} p_n(t) = \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} e^{-\lambda t} = e^{-\lambda t} \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} = 1, \forall t$$

The Poisson process is Markov, which can be stated as: let $X(t), t > 0$ be a Poisson process of rate λ. Then, for any $s > 0$, $X(s+t) - X(s), t > 0$ is also a Poisson process of rate λ independent of $X(r), r \leq s$.

Example 2.13 Examples of Poisson processes.

- the number of telephone calls at an office logged up to time t
- the number of vehicles which pass a roadside speed camera within a specified hour
- the number of students in Teaching Building 6 at time t
- ·····

Example 2.14

The number of failures $N(t)$, which occur in a computer network over the time interval $[0, t)$, can be described by a homogeneous Poisson process $\{N(t), t \geq 0\}$. On an average, there is a failure after every 4 hours, i.e. the intensity of the process is equal to $\lambda = 0.25[h^{-1}]$. Derive the probability of at most 1 failure in $[0, 8)$.

For Poisson process, we have

$$E[N(t)] = \lambda t, D[N(t)] = \lambda t.$$

As $N(0) = 0$, we have $\lambda = 0.25[h^{-1}]$.

The probability of at most 1 failure in $[0, 8)$ can be expressed as

$$\mathbf{prob}(N(8) - N(0) \leq 1) = \mathbf{prob}(N(8) = 0) + \mathbf{prob}(N(8) = 1) = 0.406$$

2.4.5 White noise

In signal processing, white noise is a random signal having equal intensity at different frequencies, giving it a constant power spectral density. In discrete time, white noise is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean and finite variance. In particular,

if each sample has a normal distribution with zero mean, the signal is said to be **Gaussian white noise**.

In order to strictly define the white noise, we first introduce the concepts of power spectral density, which is also known as power spectrum. The power spectral density (PSD) refers to the spectral energy distribution that would be found per unit time. Parseval's theorem: Summation or integration of the spectral components yields the total power (for a physical process) or variance (in a statistical process), identical to what would be obtained by integrating $x^2(t)$ over the time domain.

For a continuous time stochastic process, the power spectrum $S_X(\omega)$ of a wide-sense stationary stochastic process $X(t)$ is defined as the Fourier transform of the autocorrelation.

$$S_X(\omega) = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau} d\tau.$$

And the autocorrelation is the inverse Fourier transform of the power spectrum

$$R_X(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(\omega) e^{j\omega\tau} d\omega$$

The power of a wide-sense stationary stochastic process can be calculated as,

$$P_X = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(\omega) d\omega$$

We can also define cross power spectrum for 2 stochastic processes. The cross power spectrum of two wide-sense stationary stochastic processes $X(t)$ and $Y(t)$ is Fourier transform of the cross correlation:

$$S_{XY}(\omega) = \int_{-\infty}^{\infty} R_{XY}(\tau) e^{-j\omega\tau} d\tau$$

And the inverse Fourier transform can be used to obtain the cross correlation,

$$R_{XY}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{XY}(\omega) e^{j\omega\tau} d\omega.$$

For discrete-time random processes, the power spectrum of a discrete-time random process is defined as

$$S_X(\omega) = \sum_{k=-\infty}^{\infty} R_X(k) e^{-j\omega k}, \omega \in [-\pi, \pi]$$

and

$$R_X(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_X(\omega) e^{j\omega k} d\omega.$$

Based on these, a discrete-time stochastic process $X(t)$ is called white noise if

$$R_X(k) = \begin{cases} \sigma^2 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases} = \sigma^2 \delta_k,$$

where δ_k is the Kronecker delta function and is defined as

$$\delta_k = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases}$$

The discrete-time white noise can be interpreted as follows. If $X(k)$ is a discrete-time white noise process, then the RV $X(n)$ is uncorrelated with $X(m)$ unless $n = m$. Besides, the power spectrum of a discrete-time white noise process is equal at all frequencies:

$$S_X(\omega) = R_X(0), \forall \omega \in [-\pi, \pi].$$

This is similar to the white light, which results in the name of white noise.

For a continuous-time random process, white noise has equal power at all frequencies (like white light):

$$S_X(\omega) = R_X(0), \forall \omega.$$

In this case, we have

$$R_X(\tau) = R_X(0)\delta(\tau),$$

where $\delta(\tau)$ is the continuous-time impulse function.

It is noted that continuous-time white noise is not something that occurs in the real world because it has infinite power. In fact, many continuous-time processes approximate white noise and are useful in mathematical analyses of signals and systems.

Therefore, an infinite-bandwidth white noise signal is a purely theoretical construction. The bandwidth of white noise is limited in practice by the mechanism of noise generation, by the transmission medium and by finite observation capabilities. Thus, a random signal is considered "white noise" if it is observed to have a flat spectrum over the range of frequencies that is relevant to the context.

Example 2.15

Suppose that a zero-mean stationary stochastic process has the autocorrelation function

$$R_X(\tau) = \sigma^2 e^{-\beta|\tau|}, \beta \in \mathbf{R}_+$$

Calculate the power spectrum as well as the power of the stochastic process.

The power spectrum

$$\begin{aligned} &S_X(\omega) = \int_{-\infty}^{\infty} \sigma^2 e^{-\beta|\tau|} e^{-j\omega\tau} d\tau \\ &= \int_{-\infty}^{0} \sigma^2 e^{(\beta - j\omega)\tau} d\tau + \int_{0}^{\infty} \sigma^2 e^{-(\beta + j\omega)\tau} d\tau \\ &= \frac{\sigma^2}{\beta - j\omega} + \frac{\sigma^2}{\beta + j\omega} \\ &= \frac{2\sigma^2\beta}{\omega^2 + \beta^2} \end{aligned}$$

The variance (also power) of the stochastic process is computed as

$$\begin{aligned} &E[X^2(t)] = R_X(0) \\ &= P_X = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(\omega) d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{2\sigma^2\beta}{\omega^2 + \beta^2} d\omega \\ &= \frac{\sigma^2}{\pi} \arctan \frac{\omega}{\beta} \Big|_{-\infty}^{\infty} \\ &= \sigma^2 \end{aligned}$$

Chapter 3

Estimation Theory

3.1 Estimation problem formulation

Before introducing the estimation problem, we have to answer the question "What is optimal?". In general, the "goodness" of an estimate can be expressed in different ways, depending upon the particular engineering problem. 3 commonly-used optimality criterion are the maximum-likelihood, maximum a posteriori, and minimum mean-square error criterion

Table 3.1 gives the notations used in estimation.

Table 3.1 Notations used in estimation.

$\mathbf{s}(n)$	signal	$s(n)$	signal realization
$\mathbf{v}(n)$	noise signal	$v(n)$	noise signal realization
$\mathbf{z}(n)$	sample	$z(n)$	sample realization
$\hat{\mathbf{s}}(n)$	estimate	$\hat{s}(n)$	a specific estimate
$\tilde{\mathbf{s}}(n) = \mathbf{s} - \hat{\mathbf{s}}$	estimation error	$\tilde{s}(n)$	a specific estimation error

Given the measurements $\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(n)$, the corruption function g such that

$$\mathbf{z}(n) = g(\mathbf{s}(n), \mathbf{v}(n), n)$$

and an optimality criterion. Design an estimator that generates an optimal estimate $\hat{\mathbf{s}}(n)$ of $\mathbf{s}(n)$ given by

$$\hat{\mathbf{s}}(n) = \alpha_n(\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(n)),$$

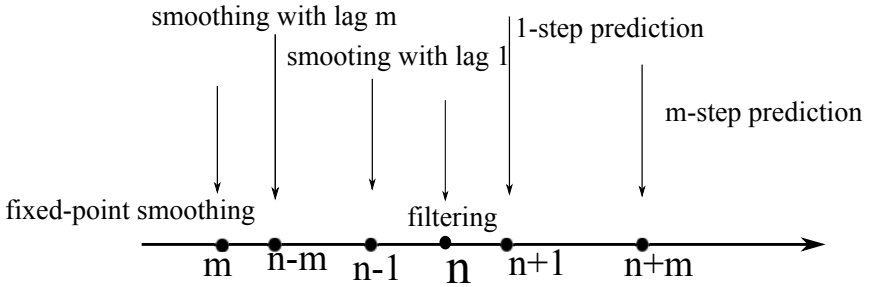
for some function α_n.

Given observations $\{\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(n)\}$, estimation aims to make the best guess of the value of $\mathbf{s}(\ell)$. There are 3 kinds of estimation problems, namely, prediction, filtering, and smoothing, which is shown in Fig. 3.1.

Given an estimate, we'll discuss the corresponding properties.

- Unbiased estimator:

$$E(\hat{\mathbf{s}}(n)) = E(\mathbf{s}(n)), E(\tilde{\mathbf{s}}(n)) = 0$$



- Asymptotically unbiased estimator:

$$\lim_{n\to\infty} E(\hat{\mathbf{s}}(n)) = E(\mathbf{s}(n)),\ \lim_{n\to\infty} E(\tilde{\mathbf{s}}(n)) = 0$$

- Consistent estimator:

$$\lim_{n\to\infty} E(\tilde{\mathbf{s}}^2(n)) = 0$$

Example 3.1 The mean filter. Assume we have constant signal $\mathbf{s}$: $\mathbf{s}(n) = \mathbf{s}$ and the measurement: $\mathbf{z}(n) = \mathbf{s} + \mathbf{v}(n)$, in which $\mathbf{v}(n)$ is a RV with mean 0 and variance σ_v^2. Assume $\mathbf{s}$ and $\mathbf{v}(1), \mathbf{v}(2), \ldots, \mathbf{v}(n)$ are independent. The mean filter can be derived as,

$$\hat{\mathbf{s}}(n) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{z}(j)$$

We can check the properties of the mean filter.

- unbiasedness: $E[\hat{\mathbf{s}}(n)] = E[\mathbf{s}]$
- consistency: $E[\tilde{\mathbf{s}}^2(n)] = \frac{\sigma_v^2}{n}$

3.2 Maximum likelihood estimation

The maximum likelihood estimation seeks to find most-probable (most likely) estimation by maximizing the likelihood function $f_{\mathbf{x}}(x)$. Therefore, the maximum likelihood estimation, i.e., the most-likely value of $\mathbf{x}$ is the one that maximizes $f_{\mathbf{x}}(x)$. For example, if we have a single measurement z, then we have to find the value of s that is **most likely** to have produced z, which equals finding the value of s that maximizes the **likelihood function**. The likelihood function in this case can be defined as,

$$f_{\mathbf{z}}(z|\mathbf{s} = s)$$

The maximum likelihood estimation problem can be formulated as,

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \text{value of } s \text{ that maximizes } f_{\mathbf{z}}(z|\mathbf{s} = s) \tag{3.1}$$

The solution to the optimization problem (3.1) can be described as,

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \text{value of } s \text{ for which } \frac{\partial f_{\mathbf{z}}(z|\mathbf{s}=s)}{\partial s} = 0$$

If we employ the log-likelihood function, the solution is as follows,

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \text{value of } s \text{ for which } \frac{\partial \ln f_{\mathbf{z}}(z|\mathbf{s}=s)}{\partial s} = 0$$

Example 3.2

Suppose $\mathbf{s}$ and $\mathbf{z}$ are random variables with joint pdf

$$f_{\mathbf{s},\mathbf{z}}(s,z) = \begin{cases} \frac{1}{12}(s+z)e^{-z}, & 0 \le s \le 4, 0 \le z \le \infty; \\ 0, & \text{otherwise;} \end{cases}$$

The goal is to compute the ML estimate of $\mathbf{s}$ based on z.

Find the likelihood function:

$$f_{\mathbf{z}}(z|\mathbf{s}=s) = \frac{f_{\mathbf{s},\mathbf{z}}(s,z)}{f_{\mathbf{s}}(s)}$$

As

$$f_{\mathbf{s}}(s) = \int_{-\infty}^{\infty} f_{\mathbf{s},\mathbf{z}}(s,z)dz = \frac{1}{12}(s+1),\ 0 \le s \le 4.$$

the likelihood function is

$$f_{\mathbf{z}}(z|\mathbf{s}=s) = \frac{s+z}{s+1}e^{-z},\ 0 \le s \le 4, 0 \le z \le \infty$$

Find the value of s that maximizes $f_{\mathbf{z}}(z|\mathbf{s}=s)$.

Calculate the partial derivative as:

$$\frac{\partial f_{\mathbf{z}}(z|\mathbf{s}=s)}{\partial s} = \frac{1-z}{(s+1)^2}e^{-z}$$

Hence

$$\hat{s}_{\mathrm{ML}} = \begin{cases} 4 & 0 \le z < 1 \\ 2 & z = 1 \\ 0 & z > 1 \end{cases}$$

Example 3.3 ML Estimation with Gaussian Noise

Suppose that $\mathbf{z} = \mathbf{s} + \mathbf{v}$, where $\mathbf{s}$ and $\mathbf{v}$ are independent and $\mathbf{v} \sim \mathcal{N}(0, \sigma^2)$, i.e.,

$$f_{\mathbf{v}}(v) = \frac{1}{\sqrt{2\pi}\sigma}e^{-v^2/2\sigma^2}$$

Given the sample realization z, derive the maximum likelihood estimation $\hat{s}_{\mathrm{ML}}$.

As $F(z|\mathbf{s}=s) = P(\mathbf{z} \le z|\mathbf{s}=s) = \frac{P(\mathbf{z}\le z, \mathbf{s}=s)}{P(\mathbf{s}=s)} = P(\mathbf{v} \le z-s)$, the likelihood function is

$$f_{\mathbf{z}}(z|\mathbf{s}=s) = f_{\mathbf{v}}(v)|_{v=z-s} = \frac{1}{\sqrt{2\pi}\sigma}e^{-(z-s)^2/2\sigma^2}$$

Thus

$$\hat{s}_{\mathrm{ML}} = z, \hat{\mathbf{s}}_{\mathrm{ML}} = \mathbf{z}.$$

3.3 Maximum a posteriori estimation

Apart from maximizing the likelihood function, another important optimality criterion is to maximizes the conditional density $f_{\mathbf{s}}(s|\mathbf{z}=z)$, resulting in the problem of Maximum *a posteriori* estimation. The density is known as the *a posteriori* density since it is the density after the measurement z becomes available.

The problem of maximum a posteriori (MAP) estimate can be described in the following optimization problem,

$$\hat{\mathbf{s}}_{\text{MAP}} = \text{value of } \mathbf{s} \text{ that maximizes } f_{\mathbf{s}}(s|\mathbf{z}=z)$$

Assume that $f_{\mathbf{s}}(s|\mathbf{z}=z)$ is differentiable and has a unique maximum in the interior of its domain, we have

$$\hat{\mathbf{s}}_{\text{MAP}} = \text{value of } \mathbf{s} \text{ for which } \frac{\partial f_{\mathbf{s}}(s|\mathbf{z}=z)}{\partial s} = 0,$$

By Bayes' formula, we then have the following,

$$f_{\mathbf{s}}(s|\mathbf{z}=z) = \frac{f_{\mathbf{z}}(z|\mathbf{s}=s)f_{\mathbf{s}}(s)}{f_{\mathbf{z}}(z)}$$

Thus,

$$\hat{\mathbf{s}}_{\text{MAP}} = \text{value of } \mathbf{s} \text{ that maximizes } f(z|\mathbf{s}=s)f_{\mathbf{s}}(s).$$

The difference between ML estimate and MAP estimate is that the objective functions those be optimized are different, i.e., $\max_s f_{\mathbf{z}}(z|\mathbf{s}=s)$ for ML estimate and $\max_s f_{\mathbf{s}}(s|\mathbf{z}=z)$ for MAP estimate, which can be also written as $\max_s f_{\mathbf{z}}(z|\mathbf{s}=s)f_{\mathbf{s}}(s)$. This means that in MAP estimate, we have to know the density $f_{\mathbf{s}}(s)$ additionally, indicating that the MAP estimate is a Bayesian estimation. In fact, in ML estimate, the objective is more appropriate to be called "likelihood function"

Example 3.4 MAP estimation with Gaussian noise

Additive-noise $\mathbf{z}=\mathbf{s}+\mathbf{v}$, where $\mathbf{v}\sim\mathcal{N}(0,\sigma_v^2)$. Assume $\mathbf{s}\sim\mathcal{N}(\eta_s,\sigma_s^2)$. Then

$$f_{\mathbf{s}}(s) = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(s-\eta_s)^2}{2\sigma_s^2}}$$

As

$$f_{\mathbf{z}}(z|\mathbf{s}=s) = \frac{1}{\sqrt{2\pi}\sigma_v} e^{-(z-s)^2/2\sigma_v^2}$$

we have

$$f(z|\mathbf{s}=s)f_{\mathbf{s}}(s) = \frac{1}{2\pi\sigma_s\sigma_v}\exp\left[-\frac{(z-s)^2}{2\sigma_v^2} - \frac{(s-\eta_s)^2}{2\sigma_s^2}\right]$$

Differentiating the term $\left[-\frac{(z-s)^2}{2\sigma_v^2} - \frac{(s-\eta_s)^2}{2\sigma_s^2}\right]$ with respect to s yields,

$$\hat{\mathbf{s}}_{\text{MAP}} = \frac{\sigma_v^2}{\sigma_v^2+\sigma_s^2}\eta_s + \frac{\sigma_s^2}{\sigma_v^2+\sigma_s^2}z$$

When the noise power is much less than the signal power, i.e., $\sigma_v^2 \ll \sigma_s^2$, we have

$$\hat{\mathbf{s}}_{\text{MAP}} = z = \hat{\mathbf{s}}_{\text{ML}}.$$

(It is implied that there is no *a priori* information about $\mathbf{s}$)

3.4 Naïve Bayes and Logistic Regression

3.4.1 Naïve Bayes

Along with decision trees, neural networks, nearest NBR, Naïve Bayes classifier is one of the most practical learning methods. For moderate or large training set , when the attributes that describe instances are **conditionally independent** given classification, Naïve Bayes classifier can be used. Successful applications of Naïve Bayes classifier includes diagnosis and classifying text documents.

For 2 RVs X and Y, assume Y is categorial, given instances X^{new}, we often want to want to find the class Y belongs to, i.e., $P(Y|X^{new})$. In order to achieve this, Naïve Bayes uses a generative method, i.e., we first learn (estimate) $P(X|Y), P(Y)$, and then using Bayes' rule to calculate $P(Y|X^{new})$.

A direct concern is that how shall we represent $P(X|Y), P(Y)$ and how many parameters should we estimate? This formulates the problem of Bayes Classifier.

In Bayes classifier, suppose $X = [X_1, ...X_n]$ where X_i and Y are boolean RVs. For each instance, we need to estimate $2(2^n - 1)$ such parameters

$$\theta_{ij} = P(X = x_i|Y = y_j)$$

in which x_i is a n-element vector. For example, if X is a vector containing 30 Boolean features, then we will need to estimate more than 3 billion parameters!

In Naïve Bayes Classifier, we assume $X = [X_1, ...X_n]$ and Y is discrete-valued, besides, suppose conditional independence holds, i.e.,

$$P(X_1 \dots X_n|Y) = \prod_i P(X_i|Y),$$

i.e., X_i and X_j are conditionally independent given Y, for all $i \neq j$. The conditionally independent can be formally described as follows.

X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z, i.e.,

$$P(X = x_i|Y = y_j, Z = z_k) = P(X = x_i|Z = z_k), \forall i, j, k,$$

which can be often written as,

$$P(X|Y, Z) = P(X|Z).$$

For example, from daily experience, we have (approximately)

$$P(Thunder|Rain, Lightening) = P(Thunder|Lightening).$$

Naïve Bayes uses the assumption that X_i are conditionally independent given Y

then

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y) = P(X_1|Y)P(X_2|Y)$$

$$P(X_1, \dots, X_n|Y) = \prod_i P(X_i|Y)$$

In this case, the number of parameters needed for computing $P(X|Y), P(Y)$ can be calculated. Basically, we need only $2n$ parameters to define $P(X_i = x_{ik}|Y = y_j)$. Denote $x_k = [x_{1k}, \ldots, x_{nk}]$, then we have

$$P(X = x_k|Y = y_j) = \prod_{i=1}^{n} P(X_i = x_{ik}|Y = y_j)$$

According to Bayes' rule, we have

$$P(Y = y_j|X_1, \ldots, X_n) = \frac{P(Y = y_j)P(X_1, \ldots, X_n|Y = y_j)}{\sum_m P(Y = y_m)P(X_1, \ldots, X_n|Y = y_m)}.$$

As the conditional independence holds, we then have

$$P(Y = y_j|X_1, \ldots, X_n) = \frac{P(Y = y_j)\prod_i P(X_i|Y = y_j)}{\sum_m P(Y = y_m)\prod_i P(X_i|Y = y_m)}$$

Therefore, the classification rule for $X^{\text{new}} = [x_1^{\text{new}}, \ldots, x_n^{\text{new}}]$ is

$$Y^{\text{new}} \leftarrow \arg\max_{y_j} P(Y = y_j)\prod_i P(X_i = x_i^{\text{new}}|Y = y_j)$$

Clearly the Naïve Bayes is an MAP estimate.

The Naïve Bayes algorithm can be described as follows.

Algorithm 3.1 *Naïve Bayes algorithm.*

Naïve_Bayes_Learn(examples)
1. For each target value y_j estimate $\pi_j = P(Y = y_j)$.
2. For each attribute value x_{ik} of each attribute X_i, estimate $\theta_{ijk} = P(X_i = x_{ik}|Y = y_j)$.

Classify_New_Instance(x)
$Y^{new} = \text{argmax}_{y_j} P(Y = y_j)\prod_i P(X_i|Y = y_j)$ ($Y^{new} = \text{argmax}_{y_j}\pi_j\theta_{ijk}$)

It is noted that the parameters must sum to 1, i.e.,

$$\sum_j \pi_j = 1,$$

$$\sum_k \theta_{ijk} = 1, \forall i, j$$

We then estimate the number of parameters needed in the Naïve Bayes algorithm when Y, X_i are discrete-valued. In the maximum likelihood estimates, we have

$$\pi_j = P(Y = y_j) = \frac{\#D\{Y = y_j\}}{|D|}$$

$$\theta_{ijk} = P(X_i = x_{ik}|Y = y_j) = \frac{\#D\{X_i = x_{ik}, Y = y_j\}}{\#D\{Y = y_j\}},$$

where $\#D()$ denotes the number of items in data set D.

If we assume a Dirichlet prior distribution over the θ_{ijk} parameters, with equal-valued parameters, we have the following MAP estimate,

$$\pi_j = P(Y = y_j) = \frac{\#D\{Y = y_j\} + l}{|D| + lR},$$

where R is the number of distinct values Y can take on, and l determines the strength of the prior assumptions relative to the observed data D.

$$\theta_{ijk} = P(X_i = x_{ik}|Y = y_j) = \frac{\#D\{X_i = x_{ik}, Y = y_j\} + l}{\#D\{Y = y_j\} + lM}, \tag{3.2}$$

where M is the number of distinct values X_i can take on, and l again determines the strength of this smoothing (i.e., the number of hallucinated examples is lM). If l is set to 1, this approach is called Laplace smoothing.

We then give an example to illustrate the principle of Naïve Bayes.

Example 3.5 An example for classification

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rainy	Mild	High	Weak	Yes
D5	Rainy	Cool	Normal	Weak	Yes
D6	Rainy	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rainy	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rainy	Mild	High	Strong	No

Given the training data listed in the table, $X = [Outlook, Temperature, Humidity, Wind]$, $Y = PlayTennis$, we want to give an optimal estimate of $P(Y|X)$.

Consider *PlayTennis*, and new instance

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong\rangle$$

Want to compute:

$$Y^{new} = \text{argmax}_j P(Y = y_j) \prod_i P(X_i = x_{ik}|Y = y_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$
$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow Y^{new} = n$$

It is noted that in Naïve Bayes, the conditional independence assumption is often violated, i.e.,

$$P(X_1, X_2 \dots X_n | Y_j) \neq \prod_i P(X_i | Y_j).$$

However, it works surprisingly well anyway. Besides, the estimated posteriors $P(Y_j|X)$ does not need to be correct, and only the following is needed,

$$\begin{aligned} &\text{argmax}_{y_j} P(Y = y_j) \textstyle\prod_i P(X_i | Y = y_j) = \\ &\text{argmax}_{y_j} P(Y = y_j) P(X_1 \dots, X_n | Y = y_j) \end{aligned}$$

Readers can see [Domingos & Pazzani, 1996] for analysis.

It is also noted that Naïve Bayes posteriors often unrealistically close to 1 or 0.

In some cases, none of the training instances with target value y_j have attribute value x_{ik}. Then we have

$$P(X_i = x_{ik} | Y = y_j) = 0, \text{ and...} P(Y = y_j) \prod_i P(X_i = x_{ik} | Y = y_j) = 0.$$

At this time we can use MAP Estimate (3.2) mentioned before.

In other cases, we have continuous X_i. For example, in image classification, assume X_i is the i-th pixel, we can use the Gaussian Naïve Bayes (GNB). One common approach in GNB is to assume that for each possible discrete value y_j of Y , the distribution of each continuous X_i is Gaussian, and is defined by a mean and standard deviation specific to X_i and y_j. In order to train such a GNB classifier we must therefore estimate the mean and standard deviation of each of these Gaussians with respect to each attribute X_i and each possible value y_j of Y, i.e.,

$$\begin{aligned} \mu_{ij} &= E[X_i | Y = y_j] \\ \sigma_{ij}^2 &= E[(X_i - \mu_{ij})^2 | Y = y_j] \end{aligned}$$

Note there are $2nK$ of these parameters, all of which must be estimated independently.

The maximum likelihood estimation for the quantities μ_{ij} can be described as,

$$\hat{\mu}_{ij} = \frac{1}{\sum_k \delta(Y^k = y_j)} \sum_k X_i^k \delta(Y^k = y_j)$$

where the superscript k refers to the kth training example, and

$$\delta(Y = y_j) = \begin{cases} 1 & Y = y_j \\ 0 & Y \neq y_j \end{cases}$$

The role of δ here is to select only those training examples for which $Y = y_k$.

The maximum likelihood estimator for σ_{ij}^2 is

$$\hat{\sigma}_{ij}^2 = \frac{1}{\sum_j \delta(Y^k = y_j)} \sum_j (X_i^k - \hat{\mu}_{ij})^2 \delta(Y^k = y_j)$$

Application of Naïve Bayes includes credit scoring; medical data classification; classify which emails are spam; classify which emails are meeting invitations; classify which web pages are student home pages (recommendation system); and sentiment analysis.

