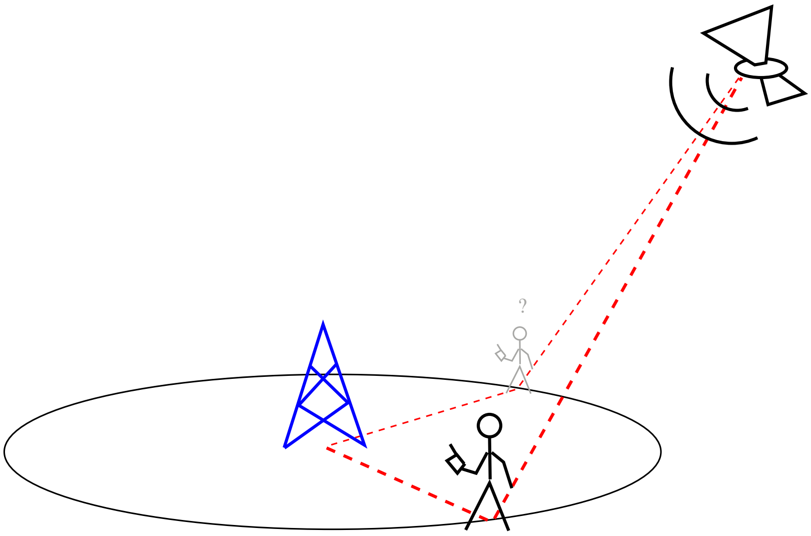
TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Mathematics

MAT-45806 Mathematics for Positioning & MAT-45807 Mathematics for Positioning

Simo Ali-Löytty
Jussi Collin
Niilo Sirola

2010



Preface

Positioning techniques and algorithms have been studied for some years at the Tampere University of Technology within several research groups. The objective of this course hand-out has been to collect together the most important algorithms and mathematical tools used in positioning including examples and starting from the basics. We do not go into details of specialized techniques and equipment, but after this course student should be able to solve application dependent problems without having to "re-invent the wheel" again and again.

This hand-out and course provide a strong basis for the course *TKT-2546 Methods for Positioning*. During the previous years courses *MAT-45806 Mathematics for Positioning* and *TKT-2546 Methods for Positioning* had a common hand-out. For practical reasons, the earlier hand-out has been divided into two parts so that both courses now have their own hand-out. Still the courses in question are tightly connected and it is strongly recommended to take both courses the same school year.

Prerequisites are first-year engineering mathematics and basics of probability. Additionally, the course *TKT-2536 Introduction to Satellite Positioning* is a useful but not compulsory prerequisite. There is no official course text book in addition to this hand-out, mostly because the authors have not managed to find a single book to cover all the material on the level of abstraction we need. The arsenal of positioning computation methods is collected from different areas of mathematics and engineering sciences, and there are often discipline and interpretation differences between them, so we have tried to use common notations and represent connections between different ways of thinking as best as we could.

The homepage of the course is `http://math.tut.fi/courses/MAT-45806/` which contains additional information about the course and if necessary errata of this hand-out.

The authors would like to thank Sami Tiainen for the initial translation of the manuscript, and professor Robert Piché, Helena Leppäkoski, Henri Pesonen, Hanna Sairo, Martti Kirkko-Jaakkola and others who have contributed to the hand-out. The sections excluded from this year's implementation have been marked with an asterisk (*).

Tampere, September 7, 2010

the authors

Contents

Chapter 1

Preliminaries

Simo Ali-Löytty

In this chapter, we review the mathematical concepts and tools necessary for digesting the rest of the material. The reader is assumed to be familiar with most of the material, and is encouraged to look up some of the cited references if this is not the case. Mathematically speaking, positioning is finding the "best"* estimator for state $x \in \mathbb{R}^n$ using the equation

$$y = f(x) + \varepsilon. \tag{1.1}$$

Here y is a vector that contains measurements and ε is called the error (unknown). An important special case of equation (1.1) is when function f is linear; this special case is handled in Section 1.2. Often the error term ε and possibly also the state x are modeled as random variables. Because of this we review some probability theory in section 1.3. At the end of the chapter we handle coordinate systems in 1.4 and moving coordinate systems in 1.5, which are a natural part of positioning. First of all we give short introduction to linear algebra in 1.1.

1.1 Linear Algebra

Here we give short introduction to linear algebra, which is covered in detail in, for example, [24, 25]. The following lists some properties of a matrix $\mathrm{A} \in \mathbb{R}^{m \times n}$.

- The null space of matrix A is $\mathcal{N}(\mathrm{A}) = \{x \in \mathbb{R}^n | \mathrm{A}x = 0\}$.
- The column space of matrix A is $\mathcal{R}(\mathrm{A}) = \{y \in \mathbb{R}^m | y = \mathrm{A}x \text{ for some } x \in \mathbb{R}^n\}$.
- $\dim(\mathcal{R}(\mathrm{A}^T)) = \dim(\mathcal{R}(\mathrm{A})) = \mathrm{rank}(\mathrm{A})$.

*It is not at all obvious what the word "best" means. One of the tasks of the mathematical modeller is to define a criterion to compare estimators. Often this criterion is a so-called cost function, for instance in the form $\|y - f(\hat{x})\|^2$ or $\mathrm{E}(\|x - \hat{x}\|^2)$.

- $\dim(\mathcal{N}(\mathrm{A})) + \mathrm{rank}(\mathrm{A}) = n$ (The Rank Theorem).
- $\mathrm{A} \in \mathbb{R}^{n\times n}$ is orthogonal if $\mathrm{A}^T\mathrm{A} = \mathrm{I}$.
- $\mathrm{A} \in \mathbb{R}^{n\times n}$ is symmetric if $\mathrm{A}^T = \mathrm{A}$.
- $\mathrm{A} \in \mathbb{R}^{n\times n}$ is idempotent if $\mathrm{AA} = \mathrm{A}$.
- If $\mathrm{A} \in \mathbb{R}^{n\times n}$ and $\mathrm{A}x = \lambda x$, where $x \neq 0$, then λ is an eigenvalue of the matrix A and the corresponding eigenvector is x.

Example 1. *Let* $\mathrm{A} \in \mathbb{R}^{n\times n}$ *be idempotent and* λ *be an arbitrary eigenvalue of* A *with corresponding eigenvector* x*. Now*

$$\lambda x = \mathrm{A}x = \mathrm{AA}x = \lambda^2 x \Longrightarrow \lambda = 1 \text{ or } \lambda = 0,$$

so eigenvalues of an idempotent matrix A *are all either zero or one.*

Let matrix $\mathrm{A} \in \mathbb{R}^{n\times n}$ be symmetric. Then

- Its eigenvalues are real.
- A is positive definite, denoted $\mathrm{A} > 0$, if $x^T\mathrm{A}x > 0$ for all $x \neq 0$.
- A is positive semi-definite, denoting $\mathrm{A} \geq 0$, if $x^T\mathrm{A}x \geq 0$ for all x.
- $\mathrm{A} > \mathrm{B}$ is interpreted as $\mathrm{A} - \mathrm{B} > 0$, and $\mathrm{A} \geq \mathrm{B}$ as $\mathrm{A} - \mathrm{B} \geq 0$.

Example 2. *If* $\mathrm{A} \geq 0$ *and* λ *is an arbitrary eigenvalue of matrix* A *corresponding to eigenvector* x*, then*

$$\lambda x = \mathrm{A}x \Longrightarrow \lambda \|x\|^2 = x^T\mathrm{A}x \geq 0 \Longrightarrow \lambda \geq 0,$$

so all eigenvalues of a positive semi-definite matrix are non-negative.

Theorem 1 (Schur decomposition)**.** *Let* $\mathrm{A} \in \mathbb{R}^{n\times n}$ *be symmetric. Then there is an orthogonal matrix* Q *and a diagonal matrix* Λ *such that*

$$\mathrm{A} = \mathrm{Q}\Lambda\mathrm{Q}^T. \tag{1.2}$$

Because Q is orthogonal, the inverse matrix of Q is Q^T. From Eq (1.2) follows that $\mathrm{AQ} = \mathrm{Q}\Lambda$, so diagonal elements of diagonal matrix Λ are eigenvalues of matrix A and columns of matrix Q are normalized eigenvectors corresponding to diagonal elements of Λ.

Definition 1 (Square root of matrix $\mathrm{A} \geq 0$)**.** *Using Schur decomposition we get*

$$\mathrm{A} = \mathrm{Q}\lceil \lambda_1, \ldots \lambda_n \rfloor \mathrm{Q}^T.$$

where $\lambda_i \geq 0$ *for all* $i \in \{1, \ldots, n\}$ *(Example 2). We define the square root*[*] *of matrix* A *as*

$$\mathrm{A}^{\frac{1}{2}} = \mathrm{Q}\lceil \sqrt{\lambda_1}, \ldots \sqrt{\lambda_n} \rfloor \mathrm{Q}^T.$$

[*] Usually matrix B is called a square root of matrix A if $\mathrm{A} = \mathrm{BB}$. In some books also matrix C is called a square root of matrix A if $\mathrm{A} = \mathrm{CC}^T$. Neither one of the above matrices (B or C) is unique. Notice that our definition of the matrix $A^{\frac{1}{2}}$ fulfills both definitions.

1.2 Overdetermined linear system of equations

Consider the linear system of equations

$$\mathrm{A}x = y, \tag{1.3}$$

where $\mathrm{A} \in \mathbb{R}^{m\times n}$ and $y \in \mathbb{R}^m$. Assume that the columns of the matrix A are linearly independent i.e.

$$\mathrm{A}x = 0 \text{ only if } x = 0.$$

If $m > n$, the system is called overdetermined, which can happen when for instance there are more measurements than unknowns. Generally the overdetermined equation has no exact solution, in which case one approach is to search for the least squares solution

$$\hat{x} = \mathrm{argmin}_x \|y - \mathrm{A}x\|^2. \tag{1.4}$$

Because (see Exercise 1.3)

$$\|y - \mathrm{A}x\|^2 = \|\mathrm{A}((\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T y - x)\|^2 + \|y - \mathrm{A}(\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T y\|^2, \tag{1.5}$$

the solution of Equation (1.4) is

$$\hat{x} = (\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T y. \tag{1.6}$$

If the system $\mathrm{A}x = y$ is to be solved with Matlab software, it is recommended to use backslash-command `A\y` instead of computing the inverse of $\mathrm{A}^T\mathrm{A}$ explicitly.

Example 3. *Let y be measurement and let x_0 be an initial guess. Compute the estimate $\hat{x}$ that minimizes $\|y - \mathrm{H}x\|^2 + \|x_0 - x\|^2$.*

$$\begin{aligned}
\hat{x} &= \mathrm{argmin}_x(\|y - \mathrm{H}x\|^2 + \|x_0 - x\|^2) \\
&= \mathrm{argmin}_x \left(\left\| \begin{bmatrix} y \\ x_0 \end{bmatrix} - \begin{bmatrix} \mathrm{H} \\ \mathrm{I} \end{bmatrix} x \right\|^2 \right) \\
&\overset{(1.6)}{=} \left(\mathrm{H}^T\mathrm{H} + \mathrm{I}\right)^{-1} \left(\mathrm{H}^T y + x_0\right).
\end{aligned}$$

Example 4. *We obtain the following two-dimensional $\begin{bmatrix} x_i \\ y_i \end{bmatrix}$ measurements*

$$\left\{ \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \textit{ and } \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right\}$$

Estimate the parameters of equation $y = ax + b$ so that the error $\sum_{i=1}^{5} \|y_i - (ax_i + b)\|^2$ is as small as possible.

Now, denoting $z = \begin{bmatrix} a \\ b \end{bmatrix}$, *we have*

$$\sum_{i=1}^{5} \|y_i - (ax_i + b)\|^2 = \left\| \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} -1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = \|y - \mathrm{A}z\|^2$$

It follows from Eq (1.6) that the solution is

$$\hat{z} = (\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T y = \begin{bmatrix} 6 & 2 \\ 2 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \frac{1}{26} \begin{bmatrix} 5 & -2 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \begin{bmatrix} \frac{17}{26} \\ \frac{14}{26} \end{bmatrix}.$$

So the estimated equation is $y = \frac{17}{26}x + \frac{7}{13}$. *See Fig. 1.1.*

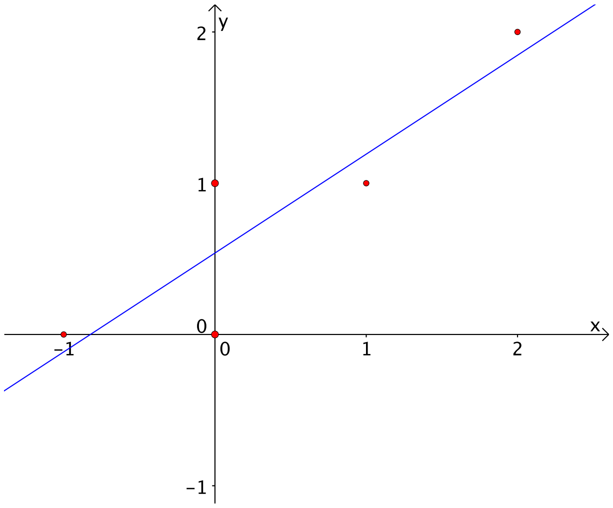


Figure 1.1: Example 4. Here red dots are two-dimensional measurements and straight line is estimated equation.