3.4.2 Logistic regression

The Naïve Bayes mentioned in Section 3.4.1 is basically the generative classifiers, in which some functional form for $P(X|Y), P(Y)$ are assumed, and the estimation of $P(X|Y), P(Y)$ can be performed directly from the training data. Then Bayes' rule is used to calculate $P(Y|X = x_i)$.

In contrast, the discriminative classifier assume some functional form for $P(Y|X)$, and estimate the parameters of $P(Y|X)$ directly from the training data.

In general, a discriminative model models the decision boundary between the classes. A generative model explicitly models the actual distribution of each class.

In fact, the generative classifier Gaussian Naïve Bayes can be generalized to Logistic Regression, which is a discriminative classifier. Consider learning $f : X \to Y$, where $X = [X_1, \ldots, X_n]^T$ is vector of real-valued features, Y is Boolean. We could use a Gaussian Naïve Bayes classifier by assuming the following assumptions are valid,

- Y is Boolean, governed by a Bernoulli distribution, with parameter $\pi = P(Y = 1)$
- $X = [X_1, \ldots, X_n]$, where each X_i is a continuous random variable
- For each X_i, $P(X_i|Y = y_k)$ is a Gaussian distribution of the form $\mathcal{N}(\mu_{ij}, \sigma_i)$
- For all i and $m \neq i$, X_i and X_m are conditionally independent given Y

thus we have the following expression for the conditional probability

$$P(X_i = x|Y = y_j) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_{ij})^2}{2\sigma_i^2}}.$$

It is apparent that the variance is independent of Y. Sometimes we can assume the variance is independent of X_i (i.e., use σ_k instead). There are also cases when the variance is independent of both X and Y, i.e., we use σ for the variance.

The form of $P(Y|X)$ can be derived as follows.

$$\begin{aligned} P(Y = 1|X) &= \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1)+P(Y=0)P(X|Y=0)} \\ &= \frac{1}{1+\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}} \\ &= \frac{1}{1+\exp\left(\ln\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}\right)} \\ &= \frac{1}{1+\exp\left[\ln\frac{1-\pi}{\pi}+\sum_i \ln\frac{P(X_i|Y=0)}{P(X_i)|Y=1}\right]} \end{aligned}$$

As $P(X_i|Y = y_j)$ is Gaussian, we have

$$\ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}x_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2},$$

then $P(Y = 1|X)$ can be expressed as,

$$P(Y = 1|X) = \frac{1}{1+\exp(\omega_0 + \sum_{i=1}^{n}\omega_i x_i)}, \tag{3.3}$$

where

$$\omega_0 = \ln \frac{1-\pi}{\pi} + \sum_i \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}, \omega_i = \frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2}$$

We also have

$$P(Y=0|X) = \frac{\exp(w_0 + \sum_{i=1}^n w_i x_i)}{1 + \exp(w_0 + \sum_{i=1}^n w_i x_i)}. \tag{3.4}$$

Therefore, the following equality is valid,

$$\frac{P(Y=0|X)}{P(Y=1|X)} = \exp(w_0 + \sum_{i=1}^n w_i x_i),$$

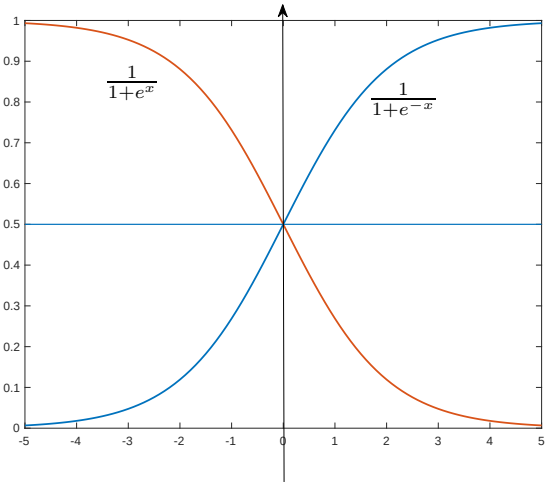
which implies

$$\ln \frac{P(Y=0|X)}{P(Y=1|X)} = w_0 + \sum_{i=1}^n w_i x_i.$$

And we can see that this is a linear classification rule!!

It is clear that the value of the weights w_i can be provided by the parameters estimated by the GNB classifier.

Indeed, the probability $P(Y=0|X)$ is now a logistic function, the shape of which is shown in Fig. 3.4.2 for illustration. Apparently, if $x > 0$, which means



$w_0 + \sum_i w_i x_i > 0$ in (3.4), we have

$$P(Y=0|X) > \frac{1}{2},$$

meaning that we should assign the label $Y = 0$.

In general, the Logistic Regression method, which is a discriminative method, we may suspect the GNB assumptions are not perfectly satisfied. However, the

form of $P(Y|X)$, which is a logistic function, still holds, and we can estimate the w_i parameters directly from the data.

Specifically, we choose parameters $W = [w_0, \ldots, w_n]$ to maximize conditional likelihood of training data

Given training data $D = \{(X^1, Y^1), \ldots, (X^L, Y^L)\}$, the data conditional likelihood is $\prod_l P(Y^l|X^l, W)$, and we obtain W through solving the following optimization problem,

$$\max_W \ln \prod_l P(Y^l|X^l, W) \tag{3.5}$$

The log likelihood can be further written as,

$$l(W) = \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

To keep our derivation consistent with common usage, we will in this section flip the assignment of the boolean variable Y so that we assign

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i x_i)}$$

and

$$P(Y = 1|X) = \frac{\exp(w_0 + \sum_{i=1}^n w_i x_i)}{1 + \exp(w_0 + \sum_{i=1}^n w_i x_i)}.$$

Then we have

$$\begin{aligned} l(W) &= \sum_l \ln P(Y^l|X^l, W) \\ &= \sum_l \left[Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W)\right] \\ &= \sum_l \left[Y^l \ln \tfrac{P(Y^l=1|X^l,W)}{P(Y^l=0|X^l,W)} + \ln P(Y^l = 0|X^l, W)\right] \\ &= \sum_l \left[Y^l (w_0 + \sum_{i=1}^n w_i x_i^l) - \ln(1 + \exp(w_0 + \sum_{i=1}^n w_i x_i^l))\right] \end{aligned}$$

Although $l(W)$ is a concave function of W, it is not easy to solve the optimization problem (3.5). Besides, there is no closed-form solution for W, and we use Gradient-based (Gradient-ascent) method.

Clearly the Logistic Regression problem (3.5) is an ML estimation, and following we'll discuss about MAP estimation, which includes regularization in regular Logistic Regression. On common approach is to define priors on W, we can assume W conforms to normal distribution, have zero mean or identity covariance. Then the MAP estimate becomes,

$$W \leftarrow \text{argmax}_W \ln P(W|\{X^l, Y^l\})$$

which can then be written as,

$$W \leftarrow \text{argmax}_W \ln P(W) P(Y^l|X^l, W),$$

i.e.,

$$\sum_l \ln P(Y^l|X^l, W) + \ln P(W).$$

If we assume $P(W)$ conforms to a zero mean Gaussian distribution, then $\ln P(W)$ yields a term proportional to $\|W\|^2$.

$$W \leftarrow \text{argmax}_W \sum_l \ln P(Y^l|X^l, W) - \frac{\lambda}{2}\|W\|^2.$$

The last term can helps avoid very large weights and overfitting.

In more general case, where $Y \in \{Y_1, \ldots, Y_R\}$, Logistic Regression can be used for functions with many discrete values, and we have to learn $R-1$ sets of weights. In this case, for $k < R$, we assume

$$P(Y = y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki}X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

and for $k = R$, we have

$$P(Y = y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}.$$

This is the activation function of the softmax layer in CNN, which is used to give multiple classes.

In the end of this section, we can give the relationship between Gaussion Naïve Bayes (GNB) classifiers and Logistic Regression. Basically, GNB is a generative classifier while Logistic Regression is a discriminative classifier. They can be transformed in some ways, however, when the GNB modeling assumptions do not hold, Logistic Regression and GNB typically learn different classifier function. In general, the asymptotic (#samples $\to \infty$) classification accuracy for Logistic Regression is often better than that of GNB. Besides, GNB and Logistic Regression converges toward their asymptotic accuracies at different rates, i.e., GNB parameter estimates converge in order $\log n$ examples and Logistic Regression requiring order n examples. In general, when many training examples are available, we choose Logistic Regression, otherwise choose GNB.

3.5 Minimum Mean-Square Error Estimation

Apart from ML estimate and MAP estimate, in this section, we introduce another estimate, which is the minimum mean-square error estimate. First, we would introduce the minimum mean-square error,

$$\text{MSE} = \text{E}[\text{E}[\tilde{\mathbf{s}}^2|\mathbf{z}]] = \text{E}[\text{E}[(\mathbf{s} - \hat{\mathbf{s}})^2|\mathbf{z}]] = \text{E}[(\mathbf{s} - \hat{\mathbf{s}})^2]$$

The above equality can be shown through the following.

Given jointly distributed RVs $\mathbf{x}$ and $\mathbf{y}$ with conditional density function $f_y(y|\mathbf{x} = x)$, the conditional expectation of $\mathbf{y}$ given $\mathbf{x} = x$, denoted by $E[\mathbf{y}|\mathbf{x} = x]$, is defined by

$$E[\mathbf{y}|\mathbf{x} = x] = \int_{-\infty}^{\infty} y f_y(y|\mathbf{x} = x) dy$$

Let Ψ denote the real-valued function given by

$$\Psi(x) = \int_{-\infty}^{\infty} y f_y(y|\mathbf{x} = x) dy,$$

which is a function of x.

Then we can define the RV

$$\Psi(\mathbf{x}) = E[\mathbf{y}|\mathbf{x}],$$

and thus the conditional expectation can be viewed as an RV. Then we have

$$E[E[\mathbf{y}|\mathbf{x}]] = \int_{-\infty}^{\infty} \Psi(x) f_x(x) dx = E[\mathbf{y}].$$

The MSE gives the average power of the error.

Given the RV $\mathbf{z}$, It can be proved that the MMSE estimate $\hat{s}_{\text{MMSE}}$ of $\hat{s}$ is the conditional expectation

$$\hat{s}_{\text{MMSE}} = E[\mathbf{s}|\mathbf{z}].$$

The proof is not shown here, interested readers can resort to reference for detail.

The properties of MMSE estimate can be shown as follows,

- MMSE estimate $\hat{\mathbf{s}}_{\text{MMSE}}$ is unique;
- MMSE estimate requires information about $\mathbf{s}$, is another type of Baysian estimation;
- MMSE estimate $\hat{\mathbf{s}}_{\text{MMSE}}$ is unbiased, i.e.,

$$E(\hat{\mathbf{s}}) = E(\mathbf{s}) \text{ or } E(\tilde{\mathbf{s}}) = 0$$

- Generalization to a finite number of measurements $\mathbf{z}(1), \ldots, \mathbf{z}(n)$,

$$\hat{\mathbf{s}}_{\text{MMSE}} = E[\mathbf{s}|\mathbf{z}(1), \ldots, \mathbf{z}(n)], \quad E[\mathbf{s} - \hat{\mathbf{s}}_{\text{MMSE}}] = 0$$

Example 3.6 MMSE estimate with Gaussian noise

Again consider the additive-noise case

$$\mathbf{z} = \mathbf{s} + \mathbf{v}$$

with $\mathbf{v} \sim \mathcal{N}(0, \sigma_v^2)$ and $\mathbf{s} \sim \mathcal{N}(\bar{s}, \sigma_s^2)$. Assume that $\mathbf{s}$ and $\mathbf{v}$ are uncorrelated, then

$$E[\mathbf{z}] = E[\mathbf{s}] = \bar{s},$$

$$\text{Var}[\mathbf{z}] = \text{Var}[\mathbf{s}] + \text{Var}[\mathbf{v}] = \sigma_s^2 + \sigma_v^2$$

The pdf of $\mathbf{z}$ is

$$f_{\mathbf{z}}(z) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma_s^2 + \sigma_v^2}} \exp\left[-\frac{(z-\bar{s})^2}{2(\sigma_s^2+\sigma_v^2)}\right],$$

which can be further written as,

$$f_{\mathbf{s}}(s|\mathbf{z}=z) = \frac{1}{2\pi\sigma_s\sigma_v f_{\mathbf{z}}(z)} \exp\left\{-\left[\frac{(z-s)^2}{2\sigma_v^2} + \frac{(s-\bar{s})^2}{2\sigma_s^2}\right]\right\},$$

i.e.,

$$f_{\mathbf{s}}(s|\mathbf{z}=z) = \frac{1}{\sqrt{2\pi}\sqrt{\frac{\sigma_s^2\sigma_v^2}{\sigma_s^2+\sigma_v^2}}} \exp\left[-\frac{(s-\hat{s}_{\text{MAP}})^2}{2\frac{\sigma_s^2\sigma_v^2}{\sigma_s^2+\sigma_v^2}}\right].$$

Therefore, the MMSE estimate can be obtained, i.e.,

$$\hat{s}_{\text{MMSE}} = E[\mathbf{s}|\mathbf{z}=z] = \hat{s}_{\text{MAP}} = \bar{s} + \frac{\sigma_s^2}{\sigma_v^2+\sigma_s^2}(z-\bar{s})$$

It is noted that if $\mathbf{s}$ and $\mathbf{v}$ are uncorrelated, then the MMSE estimate of $\mathbf{s}$ is identical to the MAP estimate.

For MMSE estimate, an important principle, called the orthogonality principle, holds. The orthogonality principle in this case shows that the error $\mathbf{s} - E[\mathbf{s}|\mathbf{z}]$ is orthogonal to every function $\gamma(\mathbf{z})$, i.e.,

$$E[(\mathbf{s} - E[\mathbf{s}|\mathbf{z}])\gamma(\mathbf{z})] = 0.$$

Sketch of proof. $E[(\mathbf{s}-E[\mathbf{s}|\mathbf{z}])\gamma(\mathbf{z})] = E\left\{E[(\mathbf{s} - E[\mathbf{s}|\mathbf{z}])\gamma(\mathbf{z})]|\mathbf{z}\right\} = E\left\{E[(\mathbf{s} - E[\mathbf{s}|\mathbf{z}])|\mathbf{z}]\gamma(\mathbf{z})\right\}$

We now explain the necessary and sufficient condition for an MMSE estimate.

The estimate given by $\hat{\mathbf{s}} = \alpha(\mathbf{z})$ is the MMSE estimate of $\mathbf{s}$ given $\mathbf{z}$ if and only if the error $\mathbf{s} - \alpha(\mathbf{z})$ is orthogonal to every function $\gamma(\mathbf{z})$; that is

$$E[(\mathbf{s} - \alpha(\mathbf{z}))\gamma(\mathbf{z})] = 0.$$

This means that $\hat{\mathbf{s}} = \alpha(\mathbf{z})$ is the MMSE estimate is equivalent to that $(\mathbf{s} - \alpha(\mathbf{z})) \perp \gamma(\mathbf{z})$.

In general, the conditional expectation $E[\mathbf{s}|\mathbf{z}]$ is difficult to obtain, this is because that $f_{\mathbf{s}}(s|\mathbf{z}=z)$ is difficult to be find. Therefore, it is difficult to find both the MAP and the MMSE estimation. Hence, we can sacrifice some of the optimality to facilitate the process of finding an estimate, i.e., restrict the estimation problem to produce a tractable solution for α, and trading overall optimality for tractability. A typical restriction is the linear restriction, i.e.,

$$\hat{\mathbf{s}} = \lambda\mathbf{z}$$

The linear MMSE (LMMSE) estimation problem can be formulated as,

$$\min_{\lambda} \text{ MSE} = E[(\mathbf{s} - \lambda\mathbf{z})^2] = E[\mathbf{s}^2 - 2\lambda\mathbf{s}\mathbf{z} + \lambda^2\mathbf{z}^2]$$

In order to find the optimal λ, we take the partial derivative with respect to λ, set the result equal to zero and get

$$-2E(\mathbf{s}\mathbf{z}) + 2\lambda E(\mathbf{z}^2) = 0, \lambda = \frac{E(\mathbf{s}\mathbf{z})}{E(\mathbf{z}^2)}$$

The LMMSE estimate is then given by

$$\hat{s}_{\text{LMMSE}} = \alpha(\mathbf{z}) = \frac{E(\mathbf{sz})}{E(\mathbf{z}^2)} \cdot \mathbf{z}$$

Apparently, there are advantages for LMMSE estimate over the former estimates. In LMMSE estimate, knowledge about any likelihood function or densities is not required, and we only need the second-order moments $E[\mathbf{sz}]$ and $E[\mathbf{z}^2]$. $E[\mathbf{sz}]$ and $E[\mathbf{z}^2]$ can be estimated from experimental training data $(s_i, z_i)_{i=1}^M$, i.e.,

$$E(\mathbf{sz}) \approx \frac{1}{M} \sum_{m=1}^{M} s_m z_m$$

and

$$E(\mathbf{z}^2) \approx \frac{1}{M} \sum_{m=1}^{M} z_m^2$$

Example 3.7 Consider again the Gaussian additive noise problem.

$$\mathbf{z} = \mathbf{s} + \mathbf{v}$$

with $\mathbf{v} \sim \mathcal{N}(0, \sigma_v^2)$ and $\mathbf{s} \sim \mathcal{N}(\bar{s}, \sigma_s^2)$. Assume that $\mathbf{s}$ and $\mathbf{v}$ are uncorrelated, we have

$$\hat{\mathbf{s}}_{\text{LMMSE}} = \frac{E(\mathbf{sz})}{E(\mathbf{z}^2)}\mathbf{z} = \frac{\bar{s}^2 + \sigma_s^2}{\bar{s}^2 + \sigma_s^2 + \sigma_v^2} \cdot \mathbf{z}$$

which is different from $\hat{\mathbf{s}}_{\text{MMSE}}$ and is biased.

Orthogonality principle also holds for 1-dimensional LMMSE estimate.

Let $\alpha(\mathbf{z})$ be the LMMSE estimate of $\mathbf{s}$ given $\mathbf{z}$. Then the error $\mathbf{s} - \alpha(\mathbf{z})$ is orthogonal to every linear function $\gamma(\mathbf{z})$, i.e.,

$$E[(\mathbf{s} - \alpha(\mathbf{z}))\gamma(\mathbf{z})] = 0.$$

Sketch of proof. Assume $\gamma(\mathbf{z}) = \beta\mathbf{z}$

For vector RVs, the following derives the LMMSE estimate.

Let $\mathbf{s} \in \mathbf{R}^m$ and $\mathbf{z} \in \mathbf{R}^q$. Assume the LMMSE takes the form $\hat{\mathbf{s}} = M\mathbf{z}$, where M is an $m \times q$ matrix to be determined.

Assume $P = E[(\mathbf{s} - \hat{\mathbf{s}})(\mathbf{s} - \hat{\mathbf{s}})^T]$, then

$$\text{MSE} = \text{tr}(P) = E[(\mathbf{s} - \hat{\mathbf{s}})^T(\mathbf{s} - \hat{\mathbf{s}})],$$

i.e., the trace of the matrix P.

Supplementary material Matrix trace, matrix derivative.

The trace of a matrix is defined only for square matrices:

$$\text{tr}(A) = \sum_{i=1}^{n} a_{ii}$$

Some useful identities for matrix trace,

$$\begin{aligned}&\mathrm{tr}(\alpha A) = \alpha \mathrm{tr}(A)\\ &\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)\\ &\mathrm{tr}(AB) = \mathrm{tr}(BA)\\ &\mathrm{tr}(xy^T) = x^T y, x, y \in \mathbf{R}^n\end{aligned}$$

Matrix derivative
$f(A) : \mathbf{R}^{m\times n} \to 1$, $A = [A_1, A_2, \ldots, A_n]$, the Jacobian matrix is given by,

$$\nabla_A f = \frac{\partial f}{\partial A} = \begin{bmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \cdots & \frac{\partial f}{\partial a_{1n}} \\ \frac{\partial f}{\partial a_{21}} & \frac{\partial f}{\partial a_{22}} & \cdots & \frac{\partial f}{\partial a_{2n}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f}{\partial a_{n1}} & \frac{\partial f}{\partial a_{n2}} & \cdots & \frac{\partial f}{\partial a_{nn}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial A_1}, \frac{\partial f}{\partial A_2}, \ldots, \frac{\partial f}{\partial A_n} \end{bmatrix}$$

A list of derivatives

$$\begin{aligned}&\tfrac{\partial (Ax)}{\partial x} = A, \tfrac{\partial (a^T A b)}{\partial A} = ab^T, \tfrac{\partial (a^T A^T b)}{\partial A} = ba^T\\ &\tfrac{\partial \mathrm{tr}(C^T A B^T)}{\partial A} = \tfrac{\partial \mathrm{tr}(BA^T C)}{\partial A} = CB\\ &\tfrac{\partial^2}{\partial x \partial x^T}(Ax+b)^T C(Dx+e) = A^T CD + D^T C^T A\\ &\tfrac{\partial^2}{\partial x \partial x^T}(x^T Cx) = C + C^T\\ &\tfrac{\partial}{\partial x}(Ax+b)^T C(Dx+e) = A^T C(Dx+e) + D^T C^T (Ax+b)\\ &\tfrac{\partial}{\partial A}(Aa+b)^T C(Aa+b) = (C+C)^T (Aa+b)a^T\end{aligned}$$

Solution of the LMMSE can be found by differentiating the MSE with respect to M, which yields,

$$\frac{\partial \mathrm{tr}(P)}{\partial M} = -2E[\mathbf{s}\mathbf{z}^T] + 2ME[\mathbf{z}\mathbf{z}^T]$$

Hence

$$M = E(\mathbf{s}\mathbf{z}^T)[E(\mathbf{z}\mathbf{z}^T)]^{-1}$$

Thus the LMMSE estimate of $\hat{s}$ given $\mathbf{z}$ is

$$\hat{\mathbf{s}}_{\mathrm{LMMSE}} = E(\mathbf{s}\mathbf{z}^T)\left[E(\mathbf{z}\mathbf{z}^T)\right]^{-1}\mathbf{z}$$

For vector LMMSE estimate, the orthogonality principle also holds.

Let $\mathbf{s} \in \mathbf{R}^m$ and $\mathbf{z} \in \mathbf{R}^q$ be jointly distributed random vectors, let $\mathbf{s} = \alpha(\mathbf{z})$ be the LMMSE estimate of $\mathbf{s}$ given $\mathbf{z}$. Then the estimation error $\mathbf{s} - \hat{\mathbf{s}}$ is orthogonal to $\mathbf{z}$, i.e.,

$$E[(\mathbf{s} - \hat{\mathbf{s}})\mathbf{z}^T] = \mathbf{0}.$$

Sketch of proof. Just use the expression for $\hat{\mathbf{s}}$.

The orthogonality principle for vector LMMSE estimate can be illustrated using Fig. 3.4. In fact, the orthogonality principle is necessary and sufficient for the estimate to be LMMSE estimate.

Let $\alpha(\mathbf{z})$ be a linear estimator of $\mathbf{s}$ given $\mathbf{z}$. Then α minimizes the MSE if and only if the error $\mathbf{s} - \alpha(\mathbf{z})$ is orthogonal to the measurement $\mathbf{z}$,

$$E\left\{[\mathbf{s} - \alpha(\mathbf{z})]\mathbf{z}^T\right\} = 0$$

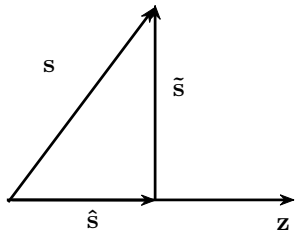


Figure 3.1 Illustration of orthogonality principle in 2-dimension.

We can use this to find the optimum linear estimator.

It is noted that under certain circumstances, the LMMSE estimate is also the MMSE estimate.

When $\mathbf{s}$ and $\mathbf{z}$ are jointly Gaussian, the LMMSE estimate is also the optimal MMSE estimate.

Proof. Suppose that $\mathbf{s}$ and $\mathbf{z}$ have a zero-mean bivariate Gaussian distribution with covariance matrix P given by

$$P = \begin{bmatrix} \sigma_s^2 & \mathrm{Cov}(\mathbf{s},\mathbf{z}) \\ \mathrm{Cov}(\mathbf{z},\mathbf{s}) & \sigma_z^2 \end{bmatrix}$$

then the joint distribution function is,

$$f(s,z) = \frac{1}{2\pi\sqrt{\det(P)}} \exp\left[-\frac{1}{2}[x\ z]P^{-1}\begin{bmatrix} x \\ z \end{bmatrix}\right],$$

which can be equivalently written as,

$$f(s,z) = \frac{1}{2\pi\sigma_s\sigma_z\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2\sigma_s^2\sigma_z^2(1-\rho^2)}\left[\sigma_z^2 s^2 - 2C_{sz}\cdot sz + \sigma_s^2 z^2\right]\right\}$$

where ρ is the correlation coefficient between s and z, i.e., $\rho = \frac{C_{sz}}{\sigma_s\sigma_z}$.

The conditional density function $f_{\mathbf{s}}(s|\mathbf{z}=z)$ is then given by

$$f_{\mathbf{s}}(s|\mathbf{z}=z) = \frac{1}{\sqrt{2\pi}\sigma_s\sqrt{(1-\rho^2)}} \exp\left\{-\frac{1}{2\sigma_s^2(1-\rho^2)}\left(s - \frac{E(\mathbf{sz})}{E(\mathbf{z}^2)}z\right)^2\right\}$$

Therefore, we have

$$\hat{s}_{\mathrm{MMSE}} = \hat{s}_{\mathrm{LMMSE}} = \frac{E(\mathbf{sz})}{E(\mathbf{z}^2)}\mathbf{z}.$$

In the last of this chapter, we'll give the comparison of different estimators.