1.3 Probability theory

In this section, we briefly review the basics of probability theory the focus being on the normal distribution and conditional probability. For a proper definition of a random variable, see [33, 11]. In this section, a random variable is denoted in boldface, e.g. **x**, and can be either vector or scalar. (In later sections, it should be clear from the context whether boldface is used to refer to a vector or to a random variable.) The density function of random variable **x** is denoted as $f_{\mathbf{x}}(x)$ or $p_{\mathbf{x}}(x)$ and cumulative distribution function as $F_{\mathbf{x}}(x)$. The independence of random variables is a central concept: it is needed for instance in filtering in Chapter 3.

Definition 2 (Independence)**.** *Random variables* $\mathbf{x}_1,\ldots,\mathbf{x}_k$ *are independent if*

$$F_{\mathbf{x}_1,\ldots,\mathbf{x}_k}(x_1,\ldots,x_k)=\prod_{i=1}^{k}F_{\mathbf{x}_i}(x_i),\quad \forall x_1,\ldots,x_k\in\mathbb{R}^n.$$

Theorem 2. *Random variables* $\mathbf{x}_1,\ldots,\mathbf{x}_k$ *are independent if and only if*

$$f_{\mathbf{x}_1,\ldots,\mathbf{x}_k}(x_1,\ldots,x_k)=\prod_{i=1}^{k}f_{\mathbf{x}_i}(x_i),\quad \forall x_1,\ldots,x_k\in\mathbb{R}^n.$$

Example 5. *Let the density function of a two-dimensional random variable* $\mathbf{x}=[\mathbf{x}_1,\mathbf{x}_2]$ *be*

$$f_{\mathbf{x}_1,\mathbf{x}_2}(x_1,x_2)=\begin{cases}\frac{1}{\pi}, & \textit{when}\quad x_1^2+x_2^2\le 1\\ 0, & \textit{otherwise}\end{cases}.$$

Are the random variables $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *independent?*

Now the density functions of the marginal distributions are

$$f_{\mathbf{x}_1}(x_1)=\int_{-\infty}^{\infty}f_{\mathbf{x}_1,\mathbf{x}_2}(x_1,x_2)dx_2=\begin{cases}\frac{2}{\pi}\sqrt{1-x_1^2}, & \textit{when}\quad |x_1|\le 1\\ 0, & \textit{otherwise}\end{cases}$$

and

$$f_{\mathbf{x}_2}(x_2)=\int_{-\infty}^{\infty}f_{\mathbf{x}_1,\mathbf{x}_2}(x_1,x_2)dx_1=\begin{cases}\frac{2}{\pi}\sqrt{1-x_2^2}, & \textit{when}\quad |x_2|\le 1\\ 0, & \textit{otherwise}\end{cases}.$$

Because

$$f_{\mathbf{x}_1}(0)f_{\mathbf{x}_2}(0)=\frac{4}{\pi^2}\neq\frac{1}{\pi}=f_{\mathbf{x}_1,\mathbf{x}_2}(0,0),$$

it follows from Theorem 2 that the random variables $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are not independent.*

Definition 3 (Expectation)**.** *Assume that* $\mathbf{x}$ *is a continuous random variable and that the integral*

$$\int_{-\infty}^{\infty}|g(u)|f_{\mathbf{x}}(u)du$$

converges. Then the expectation of the random variable $g(\mathbf{x})$ *is*

$$\mathrm{E}(g(\mathbf{x}))=\int_{-\infty}^{\infty}g(u)f_{\mathbf{x}}(u)du.$$

The next theorem follows from the linearity of integration.

Theorem 3. *If* $\mathrm{A}\in\mathbb{R}^{p\times n}$ *and* $b\in\mathbb{R}^p$ *are constant, then*

$$\mathrm{E}(\mathrm{A}\mathbf{x}+b)=\mathrm{A}\,\mathrm{E}(\mathbf{x})+b.$$

The mean $\mu_{\mathbf{x}} = \mathrm{E}(\mathbf{x})$ and the covariance matrix $\Sigma_{\mathbf{x}} = \mathrm{V}(\mathbf{x}) = \mathrm{E}((\mathbf{x}-\mu_{\mathbf{x}})(\mathbf{x}-\mu_{\mathbf{x}})^T)$, of random variable $\mathbf{x}$ are important expectation values.

The correlation matrix of random variable $\mathbf{x}$ is

$$\mathrm{R}_{\mathbf{x}} = \mathrm{D}\Sigma_{\mathbf{x}}\mathrm{D}, \tag{1.7}$$

where D is a diagonal matrix whose diagonal contains the square roots of the reciprocals of the diagonal elements of the covariance matrix $\Sigma_{\mathbf{x}}$.

Example 6. *Let $\mathbf{x}_i$ be independent identically distributed continous random variables, so that $\mathrm{E}(\mathbf{x}_i) = \mu$ and $\mathrm{V}(\mathbf{x}_i) = \Sigma$. Define random variable $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. Now*

$$\begin{aligned}
\mathrm{E}(\bar{\mathbf{x}}) &= \int \left(\frac{1}{n}\sum_{i=1}^n x_i\right) \prod_{k=1}^n f_{\mathbf{x}_k}(x_k)dx_1\cdots dx_n = \frac{1}{n}\sum_{i=1}^n \int x_i f_{\mathbf{x}_i}(x_i)dx_i = \mu \quad \textit{and} \\
\mathrm{V}(\bar{\mathbf{x}}) &= \int \left(\frac{1}{n}\sum_{i=1}^n (x_i-\mu)\right)\left(\frac{1}{n}\sum_{j=1}^n (x_j-\mu)\right)^T \prod_{k=1}^n f_{\mathbf{x}_k}(x_k)dx_1\cdots dx_n \\
&= \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n \int (x_i-\mu)(x_j-\mu)^T \prod_{k=1}^n f_{\mathbf{x}_k}(x_k)dx_1\cdots dx_n = \frac{1}{n}\Sigma.
\end{aligned}$$

The random variable $\bar{\mathbf{x}}$ is an unbiased estimator of the parameter μ because $\mathrm{E}(\bar{\mathbf{x}}-\mu) = 0$. Furthermore, we see that $\mathrm{V}(\bar{\mathbf{x}}) = \frac{1}{n}\Sigma \to 0$, when $n \to \infty$. Thus, for all $\varepsilon > 0$, the probability $\mathrm{P}(\|\bar{\mathbf{x}}-\mu\| > \varepsilon) \to 0$ when $n \to \infty$. Therefore, the estimator $\bar{\mathbf{x}}$ is also consistent.

1.3.1 Normal distribution

We assume that the reader is familiar with the one-dimensional normal distribution. We define n-dimensional normal distribution as follows.

Definition 4 (Normal distribution)**.** *An n-dimensional random variable $\mathbf{x}$ is normal distributed with parameters $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n\times n}$, $\Sigma \geq 0$, denoted $\mathbf{x} \sim \mathrm{N}(\mu,\Sigma)$ or $\mathbf{x} \sim \mathrm{N}_n(\mu,\Sigma)$, if the random variable $a^T\mathbf{x}$ is one-dimensional normal distribution or constant for all vectors $a \in \mathbb{R}^n$.*

Let $\mathbf{x} \sim \mathrm{N}(\mu,\Sigma)$. Then the parameters are the mean $\mu = \mathrm{E}(\mathbf{x})$ and the covariance $\Sigma = \mathrm{V}(\mathbf{x})$ of the random variable $\mathbf{x}$. If Σ is positive definite, then the density function of random variable $\mathbf{x}$ is

$$f_{\mathbf{x}}(x) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det(\Sigma)}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{1.8}$$

If Σ positive semi-definite but singular, then we call the distribution a singular normal distribution. For instance, in Chapter 3 the state model error $\mathbf{w}_{k-1}$ (3.1) can quite possibly follow singular normal distribution. This happens for instance when we know that the user is stationary, then the position state model is simply $\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{w}_{k-1}$, where $\mathbf{w}_{k-1} \sim \mathrm{N}(0,0)$.

Example 7 (Visualizing the normal distribution)**.** *Figure 1.2 shows the density function of the random variable*

$$\mathbf{x} \sim \mathrm{N}(0, \Sigma), \tag{1.9}$$

where $\Sigma = \begin{bmatrix} 16 & 24 \\ 24 & 52 \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 8^2 & 0 \\ 0 & 2^2 \end{bmatrix} \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}.$

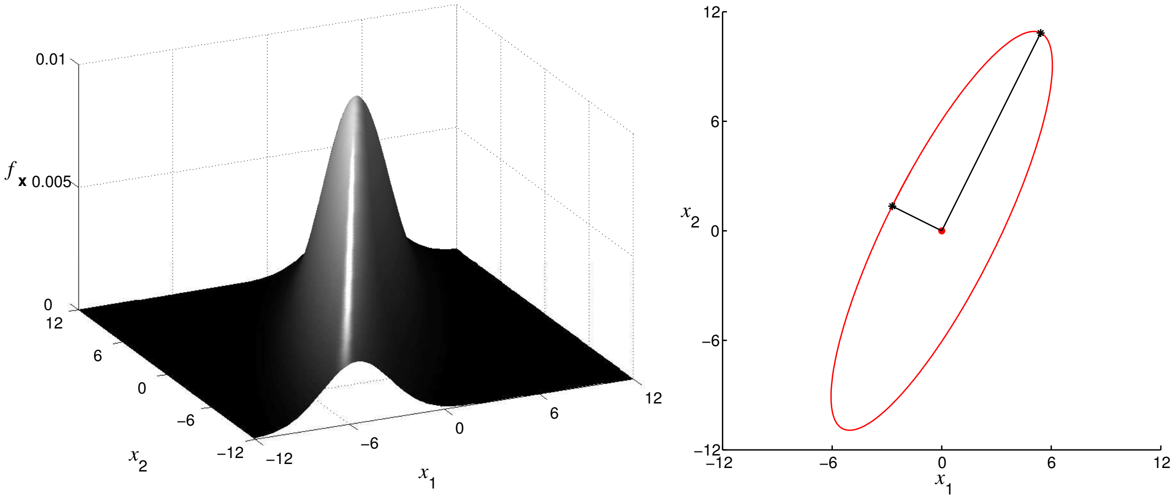


Figure 1.2: Two visualizations of the density function of normal distribution (1.9) in the area $|x_1| \leq 12, |x_2| \leq 12$

The picture on the left shows density function (1.8) values. The picture on the right shows the level curve of the density function that contains 68% *of the probability mass, see Exercise 1.10. The picture on the right also contains line segments which are directed along the eigenvectors of* Σ *and whose lengths are proportional to the square roots of corresponding eigenvalues.*

Theorem 4. *If* $\mathbf{x} \sim \mathrm{N}(\mu, \Sigma)$ *and* $\mathbf{y} = \mathrm{A}\mathbf{x} + b$ *then* $\mathbf{y} \sim \mathrm{N}(\mathrm{A}\mu + b, \mathrm{A}\Sigma\mathrm{A}^T)$.

Example 8 (Generate a sample from the normal distribution)**.** *If* $\mathbf{u} \sim \mathrm{N}_n(0, \mathrm{I})$ *then random variable* $\mathbf{x} = \Sigma^{\frac{1}{2}}\mathbf{u} + \mu$ *has normal distribution with parameters* μ *and* Σ *(Theorem 4) that is*

$$\mathbf{x} \sim \mathrm{N}_n(\mu, \Sigma).$$

If we can generate a sample from distribution $\mathrm{N}_n(0, \mathrm{I})$ *then we can generate a sample from arbitrary normal distribution* $\mathrm{N}_n(\mu, \Sigma)$. *For example, in Matlab we can use the commands:*

```
x1=sqrtm(Sigma)*randn(n,1)+mu or
x2=chol(Sigma,'lower')*randn(n,1)+mu
```

Even though both random variables x_1 *and* x_2 *have normal distribution with parameters* μ *and* Σ *they are not the same random variable because matrices* `sqrtm(Sigma)` *and* `chol(Sigma,'lower')` *are unequal (see also Theorem 4 and footnote on the page 5). Matlab Statistics Toolbox has also the command* `mvnrnd` *for generating samples from an arbitrary normal distribution. This command uses the Cholesky decomposition.*

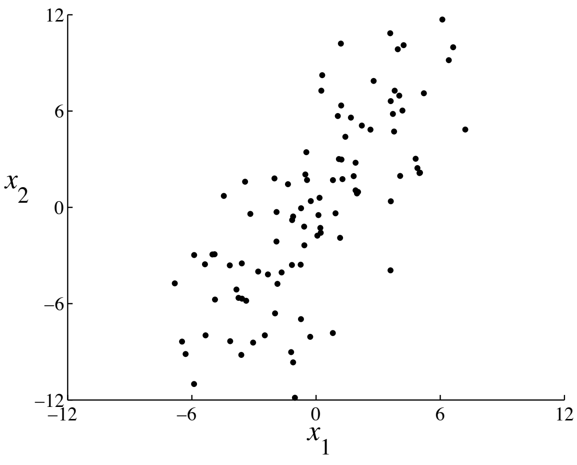


Figure 1.3: 100 samples from normal distribution (1.9).

Figure 1.3 shows one hundred samples (with some samples outside the figure) from normal distribution (1.9). Compare Figure 1.3 and Figure 1.2.