	Maximum likelihood (ML)	Maximum *a posteriori* (MAP)
Motivation	Given z, what value of $\mathbf{s}$ is most likely to have produced z?	Given z, what value of $\mathbf{s}$ is most likely to have occured?
Objective	Maximize the likelihood function $f_{\mathbf{z}}(z\|\mathbf{s}=s)$	maximize the conditional density $f_{\mathbf{s}}(s\|\mathbf{z}=z)$. via Bayes; rule, equivalently maximize $f_{\mathbf{z}}(z\|\mathbf{s}=s)f_{\mathbf{s}}(s)$.
Esitmate	$\hat{s}_{\mathrm{ML}} = \mathrm{argmax} f_{\mathbf{z}}(z\|\mathbf{s}=s)$	$\hat{s}_{\mathrm{MAP}} = \mathrm{argmax} f_{\mathbf{z}}(z\|\mathbf{s}=s)f_{\mathbf{s}}(s)$
Required knowledge	likelihood function $f_{\mathbf{z}}(z\|\mathbf{s}=s)$	Density function $f_{\mathbf{s}}(s\|z)$ (or $f_{\mathbf{z}}(z\|\mathbf{s}=s)$ and $f_{\mathbf{s}}(s)$)

	Minimum mean-square error (MMSE)	Linear MMSE (LMMSE)
Motivation	Given z, what estimate of $\mathbf{s}$ gives the smallest MSE?	Given z, what linear function $\hat{s} = \lambda\mathbf{z}$ gives the smallest MSE?
Objective	Minimize the MSE $E[(\mathbf{s}-\hat{\mathbf{s}})^2]$	find λ to minimize $E[(\mathbf{s}-\lambda\mathbf{z})^2]$.
Esitmate	$\hat{\mathbf{s}}_{\mathrm{MMSE}} = E(\mathbf{s}\|\mathbf{z}) = \int_{-\infty}^{\infty} s f_{\mathbf{s}}(s\|\mathbf{z})ds$	$\hat{s}_{\mathrm{LMMSE}} = \lambda\mathbf{z}$, where $\lambda = E[\mathbf{sz}]/E[\mathbf{z}^2]$
Required knowledge	Density $f_{\mathbf{s}}(s\|z)$	Cross-correlation of $\mathbf{s}$ and $\mathbf{z}$ $E[\mathbf{sz}]$; second moment of $\mathbf{z}$, $E(\mathbf{z}^2)$

Chapter 4

Least Squares

4.1 Linear least squares

4.1.1 Motivation

In Chapter 3, we have discussed maximum likelihood estimate, maximum a posteriori estimate, the MMSE estimate as well as the LMMSE estimate. In particular, if the second-order statistics are known, the LMMSE estimator is given by $\hat{\mathbf{s}}_{\text{LMMSE}} = E[\mathbf{s}\mathbf{z}^T](E[\mathbf{z}\mathbf{z}^T])^{-1}\mathbf{z}$. However, in may cases, the second-order statistics are unknown. An alternative approach is to estimate the coefficients from observed data, which includes 2 possible approaches, i.e., estimate the required moments from available data and build an approximate LMMSE estimator, or build an estimator that minimizes some error functional calculated from the available data.

We can compare the LMMSE estimate with the LS estimate. As mentioned in the Chapter 3, the LMMSE estimators are optimal in expectation across the ensemble of all stochastic processes with the same second order statistics. On the other hand, Least squares estimators minimize the error on a given *block* of data, in which there are no guarantees about optimality on other data sets or other stochastic processes. Under certain conditions, i.e., if the process is ergodic, the LS estimator approaches the LMMSE estimator as the size of the data set grows.

The optimality criterion employed in the LS estimate is the sum of squares, and minimizing the sum of squares gives the name of least squares. Therefore, in LS estimate, a data set where both the inputs and desired responses are known is required. LS can be applied in many areas, including plant modeling for control (system identification), prediction, inverse modeling, interference cancellation.

4.1.2 Estimation of a Constant

Now Let's give the details about the least squares problems.

Given a data set with the input $h_k(n)$ and output $y(n)$, with $k \in \{1, \ldots, M\}$

and $n \in \{1, \ldots, N\}$, assume a linear relationship holds, i.e.,

$$y(n) = \mathbf{x}^T \mathbf{h}(n) + v(n)$$

in which $\mathbf{h}(n) = [h_1(n), \ldots, h_M(n)]^T$, and $v(n)$ is the noise. The variable to be estimated is

$$\mathbf{x} = [x_1, \ldots, x_M]^T.$$

What we want to do is to estimate $\mathbf{x}$, i.e., to obtain $\hat{\mathbf{x}}$, such that the deviation between the estimated output

$$\hat{y}(n) = \hat{\mathbf{x}}^T \mathbf{h}(n)$$

and the true output $y(n)$ is minimal.

Define the estimation error as,

$$e(n) = y(n) - \hat{y}(n) = y(n) - \mathbf{x}^T \mathbf{h}(n)$$

and the sum of squared errors is,

$$E_e = \sum_{n=1}^{N} [e(n)]^2.$$

The sum of squared error can be expressed using a matrix formulation, i.e.,

$$\begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} - \begin{bmatrix} h_1(1) & \cdots & h_M(1) \\ \vdots & \ddots & \vdots \\ h_1(N) & \cdots & h_M(N) \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}$$

Denote

$$\mathbf{e} = [e(1), \ldots, e(N)]^T, \mathbf{x} = [x_1, \ldots, x_M]^T, \mathbf{y} = [y_1, \ldots, y_N]^T,$$

and let

$$H = \begin{bmatrix} h_1(1) & \cdots & h_M(1) \\ \vdots & \ddots & \vdots \\ h_1(N) & \cdots & h_M(N) \end{bmatrix}$$

then we have

$$\mathbf{e} = \mathbf{y} - H\mathbf{x}$$

What we want to solve is the following optimization problem,

$$\min \; E_e = \mathbf{e}^T \mathbf{e} = \mathbf{y}^T \mathbf{y} - \mathbf{x}^T H^T \mathbf{y} - \mathbf{y}^T H \mathbf{x} + \mathbf{x}^T H^T H \mathbf{x} \tag{4.1}$$

Again, the solving of the optimization problem can be fulfilled by the first-order condition. The necessary condition for an estimate $\hat{\mathbf{x}}$ to be optimal to the problem (4.1) is

$$\frac{\partial E_e}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}} = -\mathbf{y}^T H - \mathbf{y}^T H + 2\hat{\mathbf{x}}^T H^T H = 0,$$

then we have

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T \mathbf{y},$$

provided that the inversion of $H^T H$ exists.

The sufficient condition for $\hat{\mathbf{x}}$ to be optimal to the problem (4.1) is the second-order conditions, i.e., the matrix

$$\frac{\partial^2 E_e}{\partial \mathbf{x} \partial \mathbf{x}^T}|_{\mathbf{x}=\hat{\mathbf{x}}} = H^T H$$

is positive definite.

The rank of the matrix H affects the positive definiteness as well as the invertibility of the matrix $H^T H$, specifically, the matrix $H^T H$ is positive definite (invertible) is equivalent to the condition that H has full column rank. And in this case, we have unique optimal solution $\hat{\mathbf{x}}$. In other cases, if the columns of the matrix H are linearly dependent, then any solution $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ differ by a vector in the nullspace of H, i.e.,

$$H(\hat{\mathbf{x}}_2 - \hat{\mathbf{x}}_1) = 0.$$

The unbiasedness of the LS estimate can be checked. For the LS estimate $\hat{\mathbf{x}}$, we have

$$E[\hat{\mathbf{x}}] = (H^T H)^{-1} H^T E[y].$$

If the noise is zero-mean, then

$$E[\hat{\mathbf{x}}] = E[\mathbf{x}],$$

i.e., the LS estimator is unbiased.

Now that the LS estimate is unbiased, we also want to investigate that if LS estimate has the minimum mean square error, and under what condition does it has the MMSE. The conclusions are shown below.

Given assumptions

- $\mathbf{v}$ is zero-mean white noise, $E(\mathbf{v}\mathbf{v}^T) = \sigma^2 I$

then the LS estimate has the minimum mean square error among all the linear unbiased estimate of $\mathbf{x}$. That is, if $\bar{\mathbf{x}} = L\mathbf{y}$ and $E(\bar{\mathbf{x}}) = E(\mathbf{x})$, we have

$$E\{[\mathbf{x} - \hat{\mathbf{x}}][\mathbf{x} - \hat{\mathbf{x}}]^T\} \leq E\{[\mathbf{x} - \bar{\mathbf{x}}][\mathbf{x} - \bar{\mathbf{x}}]^T\} \tag{4.2}$$

Proof. As

$$\mathbf{y} = H\mathbf{x} + \mathbf{v}, \quad \hat{\mathbf{x}} = (H^T H)^{-1} H^T \mathbf{y}$$

we have

$$\begin{aligned} E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] &= (H^T H)^{-1} H^T E(\mathbf{v}\mathbf{v}^T) H (H^T H)^{-1} \\ &= (H^T H)^{-1} H^T \sigma^2 H (H^T H)^{-1} \\ &= \sigma^2 (H^T H)^{-1} \end{aligned}$$

As $E(\bar{\mathbf{x}}) = \mathbf{x}$, we have

$$LH = I,$$

then if we can prove that the matrix $LL^T - (H^T H)^{-1}$ is positive semidefinite, we have (4.2).

The matrix $LL^T - (H^T H)^{-1}$ can be written as,

$$L[I - H(H^H)^{-1}H]L^T,$$

and the matrix $[I - H(H^H)^{-1}H]$ equals

$$[I - H(H^H)^{-1}H] \cdot [I - H(H^H)^{-1}H]$$

Hence we can write the matrix $LL^T - (H^T H)^{-1}$ as

$$L[I - H(H^H)^{-1}H][I - H(H^H)^{-1}H]^T L^T$$

and it is clear that the above matrix is positive semidefinite.

If we include an additional assumption:

- $\mathbf{v}$ is Gaussian white noise

then we can come to the conclusion that LS estimate has the minimum mean square error among all the unbiased estimate of $\mathbf{x}$..

Example 4.1 Consider again the Gaussian additive noise problem.

$$\mathbf{z} = \mathbf{s} + \mathbf{v}.$$

Assume we have measurements $z(1), \ldots, z(N)$ and $\mathbf{v}$ is zero mean, then the LS estimate is given by

$$\hat{s}_{\mathrm{LS}} = (H^T H)^{-1} H^T [z(1), \ldots, z(N)]^T = \frac{1}{N}(z(1) + \ldots + z(N))$$

which is the same as the mean filter and is unbiased.

If $\mathbf{v}$ is Gaussian, the we can derive the maximum likelihood estimate of $\mathbf{s}$. The likelihood function can be written as

$$f(z(1), \ldots, z(N)|\mathbf{s} = s) = \prod_i f(z(i)|\mathbf{s} = s) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_v} e^{-(z(i)-s)^2/2\sigma_v^2}$$

And we have the log-likelihood function

$$\log f(z(1), \ldots, z(N)|\mathbf{s} = s) = C - \frac{1}{2\sigma_v^2} \sum_{i=1}^{N} (z(i) - s)^2,$$

where $C = \sum_i \log \frac{1}{\sqrt{2\pi}\sigma_v}$. Thus the maximum likelihood estimate is

$$\hat{s}_{\mathrm{ML}} = \hat{s}_{\mathrm{LS}}.$$

In the end of this section, we would like to give the connections between LS estimate and ML estimate. Assume

$$\mathbf{y}(n) = h(n)^T\mathbf{x} + \mathbf{v}(n)$$

Suppose the noise v conforms to the normal distribution, i.e., $\mathbf{v} \sim \mathcal{N}(0, \sigma^2)$, additionally, the errors at different times are independent, i.e., $\mathbf{v}(1), \ldots, \mathbf{v}(N)$ are independent (or uncorrelated), then the maximum likelihood estimate of $\mathbf{x}$ equals the least squares estimate

$$\hat{\mathbf{x}}_{\mathrm{ML}} = (H^TH)^{-1}H^T\mathbf{y}.$$

Proof:

$$\begin{aligned}
&P\left[\begin{pmatrix}\mathbf{y}(1)\\ \vdots \\ \mathbf{y}(N)\end{pmatrix} \le \begin{pmatrix} y_1 \\ \vdots \\ y_N\end{pmatrix} \mid \mathbf{x} = x\right] \\
&= P\left[\begin{pmatrix}\mathbf{v}(1)\\ \vdots \\ \mathbf{v}(n)\end{pmatrix} \le \begin{pmatrix} y_1 - h(1)^Tx \\ \vdots \\ y_N - h(N)^Tx\end{pmatrix}\right] \\
&= P(\mathbf{v}(1) \le y_1 - h(1)^Tx) \cdots P(\mathbf{v}(n) \le y_N - h(N)^Tx)
\end{aligned}$$

$$f(y_1, \ldots, y_N|x) = \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{\left[y_i - h(i)^Tx\right]^2}{2\sigma^2}$$

Then the log-likelihood function is

$$\log f(y_1, \ldots, y_N|x) = C - \frac{1}{2\sigma^2}\sum_i [y_i - h(i)^Tx]^2$$

Therefore the maximum likelihood estimation is

$$\hat{\mathbf{x}}_{ML} = (H^TH)^{-1}H^Ty$$

4.1.3 Weighted least squares

In the LS estimate that we discussed in Section 4.1.2, we assumed that we had an equal amount of confidence in all of our measurements. In this section, suppose we have more confidence in some measurements than others, which yields a closely related problem, i.e., the weighted least squares,

$$E_e = \sum_{n=1}^{N} w_n^2[y(n) - \mathbf{h}(n)^T\mathbf{x}]^2 = (\mathbf{y} - H\mathbf{x})^TW(\mathbf{y} - H\mathbf{x}) \tag{4.3}$$

in which $W = \mathrm{diag}\{w_1^2, \ldots, w_N^2\}$.

The cost function E_e can be written as

$$\begin{aligned}
E_e &= \mathbf{e}^TW\mathbf{e} \\
&= \mathbf{y}^TW\mathbf{y} - \mathbf{x}^TH^TW\mathbf{y} - \mathbf{y}^TWH\mathbf{x} + \mathbf{x}^TH^TWH\mathbf{x}
\end{aligned}$$

The necessary condition for a solution to be optimal for (4.3) is

$$\nabla_x f = \frac{\partial E_e}{\partial \mathbf{x}} = -\mathbf{y}^T W H + \mathbf{x}^T H^T W H = 0. \tag{4.4}$$

Additionally, the sufficient condition for an optimal solution is

$$\nabla_x^2 f = \frac{\partial^2 E_e}{\partial \mathbf{x} \partial \mathbf{x}^T} = H^T W H > 0$$

Solution of (4.4) gives rise to,

$$\hat{x} = (H^T W H)^{-1} H^T W y$$

It is noted that the uniqueness of WLS estimate requires that the matrix $H^T W H$ to be positive definite

4.2 Recursive least squares

Although the LS estimate provides a more efficient estimation than the MMSE estimate, there is a problem in the LS estimation when the number of measurements is large.

Suppose the H matrix is an $M \times n$ matrix, if we obtain the measurements sequentially and want to update our estimate of $\mathbf{x}$ with each new measurement, we need to augment the H matrix and completely recompute the estimate $\hat{\mathbf{x}}$. If the number of measurements becomes large, then the computational effort could become prohibitive. And in this case, we resort to the recursive least squares (RLS).

4.2.1 RLS I

In this kind of RLS, a linearly recursive estimator can be written in the form

$$\begin{aligned} \mathbf{y}_k &= H_k \mathbf{x} + \mathbf{v}_k \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + K_k(\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k-1}) \end{aligned} \tag{4.5}$$

Then we compute $\hat{\mathbf{x}}_k$ on the basis of the previous estimate $\hat{\mathbf{x}}_{k-1}$ and the new measurement $\mathbf{y}_k$. The matrix K_k in (4.5) is the estimator gain matrix to be determined. The quantity $(\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k-1})$ is called the correction term or innovation.

We can check whether the linear RLS estimate is unbiased. The estimation error mean can be computed as ($\hat{\mathbf{x}}_k$ is a random variable)

$$\begin{aligned} E(\epsilon_{x,k}) &= E(\mathbf{x} - \hat{\mathbf{x}}_k) \\ &= E[\mathbf{x} - \hat{\mathbf{x}}_{k-1} - K_k(\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k-1})] \\ &= E[\epsilon_{x,k-1} - K_k(H_k \mathbf{x} + \mathbf{v}_k - H_k \hat{\mathbf{x}}_{k-1})] \\ &= E[\epsilon_{x,k-1} - K_k H_k(\mathbf{x} - \hat{\mathbf{x}}_{k-1}) - K_k \mathbf{v}_k] \\ &= (I - K_k H_k) E(\epsilon_{x,k-1}) - K_k E(\mathbf{v}_k) \end{aligned}$$

where $\epsilon_{x,k} = \mathbf{x} - \hat{\mathbf{x}}_k$.

If $E(v_k) = 0$ and $E(\epsilon_{x,k-1}) = 0$, then we have$E(\epsilon_{x,k}) = 0$. Thus if the measurement noise v_k is zero-mean for all k, and the initial estimate of $\mathbf{x}$ is set equal to the expected value of $\mathbf{x}$, i.e., $\hat{\mathbf{x}}_0 = E(\mathbf{x})$, then the expected value of $\hat{\mathbf{x}}_k$ is equal to $E(\mathbf{x})$ for all k. It is noted that this property holds regardless of the value of the gain matrix K_k.

In order to find the optimal RLS estimate, we need to find the optimal K_k. The optimally criterion (the sum of the variances of the estimation errors at time k) can be written as,

$$\begin{aligned} J_k &= E[(x_1 - \hat{x}_{1,k})^2] + \ldots + E[(x_n - \hat{x}_{n,k})^2] \\ &= E(\epsilon_{x1,k}^2 + \ldots + \epsilon_{xn,k}^2) \\ &= E(\epsilon_{x,k}^T \epsilon_{x,k}) \\ &= E[\mathrm{Tr}(\epsilon_{x,k}\epsilon_{x,k}^T)] \\ &= \mathrm{Tr}P_k \end{aligned}$$

where $\epsilon_{x,k} = [\epsilon_{x1,k}, \ldots, \epsilon_{xn,k}]^T$ and P_k is the estimation error covariance, and can be recursively written as,

$$\begin{aligned} P_k &= E(\epsilon_{x,k}\epsilon_{x,k}^T) \\ &= E\left\{[(I - K_k H_k)\epsilon_{x,k-1} - K_k v_k][\cdots]^T\right\} \\ &= (I - K_k H_k)E(\epsilon_{x,k-1}\epsilon_{x,k-1}^T)(I - K_k H_k)^T - \\ &\quad K_k E(v_k \epsilon_{x,k-1}^T)(I - K_k H_k)^T - (I - K_k H_k)E(\epsilon_{x,k-1} v_k^T)K_k^T + \\ &\quad K_k E(v_k v_k^T)K_k^T \end{aligned}$$

As $\epsilon_{x,k-1}$ is independent of v_k, we have (suppose $R_k = E(v_k v_k^T)$)

$$E(v_k \epsilon_{x,k-1}^T) = E(v_k)E(\epsilon_{x,k-1}) = 0,$$

and

$$P_k = (I - K_k H_k)P_{k-1}(I - K_k H_k)^T + K_k R_k K_k^T \tag{4.6}$$

From the expression of (4.6), we can know that as the measurement noise increases (i.e., R_k increases), the uncertainty in our estimate also increases (i.e., P_k increases). Besides, as P_k is a covariance matrix, it should be positive definite, and it can be concluded from (4.6) that P_k is positive definite provided that P_{k-1} and R_k are positive definite.

The selection of K_k is to make the cost function (the trace of P_k) small then the estimation error will not only be zero-mean, but it will also be consistently close to zero. The necessary condition for a selected K_k to be optimal is,

$$\frac{\partial J_k}{\partial K_k} = 0$$

that is,

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_k H_k)P_{k-1}(-H_k^T) + 2K_k R_k = 0$$

Then we have,

$$\begin{aligned} K_k R_k &= (I - K_k H_k)P_{k-1}H_k^T \\ K_k(R_k + H_k P_{k-1} H_k^T) &= P_{k-1}H_k^T \\ K_k &= P_{k-1}H_k^T(H_k P_{k-1} H_k^T + R_k)^{-1} \end{aligned} \tag{4.7}$$

The algorithm for the recursive least squares estimation is,

Algorithm 4.1 *Recursive least squares algorithm.*

Initialization
1. $\hat{\mathbf{x}}_0 = E(\mathbf{x})$, $P_0 = E[(\mathbf{x} - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]$

Iteration (for k)
2. Obtain the measurement $\mathbf{y}_k$, assuming that $\mathbf{y}_k$ is given by the equation $\mathbf{y}_k = H_k\mathbf{x} + v_k$
3. Update the estimate of x and the estimation-error covariance P as follows
   $K_k = P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}$
   $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + K_k(y_k - H_k\hat{\mathbf{x}}_{k-1})$
   $P_k = (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T$

It is noted that there are several important assumptions in RLS.

- if no knowledge about $\mathbf{x}$ is available before measurements are taken, then $P_0 = \infty I$. If perfect knowledge about $\mathbf{x}$ is available before measurements are taken, then $P_0 = 0$.
- the measurement noise at each time step k is independent, i.e., $E(\mathbf{v}_i\mathbf{v}_k) = R_k\delta_{k-i}$. That is, the measurement noise is white.

There also alternate forms of the Algorithm 4.1, in which the covariance P_k and gain matrices K_k take different forms. Although these alternate forms are mathematically identical, they can be beneficial from a computational point of view

Introduce an intermediate variable $S_k = H_kP_{k-1}H_k^T + R_k$, then

$$K_k = P_{k-1}H_k^TS_k^{-1},$$

Substituting for K_k from the above into(4.6), we obtain

$$P_k = [I - P_{k-1}H_k^TS_k^{-1}H_k]P_{k-1}[\cdots]^T + P_{k-1}H_k^TS_k^{-1}R_kS_k^{-1}H_kP_{k-1}^T.$$

We expand the terms and get

$$\begin{aligned} P_k = \quad & P_{k-1} - P_{k-1}H_k^TS_k^{-1}H_kP_{k-1} - P_{k-1}H_k^TS_k^{-1}H_kP_{k-1} + \\ & P_{k-1}H_k^TS_k^{-1}H_kP_{k-1}H_k^TS_k^{-1}H_kP_{k-1} + P_{k-1}H_k^TS_k^{-1}R_kS_k^{-1}H_kP_{k-1} \end{aligned} \tag{4.8}$$

Combining the last two terms in (4.8) gives

$$\begin{aligned} P_k \quad &= P_{k-1} - 2P_{k-1}H_k^TS_k^{-1}H_kP_{k-1} + P_{k-1}H_k^TS_k^{-1}S_kS_k^{-1}H_kP_{k-1} \\ &= P_{k-1} - 2P_{k-1}H_k^TS_k^{-1}H_kP_{k-1} + P_kH_k^TS_k^{-1}H_kP_{k-1} \\ &= P_{k-1} - P_{k-1}H_k^TS_k^{-1}H_kP_{k-1} \end{aligned}$$

As $K_k = P_{k-1}H_k^TS_k^{-1}$, we obtain

$$\begin{aligned} P_k \quad &= P_{k-1} - K_kH_kP_{k-1} \\ &= (I - K_kH_k)P_{k-1} \end{aligned}$$

It is noted that numerical computing problems (i.e., scaling issues) may cause this expression for P_k to be not positive definite, even when P_{k-1} and R_k are positive definite.

Next we'll introduce another formula for the update of P_k. Before doing this, we review the matrix inversion lemma, which can be expressed as follows,

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1}$$

As

$$P_k = P_{k-1} - P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}H_kP_{k-1}$$

then

$$P_k^{-1} = [P_{k-1} - P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}H_kP_{k-1}]^{-1},$$

Applying the matrix inversion lemma and we have

$$\begin{aligned} P_k^{-1} = \; & P_{k-1}^{-1} + P_{k-1}^{-1}P_{k-1}H_k^T[(H_kP_{k-1}H_k^T + R_k) - \\ & H_kP_{k-1}P_{k-1}^{-1}(P_{k-1}H_k^T)]^{-1}H_kP_{k-1}P_{k-1}^{-1} \\ = \; & P_{k-1}^{-1} + H_k^TR_k^{-1}H_k \end{aligned} \tag{4.9}$$

Inverting both sides of the equation (4.9) gives

$$P_k = [P_{k-1}^{-1} + H_k^TR_k^{-1}H_k]^{-1} \tag{4.10}$$

This equation for P_k is more complicated in that it requires three matrix inversions, but it may be computationally advantageous in some situations.

After obtaining the 2 different forms for the update of P_k, next we'll derive the alternate equations for the estimator gain matrix K_k. As (4.7) shows,

$$K_k = P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}.$$

Premultiplying the right side by $P_kP_k^{-1}$ gives

$$K_k = P_kP_k^{-1}P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}$$

substituting for P_k^{-1} from (4.10) gives

$$K_k = P_k(P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}$$

multiply the factor $P_{k-1}H_k^T$ inside the first term in parentheses gives

$$K_k = P_k(H_k^T + H_k^TR_k^{-1}H_kP_{k-1}H_k^T)(H_kP_{k-1}H_k^T + R_k)^{-1}$$

Now bring H_k^T out to the left side of the parentheses to obtain

$$K_k = P_kH_k^T(I + R_k^{-1}H_kP_{k-1}H_k^T)(H_kP_{k-1}H_k^T + R_k)^{-1}$$

Premultiply the first parenthetical expression by R_k^{-1}, and multiply on the inside of the parenthetical expression by R_k, and we obtain

$$\begin{aligned} K_k \; & = P_kH_k^TR_k^{-1}(R_k + H_kP_{k-1}H_k^T)(H_kP_{k-1}H_k^T + R_k)^{-1} \\ & = P_kH_k^TR_k^{-1} \end{aligned} \tag{4.11}$$

Assume the measurement equations are given by

$$\begin{aligned} \mathbf{y}_k &= H_k\mathbf{x} + v_k \\ E(v_k) &= 0 \\ E(v_k v_i^T) &= R_k\delta_{k-i}, \end{aligned}$$

now we give the framework of the general recursive least squares estimation.

Algorithm 4.2 *General recursive least squares algorithm.*

Initialization

1. $\hat{\mathbf{x}}_0 = E(\mathbf{x})$, $P_0 = E[(\mathbf{x} - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]$

Iteration (for k)

2. Obtain the measurement $\mathbf{y}_k$, assuming that $\mathbf{y}_k$ is given by the equation $\mathbf{y}_k = H_k\mathbf{x} + v_k$
3. Update the estimate of x and the estimation-error covariance P as follows

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ &= P_kH_k^TR_k^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + K_k(y_k - H_k\hat{\mathbf{x}}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\ &= (P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\ &= (I - K_kH_k)P_{k-1} \end{aligned}$$

It is noted that if we use the equation (4.11) for K_k, then in order to update P_k, the formula (4.10) should be used.

4.2.2 RLS II

According to least squares estimation, we have

$$\hat{x} = (H^TH)^{-1}H^Ty$$

Assume we have measurements till time k,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} H_1 \\ \vdots \\ H_k \end{bmatrix} \cdot x + \begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix}$$

then the estimate is given by

$$\hat{x}(k) = [H(k)^TH(k)]^{-1}H(k)^Ty(k)$$

in which $H(k) = [H_1^T, \ldots, H_k^T]^T$, $y(k) = [y_1, \ldots, y_k]^T$.

When the time $k+1$ comes, we have

$$\hat{x}(k+1) = [H(k+1)^TH(k+1)]^{-1}H(k+1)^T\mathbf{y}(k+1)$$

in which

$$H(k+1) = \begin{bmatrix} H(k) \\ H_{k+1} \end{bmatrix}.$$

The problem is how to express the estimate $\hat{x}_{k+1}$ as an incremental expression.

In order to derive this incremental expression, we first investigate the relationship between $\hat{x}(k+1)$ and $\hat{x}(k)$. Assume

$$C(k) = (H(k)^T H(k)), \tag{4.12}$$

then

$$\begin{aligned}
\hat{x}(k+1) - \hat{x}(k) &= C(k+1)^{-1} \cdot \left[\sum_{i=1}^{k} H_i^T y_i + H_{k+1}^T y_{k+1} \right] - C(k)^{-1} \sum_{i=1}^{k} H_i^T y_i \\
&= [C(k+1)^{-1} - C(k)^{-1}] \cdot \sum_{i=1}^{k} H_i^T y_i + C(k+1)^{-1} H_{k+1}^T y_{k+1} \\
&= C(k+1)^{-1}[C(k) - C(k+1)]C(k)^{-1} \cdot \sum_{i=1}^{k} H_i^T y_i + C(k+1)^{-1} H_{k+1}^T y_{k+1} \\
&= C(k+1)^{-1} \left(-H_{k+1}^T H_{k+1} \right) \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T y_{k+1} \\
&= C(k+1)^{-1} H_{k+1}^T \left(y_{k+1} - H_{k+1} \hat{x}(k) \right)
\end{aligned}$$

Or equivalently,

$$\hat{x}(k+1) = \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T \left[y_{k+1} - H_{k+1} \hat{x}(k) \right]$$

As

$$C(k+1) = C(k) + H_{k+1}^T H_{k+1},$$

according to the matrix inversion lemma, we have

$$C(k+1)^{-1} = C(k)^{-1} - C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1} H_{k+1} C(k)^{-1}$$

Then

$$C(k+1)^{-1} H_{k+1}^T = C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1}$$

Introduce the matrix gain $\tilde{K}(k+1)$ as $\tilde{K}(k+1) = C(k+1)^{-1} H_{k+1}^T$, then we have

$$\begin{aligned}
\tilde{K}(k+1) &= C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1} \\
C(k+1)^{-1} &= \left[I - \tilde{K}(k+1) H_{k+1} \right] C(k)^{-1}
\end{aligned}$$

Thus we have another formulation of the RLS.

Algorithm 4.3 *Recursive least squares algorithm II.*

Initialization

1. $\hat{\mathbf{x}}_0 = E(\mathbf{x})$,

Iteration (for k)

2. Obtain the measurement $\mathbf{y}_k$, assuming that $\mathbf{y}_k$ is given by the equation $\mathbf{y}_k = H_k \mathbf{x} + v_k$
3. Update the estimate of x and the estimation-error covariance P as follows

$$\begin{aligned}
\tilde{K}(k+1) &= C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1} \\
&= C(k+1)^{-1} H_{k+1}^T \\
\hat{x}(k+1) &= \hat{x}(k) + \tilde{K}(k+1) \left(y_{k+1} - H_{k+1} \hat{x}(k) \right) \\
C(k+1)^{-1} &= \left[I - \tilde{K}(k+1) H_{k+1} \right] C(k)^{-1}
\end{aligned}$$

It can be seen from Algorithm 4.2 and 4.3 that by replacing P_k with $C(k)^{-1}$, and letting $R_k = I$, the 2 ways of developing the RLS are very similar. It is apparent that if the statistic properties of the noise is known, and we have the covariance of the estimate, the estimate of the state would be more precise.

If we have precise initialization, i.e.,

$$\hat{x}(1) = (H_1^T H_1)^{-1} H_1 y(1).$$

And let

$$P_1 = (H_1^T H_1)^{-1},$$

then assume the statistical properties of the noise is $R_k = 1$, we can get the same result from the 2 kinds of RLS algorithms!

The interpretation of the RLS developed can also be implemented by comparing with the stochastic gradient method (k is stochastic). According to the stochastic gradient descent algorithm. The objective function we want to minimize is

$$E = (y - Hx)^T (y - Hx),$$

the Jacobian matrix (denominator layout) is

$$\frac{\partial E}{\partial x} = -2H^T y + 2H^T H x,$$

then the update of the state should be

$$\hat{x}(k+1) = \hat{x}(k) + \rho H_{k+1}^T (y_{k+1} - H_{k+1}\hat{x}(k)),$$

in which ρ is the step size.

In RLS I algorithm, we have

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + K_{k+1}(y_{k+1} - H_{k+1}\hat{\mathbf{x}}_k).$$

As $K_{k+1} = P_{k+1} H_{k+1}^T R_{k+1}^{-1}$, the RLS I algorithm can be seen as one kind of gradient descent algorithm, in which the descent direction now becomes,

$$P_{k+1} H_{k+1}^T R_{k+1}^{-1} (y_{k+1} - H_{k+1}\hat{\mathbf{x}}_k).$$

In RLS II algorithm, we have

$$\hat{x}(k+1) = \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T \left(y_{k+1} - H_{k+1}\hat{x}(k)\right)$$

and we can see that the descent direction now is

$$C(k+1)^{-1} H_{k+1}^T \left(y_{k+1} - H_{k+1}\hat{x}(k)\right).$$

In fact, the RLS II algorithm is basically the stochastic version of the Gauss-Newton algorithm. Therefore, in RLS I and RLS II algorithm, the step size is automatically determined.

Example 4.2 Consider the problem of trying to estimate the resistance x of an unmarked resistor on the basis of noisy measurement from a multimeter. However, we do not want to wait until we have all the measurements in order to have an estimate, instead, we want to recursively modify our estimate of x each time we obtain a new measurement. At sample time k our measurement is

$$\begin{aligned} y_k &= H_k x + v_k \\ H_k &= 1 \\ R_k &= E(v_k^2) \end{aligned}$$

Assume the measurement covariance is constant, i.e., $R_k = R$ and suppose the initial estimate is

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x_0 - \hat{x}_0)(x - \hat{x}_0)^T] \end{aligned}$$

If we have absolutely no idea about the resistance value, then $P(0) = \infty$. If we are 100% certain about the resistance value before taking any measurements, then $P(0) = 0$ (but then, of course, there would not be any need to take measurements)

After the first measurement ($k = 1$), we have

$$\begin{aligned} K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\ K_1 &= P_0 (P_0 + R)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1}) \\ \hat{x}_1 &= \hat{x}_0 + \tfrac{P_0}{P_0+R}(y_1 - \hat{x}_0) \\ P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \\ P_1 &= \tfrac{P_0 R}{P_0+R} \end{aligned}$$

Repeating these calculations to find these quantities after the second measurement ($k = 2$) gives

$$\begin{aligned} K_2 &= \tfrac{P_1}{P_1+R} = \tfrac{P_0}{2P_0+R} \\ P_2 &= \tfrac{P_1 R}{P_1+R} = \tfrac{P_0 R}{2P_0+R} \\ \hat{x}_2 &= \hat{x}_1 + \tfrac{P_1}{P_1+R}(y_2 - \hat{x}_1) \\ &= \tfrac{P_0+R}{2P_0+R}\hat{x}_1 + \tfrac{P_0}{2P_0+R} y_2 \end{aligned}$$

By induction we can find general expressions for P_{k-1}, K_k, and $\hat{x}_k$ as follows:

$$\begin{aligned} P_{k-1} &= \tfrac{P_0 R}{(k-1)P_0+R} \\ K_k &= \tfrac{P_0}{kP_0+R} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - \hat{x}_{k-1}) \\ &= (1 - K_k)\hat{x}_{k-1} + K_k y_k \\ &= \tfrac{(k-1)P_0+R}{kP_0+R}\hat{x}_{k-1} + \tfrac{P_0}{kP_0+R} y_k \end{aligned}$$

If x is known perfectly a priori (i.e., before any measurements are obtained) then $P_0 = 0$, and then $K_k = 0$ and $\hat{x}_k = \hat{x}_0$, i.e., the optimal estimate of x is independent of any measurements that are obtained.

On the other hand, if x is completely unknown a priori, then $P_0 \to \infty$, and then

$$\begin{aligned} \hat{x}_k &= \tfrac{(k-1)P_0}{kP_0}\hat{x}_{k-1} + \tfrac{P_0}{kP_0} y_k \\ &= \tfrac{k-1}{k}\hat{x}_{k-1} + \tfrac{1}{k} y_k \\ &= \tfrac{1}{k}[(k-1)\hat{x}_{k-1} + y_k] \end{aligned}$$

In other words, the optimal estimate of x is equal to the cumulative moving average of the measurements y_k. As we have introduced before, the cumulative moving average of the measurements y_k is,

$$\begin{aligned}\bar{y}_k &= \tfrac{1}{k}\sum_{j=1}^{k} y_j \\ &= \tfrac{1}{k}\left(\sum_{j=1}^{k-1} y_j + y_k\right) \\ &= \tfrac{1}{k}\left[(k-1)\left(\tfrac{1}{k-1}\sum_{j=1}^{k-1} y_j\right) + y_k\right] \\ &= \tfrac{1}{k}[(k-1)\bar{y}_{k-1} + y_k]\end{aligned}$$

We can also use RLS II to solve this problem. The result is as follows.

$$\hat{x}(k+1) = \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T (y_{k+1} - H_{k+1}\hat{x}(k))$$

$$H_1 = 1, \hat{x}(1) = y_1, C(1) = 1, C(2) = 2, \hat{x}(2) = \frac{1}{2}[\hat{x}(1) + y_2]$$

$$\hat{x}(k) = \frac{1}{k}\left[\sum_{j=1}^{k-1} y_j + y_k\right]$$

which is the same as RLS I if $P_0 = \infty$.

About the moving average.

In financial applications a simple moving average (SMA) is the unweighted mean of the previous n data. However, in science and engineering, the mean is normally taken from an equal number of data on either side of a central value. An example of a simple equally weighted running mean for a n-day sample of closing price is the mean of the previous n days' closing prices. If those prices are $p_M, p_{M-1}, \ldots, p_{M-(n-1)}$, then the formula is

$$\bar{p}_{\mathrm{SM}} = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n} = \frac{1}{n}\sum_{i=0}^{n-1} p_{M-i}$$

When calculating successive values, a new value comes into the sum, and the oldest value drops out, meaning that a full summation each time is unnecessary for this simple case:

$$\bar{p}_{\mathrm{SM}} = \bar{p}_{\mathrm{SM,prev}} + \frac{1}{n}(p_M - p_{M-n}).$$

In a cumulative moving average (CMA), the data arrive in an ordered datum stream, and the user would like to get the average of all of the data up until the current datum point. For example, an investor may want the average price of all of the stock transactions for a particular stock up until the current time. As each new transaction occurs, the average price at the time of the transaction can be calculated for all of the transactions up to that point using the cumulative average, typically an equally weighted average of the sequence of n values $x_1 \ldots, x_n$ up to the current time:

$$\mathrm{CMA}_n = \frac{x_1 + \cdots + x_n}{n}$$

The brute-force method to calculate this would be to store all of the data and calculate the sum and divide by the number of datum points every time a new datum point arrived. However, it is possible to simply update cumulative average as a new value, x_{n+1} becomes available, using the formula

$$\mathrm{CMA}_{n+1} = \frac{x_{n+1} + n \cdot \mathrm{CMA}_n}{n+1}.$$

Thus the current cumulative average for a new datum point is equal to the previous cumulative average, times n, plus the latest datum point, all divided by the number of points received so far, $n+1$. When all of the datum points arrive ($n = N$), then the cumulative average will equal the final average.

Example 4.3 Suppose we have a scalar parameter x and a perfect measurement of it, i.e., $H_1 = 1$ and $R_1 = 0$. Assume that our initial estimation covariance $P_0 = 6$, and that our computer provides precision of three digits to the right of the decimal point for each quantity that it computes.

Using RLS I, the estimator gain K_1 is:

$$\begin{aligned} K_1 = &\quad P_0(P_0 + R_1)^{-1} \\ = &\quad 6 * 1/6 \\ = &\quad 6 * 0.167 \\ = &\quad 1.002 \end{aligned}$$

Using the first form in RLS I we obtain

$$\begin{aligned} P_1 &= (1 - K_1)P_0(1 - K_1) + K_1 R_1 K_1 \\ &= (1 - K_1)^2 P_0 + K_1^2 R_1 \\ &= 0 \end{aligned}$$

Then if we use the third term in RLS I, the covariance update is

$$\begin{aligned} P_1 &= (1 - K_1)P_0 \\ &= (-0.002) * 6 \\ &= -0.012 \end{aligned}$$

We can see that the covariance after the first measurement is negative, which is physically impossible. However, for the first form, the covariance matrix will never be negative, regardless of any numerical errors in P_0, R_1, and K_1.

4.3 Curve fitting

We can also apply the recursive least squares theory to the curve fitting problem.

Assume the measured data comes one sample at a time $(y_1, y_2, \cdots)$, the problem is to find the best fit of a curve to the data. The curve that we want to fit to the data could be constrained to be linear, or quadratic, or sinusoid, or some other shape.

Example 4.4 Fit a straight line to a set of data points.

The linear data fitting problem can be written as

$$\begin{aligned} y_k &= x_1 + x_2 t_k + v_k \\ E(v_k^2) &= R_k \end{aligned}$$

in which $x = [x_1, x_2]^T$, t_k is the independent variable, y_k is the noisy data. What we want to estimate are the constants x_1 and x_2. Denote the measurement matrix as $H_k = [1 \;\; t_k]$, then the linear data fitting equation is

$$y_k = H_k x + v_k$$

According to RLS I, we can initialize our recursive estimator as,

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T] \end{aligned}$$

The iteration for $k = 1, 2, \ldots$, can be described as,

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \end{aligned}$$

If we use RLS II, the recursive estimator is initialized as,

$$\begin{aligned} \hat{x}_1 &= (H_1^TH_1)^{-1}H_1y_1 \\ C(1)^{-1} &= (H_1^TH_1)^{-1} \end{aligned}$$

For $k = 2, \ldots,$, we have

$$\begin{aligned} \tilde{K}_k &= C_{k-1}^{-1}H_k^T(I + H_kC_k^{-1}H_k^T)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + \tilde{K}_k(y_k - H_k\hat{x}_{k-1}) \\ C(k)^{-1} &= (I - \tilde{K}_kH_k)C(k-1)^{-1} \end{aligned}$$

Example 4.5 Fit a neural network to a set of data points.

Suppose we want to fit a neural network to a set of data points,

$$\begin{aligned} y_k &= x_0 + \sum_{i=1}^{M} x_iB_i(t_k) + v_k \\ E(v_k^2) &= R_k \end{aligned}$$

in which $x = [x_0, x_1, \ldots, x_M]^T$, t_k is the independent variable, y_k is the noisy data, and $B_i(t_k)$ is called the basis (kernel) function. What we want to estimate are the constants $x_0, x_1, \ldots, x_M$. The measurement matrix here can be described as $H_k = [1, B_1(t_k), \ldots, B_M(t_k)]$, and we also have the linear data fitting equation: $y_k = H_kx + v_k$.

The structure of a neural network can be depicted in Figure 4.1.

Popular basis functions are

- linear function
- step function
- polynomial function
$$B_i(t_k) = t_k^{\alpha_i}$$
- RBF: (Gaussian)radial basis function:
$$B_i(t_k) = \exp\left\{-\frac{t_k - \alpha_i}{2\sigma_i^2}\right\}$$
- sigmoid function (S-shape):
$$B_i(t_k) = \frac{1}{1 + e^{-\beta_i t_k}}$$

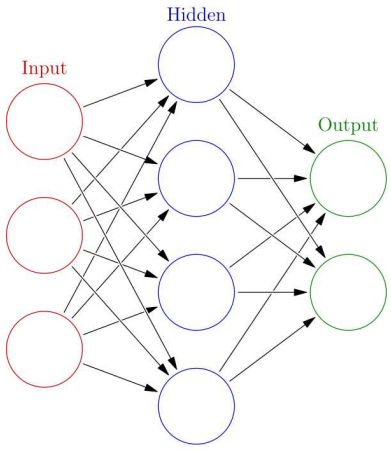


Figure 4.1 Typical single-layer neural network

Using RLS I. The recursive estimator is initialized as,

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T] \end{aligned}$$

For $k = 1, 2, \ldots,$, we have

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \end{aligned}$$

Using RLS II. The recursive estimator is initialized as,

$$\begin{aligned} \hat{x}_1 &= (H_1^TH_1)^{-1}H_1y_1 \\ C(1)^{-1} &= (H_1^TH_1)^{-1} \end{aligned}$$

For $k = 2, \ldots,$, we have,

$$\begin{aligned} \tilde{K}_k &= C_{k-1}^{-1}H_k^T(I + H_kC_k^{-1}H_k^T)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + \tilde{K}_k(y_k - H_k\hat{x}_{k-1}) \\ C(k)^{-1} &= (I - \tilde{K}_kH_k)C(k-1)^{-1} \end{aligned}$$

It is noted that by choosing different basis functions, we get different estimations, which differs in structure, parameter and performance. If we also want to determine the suitable parameters $\alpha_i, \sigma^2, \beta_i$, we have to use back-propagation method to train the neural network.

Chapter 5

Kalman Filter

5.1 Propagation of states and covariances

In this section, we will give the mathematical description of a dynamic system. Based on this, the equations that govern the propagation of the state mean and covariance are derived, which are fundamental to the state estimation algorithm (the Kalman filter).

5.1.1 Discrete-time systems

Suppose we have the following linear discrete-time system:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}, \tag{5.1}$$

in which u_k is a known input and w_k is the process noise drawn from a zero-mean multivariate normal distribution with covariance Q_k. Besides, the initial state, and the noise vector at each step $\{x_0, w_1, \ldots, w_k, v_1, \ldots, v_k\}$ are all assumed to be mutually independent.

For the system (5.1), the mean of x_k can be obtained by taking the expected value of both sides of the equation (5.1), i.e.,

$$\bar{x}_k = E(x_k) = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$$

In order to obtain the covariance, we first study $(x_k - \bar{x}_k)(x_k - \bar{x}_k)^T$,

$$\begin{aligned}(x_k - \bar{x}_k)(\cdots)^T &= (F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} - \bar{x}_k)(\cdots)^T \\ &= [F_{k-1}(x_{k-1} - \bar{x}_{k-1}) + w_{k-1}][\cdots]^T \\ &= F_{k-1}(x_{k-1} - \bar{x}_{k-1})(x_{k-1} - \bar{x}_{k-1})^T F_{k-1}^T + w_{k-1}w_{k-1}^T + \\ &\quad F_{k-1}(x_{k-1} - \bar{x}_{k-1})w_{k-1}^T + w_{k-1}(x_{k-1} - \bar{x}_{k-1})^T F_{k-1}^T\end{aligned}$$

According to the assumption, x_0 is uncorrelated with $w_k, k = 0, 1, 2, \ldots$, it is clear that the term $(x_{k-1} - \bar{x}_{k-1})$ is uncorrelated with w_{k-1}. Thus the covariance matrix can be expressed as,

$$P_k = E[(x_k - \bar{x}_k)(\cdots)^T] = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}.$$

This is called a discrete-time Lyapunov equation, or a Stein equation, which is fundamental in the derivation of the Kalman filter.

For the discrete-time Lyapunov equation, we can derive the steady-state solution.

Consider the equation $P = FPF^T + Q$ where F and Q are real matrices. Denote by $\lambda_i(F)$ the eigenvalues of the F matrix.

- A unique solution P exists iff $\lambda_i(F) \cdot \lambda_j(F) \neq 1$ for all i, j. The unique solution is symmetric.
- If F is stable then the discrete-time Lyapunov equation has a solution P that is unique and symmetric:

$$P = \sum_{i=0}^{\infty} F^i Q (F^T)^i$$

We can further write the solution of the linear system with respect to the initial state and the input and process noise at each sample time.

$$x_k = F_{k,0} x_0 + \sum_{i=0}^{k-1} (F_{k,i+1} w_i + F_{k,i+1} G_i u_i) \tag{5.2}$$

in which the state transition matrix of the system is,

$$F_{k,i} = \begin{cases} F_{k-1} F_{k-2} \cdots F_i & k > i \\ I & k = i \\ 0 & k < i \end{cases}$$

It can be seen from (5.2) that the state x_k is a linear combination of $x_0, \{w_i\}$ and $\{u_i\}$. Then if the input sequence $\{u_i\}$ is known, x_0 and w_i are unknown but are Gaussian random variables, then x_k is itself a Gaussian random variable, i.e.,

$$x_k \sim \mathcal{N}(\bar{x}_k, P_k).$$

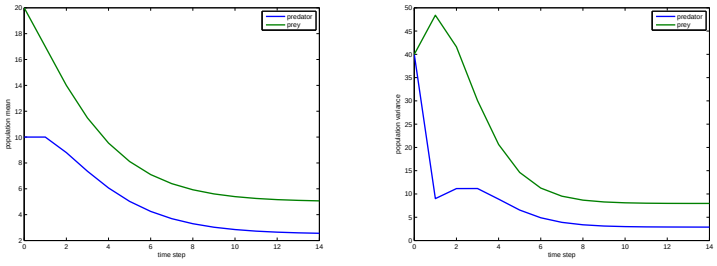
Example 5.1 A linear system describing the population of a predator $x(1)$ and that of its prey $x(2)$ can be written as

$$\begin{array}{rcl} x_{k+1}(1) & = & x_k(1) - 0.8x_k(1) + 0.4x_k(2) + w_k(1) \\ x_{k+1}(2) & = & x_k(2) - 0.4x_k(1) + u_k + w_k(2) \end{array} \tag{5.3}$$

The difference equation can be explained according to the real life experience. In general, the predator population causes itself to decrease because of overcrowding. As the prey serves as food to the predator, the prey population causes the predator population to increase, the prey population decreases due to the predator population. Besides, the prey population increases due to an external food supply u_k, and the populations are also subject to random disturbances due to environmental factors.

The state-space form of the linear system (5.3) is,

$$\begin{array}{rl} x_{k+1} & = \begin{bmatrix} 0.2 & 0.4 \\ -0.4 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k + w_k \\ w_k & \sim N(0, Q) \qquad Q = \mathrm{diag}(1, 2) \end{array}$$



Assume $\bar{x}_0 = [10, 20]^T$, $P_0 = \text{diag}(40, 40)$ and $u_k = 1$, we can obtain the two means and the two diagonal elements of the covariance matrix.

The steady-state values is as follows,

$$\begin{aligned} \bar{x} &= (I - F)^{-1} G u \\ &= [2.5, 5]^T \\ P &\sim \begin{bmatrix} 2.88 & 3.08 \\ 3.08 & 7.96 \end{bmatrix} \end{aligned}$$

Consider another case when the process noise is multiplied by some matrix. Assume the linear discrete-time system can be written as,

$$x_k = F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + L_{k-1} \tilde{w}_{k-1}, \ \tilde{w}_k \sim \mathcal{N}(0, \tilde{Q}_k) \tag{5.4}$$

As the rightmost term of the above equation has a covariance given by

$$\begin{aligned} E[(L_{k-1}\tilde{w}_{k-1})(L_{k-1}\tilde{w}_{k-1})^T] &= L_{k-1} E(\tilde{w}_{k-1}\tilde{w}_{k-1}^T) L_{k-1}^T \\ &= L_{k-1} \tilde{Q}_{k-1} L_{k-1}^T \end{aligned}$$

Therefore, the equation (5.4) is equivalent to the equation

$$x_k = F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1}, w_k \sim \mathcal{N}(0, L_k \tilde{Q}_k L_k^T)$$

Similar to the case when the process noise is multiplied by some matrix, the measurement noise can also be multiplied by some matrix. Correspondingly, the same type of transformation can be made with noisy measurement equations, i.e.,

$$y_k = H_k x_k + L_k \tilde{v}_k, \tilde{v}_k \sim \mathcal{N}(0, \tilde{R}_k)$$

is equivalent to the measurement equation

$$y_k = H_k x_k + v_k, v_k \sim \mathcal{N}(0, L_k \tilde{R}_k L_k^T)$$

5.1.2 Sampled-data systems

In this section, we consider the propagation of the state mean and covariance for sampled-data systems. A sampled-data system is a system whose dynamics are

described by a continuous-time differential equation, but the input only changes at discrete time instants. For this kind of system, we are interested in obtaining the mean and covariance of the state only at discrete time instants. Assume the continuous-time dynamics are described as

$$\dot{x} = Ax + Bu + w$$

The solution of $x(t)$ at some arbitrary time, say t_k, is given as

$$x(t_k) = e^{A(t_k - t_{k-1})}x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}[Bu(\tau) + w(\tau)]d\tau$$

The continuous-time system can be transformed the discrete-time version. Assume $u(t) = u_{k-1}$ for $t \in [t_{k-1}, t_k]$, $\Delta t = t_k - t_{k-1}$, $x_k = x(t_k)$ and $u_k = u(t_k)$, we have

$$x_k = e^{A\Delta t}x_{k-1} + \left[\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Bd\tau\right] u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau$$

Define F_{k-1} and G_{k-1} as

$$\begin{aligned} F_{k-1} &= e^{A\Delta t} \\ G_{k-1} &= \textstyle\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Bd\tau \end{aligned}$$

then we have

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau$$

Suppose $w(t)$ is zero-mean, the propagation of the state mean can be expressed as,

$$\begin{aligned} \bar{x}_k &= E(x_k) \\ &= F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1} \end{aligned}$$

Additionally assume $w(t)$ is white noise, i.e.,

$$w(t) \sim \mathcal{N}(0, Q(t)), E[w(t)w^T(\tau)] = Q_c(t)\delta(t-\tau),$$

then we have

$$\begin{aligned} P_k &= E[(x_k - \bar{x}_k)(x_k - \bar{x}_k)^T] \\ &= E\left[\left(F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + \textstyle\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau - \bar{x}_k\right)(\cdots)^T\right] \\ &= F_{k-1}P_{k-1}F_{k-1}^T + E\left[\left(\textstyle\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau\right)(\cdots)^T\right] \\ &= F_{k-1}P_{k-1}F_{k-1}^T + \textstyle\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}E[w(\tau)w^T(\alpha)]e^{A^T(t_k-\alpha)}d\tau d\alpha \\ &= F_{k-1}P_{k-1}F_{k-1}^T + \textstyle\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Q_c(\tau)e^{A^T(t_k-\tau)}d\tau \end{aligned}$$

Define

$$Q_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Q_c(\tau)e^{A^T(t_k-\tau)}d\tau,$$

then we have

$$P_k = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1}.$$

For small values of $(t_k - t_{k-1})$ we have

$$\begin{aligned} e^{A(t_k-\tau)} &\approx I \text{ for } \tau \in [t_{k-1}, t_k] \\ Q_{k-1} &\approx Q_c(t_k)\Delta t \end{aligned}$$

Example 5.2 Suppose we have a first-order, continuous-time dynamic system (e.g. the behaviour of the current through a series RL circuit that is driven by a random voltage $w(t)$, where $f = -R/L$) given by the equation

$$\begin{aligned} \dot{x} &= fx + w \\ E[w(t)w(t+\tau)] &= q_c\delta(\tau) \end{aligned}$$

where $w(t)$ is zero-mean noise.