Theorem 5. *Let* $\mathbf{x} \sim \mathrm{N}(\mu, \Sigma)$. *Then* $\mathrm{A}\mathbf{x}$ *and* $\mathrm{B}\mathbf{x}$ *are independent if and only if* $\mathrm{A}\Sigma\mathrm{B}^T = 0$.

Example 9. *Let*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \right) \quad \textit{and} \quad \mathbf{z} = \mathbf{x} - \Sigma_{xy}\Sigma_{yy}^{-1}\mathbf{y}.$$

Show that random variables $\mathbf{z}$ *and* $\mathbf{y}$ *are independent.*

Note that

$$\begin{aligned} &\begin{bmatrix} \mathrm{I} & -\Sigma_{xy}\Sigma_{yy}^{-1} \end{bmatrix} \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} & \Sigma_{xy} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yy} \end{bmatrix} \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix} = 0. \end{aligned}$$

Thus, the random variables $\mathbf{z}$ *and* $\mathbf{y} = \begin{bmatrix} 0 & \mathrm{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$ *are independent (Theorem 5).*

1.3.2 χ^2 distribution *

Definition 5. *Random variable* $\mathbf{z}$ *is* χ^2 *distributed with n degrees of freedom, denoted* $\mathbf{z} \sim \chi^2(n)$, *if it has the same distribution as random variable* $\mathbf{x}^T\mathbf{x}$, *where* $\mathbf{x} \sim \mathrm{N}_n(0, \mathrm{I})$.

Example 10. *Let* $\mathbf{x} \sim \mathrm{N}_n(\mu, \Sigma)$ *be a non-singular normal random variable. Now* $\Sigma^{-\frac{1}{2}}(\mathbf{x} - \mu) \sim \mathrm{N}(0, \mathrm{I})$, *and so*

$$(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \sim \chi^2(n).$$

Definition 6. *Random variable $\mathbf{z}$ is noncentral χ^2 distributed with n degrees of freedom and noncentral parameter λ, denoted $\mathbf{z} \sim \chi^2(n,\lambda)$, if it has the same distribution as the random variable $\mathbf{x}^T\mathbf{x}$, where $\mathbf{x} \sim \mathrm{N}_n(\mu,\mathrm{I})$ and $\lambda = \mu^T\mu$.*

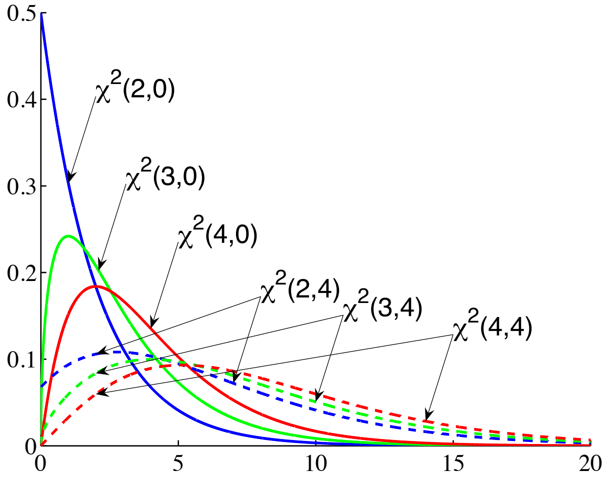


Figure 1.4: Density functions of noncentral χ^2 distributions.

Theorem 6. *Let $\mathbf{x} \sim \mathrm{N}_n(\mu,\Sigma)$ be a non-singular normal random variable, let matrix A be symmetric and let matrix $\mathrm{A}\Sigma$ be idempotent. Then*

$$\mathbf{x}^T\mathrm{A}\mathbf{x} \sim \chi^2\left(\mathrm{rank}(\mathrm{A}), \mu^T\mathrm{A}\mu\right).$$

Proof. Let $\mathrm{B} = \Sigma^{\frac{1}{2}}\mathrm{A}\Sigma^{\frac{1}{2}}$. Matrix B is symmetric because matrix A and $\Sigma^{\frac{1}{2}}$ are symmetric, and B is idempotent because

$$\mathrm{BB} = \Sigma^{\frac{1}{2}}\mathrm{A}\Sigma\mathrm{A}\Sigma^{\frac{1}{2}} = \Sigma^{\frac{1}{2}}\mathrm{A}\Sigma\mathrm{A}\Sigma\Sigma^{-\frac{1}{2}} \overset{idemp.}{=} \Sigma^{\frac{1}{2}}\mathrm{A}\Sigma\Sigma^{-\frac{1}{2}} = \mathrm{B}.$$

Using Schur decomposition (Theorem 1) and Example 1 we get

$$\mathrm{B} = \mathrm{Q}\begin{bmatrix} \mathrm{I}_{\mathrm{rank(B)}} & 0 \\ 0 & 0 \end{bmatrix}\mathrm{Q}^T,$$

where Q orthogonal. In Exercise 1.2, we show that rank(B) = rank(A). Define rank(A)-dimensional random variable $\mathbf{y} = [\mathrm{I}_{\mathrm{rank(A)}}, 0]\mathrm{Q}^T\Sigma^{-\frac{1}{2}}\mathbf{x}$. Now using Theorem 4 we get

$$\mathbf{y} \sim \mathrm{N}([\mathrm{I}_{\mathrm{rank(A)}}, 0]\mathrm{Q}^T\Sigma^{-\frac{1}{2}}\mu, \mathrm{I}_{\mathrm{rank(A)}}).$$

and so, by Definition 6,

$$\mathbf{x}^T\mathrm{A}\mathbf{x} = \mathbf{x}^T\Sigma^{-\frac{1}{2}}\mathrm{Q}\begin{bmatrix} \mathrm{I}_{\mathrm{rank(A)}} & 0 \\ 0 & 0 \end{bmatrix}\mathrm{Q}^T\Sigma^{-\frac{1}{2}}\mathbf{x} = \mathbf{y}^T\mathbf{y} \sim \chi^2(\mathrm{rank(A)}, \mu^T\mathrm{A}\mu).$$

□

1.3.3 Conditional density function

The conditional density function plays a central role in Bayesian probability theory.

Definition 7 (Conditional density function)**.** *The conditional density function of a random variable* $\mathbf{x}$ *with condition* $\mathbf{y} = y$ *is defined with equation*

$$f_{\mathbf{x}|\mathbf{y}}(x|y) = \frac{f_{\mathbf{x},\mathbf{y}}(x,y)}{f_{\mathbf{y}}(y)},$$

at points where the denominator is positive.

According to Theorem 2 we see that random variables $\mathbf{x}$ and $\mathbf{y}$ are independent if and only if $f_{\mathbf{x}|\mathbf{y}}(x|y) = f_{\mathbf{x}}(x)$. Conditional expectation is defined correspondingly:

Definition 8 (Conditional expectation)**.** *The conditional expectation of a random variable* $\mathbf{x}$ *with condition* $\mathbf{y} = y$ *is defined with equation*

$$\mathrm{E}(\mathbf{x}|\mathbf{y} = y) = \int_{-\infty}^{\infty} x f_{\mathbf{x}|\mathbf{y}}(x|y)dx.$$

Notice that conditional expectation $\mathrm{E}(\mathbf{x}|\mathbf{y} = y)$ depends on the value y of the random variable $\mathbf{y}$ and it is therefore also a random variable, defined in the same probability space as $\mathbf{y}$.

Example 11 (Conditional density function)**.** *Assume that the user's state* $\mathbf{x}$ *is normally distributed according to* $\mathbf{x} \sim \mathrm{N}(0,16)$*, and we get measurement*

$$\mathbf{y} = \frac{3}{2}\mathbf{x} + \mathbf{v},$$

where the error $\mathbf{v} \sim \mathrm{N}(0,16)$ *is independent of the state (of the random variable* $\mathbf{x}$*). Then the density function of joint distribution* $\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix}$ *is the product of the density functions of random variables* $\mathbf{x}$ *and* $\mathbf{v}$*. Easy calculation shows that*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} \sim \mathrm{N}(0, 16 \cdot \mathrm{I}),$$

then according to Theorem 4

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathrm{N}\left(0, \begin{bmatrix} 16 & 24 \\ 24 & 52 \end{bmatrix}\right).$$

It can then be shown (Exercise 1.14) that

$$\mathbf{x}|\mathbf{y} \sim \mathrm{N}\left(\frac{6}{13}y, \frac{64}{13}\right). \tag{1.10}$$

For instance, if we get an observation $\mathbf{y} = -12$*, then the conditional distribution of random variable* $\mathbf{x}$ *with condition* $\mathbf{y} = -12$ *is* $\mathrm{N}(-5\frac{7}{13}, 4\frac{12}{13})$*. Graphically, the conditional density function is a normalized slice of the density function in the plane* $\mathbf{y} = y$*. Compare the function in plane* $\mathbf{y} = -12$ *on the left side of Figure 1.2 with the calculated density function of normal distribution (Figure 1.5).*

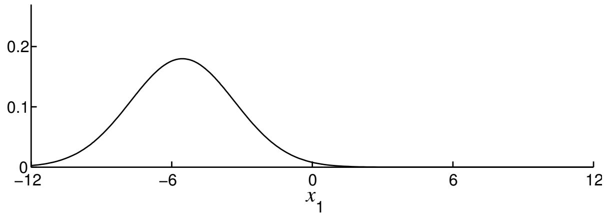


Figure 1.5: The density function of normal distribution $\mathrm{N}(-5\frac{7}{13}, 4\frac{12}{13})$ within interval $[-12, 12]$, which can be compared to the left hand side of Figure 1.2.

1.4 Coordinate systems*

JUSSI COLLIN

The precise definition of the coordinate systems used is a fundamental part of navigation. A linear coordinate system consists of an origin and a basis, for instance $(O, \mathbf{u}, \mathbf{v}, \mathbf{w})$, where vectors $\mathbf{u}, \mathbf{v}$ and $\mathbf{w}$ are linearly independent. Now any position can be given in component representation $\mathbf{r} = x\mathbf{u} + y\mathbf{v} + z\mathbf{w}$, and coordinates x, y and z are uniquely defined.

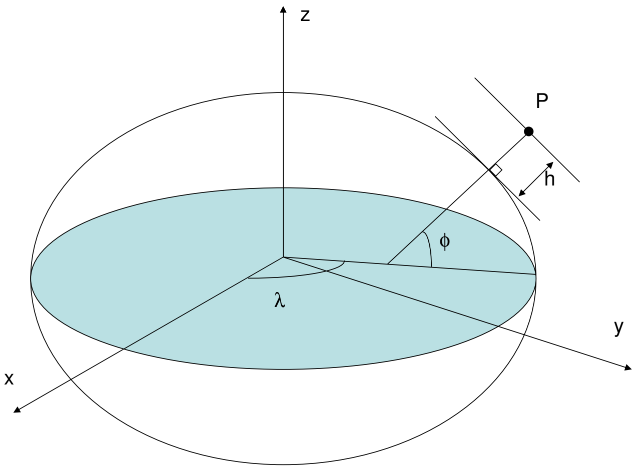


Figure 1.6: Geodetic coordinates and x-, y- and z-axis of the corresponding rectangular coordinate system.

When positioning with respect to Earth, curvilinear coordinate systems are often necessary. For example, in a spherical coordinate system, position is given as coordinates (r, θ, φ). Here r is the point's distance from origin, angle θ is the colatitude and φ is the azimuth angle. The commonly used geodetic coordinate system is slightly different from this. The geodetic coordinate system is generated by rotating an ellipse around its minor axis. This ellipsoid is usually characterized by two parameters, the major axis length a and flattening ratio $f = \frac{a-b}{a}$ (b is minor axis length). Geodetic coordinates in this coordinate system are defined as shown in Figure 1.6, i.e.

- ϕ, geodetic latitude. Notice that the angle in question is the angle between the equatorial plane and the normal vector of the ellipsoid surface.

- λ, geodetic longitude. The angle between reference meridian and meridian of the location P.

- h, geodetic height. Signed distance from the ellipsoid surface.

In addition to mathematical definition, a coordinate system has to be fixed to some physical entity. The problem here is the dynamic behavior of Earth – a planet is not a rigid body. An *realization* of a coordinate is therefore fixed to some time instant, i.e. epoch. Typically the origin is fixed to be the Earth's mass center, the positive z-axis goes through the North Pole, the reference meridian goes through the Greenwich observatory and major axis length as well as flattening ratio are found by fitting an ellipsoid to the mean sea level of Earth. GPS uses World Geodetic System 84 (WGS-84) coordinate system [32]. The Finnish coordinate system realization (EUREF-FIN) has been defined in the Finnish Government Recommendation 153 [10].

1.4.1 Linear coordinate transformation

Navigation algorithms often require transformations between different coordinate systems. For instance, GPS computations start from inertial coordinates (satellites' orbits), switch to geodetic coordinates (coordinate system rotating with Earth) and finally to local coordinates for showing the position on a 2D map. In inertial positioning computations the coordinates have to be transformed perhaps even more often; sensors form their own coordinate system, sensor measurements are with respect to inertial coordinates, and gravitation computation requires position in geodetic coordinates.