Suppose we are interested in obtaining the mean and covariance of the state $x(t)$ every Δt time units, i.e., $t_k - t_{k-1} = \Delta t$, for this simple scalar example, we can explicitly calculate Q_{k-1} as

$$Q_{k-1} = \frac{q_c}{2f}[\exp(2f\Delta t) - 1]$$

Expanding Q_{k-1} in a Taylor series around $\Delta t = 0$ results,

$$\begin{aligned} Q_{k-1} &= \tfrac{q_c}{2f}[\exp(2f\Delta t) - 1] \\ &\approx q_c 2f \left[\left(1 + 2f\Delta t + \tfrac{(2f\Delta t)^2}{2!}\right) - 1\right] \\ &\approx \tfrac{q_c}{2f}[1 + 2f\Delta t - 1] \\ &= q_c\Delta t \end{aligned}$$

The sampled mean of the state is computed as (noting that the control input is zero)

$$\begin{aligned} \bar{x}_k &= F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1} \\ &= \exp[f(t_k - t_{k-1})]\bar{x}_{k-1} + 0 \\ &= \exp(f\Delta t)\bar{x}_{k-1} \\ &= \exp(kf\Delta t)\bar{x_0} \end{aligned}$$

We can discuss the dynamic of the system according to the value of f.

- If $f > 0$ (i.e., the system is unstable) then the mean $\bar{x}_k$ will increase without bound (unless $\bar{x}_0 = 0$)
- If $f < 0$ (i.e., the system is stable) then the mean $\bar{x}_k$ will decay to zero regardless of the value of $\bar{x}_0$

The sampled covariance of the state is computed as

$$\begin{aligned} P_k &= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \\ &\approx (1 + 2f\Delta t)P_{k-1} + q_c\Delta t \\ P_k - P_{k-1} &= (2fP_{k-1} + q_c)\Delta t \end{aligned}$$

The propagation of the sampled covariance can also be discussed according to the value of f.

- assume $f < 0$, when $P_{k-1} = -q_c/2f$, P_k reaches steady state, i.e., $P_k - P_{k-1} = 0$
- if $f \geq 0$, then $P_k - P_{k-1}$ will always be greater than 0, which means that $\lim_{k\to\infty} P_k = \infty$

5.1.3 Continuous-time systems

Consider the continuous-time system

$$\dot{x} = Ax + Bu + w \tag{5.5}$$

where $u(t)$ is a known control input and $w(t)$ is zero-mean noise with a covariance of

$$E[w(t)w^T(\tau)] = Q_c\delta(t-\tau).$$

Taking the mean of (5.5) yields,

$$\dot{\bar{x}} = A\bar{x} + Bu$$

In order to describe the propagation of the state covariance, we can use the equation

$$P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

that describes the covariance of a sampled-data system and taking the limit as $\Delta t = t_k - t_{k-1} \to 0$. As

$$\begin{aligned} F &= e^{A\Delta t} \\ &= I + A\Delta t + \tfrac{(A\Delta t)^2}{2!} + \cdots \end{aligned}$$

For small values of Δt, this can be approximated as

$$F \approx I + A\Delta t$$

Thus we obtain

$$\begin{aligned} P_k &\approx (I + A\Delta t)P_{k-1}(I + A\Delta t)^T + Q_{k-1} \\ &= P_{k-1} + AP_{k-1}\Delta t + P_{k-1}A^T\Delta t + AP_{k-1}A^T(\Delta t)^2 + Q_{k-1} \end{aligned}$$

Subtracting P_{k-1} from both sides and dividing by Δt gives

$$\frac{P_k - P_{k-1}}{\Delta t} = AP_{k-1} + P_{k-1}A^T + AP_{k-1}A^T\Delta t + \frac{Q_{k-1}}{\Delta t} \tag{5.6}$$

Recall that for small Δt

$$Q_{k-1} \approx Q_c(t_k)\Delta t$$

Taking the limit of the equation (5.6) as Δt goes to zero gives the continuous-time Lyapunov equation

$$\dot{P} = AP + PA^T + Q_c$$

The conditions under which the continuous-time Lyapunov equation has a steady-state solution, i.e.,

$$AP + PA^T + Q_c = 0 \tag{5.7}$$

can be listed as follows.

- A unique solution P exists iff $\lambda_i(A) + \lambda_j(A) \neq 0, \forall i, j$. This unique solution is symmetric.

- If A is stable, then there is a unique and symmetric P

$$P = \int_0^\infty e^{A^T\tau} Q_c e^{A\tau} d\tau$$

- If A is stable and Q_c is positive (semi) definite, then the unique solution P is symmetric and positive (semi) definite
- .....

Example 5.3 Suppose we have the first-order, continuous-time dynamic system given by the equation

$$\begin{aligned} \dot{x} &= fx + w \\ E[w(t)w(t+\tau)] &= q_c\delta(\tau) \end{aligned}$$

where $w(t)$ is zero-mean noise.

The equation for the continuous-time propagation of the mean of state is

$$\dot{\bar{x}} = f\bar{x}$$

Solving this equation for $\bar{x}(t)$ gives

$$\bar{x}(t) = \exp(ft)\bar{x}(0)$$

The dynamic of the system can be discussed according to the value of f.

- The mean will increase without bound if $f > 0$ (i.e., if the system is unstable)
- The mean will asymptotically tend to zero if $f < 0$ (i.e., if the system is stable)

The equation for the continuous-time propagation of the covariance of the state is

$$\dot{P} = 2fP + q_c$$

Solving this equation for $P(t)$ gives

$$P(t) = \left(P(0) + \frac{q_c}{2f}\right)\exp(2ft) - \frac{q_c}{2f}$$

For different f, the propagation of the state covariance can be discussed.

- The covariance will increase without bound if $f > 0$ (i.e., if the system is unstable)
- The covariance will asymptotically tend to $-q_c/2f$ if $f < 0$ (i.e., if the system is stable)

The steady-state value of P can also be computed (provided that $f < 0$) as

$$\begin{aligned} P &= \int_0^\infty e^{2f\tau} q_c d\tau \\ &= \frac{q_c}{2f} e^{2f\tau}|_0^\infty \\ &= -\frac{q_c}{2f} \end{aligned}$$

Compare the results with those of the previous example.

5.2 Discrete-time Kalman filter

Kalman filter was originated in late 1950s, when James Follin, A. G. Carlton, James Hanson, and Richard Bucy developed the continuous-time Kalman filter in an unpublished work for the Johns Hopkins Applied Physics Lab. Then Rudolph Kalman independently developed the discrete-time Kalman filter in 1960. In April 1960 Kalman and Bucy became aware of each other's work and collaborated on the publication of the continuous-time Kalman filter, hence the filter is sometimes referred to as the Kalman-Bucy filter.

Rudolf Emil Kálmán was born in Budapest in 1930. He earned his bachelor's degree in 1953 and his master's degree in 1954, both from the Massachusetts Institute of Technology, completed his doctorate in 1957 at Columbia University in New York City. He has been worked in at the Research Institute for Advanced Studies in Baltimore, Maryland, Stanford University, University of Florida, Swiss Federal Institute of Technology in Zürich, Switzerland. In the morning of July 2, 2016, he died at his home in Gainesville, Florida.

Kalman filter is a mathematical technique widely used in the digital computers of control systems, navigation systems, avionics, and outerspace vehicles It can extract a signal from a long sequence of noisy or incomplete measurements, usually those done by electronic and gyroscopic systems. It was initially used in vast skepticism, **the Apollo program**, and furthermore, in the NASA Space Shuttle, in Navy submarines, and in unmanned aerospace vehicles and weapons, such as cruise missiles.

The Kalman filter operates by propagating the mean and covariance of the state through time.

1. start with a mathematical description of a dynamic system whose states we want to estimate

2. implement equations that describe how the mean of the state and the covariance of the state propagate with time

3. take the dynamic system that describes the propagation of the state mean and covariance, and implement the equations on a computer

4. every time that we get a measurement, we update the mean and covariance of the state

Following gives the details for deriving the discrete-time Kalman filter.

5.2.1 Deriving the discrete-time Kalman filter

Suppose we have a linear discrete-time system given as follows:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned}$$

The noise processes $\{w_k\}$ and $\{v_k\}$ are zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k, respectively:

$$\begin{array}{ll} w_k \sim \mathcal{N}(0, Q_k), & v_k \sim \mathcal{N}(0, R_k) \\ E[w_k w_j^T] = Q_k \delta_{k-j}, & E[v_k v_j^T] = R_k \delta_{k-j} \\ E[v_k w_j^T] = 0 & \end{array}$$

Our goal is to estimate the state x_k based on our knowledge of the system dynamics and the availability of the noisy measurement $\{y_k\}$. The amount of information available to us for our state estimate varies depending on the particular problem that we are trying to solve.

There are 4 kinds of estimate, i.e., *a posteriori* state estimate, *a priori* state estimate, smoothed estimate and predicted estimate.

1. *a posteriori* state estimate: if we have all of the measurements up to and including time k available for use in our estimate of x_k, then we can form an a posteriori estimate, $\hat{x}_k$. One way to formulate the is

$$\hat{x}_k = E[x_k | y_1, y_2, \cdots, y_k] = \textit{ a posteriori} \text{ estimate}$$

2. *a priori* state estimate: if we have all of the measurements up to but not including time k available for use in our estimate of x_k, then we can form an a priori estimate, $\check{x}_k$. One way to formulate the a priori state estimate is

$$\check{x}_k = E[x_k | y_1, y_2, \cdots, y_{k-1}] = \textit{ a priori} \text{ estimate}$$

3. smoothed estimate: if we have measurements after time k available for use in our estimate of x_k, then we can form a smoothed estimate. One way to formulate the smoothed state estimate is

$$\hat{x}_{k|k+N} = E[x_k | y_1, y_2, \cdots, y_k, \cdots, y_{k+N}] = \text{smoothed estimate}$$

4. predicted estimate: if we want to find the best prediction of x_k more than one time step ahead of the available measurements, then we can form a predicted estimate. One way to form the predicted state estimate is to compute the expected value of x_k is:

$$\hat{x}_{k|k-M} = E[x_k | y_1, y_2, \cdots, y_{k-M}] = \text{predicted estimate}$$

Before any measurements are available, the estimate is initialized as $\hat{x}_0$. In general, $\hat{x}_0 = E(x_0)$. Assume $\check{P}_k$ is the covariance of the estimation error of $\check{x}_k$, $\check{P}_k = E[(x_k - \check{x}_k)(x_k - \check{x}_k)^T]$, and suppose $\hat{P}_k$ is the covariance of the estimation error of $\hat{x}_k$, $\hat{P}_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$. Fig. 5.1 gives the illustration of the Kalman filter process.

After we process the measurement at time $(k-1)$, we have an estimate of x_{k-1} (denoted as $\hat{x}_{k-1}$) and the covariance of that estimate (denoted as $\hat{P}_{k-1}$). When time k arrives, before we process the measurement at time k we compute an estimate of x_k (denoted as $\check{x}_k$) and the covariance of that estimate (denoted

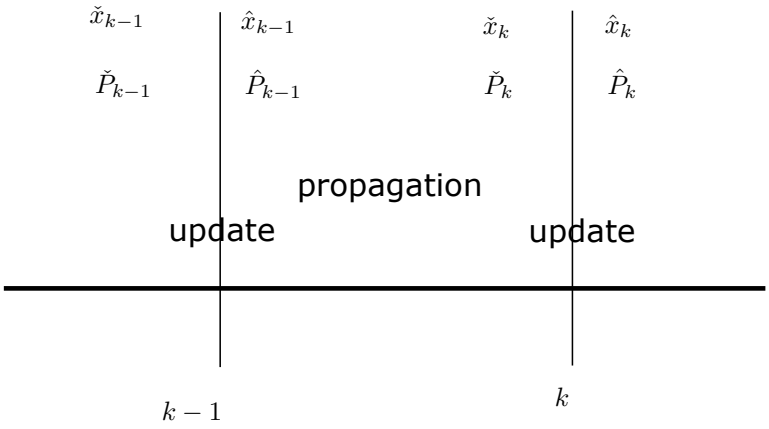


Figure 5.1 Propagation and update in the Kalman filter.

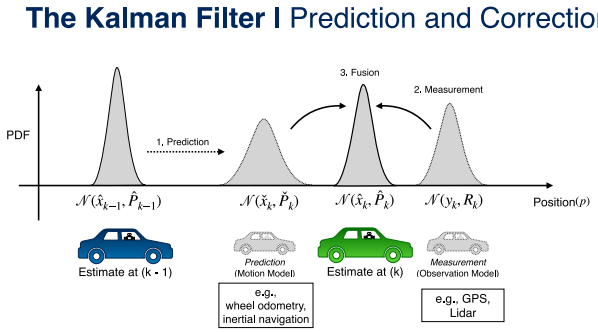


Figure 5.2 Application of the Kalman filter to estimate the position of a car.

as $\check{P}_k$). We process the measurement at time k to refine our estimate of x_k, the resulting estimate of x_k is denoted as $\hat{x}_k$, and its covariance is denoted as $\hat{P}_k$.

Fig. 5.2 gives an illustration of using the Kalman filter to estimate the position of a car.

Suppose we begin with $\hat{x}_0 = E(x_0)$. We want to set $\check{x}_1 = E(x_1)$, as $\bar{x}_k = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$, we therefore obtain

$$\check{x}_1 = F_0\hat{x}_0 + G_0u_0.$$

The time update equation for x (from $\hat{x}_{k-1}$ to $\check{x}_k$) can be described as,

$$\check{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \tag{5.8}$$

The reason we use the equation (5.8) is that we do not have any additional measurements available to help us update our state estimates after time $(k-1)$ and before time k, we should just update the state estimate based on our knowledge of the system dynamics.

Then we gives the formation to calculate $\check{P}_k$ based on $\hat{P}_{k-1}$. We begin with $\hat{P}_0 = E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T] = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$, in which $\hat{P}_0$ represents the uncertainty in our initial estimate of x_0, so if we know the initial state perfectly, $\hat{P}_0 = 0$, if we have absolutely no idea of the value of x_0, then $\hat{P}_0 = \infty I$. As the time-update equation for P is

$$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1},$$

we have

$$\check{P}_1 = F_0\hat{P}_0F_0^T + Q_0.$$

If the measurement is available, we can update the state x and covariance P based on the measurement. If we take the measurement y_k into account, we can get the estimate $\hat{x}_k$. Both the quantity $\check{x}_k$ and $\hat{x}_k$ are estimate of x_k, and the only difference is that the measurement information is available for the latter estimate.

Remember the update expression for recursive least squares has the following form,

$$\begin{aligned}
K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\
&= P_kH_k^TR_k^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\
P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\
&= (P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\
&= (I - K_kH_k)P_{k-1}
\end{aligned}$$

The update of x and P based on the measurement y_k can be described as follows.

$$\begin{aligned}
K_k &= \check{P}_kH_k^T(H_k\check{P}_kH_k^T + R_k)^{-1} \\
&= \hat{P}_kH_k^TR_k^{-1} \\
\hat{x}_k &= \check{x}_k + K_k(y_k - H_k\check{x}_k) \\
\hat{P}_k &= (I - K_kH_k)\check{P}_k(I - K_kH_k)^T + K_kR_kK_k^T \\
&= ((\check{P}_k)^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\
&= (I - K_kH_k)\check{P}_k
\end{aligned}$$

Table 5.1 Comparison of the update formation for RLS and Kalman filter.

RLS	KF
$\hat{x}_{k-1}$: estimate before y_k is processed	$\check{x}_k$: a priori estimate
P_{k-1}: covariance before y_k is processed	$\check{P}_k$: a priori covariance
$\hat{x}_k$: estimate after y_k is processed	$\hat{x}_k$: a posterior estimate
P_k: covariance after y_k is processed	$\hat{P}_k$: a posterior covariance

The relationships between estimates and covariances in recursive least-square and Kalman filtering is shown in Table 5.1.

Assume the dynamic system is given by the following equations:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ E(w_k w_j^T) &= Q_k \delta_{k-j} \\ E(v_k v_j^T) &= R_k \delta_{k-j} \\ E(w_k v_j^T) &= 0, \end{aligned}$$

the discrete-time Kalman filter can then be listed in Algorithm 5.5.

Algorithm 5.1 *Discrete-time Kalman filter.*

Initialization

1. $\hat{x}_0 = E(x_0)$, $\hat{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$

Iteration (for k)

2. State and covariance propagation (*a priori* estimate).

$\check{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}$

$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$

3. Obtain the measurement $\mathbf{y}_k$, update the estimate of x and the estimation-error covariance P as follows (*a posteriori* estimate)

$K_k = \check{P}_k H_k^T (H_k \check{P}_k H_k^T + R_k)^{-1}$ or $K_k = \hat{P}_k H_k^T R_k^{-1}$

$\hat{x}_k = \check{x}_k + K_k(y_k - H_k\check{x}_k)$

$\hat{P}_k = (I - K_k H_k)\check{P}_k(I - K_k H_k)^T + K_k R_k K_k^T$

or $\hat{P}_k = ((\check{P}_k)^{-1} + H_k^T R_k^{-1} H_k)^{-1}$ or $\hat{P}_k = (I - K_k H_k)\check{P}_k$

In Algorithm 5.5, the first expression for $\hat{P}_k$ is called the Joseph stabilized version of the covariance measurement update equation, which is more stable and robust than the third expression for $\hat{P}_k$. This is because that the first expression for $\hat{P}_k$ guarantees that $\hat{P}_k$ will always be symmetric semi-positive definite, as long as $\check{P}_k$ is symmetric semi-positive definite. The third expression for $\hat{P}_k$ is computationally simpler than the first expression, but its form does not guarantee symmetry or semi-positive definiteness for $\hat{P}_k$. For the second form of $\hat{P}_k$, we rarely implement it but it is useful in the derivation of the information filter.

It should be noted that if the second expression for K_k is used, then the second expression for $\hat{P}_k$ must be used. Besides, if x_k is a constant (random variable unchanged), then $F_k = I, Q_k = 0$ and $u_k = 0$, and **the Kalman filter reduces**

to the recursive least squares algorithm for the estimation of a constant vector. It should also be noted that as the the calculation of $\check{P}_k, K_k$, and $\hat{P}_k$ does not depend on the measurements y_k, but depends only on the system parameters F_k, H_k, Q_k and R_k, the computational effort of calculating K_k can be saved during real-time operation by precomputing it, hence the performance of the filter can be investigated and evaluated before the filter is actually run ($\hat{P}_k$ indicates the accuracy).

Example 5.4 For the moving vehicle shown in Fig. 5.3, the position and velocity constitutes the state of the vehicle, i.e.,

$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} = \frac{dp}{dt} \end{bmatrix}$$

and the input is the accelerator provided by the car, i.e.,

$$u = a = \frac{d^2}{dt^2}.$$

The process dynamic is,

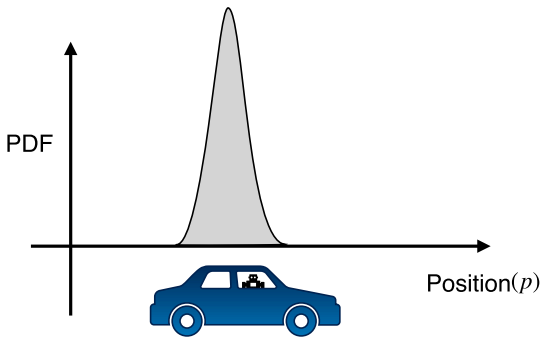


Figure 5.3 A moving vehicle.

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} u_{k-1} + w_{k-1}$$

The position observation can be described as

$$y_k = [1\ 0]\mathbf{x}_k + v_k$$

The process noise and measurement noise are assumed to be white noise, i.e.,

$$v_k \sim \mathcal{N}(0, 0.05), \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, 0.1\mathbf{1}_{2\times 2})$$

The initial state is

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

The sample instant is $\Delta t = 0.5\text{s}$, the initial input is $u_0 = -2\text{m/s}^2$, the measurements available are

$$y_1 = 2.2\text{m}$$

The prediction can be performed,

$$\check{\mathbf{x}}_k = F_{k-1}\hat{\mathbf{x}}_{k-1} + G_{k-1}u_{k-1}$$
$$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$$

thus we have

$$\begin{bmatrix} \check{p}_1 \\ \check{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \cdot (-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

$$\check{P}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

The corrections can be expressed as,

$$\begin{aligned} K_1 &= \check{P}_1 H_1^T (H_1 \check{P}_1 H_1^T + R_1)^{-1} \\ &= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left([1\ 0] \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.05 \right)^{-1} \\ &= \begin{bmatrix} 0.88 \\ 1.22 \end{bmatrix}, \end{aligned}$$

$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + K_1(y_1 - H_1\check{\mathbf{x}}_1)$$

i.e.,

$$\begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.88 \\ 1.22 \end{bmatrix} \left(2.2 - [1\ 0] \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} \right) = \begin{bmatrix} 2.24 \\ 3.63 \end{bmatrix}$$

Then the covariance $\hat{P}_1$ is

$$\hat{P}_1 = (\mathbf{1} - K_1 H_1)\check{P}_1 = \begin{bmatrix} 0.04 & 0.06 \\ 0.06 & 0.49 \end{bmatrix}$$

5.2.2 Kalman filter properties

Unbiasedness

We first check whether a Kalman filter is biased. We say an estimator or filter is unbiased if it produces an "average" error of zero at a particular time step k, over many trials.

The error dynamics is,

$$\check{e}_k = \check{x}_k - x_k, \hat{e}_k = \hat{x}_k - x_k$$

Using the Kalman filter equations, we can derive:

$$\check{e}_k = F_{k-1}\hat{e}_{k-1} - w_{k-1}, \hat{e}_k = (I - K_k H_k)\check{e}_k + K_k v_k$$

$$\hat{e}_k = (I - K_k H_k)F_{k-1}\hat{e}_{k-1} - (I - K_k H_k)w_{k-1} + K_k v_k$$

For the Kalman filter, for all k, we have

$$\begin{aligned} E[\check{e}_k] &= E[F_{k-1}\check{e}_{k-1} - w_k] \\ &= F_{k-1}E[\check{e}_{k-1}] - E[w_k] \\ &= F_{k-1}E[\check{e}_{k-1}] \end{aligned}$$

$$\begin{aligned} E[\hat{e}_k] &= E[(\mathbf{1} - K_k H_k)\check{e}_k + K_k v_k] \\ &= (\mathbf{1} - K_k H_k)E[\check{e}_k] + K_k E[v_k] \\ &= (\mathbf{1} - K_k H_k)E[\check{e}_k] = (\mathbf{1} - K_k H_k)F_{k-1}E[\check{e}_{k-1}] \end{aligned}$$

The above dynamic determines whether a Kalman filter is stable.

If $E[\hat{e}_0 = 0], E[v] = 0, E[w] = 0$, v and w are uncorrelated white noise, we have

$$E[\check{e}_k] = E[\hat{e}_k] = 0$$

for all k. This means that the Kalman filter is unbiased. It is noted that this does not mean that the error on a given trial will be zero, but that, with enough trials, our expected error is zero.

Overall optimality

Then we discuss the overall optimality of the Kalman filter. The error $e_k = x_k - \hat{x}_k$ is a random variable determined by the stochastic process $\{w_k\}$ and $\{v_k\}$. Suppose we want to find the estimator that minimizes (at each time step) a weighted two-norm of the expected value of the estimation error e_k:

$$\min \quad E[e_k^T S_k e_k] \tag{5.9}$$

where S_k is a positive definite user-defined weighting matrix. The conclusions are,

1. if $\{w_k\}$ and $\{v_k\}$ are Gaussian, zero-mean, uncorrelated, and white, then the Kalman filter is the solution to the problem (5.9), i.e., MMSE estimate.

2. if $\{w_k\}$ and $\{v_k\}$ are zero-mean, uncorrelated, and white, the Kalman filter is the best linear solution to the problem (5.9), i.e., linear MMSE estimate.