Let A-coordinates be given as vector $\mathbf{m}^A = [\alpha_1\ \alpha_2\ \alpha_3]^T$ and B-coordinates as vector $\mathbf{m}^B = [\beta_1\ \beta_2\ \beta_3]^T$. Here α_i are coordinates of position P in coordinate system A and β_i are the same position's coordinates in coordinate system B. Coordinate transformation is then the function

$$\mathbf{f}_A^B(\mathbf{m}^A) = \mathbf{m}^B. \tag{1.11}$$

When the transformation is between linear coordinate systems this function is easy to find. Assume that both coordinate systems have the same origin. The position vector $\mathbf{q}$ can be written using the basis vectors and coordinates:

$$\begin{array}{rcl} \mathbf{q} & = & \mathrm{X}_A\mathbf{m}^A \\ \mathbf{q} & = & X_B\mathbf{m}^B \end{array},$$

where columns of X are the basis vectors of the coordinate system in question. A matrix formed from the basis vectors is non-singular, and thus the transformation is just

$$\mathbf{m}^B = \mathrm{X}_B^{-1}\mathrm{X}_A\mathbf{m}^A.$$

In the following equations, the basis vectors are assumed orthonormal, and therefore $\mathrm{X}^{-1} = \mathrm{X}^T$. Additionally, we set a natural base $\mathrm{X}_A = \mathrm{I} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ for A. Then the transformation is simplified into form

$$\mathbf{m}^B = \mathrm{X}_B^T\mathbf{m}^A.$$

Now we can define a matrix often used in positioning, called the *direction cosine matrix* C_A^B:

$$\mathbf{m}^B = \mathrm{C}_A^B\mathbf{m}^A, \tag{1.12}$$

where $\mathbf{m}^A$ is a vector given in coordinate system A and $\mathbf{m}^B$ is the same vector in coordinate system B. It can be shown (Exercise 1.15) that

$$C_A^B = \begin{bmatrix} \cos(\mathbf{u},\mathbf{e}_1) & \cos(\mathbf{u},\mathbf{e}_2) & \cos(\mathbf{u},\mathbf{e}_3) \\ \cos(\mathbf{v},\mathbf{e}_1) & \cos(\mathbf{v},\mathbf{e}_2) & \cos(\mathbf{v},\mathbf{e}_3) \\ \cos(\mathbf{w},\mathbf{e}_1) & \cos(\mathbf{w},\mathbf{e}_2) & \cos(\mathbf{w},\mathbf{e}_3) \end{bmatrix}, \tag{1.13}$$

where ($\mathbf{u}$,$\mathbf{v}$,$\mathbf{w}$) is the orthonormal basis of B. Coordinate transformation back to A is easy, because $C_B^A = (C_A^B)^{-1} = (C_A^B)^T$. The following chain rule is needed especially in inertial navigation algorithms:

$$C_A^D = C_B^D C_A^B. \tag{1.14}$$

Equation (1.13) is in quite nice form for handling coordinate frame rotations.

Example 12. *Assume that coordinate frames $A1$ and $A2$ are initially identical. Rotate the frame $A2$ around the z-axis by angle θ. An observer using frame $A1$ sees an interesting object in location $\mathbf{a}^{A1} = [1\ 1\ 1]^T$. Where does an observer using frame $A2$ find this object?*

Answer: Define rotations as right handed, i.e. a positive rotation fo the vector $\mathbf{v}$ around the vector $\mathbf{w}$ takes the tip of $\mathbf{v}$ towards the direction $\mathbf{w}\times\mathbf{v}$. Note that every base vector is rotated in coordinate transformation. Now it is easy to see that the z-axis is not transformed. The other angles in equation (1.13) are also easy to find:

$$C_{A1}^{A2} = \begin{bmatrix} \cos(\theta) & \cos(\frac{\pi}{2}-\theta) & \cos(\frac{\pi}{2}) \\ \cos(\frac{\pi}{2}+\theta) & \cos(\theta) & \cos(\frac{\pi}{2}) \\ \cos(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) & \cos(0) \end{bmatrix} \tag{1.15}$$

The solution is therefore $\mathbf{a}^{A2} = C_{A1}^{A2}\mathbf{a}^{A1} = \begin{bmatrix} \cos(\theta)+\sin(\theta) \\ \cos(\theta)-\sin(\theta) \\ 1 \end{bmatrix}$.

In order to deal with more general rotations, it is good to introduce some mathematical tools that make things easier. Denote with $(\mathbf{a}\times)$ the matrix form of the cross product:

$$\mathbf{a}\times\mathbf{b} = (\mathbf{a}\times)\mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\mathbf{b} \tag{1.16}$$

Direction cosine matrix is not the only way to describe orientations of the coordinates. It can be shown (see for instance [29]), that direction cosine matrix can be represented in the form

$$C_{A2}^{A1} = I + \frac{\sin(p)}{p}(\mathbf{p}\times) + \frac{1-\cos(p)}{p^2}(\mathbf{p}\times)(\mathbf{p}\times), \tag{1.17}$$

where $\mathbf{p}$ is the rotation vector and p its length. When coordinate frame $A1$ is rotated around axis $\mathbf{p}$ by angle p, we get a new coordinate frame $A2$, and Equation (1.17) shows the connection between the direction cosine matrix and the rotation vector. As an exercise, you will verify that it does not matter whether the rotation vector is given in coordinates $A1$ or $A2$. In Example 12, the rotation vector is $\mathbf{p} = [0\,0\,\theta]^T$, and it is left as an exercise to show that when this is substituted into formula (1.17) we get the transpose of the matrix (1.15), as expected.

Here we have represented coordinate transformations in matrix forms. Another common approach is quaternion algebra, so that for example the matrix (1.17) corresponds to the quaternion (see e.g.[6, 28, 18])

$$q_{A2}^{A1} = \left[\begin{array}{c} \cos(\frac{1}{2}p) \\ \sin(\frac{1}{2}p)\mathbf{p}/p \end{array} \right]. \tag{1.18}$$

1.5 Rotating coordinate frames *

Transformations between coordinate frames that are moving linearly relative to each other are easily handled by adding velocities and translating the origin. In this section we rather focus on rotating coordinate frames. We start with simple rotation, i.e. keep the angular velocity and direction of rotation vector constant. Assume again that coordinate frames $A1$ and $A2$ are initially $(t=0)$ equivalent. Then the rotation vector can be given in the form

$$\mathbf{p} = \omega t \mathbf{u}, \tag{1.19}$$

where ω $(= \dot{p})$ is angular speed (rad/s), t is time and $\mathbf{u}$ is unit vector. The derivative of Equation (1.19) with respect to time is thus the angular velocity

$$\mathbf{w} = \dot{\mathbf{p}} = \omega \mathbf{u}. \tag{1.20}$$

Now the derivative of Equation (1.17) with respect to time is

$$\begin{array}{rcl} \dot{\mathrm{C}}^{A1}_{A2} & = & \omega\cos(\omega t)(\mathbf{u}\times) + \omega\sin(\omega t)(\mathbf{u}\times)(\mathbf{u}\times) \\ & = & [\cos(\omega t)\mathrm{I} + \sin(\omega t)(\mathbf{u}\times)](\mathbf{w}\times) \end{array}. \tag{1.21}$$

Here you should notice that $\mathrm{C}^{A1}_{A2}(\mathbf{w}\times) = [\mathrm{I}\cos(\omega t) + \sin(\omega t)(\mathbf{u}\times)](\mathbf{w}\times)$ *.