Orthogonality principle

Last, we discuss the orthogonality principle in discrete-time Kalman filter. Given the initial estimate

$$\hat{x}_0 = E(x_0),$$

and the uncorrelated properties of the noise, which can be stated as follows, w_i, v_i are uncorrelated with all past or present states, i.e.,

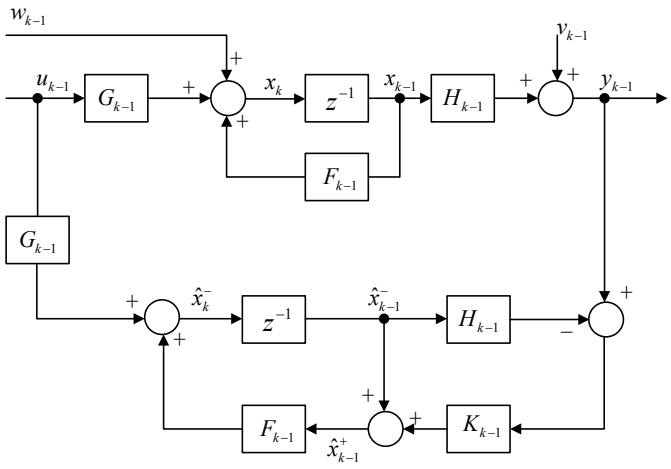
$$E[w_i x_j^T] = 0, E[v_i x_j^T] = 0, \forall j \leq i$$

and w_i, v_i are orthogonal to past outputs, i.e.,

$$E[w_i y_j^T] = 0, E[v_i y_j^T] = 0, \forall j < i$$

The propagation and update of the error can be described as,

$$\check{e}_k = F_{k-1}\hat{e}_{k-1} - w_{k-1}$$



$$\hat{e}_k = (I - K_k H_k)F_{k-1}\hat{e}_{k-1} - (I - K_k H_k)w_{k-1} + K_k v_k \tag{5.10}$$

We then use mathematical induction to derive the orthogonality principle in Kalman filtering. It can be seen from (5.10) that

$$\hat{e}_1 = (I - K_1 H_1)F_0\hat{e}_0 - (I - K_1 H_1)w_0 + K_1 v_1.$$

As K_1 solves the optimization problem

$$\min_{K_k} E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

with $k = 1$, according to the orthogonal principle for the LMMSE estimate, we have

$$E[\hat{e}_1 y_1^T] = 0$$

Assume $E[\hat{e}_k y_1^T] = 0, \ldots, E[\hat{e}_k y_k^T] = 0$, we have

$$E[\check{e}_{k+1} y_j^T] = E[(F_k\hat{e}_k - w_k)y_j^T] = 0, \forall j = 1, \ldots, k$$

and as K_k also solves the above optimization problem, we have

$$E[\hat{e}_{k+1} y_j^T] = 0, \forall j = 1, \ldots, k, k+1.$$

The block diagram of Kalman filter an be shown in Fig. 5.2.2.

The following shows another example for the discrete-time Kalman filter, in which we can find that the Kalman filter gain as well as the covariance matrix approaches a steady-state value when time approaches infinity.

Example 5.5 Considering a simple 1-dimensional example,

$$x_{k+1} = 0.5x_k + w_k$$
$$y_k = x_k + v_k$$

in which w_k and v_k are uncorrelated white noise with zero mean, i.e.,

$$E\{w_k\} = 0,\ E\{v_k\} = 0,\ E\{w_k w_j\} = 1 \cdot \delta_{k-j},\ E\{v_k v_j\} = 2 \cdot \delta_{k-j}$$

Initial values are $\hat{x}_0 = 0$, $P_0 = 1$; Observations are $y_1 = 4,\ y_2 = 2$. Find the optimal linear estimate of x_k.

1. Initialization: $\hat{x}_0 = 0$, $\hat{P}_0 = P_0 = 1$
2. Computation of the gain K_k as well as $\check{P}_k$, $\hat{P}_k$, i.e.,

$$\begin{aligned}
&\check{P}_1 = F_0\hat{P}_0F_0^T + Q_0 = 1.25\\
&K_1 = \check{P}_1H_1^T(H_1\check{P}_1H_1^T + R_1)^{-1} = 0.3846\\
&\hat{P}_1 = [I - K_1H_1]\check{P}_1[I - K_1H_1]^T + K_1R_1K_1^T = 0.7692\\
&\check{P}_2 = F_1\hat{P}_1F_1^T + Q_1 = 1.1923\\
&K_2 = \check{P}_2H_2^T(H_2\check{P}_2H_2^T + R_2)^{-1} = 0.3735
\end{aligned}$$

3. Estimations of the state sequence are

$$\begin{aligned}
&\check{x}_1 = F_0\hat{x}_0 = 0\\
&\hat{x}_1 = \check{x}_1 + K_1(y_1 - H_1\check{x}_1) = 1.5385\\
&\check{x}_2 = F_1\hat{x}_1 = 0.7692\\
&\hat{x}_2 = \check{x}_2 + K_2(y_2 - H_2\check{x}_2) = 1.2289
\end{aligned}$$

Computation of K_k when time increases is,

$$\begin{array}{lll}
K_1 = 0.3846 & \check{P}_1 = 1.2500 & \hat{P}_1 = 0.7692\\
K_2 = 0.3735 & \check{P}_2 = 1.1923 & \hat{P}_2 = 0.7470\\
K_3 = 0.3724 & \check{P}_3 = 1.1867 & \hat{P}_3 = 0.7448\\
K_4 = 0.3723 & \check{P}_4 = 1.1862 & \hat{P}_4 = 0.7446\\
K_5 = 0.3723 & \check{P}_5 = 1.1861 & \hat{P}_5 = 0.7446\\
K_6 = 0.3723 & \check{P}_6 = 1.1861 & \hat{P}_6 = 0.7446\\
\vdots & \vdots & \vdots
\end{array}$$

5.2.3 Steady-state Kalman filter

It can be seen from Example 5.5 that when time increases, the Kalman filter gain and error covariance reaches a steady-state value. The term "steady-state" Kalman filtering means that the Kalman filter is time-invariant and the Kalman gain is in steady-state. If we can calculate the steady-state Kalman filter, the savings in computations deserve any loss in the estimated state quality.

Assume the dynamic system is time-invariant, and a constant gain K exists, i.e.,

$$K = PH^T(HPH^T + R)^{-1},$$

Assume P reaches a steady-state

$$P = FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q \tag{5.11}$$

and the above equation is called discrete time algebraic Riccati equation (DARE). Then the existence of the steady-state Kalman filter is equivalent to the condition that the DARE has a unique positive semidefinite solution. It can be proved that the DARE has a unique positive semidefinite solution P iff both of the following conditions hold.

- (F, H) is detectable

  - A system is detectable if all the unobservable states are stable.

- (F, J) is stabilizable (J is any matrix such that $JJ^T = Q$)
  - A system is said to be stabilizable when all uncontrollable state variables can be made to have stable dynamics.

A sufficient condition for the existence of the steady-state Kalman filter is that the system is time invariant, (F, J) is controllable, and (F, H) is observable. Then for any nonnegative symmetric initial condition $\hat{P}_0$, one has

$$\lim_{k\to\infty} \hat{P}_k = P$$

and P satisfies the DARE (5.11). Besides, the Kalman filter gain K reaches a constant value and the matrix $(I - KH)F$ is stable.

In this case, the steady-state Kalman filter is a time-invariant observer (also time-invariant system):

$$\begin{aligned}\check{x}_k &= F\hat{x}_{k-1} + Gu_{k-1}\\ \hat{x}_k &= \check{x}_k + K(y_k - H\check{x}_k)\\ &= (I - KH)F\hat{x}_{k-1} + (I - KH)Gu_{k-1} + KHFx_{k-1} + Kv_k\end{aligned}$$

compared with the state space expression

$$\begin{aligned}x_k &= Fx_{k-1} + Gu_{k-1} + w_{k-1}\\ y_k &= Hx_k + v_k\end{aligned}$$

the state estimation error is

$$\tilde{x}_k^+ = x_k - \hat{x}_k = (I - KH)F\tilde{x}_{k-1}^+ + KHGu_{k-1} + w_{k-1} - Kv_k \tag{5.12}$$

The stability of the steady-state Kalman filter can also be considered. It can be seen from (5.12) that the estimation error propagates according to a linear system, with closed-loop dynamics $(I-KH)F$, driven by the process $w_{k-1} - Kv_k$, which is IID with zero mean and covariance $KRK^T + Q$ (provided that $u_k = 0$). Therefore, the stability of $(I - KH)F$ is requisite for the stability of the filter. This is to say, if the DARE has a unique positive semidefinite solution, then the steady-state Kalman filter is stable.

Example 5.6 *A simple example in 1-dimension.*

The truth model is given by the following equation,

$$\begin{aligned}x_{k+1} &= \varphi x_k + w_k\\ y_k &= hx_k + v_k\end{aligned}$$

in which w_k and v_k are stationary random process, with $w_k \sim \mathcal{N}(0, q)$ and $v_k \sim \mathcal{N}(0, r)$.

Considering the DARE, we have,

$$p = \varphi^2 p - \varphi ph\frac{1}{h^2 p + r}hp\varphi + q$$

By reordering, we have,

$$h^2p^2 + (r - \varphi^2 r - h^2 q)p - qr = 0 \tag{5.13}$$

Solving the second-order equation (5.13) gives the solution of steady-state p.

Next consider two special cases.

- no measurement noise: $r = 0$. then we have $p = q$ and $k = \frac{1}{h}$ and

  $$\hat{x}_{k+1} = \frac{\varphi}{h} y_k$$

  at this time, the estimate $\hat{x}_{k+1}$ depends entirely on the measurement, and does not depend on past estimate $\hat{x}_k$. This is because no measurement error exists, and the state can be estimated without the dynamic model.
- the model is accurate: $q = 0$. then we have $p = 0$. at this time, we have

  $$\hat{x}_{k+1} = \varphi \hat{x}_k$$

  the estimate depends entirely on the dynamic model, which is due to the model is precise.

5.3 Continuous-time Kalman filter

The continuous-time Kalman filter is also important. Although the vast majority of Kalman filter applications are implemented in digital computers, there are still opportunities to implement Kalman filters in continuous time (i.e., in analog circuits). Besides, the derivation of the continuous-time filter is instructive from a pedagogical point of view and the steady-state continuous-time estimators can be analyzed using conventional frequency-domain concepts, providing an advantage over discrete-time estimators.

5.3.1 Derivation of the continuous-time Kalman filter

Suppose that we have a continuous-time system given as

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ w &\sim \mathcal{N}(0, Q_c) \\ v &\sim \mathcal{N}(0, R_c) \end{aligned}$$

Assume $t = t_k$ and $x_k = x(t_k)$, the continuous-time system can be approximated by the following discrete-time system:

$$\begin{aligned} x_k &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned}$$

Next we will derive the expression for F_{k-1}, G_{k-1}, H_k and the stochastic properties of $\{w_k\}$ and $\{v_k\}$.

Recalling from the sampled-data system, the solution of $x(t)$ when $t = t_k$ can be computed as

$$x(t_k) = e^{A(t_k - t_{k-1})}x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}[Bu(\tau) + w(\tau)]d\tau$$

Suppose $u(t) = u(t_{k-1}), \forall t \in [t_{k-1}, t_k]$. This is reasonable if $t_k - t_{k-1} \to 0$. Further, we have

$$x_k = e^{A(t_k - t_{k-1})}x_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Bd\tau u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau$$

Suppose the sample time is T, i.e., $T = t_k - t_{k-1}$, Define

$$\begin{aligned} F &= \exp(AT) \\ G &= \textstyle\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Bd\tau \end{aligned}$$

then for small T, we have

$$F \approx I + AT$$

and

$$G = \int_{t_{k-1}}^{t_k} (I + A(t_k - \tau))Bd\tau \approx BT$$

Define $w_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau$, then we have

$$\bar{w}_{k-1} = 0$$

and

$$\begin{aligned} E[w_k w_j^T] &= \textstyle\int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} e^{A(t_{k+1}-\tau)}E[w(\tau)w^T(t)]e^{A^T(t_{j+1}-t)}d\tau dt \\ &= \textstyle\int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} 0 d\tau dt \quad \forall k \neq j \\ &= 0, \forall k \neq j. \end{aligned}$$

The covariance of $\{w_k\}$ is:

$$\begin{aligned} E[w_k w_k^T] &= \textstyle\int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}E[w(\tau)w^T(t)]e^{A^T(t_{k+1}-t)}dtd\tau \\ &= \textstyle\int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Q_c(\tau)\delta(t-\tau)e^{A^T(t_{k+1}-t)}dtd\tau \\ &= \textstyle\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-t)}Q_c(t)e^{A^T(t_{k+1}-t)}dt \\ &\approx Q_c T \text{ assume } T \text{ is small} \end{aligned}$$

The discretization of the measurement equation can be interpreted as:

$$y_k \approx Cx_k + \frac{1}{T}\int_{t_k}^{t_{k+1}} v(t)dt$$

Define $v_k = \frac{1}{T}\int_{t_k}^{t_{k+1}} v(t)dt$, then we have

$$\bar{v}_k = 0$$

and

$$\begin{aligned} E[v_k v_j^T] &= \tfrac{1}{T^2}\int_{t_k}^{t_{k+1}}\int_{t_j}^{t_{j+1}} E[v(\tau)v^T(t)]d\tau dt \\ &= \tfrac{1}{T^2}\int_{t_k}^{t_{k+1}}\int_{t_j}^{t_{j+1}} 0 d\tau dt \\ &= 0 \end{aligned}$$

Besides, the following holds,

$$\begin{aligned} E[v_k v_k^T] &= \tfrac{1}{T^2}\int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} E[v(t)v^T(\tau)]dtd\tau \\ &= \tfrac{1}{T^2}\int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} R_c\delta(t-\tau)dtd\tau \\ &= \tfrac{1}{T^2}\int_{t_k}^{t_{k+1}} R_c d\tau \\ &= \tfrac{R_c}{T} \end{aligned}$$

Define

$$H = C$$

we have the discretization of the continuous-time dynamic system with a sample time T:

$$\begin{aligned} x_k &= Fx_{k-1} + Gu_{k-1} + w_{k-1} \\ y_k &= Hx_k + v_k \end{aligned}$$

with the process noise and measurement noise as:

$$w_k \sim \mathcal{N}(0, Q), Q = Q_c T, \quad v_k \sim \mathcal{N}(0, R), R = R_c/T$$

The discrete-time Kalman filter gain for this system was derived as:

$$\begin{aligned} K_k &= \check{P}_k H^T (H\check{P}_k H^T + R)^{-1} \\ &= \check{P}_k C^T (C\check{P}_k C^T + R_c/T)^{-1} \end{aligned}$$

then

$$\begin{aligned} \tfrac{K_k}{T} &= \check{P}_k C^T (C\check{P}_k C^T T + R_c)^{-1} \\ \lim_{T\to 0} \tfrac{K_k}{T} &= \check{P}_k C^T R_c^{-1} \end{aligned}$$

The estimation-error covariances were derived as

$$\begin{aligned} \hat{P}_k &= (I - K_k H)\check{P}_k \\ \check{P}_{k+1} &= F\hat{P}_k F^T + Q \end{aligned}$$

For small values of T, this can be written as

$$\begin{aligned} \check{P}_{k+1} &= (I + AT)\hat{P}_k(I + AT)^T + Q_c T \\ &= \hat{P}_k + (A\hat{P}_k + \hat{P}_k A^T + Q_c)T + A\hat{P}_k A^T T^2 \end{aligned}$$

Substituting for $\hat{P}_k$ gives

$$\begin{aligned} \check{P}_{k+1} &= (I - K_k C)\check{P}_k + A\hat{P}_k A^T T^2 + \\ &\quad [A(I - K_k C)\check{P}_k + (I - K_k C)\check{P}_k A^T + Q_c]T \end{aligned}$$

Substracting $\check{P}_k$ from both sides and then dividing by T gives

$$\begin{aligned} \tfrac{\check{P}_{k+1} - \check{P}_k}{T} = &\quad \tfrac{-K_k C\check{P}_k}{T} + A\hat{P}_k A^T T + \\ &\quad (A\check{P}_k + AK_k C\check{P}_k + \check{P}_k A^T - K_k C\check{P}_k A^T + Q_c) \end{aligned}$$

Taking the limit as $T \to 0$ and using the expression for K_k gives

$$\begin{aligned}\dot{P} &= \lim_{T\to 0} \frac{\check{P}_{k+1}-\check{P}_k}{T} \\ &= -PC^T R_c^{-1} CP + AP + PA^T + Q_c\end{aligned}$$

The equation

$$\dot{P} = -PC^T R_c^{-1} CP + AP + PA^T + Q_c$$

is called a **differential Riccati equation** and can be used to compute the estimation-error covariance for the continuous-time Kalman filter. The computation requires n^2 integrations because P is an $n \times n$ matrix. As P is symmetric, so in practice we only need to integrate $n(n+1)/2$ equations in order to solve for P.

Now we will derive the continuous-time version for updating the estimate $\hat{x}$. Recall the discrete-time version of update is,

$$\begin{aligned}\check{x}_k &= F\hat{x}_{k-1} + Gu_{k-1} \\ \hat{x}_k &= \check{x}_k + K_k(y_k - H\check{x}_k)\end{aligned}$$

Assume that T is small, the measurement update equation can be written as

$$\begin{aligned}\hat{x}_k &= F\hat{x}_{k-1} + Gu_{k-1} + K_k(y_k - HF\hat{x}_{k-1} - HGu_{k-1}) \\ &\approx (I+AT)\hat{x}_{k-1} + BTu_{k-1} + K_k(y_k - C(I+AT)\hat{x}_{k-1} - CBTu_{k-1})\end{aligned}$$

Now subtract $\hat{x}_{k-1}$ from both sides, divide by T to obtain

$$\begin{aligned}\frac{\hat{x}_k - \hat{x}_{k-1}}{T} &= A\hat{x}_{k-1} + Bu_{k-1} \\ &+\check{P}_k C^T (C\check{P}_k C^T T + R_c)^{-1}(y_k - C(I+AT)\hat{x}_{k-1} - CBTu_{k-1})\end{aligned}$$

Taking the limit as $T \to 0$ gives

$$\dot{\hat{x}} = A\hat{x} + Bu + PC^T R_c^{-1}(y - C\hat{x})$$

Assume the continuous-time system dynamics and measurement equations are given as

$$\begin{aligned}\dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ w &\sim \mathcal{N}(0, Q_c) \\ v &\sim \mathcal{N}(0, R_c)\end{aligned}$$

in which $w(t)$ and $v(t)$ are continuous-time white noise processes. The continuous-time Kalman filter can be described using Algorithm.

Algorithm 5.2 *Continuous-time Kalman filter.*

Initialization

1. $\hat{x}_0 = E(x_0)$, $\hat{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$

Iteration (for k)

2. State and covariance propagation.

$K = PC^T R_c^{-1}$ (This K is not the limit of K_k as $T \to 0$)

$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x})$

$\dot{P} = -PC^T R_c^{-1} CP + AP + PA^T + Q_c$.

Example 5.7

Suppose the system dynamic is,

$$\begin{aligned}\dot{x}(t) &= -x(t) + w(t)\\ y(t) &= x(t) + v(t)\end{aligned}$$

in which $w(t)$ and $v(t)$ are white noises with zero mean, and the statistical property is as follows,

$$\begin{aligned}&E\{w(t)\} = E\{v(t)\} = 0\\ &E\{w(t)w(\tau)\} = 2.5\delta(t-\tau)\\ &E\{v(t)v(\tau)\} = 2\delta(t-\tau)\\ &E\{w(t)v(\tau)\} = 0\end{aligned}$$

Assume $P(0) = 3$, $E\{x(0)\} = m_0$. Design the continuous-time Kalman filter.

According to the filter design principle, we have

$$K(t) = 0.5P(t)$$

and the derivation of $P(t)$ is,

$$\dot{P}(t) = -2P(t) - 0.5P^2(t) + 2.5$$

Denote $P = P(t)$, then the above differential equation can be written as,

$$\begin{aligned}&\tfrac{dP}{dt} = -0.5(P-1)(P+5)\\ &\tfrac{dP}{P-1} - \tfrac{dP}{P+5} = -3dt\\ &\int_3^P \left(\tfrac{dP}{P-1} - \tfrac{dP}{P+5}\right) = \int_0^t -3dt\\ &\ln(P-1) - \ln 2 - \ln(P+5) + \ln 8 = -3t\\ &\tfrac{P-1}{P+5} = e^{-3t-2\ln 2}\\ &\tfrac{P-1}{P+5} = \tfrac{1}{4}e^{-3t}\end{aligned}$$

Therefore, we have,

$$P(t) = \frac{1 + \frac{5}{4}e^{-3t}}{1 - \frac{1}{4}e^{-3t}}$$

As $t \longrightarrow \infty$, $P(t) \longrightarrow 1$.

The Kalman gain $K(t)$ is,

$$K(t) = \frac{1 + \frac{5}{4}e^{-3t}}{2\left(1 - \frac{1}{4}e^{-3t}\right)}$$

and $K(t) \longrightarrow 0.5$ as $t \longrightarrow \infty$.

When time increases, the Kalman filter gain and error covirance reaches a steady-state value.

Also, if we want to directly know the steady-state covariance, just let $\dot{P}(t) = 0$, which gives,

$$-2P(t) - 0.5P^2(t) + 2.5 = 0$$

that is,

$$P(t) = 1, \quad \text{or} \quad P(t) = -5(\text{is not reasonable, hence delete})$$

The steady-state covariance and gain is independent of the initial value $P(0)$.

Example 5.8 Assume we obtain measurements of the velocity of an object that is moving in one dimension, and the object is subject to random accelerations. What we want to estimate is the velocity x from noisy velocity measurements. The system and measurement equations are given as

$$\begin{aligned}
&\dot{x} = w \\
&y = x + v \\
&w \sim \mathcal{N}(0, Q) \\
&v \sim \mathcal{N}(0, R)
\end{aligned}$$

The covariance matrix is

$$\begin{aligned}
\dot{P} &= -PC^T R^{-1} CP + AP + PA^T + Q \\
&= -P^2/R + Q
\end{aligned}$$

Integrate both sides from 0 to t yields:

$$\int_{P(0)}^{P(t)} \frac{dP}{Q - P^2/R} = \int_0^t d\tau$$

Then we have (assume $\sqrt{Q} > P/\sqrt{R}$)

$$\frac{\sqrt{R}}{2\sqrt{Q}} \ln\left(\frac{\sqrt{Q} + P/\sqrt{R}}{\sqrt{Q} - P/\sqrt{R}}\right) |_{P(0)}^{P(t)} = t$$

Solving the differential equation for P gives

$$P = \sqrt{QR}\left[\frac{P_0 - \sqrt{QR} + (\sqrt{QR} + P_0)\exp(2t\sqrt{Q/R})}{\sqrt{QR} - P_0 + (\sqrt{QR} + P_0)\exp(2t\sqrt{Q/R})}\right]$$

Take the limit as $t \to \infty$ we have

$$\lim_{t\to\infty} P = \sqrt{QR}$$

The Kalman gain is

$$K = PC^T R^{-1} = P/R$$

Take the limit as $t \to \infty$ and we have

$$\lim_{t\to\infty} K = \sqrt{Q/R}$$

The state estimate update expression is

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) = K(y - \hat{x}) \to \sqrt{\frac{Q}{R}}(y - \hat{x})$$

It is noted that if the process noise increases (i.e., Q increases) then K increases, meaning that we have less confidence in our system model, and relatively more confidence in our measurements. So we change $\hat{x}$ more aggressively to be consistent with our measurements. Besides, if we have large measurement noise (i.e., R is large) then K decreases, meaning that we have less confidence in our measurements. So we change $\hat{x}$ less aggressively to be consistent with our measurements. If either Q or R increase then P increases, i.e., an increase in the noise in either the system model or the measurements will degrade our confidence in our state estimate.

Similar to the steady-state discrete-time Kalman filter, here we develop the steady-state continuous-time Kalman filter.

- If (A, C) is detectable and $(A, Q_c^{\frac{1}{2}})$ is stabilizable, then for any $P_0 > 0$, the time-variant Kalman filter approaches steady state as $t \longrightarrow \infty$, the time-invariant implementation,

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(y(t) - C\hat{x}(t)) \tag{5.14}$$

  where $K = PC^T R^{-1}$ and P is the unique stabilizing solution of the continuous time algebraic Riccati equation (CARE),

$$0 = AP + PA^T + Q_c - PC^T R_c^{-1} CP \tag{5.15}$$

- At least one such solution results in a marginally stable steady-state Kalman filter.

Example 5.9 We consider the following two-state system:

$$\begin{aligned}
\dot{x} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + w \\
y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + v \\
Q &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
R &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

Solve the steady-state continuous-time Kalman filter.

The differential Riccati equation for the Kalman filter is given as

$$\dot{P} = -PC^T R^{-1} CP + AP + PA^T + Q$$

This can be written as the following three coupled differential equations:

$$\begin{aligned}
\dot{p}_{11} &= 2p_{11} - p_{11}^2 - p_{12}^2 \\
\dot{p}_{12} &= 2p_{12} - p_{11}p_{12} - p_{12}p_{22} \\
\dot{p}_{22} &= 2p_{22} - p_{12}^2 - p_{22}^2
\end{aligned}$$

Let $\dot{P} = 0$ and calculate the steady-state value for P yields,

$$P = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ or } P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ or } P = \begin{bmatrix} c & \pm\sqrt{2c - c^2} \\ \pm\sqrt{2c - c^2} & 2 - c \end{bmatrix}$$

in which $c \in [0, 2]$ is a scalar.

Then we have

$$\begin{aligned}
K &= PC^T R^{-1} = P \\
&= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ or } K = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ or } K = \begin{bmatrix} c & \pm\sqrt{2c - c^2} \\ \pm\sqrt{2c - c^2} & 2 - c \end{bmatrix}
\end{aligned}$$

Hence the estimate for x is

$$\begin{aligned}\dot{\hat{x}} &= (A-KC)\hat{x}+Ky \\ &= (-\hat{x}+Ky) \text{ or } (\hat{x}+Ky) \text{ or } \begin{bmatrix} 1-c & \mp\sqrt{2c-c^2} \\ \mp\sqrt{2c-c^2} & c-1 \end{bmatrix}\hat{x}+Ky\end{aligned}$$

in which only the first steady-state continuous-time Kalman filter is stable (the eigenvalues of the first are -1, -1, of the second are 1, 1, of the third are 1, -1). The other two filters are unstable Kalman filters.

5.4 Kalman filter generalizations

The discrete-time Kalman filter can be generalized to systems with correlated process or measurement noise.

5.4.1 Correlated process and measurement noise

Suppose that we have a system given by

$$\begin{aligned} x_k &= F_{k-1}x_{k-1}+G_{k-1}u_{k-1}+w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k\delta_{k-j} \\ E[v_k v_j^T] &= R_k\delta_{k-j} \\ E[w_k v_j^T] &= M_j\delta_{k-j+1} \end{aligned} \tag{5.16}$$

If $M_j \neq 0$, we way that the process noise and measurement noise are correlated.

We can use an example to explain this. Suppose our system is an airplane and winds are buffeting the plane, we are using an anemometer to measure wind speed as an input to our Kalman filter. The random gusts of wind affect both the process (i.e., the airplane dynamics) and the measurement (i.e., the sensed wind speed), i.e., the process noise w_k affects the state x_{k+1}, while v_{k+1} affects the measurement y_{k+1}, and w_k is correlated with v_{k+1}.

In this case, we list the update equation for the state estimate.

$$\begin{aligned}\check{x}_k &= F_{k-1}\hat{x}_{k-1}+G_{k-1}u_{k-1} \\ \hat{x}_k &= \check{x}_k + K_k(y_k - H_k\check{x}_k)\end{aligned}$$

It is noted that the gain matrix K_k will not the same as that in the white noise case. Define the estimation error as

$$\begin{aligned}\check{e}_k &= x_k - \check{x}_k \\ \hat{e}_k &= x_k - \hat{x}_k\end{aligned}$$

The dynamic of the estimation error is similar to the white noise case, i.e.,

$$\begin{aligned}
\check{e}_k &= x_k - \check{x}_k \\
&= (F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}) - (F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}) \\
&= F_{k-1}\hat{e}_{k-1} + w_{k-1} \\
\hat{e}_k &= x_k - [\check{x}_k + K_k(y_k - H_k\check{x}_k)] \\
&= \check{e}_k - K_k(H_k x_k + v_k - H_k\check{x}_k) \\
&= \check{e}_k - K_k(H_k\check{e}_k + v_k) \\
&= (I - K_k H_k)\check{e}_k - K_k v_k
\end{aligned}$$

We can express the *a priori* and *a posteriori* estimation error covariance as follows.

$$\begin{aligned}
\check{P}_k &= E[\check{e}_k(\check{e}_k)^T] \\
&= F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1} \\
\hat{P}_k &= E[\hat{e}_k(\hat{e}_k)^T] \\
&= E\{[\check{e}_k - K_k(H_k\check{e}_k + v_k)][\cdots]^T\} \\
&= \check{P}_k - K_k H_k\check{P}_k - K_k E[v_k(\check{e}_k)^T] - \check{P}_k H_k^T K_k^T + \\
&\quad K_k H_k\check{P}_k H_k^T K_k^T + K_k E[v_k(\check{e}_k)^T]H_k^T K_k^T - \\
&\quad E(\check{e}_k v_k^T)K_k^T + K_k H_k E(\check{e}_k v_k^T)K_k^T + K_k E(v_k v_k^T)K_k^T
\end{aligned}$$

Notice that

$$\begin{aligned}
E(\check{e}_k v_k^T) &= E[(x_k - \check{x}_k)v_k^T] \\
&= E(x_k v_k^T - \check{x}_k v_k^T) \\
&= E[(F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1})v_k^T] - E[\check{x}_k v_k^T] \\
&= 0 + 0 + M_k - 0
\end{aligned}$$

in which the last term is 0 because the a priori state estimate at time k is independent of v_k. Hence we can simplify the expression for $\hat{P}_k$, i.e.,

$$\begin{aligned}
\hat{P}_k &= \check{P}_k - K_k H_k\check{P}_k - K_k M_k^T - \check{P}_k H_k^T K_k^T + K_k H_k\check{P}_k H_k^T K_k^T + \\
&\quad + K_k M_k^T H_k^T K_k^T - M_k K_k^T + K_k H_k M_k K_k^T + K_k R_k K_k^T \\
&= (I - K_k H_k)\check{P}_k(I - K_k H_k)^T + K_k R_k K_k^T + \\
&\quad K_k(H_k M_k + M_k^T H_k^T)K_k^T - M_k K_k^T - K_k M_k^T
\end{aligned}$$

The optimal gain matrix K_k can be obtained by minimizing $\mathrm{Tr}(\hat{P}_k)$, which is basically MSE. Recall that

$$\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} = 2AB \text{ if } B \text{ is symmetric.}$$

We can use this fact to derive

$$\begin{aligned}
\frac{\partial \mathrm{Tr}(\hat{P}_k)}{\partial K_k} &= -2(I - K_k H_k)\check{P}_k H_k^T + 2K_k R_k + \\
&\quad 2K_k(H_k M_k + M_k^T H_k^T) - M_k - K_k \\
&= 2[K_k(H_k\check{P}_k H_k^T + H_k M_k + M_k^T H_k^T + R_k) - \check{P}_k H_k^T - M_k]
\end{aligned}$$

Setting the partial derivative to be zero gives the optimal gain K_k as

$$K_k = (\check{P}_k H_k^T + M_k)(H_k\check{P}_k H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1}$$

Then the *a posteriori* covariance $\hat{P}_k$ can be expressed as,

$$\begin{aligned}\hat{P}_k &= (I - K_k H_k)\check{P}_k(I - K_k H_k)^T + K_k R_k K_k^T + \\ &\quad K_k(H_k M_k + M_k^T H_k^T)K_k^T - M_k K_k^T - K_k M_k^T \\ &= \check{P}_k - K_k(H_k \check{P}_k + M_k^T)\end{aligned}$$

Suppose the system and measurement equations are given in (5.16), the general discrete-time Kalman filter when the process noise is correlated with the measurement noise can be described as follows.

Algorithm 5.3 *Discrete-time Kalman filter.*

Initialization

1. $\hat{x}_0 = E(x_0)$, $\hat{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$

Iteration (for k)

2. State and covariance propagation (*a priori* estimate).

$\check{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}$

$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$

3. Obtain the measurement $\mathbf{y}_k$, update the estimate of x and the estimation-error covariance P as follows (*a posteriori* estimate)

$K_k = (\check{P}_k H_k^T + M_k)(H_k \check{P}_k H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1}$

$\hat{x}_k = \check{x}_k + K_k(y_k - H_k \check{x}_k)$

$\hat{P}_k = (I - K_k H_k)\check{P}_k(I - K_k H_k)^T + K_k R_k K_k^T + K_k(H_k M_k + M_k^T H_k^T)K_k^T - M_k K_k^T - K_k M_k^T$

or $\hat{P}_k = \check{P}_k - K_k(H_k \check{P}_k + M_k^T)$

We can also use the orthogonality principle to derive the Kalman filter in this case. The estimate is initialized as,

$$\hat{x}_0 = E(x_0), \hat{P}_0 = E(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T$$

the *a priori* estimate at time $k = 1$ is

$$\check{x}_1 = E(x_1) = F_0\hat{x}_0 + G_0 u_0, \check{P}_1 = F_0 \hat{P}_0 F_0^T + Q_0$$

At time $k = 1$, according to the orthogonality principle, the *a posteriori* estimate should satisfy,

$$E(\hat{e}_1 y_1^T) = 0$$

in which $\hat{e}_1 = x_1 - \hat{x}_1$.

As

$$\hat{e}_1 = (I - K_1 H_1)\check{e}_1 + K_1 v_1$$

we have

$$E(\hat{e}_1 y_1^T) = E[(I - K_1 H_1)\check{e}_1 + K_1 v_1][H_1 x_1 + v_1]^T = 0$$

Hence

$$E[(I - K_1 H_1)\check{e}_1 + K_1 v_1][H_1(\check{x}_1 - \check{e}_1) + v_1]^T = 0$$

As

$$E(\check{e}_1 v_1^T) = M_1, E(\check{e}_1 \check{x}_1^T) = 0, E(v_1 \check{x}_1^T) = 0$$

We can find K_1 as,

Example 5.10

Consider the following scalar system:

$$\begin{aligned} x_k &= 0.8x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \\ E[w_k w_j^T] &= 1 \cdot \delta_{k-j} \\ E[v_k v_j^T] &= 0.1 \cdot \delta_{k-j} \\ E[w_k v_j^T] &= M \cdot \delta_{k-j+1} \end{aligned}$$

Correlation M	Standard Filter (M=0 assumed)	Correlated Filter (correct M used)
0	0.076	0.076
0.25	0.030	0.019
-0.25	0.117	0.052

5.4.2 Colored process and measurement noise

In this section, we consider cases when the process itself or the measurement itself is correlated.

Colored process noise

Suppose we have an LTI system given as

$$x_k = Fx_{k-1} + w_{k-1}$$

where the covariance of w_k is equal to Q_k. Further suppose that the process noise is the output of a dynamic system:

$$w_k = \psi w_{k-1} + \zeta_{k-1}$$

where ζ_{k-1} is zero-mean white noise that is uncorrelated with w_{k-1}. Hence, we have

$$E(w_k w_{k-1}^T) = E(\psi w_{k-1} w_{k-1}^T + \zeta_{k-1} w_{k-1}^T) = \psi Q_{k-1}$$

i.e., w_k and w_{k-1} are correlated.

The state is augmented to solve this kind of problems. Suppose $x'_k = [x_k^T, w_k^T]^T$, we have

$$\begin{bmatrix} x_k \\ w_k \end{bmatrix} = \begin{bmatrix} F & I \\ 0 & \psi \end{bmatrix} \begin{bmatrix} x_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \zeta_{k-1} \end{bmatrix}$$

i.e.,

$$x'_k = F' x'_{k-1} + w'_{k-1}$$

This is an augmented system with a new state x', a new system matrix F' and a new process noise vector w' whose covariance is given as follows:

$$E(w'_k (w'_k)^T) = \begin{bmatrix} 0 & 0 \\ 0 & E(\zeta_k \zeta_k^T) \end{bmatrix} = Q'_k$$

It is noted that the computational effort increases because the state dimension has doubled.

Colored measurement noise Now consider the case when we have colored measurement noise, i.e., the measurement is correlated. The system and measurement equations are given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ v_k &= \psi_{k-1}v_{k-1} + \eta_{k-1}, \\ w_k &\sim \mathcal{N}(0, Q_k), \quad \eta_k \sim \mathcal{N}(0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[\eta_k \eta_j^T] &= Q_{\eta k}\delta_{k-j} \\ E[w_k \eta_j^T] &= 0 \end{aligned}$$

The measurement noise is itself the output of a linear system with

$$E[v_k v_{k-1}^T] = E[(\psi_{k-1}v_{k-1} + \eta_{k-1})v_{k-1}^T] = \psi_{k-1}E[v_{k-1}v_{k-1}^T]$$

Similar to the case of colored process noise, we augment the original system model as follows,

$$\begin{aligned} \begin{bmatrix} x_k \\ v_k \end{bmatrix} &= \begin{bmatrix} F_{k-1} & 0 \\ 0 & \psi_{k-1} \end{bmatrix}\begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k-1} \\ \eta_{k-1} \end{bmatrix} \\ y_k &= \begin{bmatrix} H_k & I \end{bmatrix}\begin{bmatrix} x_k \\ v_k \end{bmatrix} + 0 \end{aligned}$$

This can be written as

$$\begin{aligned} x'_k &= F'_{k-1}x'_{k-1} + w'_{k-1} \\ y_k &= H'_k x'_k + v'_k \end{aligned}$$

For the newly generated system, the covariance of the process noise is,

$$\begin{aligned} E[w'_k {w'_k}^T] &= E\left[\begin{pmatrix} w_k \\ \eta_k \end{pmatrix}\begin{pmatrix} w_k^T & \eta_k^T \end{pmatrix}\right] = \begin{bmatrix} Q_k & 0 \\ 0 & Q_{\eta k} \end{bmatrix} \\ E[v'_k {v'_k}^T] &= 0 \end{aligned}$$

We can find that there is no measurement noise. In practice, a singular measurement-noise covariance often results in numerical problems. Hence we solve this problem from another point of view.

Define an auxiliary signal y'_k as follows:

$$y'_{k-1} = y_k - \psi_{k-1}y_{k-1}$$

Substitute for y_k and y_{k-1} in the above definition, and we have

$$\begin{aligned} y'_{k-1} &= H_k x_k + v_k - \psi_{k-1}(H_{k-1}x_{k-1} + v_{k-1}) \\ &= H_k(F_{k-1}x_{k-1} + w_{k-1}) + v_k - \psi_{k-1}(H_{k-1}x_{k-1} + v_{k-1}) \\ &= (H_k F_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + H_k w_{k-1} + v_k - \psi_{k-1}v_{k-1} \\ &= (H_k F_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + (H_k w_{k-1} + \eta_{k-1}) \\ &= H'_{k-1}x_{k-1} + v'_{k-1} \end{aligned}$$

The equivalent system can be expressed as,

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + w_{k-1} \\ y'_k &= H'_{k-1}x_{k-1} + v'_{k-1} \end{aligned}$$

The covariance of the new measurement noise v' is,

$$\begin{aligned} E[v'_k {v'_k}^T] &= E[(H_{k+1}w_k + \eta_k)(H_{k+1}w_k + \eta_k)^T] \\ &= H_{k+1}Q_kH_{k+1}^T + Q_{\eta_k} \\ E[w_k {v'_k}^T] &= E[w_k(H_{k+1}w_k + \eta_k)^T] \\ &= Q_kH_{k+1}^T \end{aligned}$$

Define the a priori and a posteriori state estimates for the equivalent system as:

$$\check{x}_k = E[x_k|y_1, \ldots, y_k]$$

$$\hat{x}_k = E[x_k|y_1, \ldots, y_k, y_{k+1}] = \check{x}_k + K_k(y'_k - H'_k\check{x}_k)$$

this definition is slightly different, as $y'_{k-1} = y_k - \psi_{k-1}y_{k-1}$.

The optimal gain matrix K_k can be find through minimizing the trace of the covariance of the estimation error, i.e.,

$$K_k = \text{argmin Tr } E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

Example 5.11 Consider the linear system with colored measurement noise

$$\begin{aligned} x_k &= \begin{bmatrix} 0.70 & -0.15 \\ 0.03 & 0.79 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0.15 \\ 0.21 \end{bmatrix} w_{k-1} \\ y_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \\ v_k &= \psi v_{k-1} + \zeta_{k-1} \\ E[w_k w_j^T] &= 1 \cdot \delta_{k-j} \\ E[\zeta_k \zeta_j^T] &= \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \delta_{k-j} \\ E[w_k \zeta_j^T] &= 0 \end{aligned}$$

Color ψ	Standard Filter	Augmented Filter	Measurement Differencing
0	0.245	0.245	0.247
0.2	0.260	0.258	0.259
0.5	0.308	0.294	0.295
0.9	0.631	0.407	0.406

5.5 Nonlinear Kalman filter

In Section 5.2-5.4, we have developed the Kalman filter for linear systems. Actually, all systems are ultimately nonlinear (even a device as simple as a resistor is

only approximately linear, and even then only in a limited range of operation), in which many systems are close enough to linear that linear estimation approaches give satisfactory results. However, there is some system does not behave linearly even over a small range of operation, and our linear approaches for estimation no longer give good results. Then we need to explore nonlinear estimators, and some nonlinear estimation methods including nonlinear extensions of the Kalman filter, unscented filtering, and particle filtering have become widespread.

5.5.1 Extended Kalman filter

For the nonlinear discrete-time system, we now discuss how can we adapt the Kalman filter to this case.

For a 1-dimensional nonlinear function $f(x) : \mathbf{R} \to \mathbf{R}$, a natural thought is to choose an operating point x_0 and approximate the nonlinear function by a tangent line (hyperplane) at that point. Mathematically, we compute this linear approximation using a first-order Taylor expansion,

$$f(x) \approx f(x_0) + \frac{\partial f(x)}{\partial x}|_{x=x_0}(x - x_0) + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2}|_{x=x_0}(x - x_0)^2 + \cdots$$

in which the third term in the right-hand side refers to higher-order term.

Suppose we have the system model

$$\begin{array}{ll} x_k & = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k & = h_k(x_k, v_k) \\ w_k & \sim \mathcal{N}(0, Q_k) \\ v_k & \sim \mathcal{N}(0, R_k) \end{array} \tag{5.17}$$

We perform a Taylor series expansion of the state equation around $x_{k-1} = \hat{x}_{k-1}$ and $w_{k-1} = 0$ to obtain the following:

$$\begin{array}{rcl} x_k & \approx & f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) + \frac{\partial f_{k-1}}{\partial x}|_{(\hat{x}_{k-1},0)}(x_{k-1} - \hat{x}_{k-1}) \\ & & + \frac{\partial f_{k-1}}{\partial w}|_{(\hat{x}_{k-1},0)} w_{k-1} \\ & = & f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + L_{k-1}w_{k-1} \\ & = & F_{k-1}x_{k-1} + [f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) - F_{k-1}\hat{x}_{k-1}] + L_{k-1}w_{k-1} \\ & = & F_{k-1}x_{k-1} + \tilde{u}_{k-1} + \tilde{w}_{k-1} \end{array}$$

where $F_{k-1} = \frac{\partial f_{k-1}}{\partial x}|_{(\hat{x}_{k-1},0)}$, $L_{k-1} = \frac{\partial f_{k-1}}{\partial w}|_{(\hat{x}_{k-1},0)}$. The input is $\tilde{u}_k = f_k(\hat{x}_k, u_k, 0) - F_k\hat{x}_k$. The process noise $\tilde{w}_k \sim (0, L_k Q_k L_k^T)$.

We linearize the measurement equation around $x_k = \check{x}_k$ and $v_k = 0$ to obtain

$$\begin{array}{rcl} y_k & \approx & h_k(\check{x}_k, 0) + \frac{\partial h_k}{\partial x}|_{(\check{x}_k,0)}(x_k - \check{x}_k) + \frac{\partial h_k}{\partial v}|_{(\check{x}_k,0)} v_k \\ & = & H_k x_k + [h_k(\check{x}_k, 0) - H_k\check{x}_k] + M_k v_k \\ & = & H_k x_k + z_k + \tilde{v}_k \end{array}$$

where $H_k = \frac{\partial h_k}{\partial x}|_{(\check{x}_k,0)}$ and $M_k = \frac{\partial h_k}{\partial v}|_{(\check{x}_k,0)}$. The signal z_k and the noise signal $\tilde{v}_k$ are defined as

$$\begin{array}{ll} z_k & = h_k(\check{x}_k, 0) - H_k\check{x}_k \\ \tilde{v}_k & \sim \mathcal{N}(0, M_k R_k M_k^T) \end{array}$$

Based on the above linearization, suppose the system dynamic is given as (5.17), the discrete-time extended Kalman filter can be described as follows.

Algorithm 5.4 *Discrete-time Kalman filter.*

Initialization
1. $\hat{x}_0 = E(x_0)$, $\hat{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$
Iteration (for k)
2. Compute the partial derivative matrices:
$F_{k-1} = \frac{\partial f_{k-1}}{\partial x}|_{(\hat{x}_{k-1},0)}$
$L_{k-1} = \frac{\partial f_{k-1}}{\partial w}|_{(\hat{x}_{k-1},0)}$
3. Perform the time update (*a priori* estimate).
$\check{x}_k = f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0)$
$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T$
4. Compute the partial derivative matrices:
$H_k = \frac{\partial h_k}{\partial x}|_{(\check{x}_k,0)}$
$M_k = \frac{\partial h_k}{\partial v}|_{(\check{x}_k,0)}$
5. Obtain the measurement $\mathbf{y}_k$, update the estimate of x and the estimation-error covariance P as follows (*a posteriori* estimate)
$K_k = \check{P}_k H_k^T (H_k \check{P}_k H_k^T + M_k R_k M_k^T)^{-1}$
$\hat{x}_k = \check{x}_k + K_k(y_k - h_k(\check{x}_k, 0))$
$\hat{P}_k = (I - K_k H_k)\check{P}_k$

It is noted that **The key to the EKF** lies in the **linearization of the original nonlinear dynamic system.** If the linearization does not provide a reasonably accurate description of the system dynamics, the state estimates may diverge. The computation of partial derivative matrices as well as the error covariance requires the estimates $\hat{x}_k$ and $\check{x}_k$. As a result, the EKF can not be tested off-line; it requires real or simulated data.

Example 5.12 Estimating the position of a car. Fig. 5.4 shows the estimation problem using a nonlinear measurement. The process model has been linearized and discretized as,

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}\end{aligned}$$

The nonlinear measurement equation is,

$$y_k = \phi_k = h(p_k, v_k) = \arctan\left(\frac{S}{D - p_k}\right) + v_k$$

The process noise and measurement noise are assumed to be white noise, i.e.,

$$v_k \sim \mathcal{N}(0, 0.05), \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, 0.1\mathbf{1}_{2\times 2})$$

We can calculate the motion model Jacobians as,

$$F_{k-1} = \frac{\partial f}{\partial x_{k-1}}|_{\hat{x}_{k-1}, u_{k-1}, 0} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

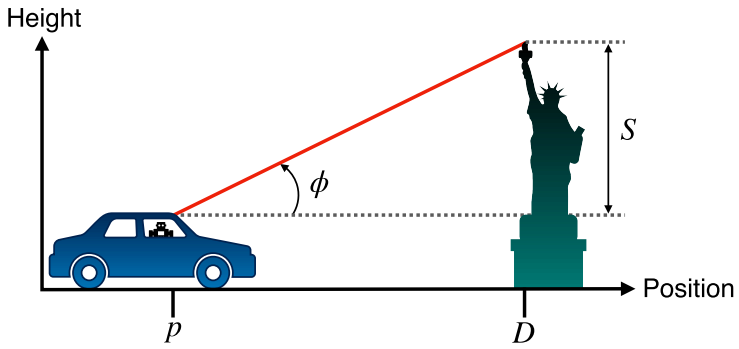


Figure 5.4 Application of the extended Kalman filter to estimate the position of a car.

$$L_{k-1} = \frac{\partial f}{\partial w_{k-1}}|_{\hat{x}_{k-1},u_{k-1},0} = \mathbf{1}_{2\times 2}$$

The measurement model Jacobians are

$$H_k = \frac{\partial h}{\partial x_k}|_{\check{x}_k,0} = \begin{bmatrix} \frac{S}{(D-\check{p}_k)^2+S^2} & 0 \end{bmatrix}$$

$$M_k = \frac{\partial h}{\partial v_k}|_{\check{x}_k,0} = 1$$

The initial state is

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

The sample instant is $\Delta t = 0.5\text{s}$, the initial input is $u_0 = -2\text{m/s}^2$, the measurements available are

$$y_1 = 30\text{deg}, S = 20\text{m}, D = 40\text{m}$$

The prediction can be performed,

$$\begin{aligned} \check{\mathbf{x}}_k &= f(\hat{x}_{k-1}, u_{k-1}, 0) = F_{k-1}\hat{\mathbf{x}}_{k-1} + G_{k-1}u_{k-1} \\ \check{P}_k &= F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1} \end{aligned}$$

thus we have

$$\begin{bmatrix} \check{p}_1 \\ \check{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \cdot (-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

$$\check{P}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

This is the same result as in the linear Kalman filter example because the motion model is already linear.

The corrections can be expressed as,

$$\begin{aligned} K_1 &= \check{P}_1 H_1^T (H_1 \check{P}_1 H_1^T + M_1 R_1 M_1^T)^{-1} \\ &= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 \\ 0 \end{bmatrix} \left([0.011\ 0] \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 \\ 0 \end{bmatrix} + 1(0.01)1 \right)^{-1} \\ &= \begin{bmatrix} 0.39 \\ 0.55 \end{bmatrix}, \end{aligned}$$

$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + K_1(y_1 - h_1(\check{\mathbf{x}}_1, 0)$$

i.e.,

$$\begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.39 \\ 0.55 \end{bmatrix} (0.52 - 0.49) = \begin{bmatrix} 2.51 \\ 4.02 \end{bmatrix}$$

Then the covariance $\hat{P}_1$ is

$$\hat{P}_1 = (\mathbf{1} - K_1 H_1)\check{P}_1 = \begin{bmatrix} 0.3585 & 0.4979 \\ -0.4978 & 1.0970 \end{bmatrix}$$

5.5.2 Nonlinear system identification using neural networks

First we discuss the linear parameters identificaiton. Consider a single-input single-output discrete-time system given by the difference equation

$$y_k = f(y_{k-1}, \ldots, y_{k-n_a}, u_{k-1}, \ldots, u_{k-n_b}) \tag{5.18}$$

We can approximate it by a linear relationship

$$y_k = \sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=1}^{n_b} b_i u(n-i) + v_k$$

where v_k is the model error term.

The linear relationship can be expressed as the state-space formulation,

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= c_k x_k + v_k \end{aligned}$$

where

$$x_k = [a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}]^T$$

and

$$c_k = [y_{k-1}, \ldots, y_{k-n_a}, u_{k-1}, \ldots, u_{k-n_b}]$$

Then we can use linear discrete-time Kalman filter to estimate x_k, which is actually the system parameters. As $x_{k+1} = x_k$, this is equivalent to the recursive least squares. The readers can compare this with system identification method you have learned.

We can then discuss the nonlinear system identification. If the function f in (5.18) is nonlinear, an accurate model can be constructed by using a neural network. For a simple 2 neurons neural network model, we have

$$\begin{aligned} y_k &= c_1\chi_k + c_2\lambda_k + v_k \\ \chi_k &= \text{act}[a_{11}y_{k-1} + b_{11}u_k] \\ \lambda_k &= \text{act}[a_{21}y_{k-1} + b_{21}u_k] \end{aligned}$$

where act denotes the activation function.

The state-space formulation can be written as,

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= \gamma(x_k) + v_k \end{aligned}$$

where

$$\gamma(x_k) = \sum_{j=1}^{q} c_j \text{act}[\phi_k\theta_j]$$

The parameter (state) to be estimated is

$$x_k = [c_1, \ldots, c_q, \theta_1, \ldots, \theta_q]$$

where θ_j is a vector with the length length(ϕ_k).

The extended Kalman filter can be described as follows,

Time update (Propagate)

$$\begin{aligned} \check{x}_{k+1} &= \hat{x}_k \\ \check{P}_{k+1} &= \hat{P}_k \end{aligned}$$

Measurement update (Correct)

- Compute the partial derivative matrices:

$$H_k = \frac{\partial\gamma}{\partial x}|_{\check{x}_k} = \left[\begin{array}{cccc} \frac{\partial\gamma}{\partial\theta_1} & \ldots & \frac{\partial\gamma}{\partial\theta_q} & \frac{\partial\gamma}{\partial c} \end{array} \right]^T |_{\check{x}_k}$$

  while

$$\frac{\partial\gamma}{\partial\theta_j} = c_j \frac{\partial \text{act}(\eta)}{\partial\eta}\theta_j, j = 1, \ldots, q, \eta = \phi_k\theta_j$$

  and

$$\frac{\partial\gamma}{\partial c} = \left[\begin{array}{ccc} \text{act}(\phi_k\theta_1) & \ldots & \text{act}(\phi_k\theta_q) \end{array} \right]$$

- Update the matrices

$$\begin{aligned} K_k &= \check{P}_k H_k^T (H_k \check{P}_k H_k^T + R_k)^{-1} \\ \hat{x}_k &= \check{x}_k + K_k[y_k - \gamma(\check{x}_k, 0)] \\ \hat{P}_k &= (I - K_k H_k)\check{P}_k \end{aligned}$$

We can compare this with what we have learned in recursive least squares estimation, in which only the linear parameters are updated. We can also compare this with the dominant method for neural network training, we can say that neural network training based on EKF can select the step size automatically.

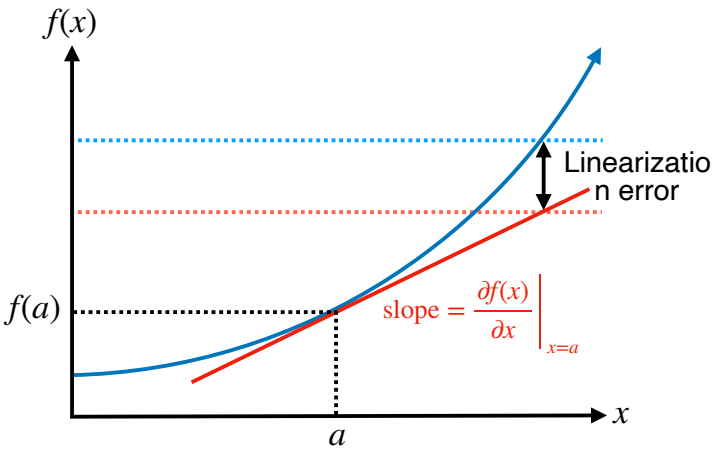


Figure 5.5 Linearization error induced by first-order linearization $f(x) \approx f(a) + \frac{\partial f(x)}{\partial x}|_{x=a}(x-a)$

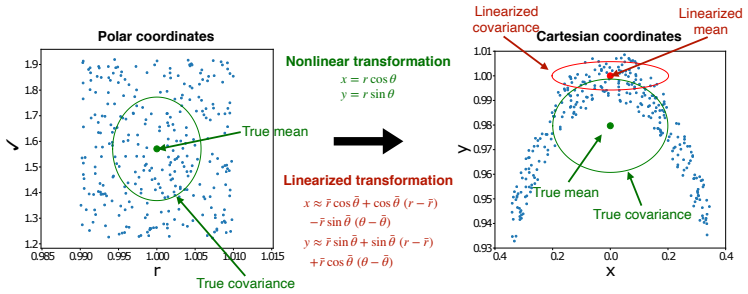


Figure 5.6 Mean and covariance of a random variable transformed by a nonlinear function

5.5.3 Unscented Kalman filter

The EKF works by linearizing the nonlinear motion and measurement models to update the mean and covariance of the state. The difference between the linear approximation and the nonlinear function is called linearization error, which is shown in Fig 5.5.

In general, linearization error depends on how nonlinear the function is and how far away from the operating point the linear approximation is being used. Large linearization error can introduce large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to sub-optimal performance and sometimes divergence of the filter.

Following presents an example, in which we can see how linearization error affects the mean and covariance of a random variable transformed by a nonlinear function.

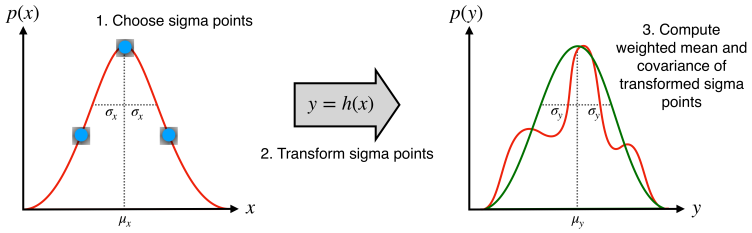


Figure 5.7 Principle of unscented transform.

Assume we have a nonlinear transformation,

$$x = r\cos\theta$$
$$y = r\sin\theta$$

The linearized transformation around $(\bar{r}, \bar{\theta})$ can be listed as,

$$x \approx \bar{r}\cos\bar{\theta}(r - \bar{r}) - \bar{r}\sin\bar{\theta}(\theta - \bar{\theta})$$
$$y \approx \bar{r}\sin\bar{\theta} + \sin\bar{\theta}(r - \bar{r}) + \bar{r}\cos\bar{\theta}(\theta - \bar{\theta})$$

The mean and covariance after the nonlinear transformation as well as linearized transformation are shown in Fig. 5.6. It is clear that the linearized transformation induces a large error in the mean and covariance of the random variable.

Hence, the unscented Kalman filter (UKF) is proposed, which approximates the probability distribution instead of the system dynamics.

In 1994 Jeffrey Uhlmann noted that the EKF takes a nonlinear function and partial distribution information of the state of a system but applies an approximation to the known function rather than to the imprecisely-known probability distribution. He suggested that a better approach would be to use the exact nonlinear function applied to an approximating probability distribution. Jeffrey Uhlmann explained that "unscented" was an arbitrary name that he adopted to avoid it being referred to as the "Uhlmann filter".

Basically, the unscented transformation is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. It uses the intuition (which also applies to the particle filter) that "It is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function– S. Julier, J. Uhlmann, and H. Durrant-Whyte (2000)"

The steps for approximating a Gaussian distribution can be shown in Fig. 5.7.

Assume we have a random variable $\mathbf{x} \sim \mathcal{N}(\mu, \sigma^2)$, and the random variable $\mathbf{y}$ is $\mathbf{x}$ undergoes a nonlinear transformation,

$$y = \sin(\mathbf{x})$$

Linearization based approximation can be described as,

$$\mathbf{y} = \sin(\mu) + \frac{\partial \sin(\mu)}{\partial \mu}(\mathbf{x} - \mu) + \dots$$

which gives

$$E(\mathbf{y}) \approx E(\sin(\mu) + \cos(\mu)(\mathbf{x} - \mu)) = \sin(\mu)$$
$$Cov(\mathbf{y}) \approx E[(\sin(\mu) + \cos(\mu)(\mathbf{x} - \mu) - \sin(\mu))^2] = \cos^2(\mu)\sigma^2$$

Alternatively, we can choose 3 sigma points as follows:

$$X_0 = \mu, X_1 = \mu + \sigma, X_2 = \mu - \sigma$$

We may now select some weights W_0, W_1, W_2 such that the original mean and covariance can be always recovered by

$$\mu = W_0 X_0 + W_1 X_1 + W_2 X_2$$
$$\sigma^2 = \sum_{i=0}^{2} W_i (X_i - \mu)^2$$

Then approximating the distribution of $\mathbf{y} = \sin(\mathbf{x})$ as follows,

$$\mu_y = \sum_{i=0}^{2} W_i \sin(X_i)$$
$$\sigma_y^2 = \sum_{i=0}^{2} W_i \left(\sin(X_i) - \mu_y\right)^2$$

Set $W_0 = 0, W_1 = W_2 = \frac{1}{2}$, we can get

$$\mu_y \approx \sin\mu\cos\sigma, \sigma_y^2 \approx \cos^2\mu\sin^2\sigma.$$

We can compare this with the first-order linearization, shown in Fig. 5.8. When $\mu = 1, \sigma = 0.1$, then mean and standard deviation of the original nonlinear output, linearized output, unscented transformation output are 0.8412, 0.8460, 0.8381 and 0.0568, 0.0578, 0.0545, respectively. If $\mu = 1, \sigma = 1$, then mean and standard deviation of the original nonlinear output, linearized output, unscented transformation output are 0.5853, 1.0188, 0.4777 and 0.4567, 0.5394, 0.3845, respectively.

It can be seen from Fig. 5.8 that when the points distributed in a wider range, the unscented transform gives a more accurate estimate of the mean and covariance of the resulted random variable.

Fig. 5.9 shows the unscented transform for the points shown in Fig. 5.6.

The principle of the unscented transform can be listed as follows.

1. For vectors $x \sim \mathcal{N}(m, P)$, the generalization of standard deviation σ is the Cholesky factor $L = \sqrt{P}$:
$$P = LL^T$$

2. The $(2n + 1)$ sigma points can be formed using **columns** of L:
$$X_0 = m$$
$$X_i = m + \sqrt{n + \lambda} L_i$$
$$X_{n+i} = m - \sqrt{n + \lambda} L_i$$
where $[]_i$ denotes the i-th column of the matrix.

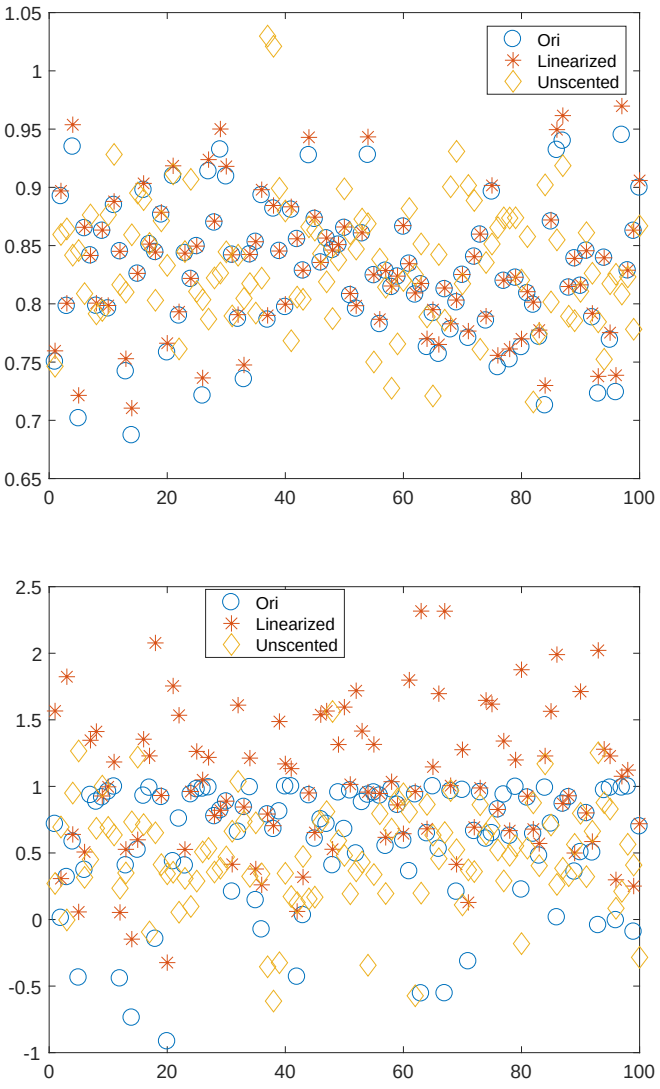


Figure 5.8 Comparison of the linearized transformation and unscented transformation.Top: $\mu = 1, \sigma = 0.1$. Bottom: $\mu = 1, \sigma = 1$.

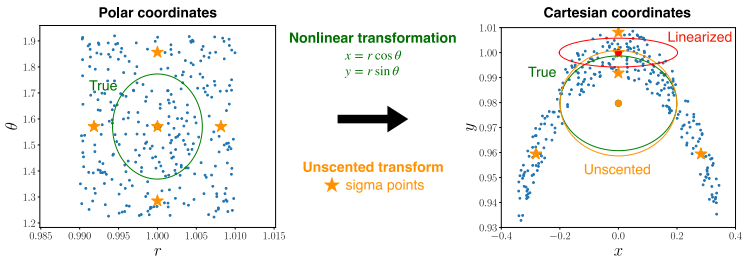


Figure 5.9 Comparison of linearized transform and unscented transform in 2 dimension.

3. For transformation $y = g(x)$ the approximation is:

$$E[g(x)] = \sum_{i=0}^{2n} W_i^{(m)} g(X_i)$$
$$Cov[g(x)] = \sum_{i=0}^{2n} W_i^{(c)} (g(X_i) - \mu_y)(g(X_i) - \mu_y)^T.$$

In the unscented transform, the parameters are set as follows. λ is a scaling parameter defined as $\lambda = \alpha^2(n + \kappa) - n$, in which α and κ determine the spread of the sigma points. The weights $W_i^{(m)}$ and $W_i^{(c)}$ are given as follows:

$$W_0^{(m)} = \frac{\lambda}{n+\lambda}, W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n+\lambda)}, i = 1, \ldots, 2n$$

where β can be used for incorporating priori information on the (non-Gaussian) distribution of $\mathbf{x}$.

For the nonlinear augmented function, we can also describe the unscented transform approximation. Assume we have a random variable $\mathbf{x}$ and $\mathbf{y} = g(\mathbf{x}) + \mathbf{q}$, in which $\mathbf{x} \sim \mathcal{N}(m, P)$ and $\mathbf{q} \sim \mathcal{N}(0, Q)$.

The unscented transform approximation to the joint distribution of x and $y = g(x) + q$ where $x \sim \mathcal{N}(m, P)$ and $q \sim \mathcal{N}(0, Q)$ is

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m \\ \mu \end{pmatrix}, \begin{pmatrix} P & C \\ C^T & S \end{pmatrix}\right)$$

The sub-matrices are formed as follows:

- Form the set of $(2n + 1)$ sigma points as follows:

$$\begin{aligned} X_0 &= m, \\ X_i &= m + \sqrt{n+\lambda} L_i, \\ X_{n+i} &= m - \sqrt{n+\lambda} L_i, i = 1, \ldots, n \end{aligned}$$

- Propagate the sigma points through $g(\cdot)$:

$$Y_i = g(X_i), i = 0, \ldots, 2n$$

- The sub-matrices are then given as:

$$\begin{aligned}
\mu &= \sum_{i=0}^{2n} W_i^{(m)} Y_i \\
S &= \sum_{i=0}^{2n} W_i^{(c)} (Y_i - \mu)(Y_i - \mu)^T + Q \\
C &= \sum_{i=0}^{2n} W_i^{(c)} (X_i - m)(Y_i - \mu)^T
\end{aligned}$$

Now we can derive the unscented Kalman filter.
Assume that the filtering distribution of previous step is Gaussian, i.e.,

$$p(x_{k-1}|y_{1:k-1}) \approx \mathcal{N}(x_{k-1}|m_{k-1}, P_{k-1})$$

The joint distribution of x_k and $x_k = f(x_{k-1}) + q_{k-1}$ can be approximated with UT as Gaussian

$$p(x_{k-1}, x_k|y_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix} | \begin{pmatrix} m_1' \\ m_2' \end{pmatrix}, \begin{pmatrix} P_{11}' & P_{12}' \\ (P_{12}')^T & P_{22}' \end{pmatrix}\right)$$

Select the sigma points X_i of $x_{k-1} \sim \mathcal{N}(m_{k-1}, P_{k-1})$ and compute the transformed sigma points as $\hat{X}_i = f(X_i)$. The expected values can now be expressed as

$$\begin{aligned}
m_1' &= m_{k-1} \\
m_2' &= \sum_i W_{i-1}^{(m)} \hat{X}_i
\end{aligned}$$

The blocks of covariance can be expressed as:

$$\begin{aligned}
P_{11}' &= P_{k-1} \\
P_{12}' &= \sum_i W_{i-1}^{(c)} (X_i - m_{k-1})(\hat{X}_i - m_2')^T \\
P_{22}' &= \sum_i W_{i-1}^{(c)} (\hat{X}_i - m_2')(\hat{X}_i - m_2')^T + Q_{k-1}
\end{aligned}$$

The prediction mean and covariance of x_k are then m_2' and P_{22}' and thus we get

$$\begin{aligned}
\breve{m}_k &= \sum_i W_{i-1}^{(m)} \hat{X}_i \\
\breve{P}_k &= \sum_i W_{i-1}^{(c)} (\hat{X}_i - m_k^-)(\hat{X}_i - m_k^-)^T + Q_{k-1}
\end{aligned}$$

For the joint distribution of x_k and $y_k = h(x_k) + r_k$ we similarly get

$$p(x_k, y_k|y_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} | \begin{pmatrix} m_1'' \\ m_2'' \end{pmatrix}, \begin{pmatrix} P_{11}'' & P_{12}'' \\ (P_{12}'')^T & P_{22}'' \end{pmatrix}\right)$$

If $\check{X}_i$ are the sigma points of $x_k \sim \mathcal{N}(\check{m}_k, \check{P}_k)$ and $\hat{Y}_i = f(\check{X}_i)$, we get

$$\begin{aligned} m_1'' &= \check{m}_k \\ m_2'' &= \sum_i W_i^{(m)} \hat{Y}_i \\ P_{11}'' &= \check{P}_k \\ P_{12}'' &= \sum_i W_i^{(c)} (\check{X}_i - \check{m}_k)(\hat{Y}_i - m_2'')^T \\ P_{22}'' &= \sum_i W_i^{(c)} (\hat{Y}_i - m_2'')(\hat{Y}_i - m_2'')^T + R_k \end{aligned}$$

Recall that if

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} A & C \\ C^T & B \end{pmatrix} \right)$$

then

$$x|y \sim \mathcal{N}(a + CB^{-1}(y-b), A - CB^{-1}C^T)$$

Thus we get the conditional mean and covariance

$$\begin{aligned} \hat{m}_k &= \check{m}_k + P_{12}''(P_{22}'')^{-1}(y_k - m_{22}'') \\ \hat{P}_k &= \check{P}_k - P_{12}''(P_{22}'')^{-1}(P_{12}'')^T \end{aligned}$$

The unscented Kalman Filter (UKF) can then be shown in the following algorithm.

Algorithm 5.5 *Unscented Kalman filter.*

Initialization

Prediction

1. Form the matrix of sigma points:

$$X_{k-1} = [\ \hat{x}_{k-1} \quad \cdots \quad \hat{x}_{k-1}\ ] + \sqrt{n+\lambda}[\ 0 \quad \sqrt{\hat{P}_{k-1}} \quad -\sqrt{\hat{P}_{k-1}}\ ]$$

2. Propagate the sigma points through the dynamic model.

$$\hat{X}_{k,i} = f(X_{k-1,i}), i = 1, \ldots, 2n+1$$

3. Compute the predicted mean and covariance

$$\begin{aligned} \check{x}_k &= \sum_i W_{i-1}^{(m)} \hat{X}_{k,i} \\ \check{P}_k &= \sum_i W_{i-1}^{(c)} (\hat{X}_{k,i} - \check{x}_k)(\hat{X}_{k,i} - \check{x}_k)^T + Q_{k-1} \end{aligned}$$

Update

4. Form the matrix of sigma points:

$$\check{X}_k = [\ \check{x}_k \quad \cdots \quad \check{x}_k\ ] + \sqrt{n+\lambda}[\ 0 \quad \sqrt{\check{P}_k} \quad -\sqrt{\check{P}_k}\ ]$$

5. Propagate the sigma points through the measurement model:

$$\hat{Y}_{k,i} = h(\check{X}_{k,i}), i = 1, \ldots, 2n+1$$

6. Compute the following terms:

$$\mu_k = \sum_i W_{i-1}^{(m)} \hat{Y}_{k,i}, S_k = \sum_i W_{i-1}^{(c)} (\hat{Y}_{k,i} - \mu_k)(\hat{Y}_{k,i} - \mu_k)^T + R_k$$

$$C_k = \sum_i W_{i-1}^{(c)} (\check{X}_{k,i} - \check{x}_k)(\hat{Y}_{k,i} - \mu_k)$$

7. Compute the filter gain K_k and the filtered state mean m_k and covariance $\hat{P}_k$, conditional to the measurement y_k:

$$K_k = C_k S_k^{-1}, \hat{x}_k = \check{x}_k + K_k(y_k - \mu_k)$$

$$\hat{P}_k = \check{P}_k - K_k S_k K_k^T$$

Example 5.13 UKF: Short example solution

Fig. 5.4 shows the estimation problem using a nonlinear measurement. The process model has been linearized and discretized as,

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ &\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}\end{aligned}$$

The nonlinear measurement equation is,

$$y_k = \phi_k = h(p_k, v_k) = \arctan\left(\frac{S}{D - p_k}\right) + v_k$$

The process noise and measurement noise are assumed to be white noise, i.e.,

$$v_k \sim \mathcal{N}(0, 0.05), \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, 0.1\mathbf{1}_{2\times 2})$$

The initial state is

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

The sample instant is $\Delta t = 0.5$s, the initial input is $u_0 = -2\text{m/s}^2$, the measurements available are

$$y_1 = 30\text{deg}, S = 20\text{m}, D = 40\text{m}$$

The prediction is performed.

1. $n = 2$, choose $\lambda = 1$
2. $\sqrt{\hat{P}_0} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$
3. choose 5 sigma points
   - $\hat{x}_0^{(0)} = \hat{x}_0, \hat{x}_0^{(i)} = \hat{x}_0 + \sqrt{3}[\sqrt{\hat{P}_0}]_i, i = 1, 2$
   - $\hat{x}_0^{(i+2)} = \hat{x}_0 - \sqrt{3}[\sqrt{\hat{P}_0}]_i, i = 1, 2$
   - $\hat{x}_0^{(0)} = [0, 5]^T, \hat{x}_0^{(1)} = [0.2, 5]^T, \hat{x}_0^{(2)} = [0, 6.7]^T, \hat{x}_0^{(3)} = [-0.2, 5]^T, \hat{x}_0^{(4)} = [0, 3.3]^T$
4. $\check{x}_1^{(i)} = f_0(\hat{x}_0^{(i)}, u_0, 0), i = 0, 1, \ldots, 4$
5. $\check{x}_1^{(0)} = [2.5, 4]^T, \check{x}_1^{(1)} = [2.7, 4]^T, \check{x}_1^{(2)} = [3.4, 5.7]^T, \check{x}_1^{(3)} = [2.3, 4]^T, \check{x}_1^{(4)} = [1.6, 2.3]^T$
6. $W_0^{(m)} = W_0^{(c)} = 1/3, W_i^{(m)} = W_i^{(c)} = 1/6, i = 1, \ldots, 4$
7. $\check{x}_1 = \sum_{i=0}^{4} W_i \check{x}_1^{(i)} = [2.5, 4]^T$
8. $\check{P}_k = \sum_{i=0}^{4} W_i(\check{x}_k^i - \check{x}_k)(\check{x}_k^i - \check{x}_k)^T + Q_{k-1} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$

Then we do the correction as follows,

1. $\sqrt{\check{P}_1} = \begin{bmatrix} 0.51 & 0 \\ 0.98 & 0.20 \end{bmatrix}$

2. choose 5 sigma points
   - $\check{x}_1^{(0)} = \check{x}_1, \check{x}_1^{(i)} = \check{x}_1 + \sqrt{3}[\sqrt{\check{P}_1}]_i, i = 1, 2$
   - $\check{x}_1^{(i+2)} = \check{x}_1 - \sqrt{3}[\sqrt{\check{P}_1}]_i, i = 1, 2$
   - $\check{x}_1^{(0)} = [2.5, 4]^T, \check{x}_1^{(1)} = [3.54, 5.44]^T, \check{x}_1^{(2)} = [2.5, 5.10]^T, \check{x}_1^{(3)} = [1.46, 2.56]^T, \check{x}_1^{(4)} = [2.5, 2.90]^T$
3. the output $\hat{y}_1^{(i)} = h_1(\check{x}_1^{(i)}, 0), i = 0, \ldots, 2n$
   - $\hat{y}_1^{(0)} = 28.1, \hat{y}_1^{(1)} = 28.7, \hat{y}_1^{(2)} = 28.1, \hat{y}_1^{(3)} = 27.4, \hat{y}_1^{(4)} = 28.1$
4. $\hat{y}_1 = \sum_{i=0}^{2n} W_i^{(m)} \hat{y}_1^{(i)} = 28.1$
5. $S_1 = \sum_{i=0}^{2n} W_i^{(c)} (\hat{y}_k^{(i)} - \hat{y}_k)(\hat{y}_k^{(i)} - \hat{y}_k)^T + R_k = 0.16$
6. $C_1 = \sum_{i=0}^{2n} W_c^{(i)} (\check{x}_k^{(i)} - \check{x}_k)(\hat{y}_k^{(i)} - \hat{y_k})^T = [0.23, 0.32]^T$
7. $K_1 = C_1 S_1^{-1} = [1.47, 2.05]^T$
8. $\hat{x}_1 = \check{x}_1 + K_1(y_1 - \hat{y}_1) = [2.55, 4.07]^T$
9. $\hat{P}_1 = \check{P}_1 - K_1 S_1 K_1^T = \begin{bmatrix} 0.0143 & 0.0178 \\ 0.0178 & 0.4276 \end{bmatrix}$

We can compare the EKF and UKF. In EKF, we perform the local approximation, while larger area approximation is employed in UKF. Besides, in EKF, the differentiability of F and h are required and we have closed form derivatives or expectations, which are not included in UKF. For EKF, we perform a first order approximation of the nonlinear dynamics, and UKF captures higher order moments of distribution (up to 3rd order).

However, there are also disadvantages of UKF. First, it is based on a small set of trial points, and thus not a truly global approximation. Besides, when the covariances is singular or nearly singular, i.e., with nearly deterministic systems, the UKF does not work well. It also requires more computations than EKF, e.g., Cholesky factorizations on every step and it can **only** be applied to models **driven by Gaussian noises**.

Therefore, the particle filter is used in some cases. In a linear system with Gaussian noise, the Kalman filter is optimal. In a system that is **nonlinear**, the Kalman filter can be used for state estimation, but the **particle filter** may give **better** results at the price of **additional computational effort.** In a system that has **non-Gaussian noise**, the Kalman filter is the **optimal linear filter,** but again the **particle filter may perform better.** The UKF provides a **balance** between the **low computational effort** of the Kalman filter and the **high performance** of the particle filter.

The particle filter has some similarities with the UKF in that it transforms **a set of points** via known nonlinear equations and combines the results to estimate the mean and covariance of the state. However, in the particle filter the points are chosen **randomly,** whereas in the UKF the points are chosen on the basis of a specific algorithm (**unscented transform**). In general, the number of points used in a particle filter needs to be much **greater** than the number of points in a

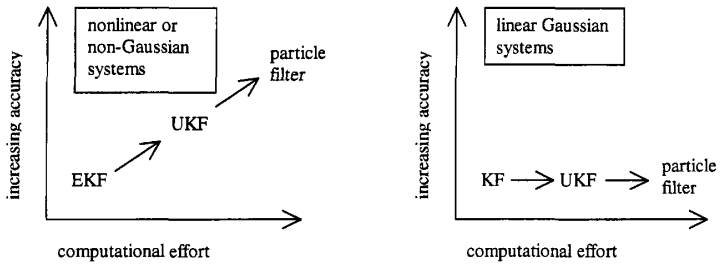


Figure 5.10 Comparisons of EKF, UKF and particle filter.

UKF. In UKF, the estimation error is not guaranteed to converge to zero in any sense, but the **estimation error in a particle filter does converge to zero as the number of particles (and hence the computational effort) approaches infinity.** Fig. 5.10 gives an illustration of these 3 kinds of filters.