Example 13. *Assume that the rotation angle in Example 12 depends on time: $\theta(t) = \omega t$. How does the object seem to move when observed from frame $A2$?*

Start with

$$\begin{array}{rcll} \mathbf{a}^{A1} & = & \mathrm{C}^{A1}_{A2}\mathbf{a}^{A2} & \Rightarrow \\ \frac{d\mathbf{a}^{A1}}{dt} & = & \frac{d\mathrm{C}^{A1}_{A2}\mathbf{a}^{A2}}{dt} & \Rightarrow \\ \frac{d\mathbf{a}^{A1}}{dt} & = & \frac{d\mathrm{C}^{A1}_{A2}}{dt}\mathbf{a}^{A2} + \mathrm{C}^{A1}_{A2}\frac{d\mathbf{a}^{A2}}{dt} & \Rightarrow \\ \frac{d\mathbf{a}^{A1}}{dt} & = & \mathrm{C}^{A1}_{A2}(\mathbf{w}\times)\mathbf{a}^{A2} + \mathrm{C}^{A1}_{A2}\frac{d\mathbf{a}^{A2}}{dt} & \Rightarrow \\ \frac{d\mathbf{a}^{A1}}{dt} & = & \mathrm{C}^{A1}_{A2}\left(\frac{d\mathbf{a}^{A2}}{dt} + \mathbf{w}\times\mathbf{a}^{A2}\right) & \end{array} \tag{1.22}$$

This Coriolis equation describes how to transform velocity vectors from between rotating coordinate frames. Similarly the derivative of **a** *in A*2 *frame is*

$$\frac{d\mathbf{a}^{A2}}{dt} = \mathrm{C}^{A2}_{A1}\left(\frac{d\mathbf{a}^{A1}}{dt} - \mathbf{w}\times\mathbf{a}^{A1}\right). \tag{1.23}$$

Now $\mathbf{a}^{A1}$ *is constant (the object is stationary with respect to frame A*1*), so*

$$\frac{d\mathbf{a}^{A2}}{dt} = -\mathrm{C}^{A2}_{A1}(\mathbf{w}\times)\mathbf{a}^{A1}. \tag{1.24}$$

Substitute $\theta(t) = \omega t$ *to Equation (1.15), and because* $\mathbf{w} = [0\ 0\ \omega]^T$*, we obtain the result*

$$\frac{d\mathbf{a}^{A2}}{dt} = \begin{bmatrix}\omega\cos(\omega t) - \omega\sin(\omega t) & -\omega\cos(\omega t) - \omega\sin(\omega t) & 0\end{bmatrix}^T. \tag{1.25}$$

*This can be verified by using the formula $(\mathbf{a}\times)(\mathbf{a}\times) = -\|\mathbf{a}\|^2\mathrm{I} + \mathbf{a}\mathbf{a}^T$, for example

The following coordinate frames are used in inertial navigation:

- *Inertial frame (I)*, non-rotating, non-accelerating coordinate system. Newton's laws of motion apply in this coordinate system. In some cases (relativity theory) we additionally need to assume that there is no gravitation in I-frame, but in this course the computations in I-frame can be approximated with the ECI (Earth Centered Inertial) coordinate system. The origin of ECI is the Earth's mass center, and the axes keep their direction with respect to stars.
- *Earth frame (E)*, coordinate axes are fixed to Earth, z-axis has the same direction as Earth rotation axis, x- and y-axes are in equatorial plane. In literature one often sees abbreviation ECEF (Earth Centered Earth Fixed). The Earth-fixed frame rotates with respect to the inertial fram with $\mathbf{w}_{IE}^{E} \approx [0\ 0\ 7.29 \times 10^{-5}]^T$(rad/s).
- *Local frame (L)*, axes define directions "up–down", "north–south" and "west–east". It is common to use ENU (East, North, Up) ordering, i.e. x-axis points east, y-axis north and z-axis points up. When the origin is fixed to user's position, C_L^E is a function of time and $\mathbf{w}_{IL}^{L} = \mathbf{w}_{IE}^{L} + \mathbf{w}_{EL}^{L}$, where the last term describes user's movement with respect to Earth.
- *Body frame (B)*, frame fixed to the vehicle in which the navigation system is used, or the mobile unit in the case of personal navigation. This is the main coordinate system in inertial measurements. Gyro triad measurement is $\mathbf{w}_{IB}^{B}$ and the acceleration triad outputs $\mathbf{a}^B - \mathbf{g}^B$, where $\mathbf{a}$ is acceleration in ECI and $\mathbf{g}$ is local gravitation acceleration.

INS literature often uses the notation $\mathbf{w}_{IB}^{B}$, which reads as the angular velocity in B-coordinate frame with respect to I-coordinate frame ($\mathbf{w}_{IB}$), and the vector is expressed in B-frame coordinates ($\mathbf{w}^B$). The subscript conventions thus differ from those in the direction cosine matrix. I-coordinate frame means inertial coordinate frame (it does not rotate nor accelerate in space), and B-coordinate frame is a coordinate system fixed to the measurement unit. The vector $\mathbf{w}_{IB}^{B}$ is in fact the output of an ideal gyro triad.

In general, integrating the gyro measurements through Eq. (1.20) does not produce a rotation vector because the equation does not hold if the direction of rotation changes. The equation connecting the measurable angular velocity vector and the change of rotation axis is [4]

$$\dot{\mathbf{p}} = \mathbf{w}_{IB}^{B} + \frac{1}{2}\mathbf{p} \times \mathbf{w}_{IB}^{B} + \frac{1}{p^2}(1 - \frac{p\sin(p)}{2(1-\cos(p))})\mathbf{p} \times (\mathbf{p} \times \mathbf{w}_{IB}^{B}), \tag{1.26}$$

where, in the case of above treated simple rotation, the two last terms are zeros. The general case will be treated on a separate course on INS.

Exercises

1.1. Let $\mathrm{A} \in \mathbb{R}^{n \times n}$ be symmetric. Show that

$$\begin{aligned}
&\mathrm{A} > 0 \Longleftrightarrow \text{the eigenvalues of A are positive}\\
&\mathrm{A} > 0 \Rightarrow \det(\mathrm{A}) > 0\\
&\mathrm{A} > 0 \Longleftrightarrow \mathrm{A}^{-1} > 0
\end{aligned}$$

1.2. Suppose that matrices $\mathrm{B} > 0$ and A have the same size. Show that

$$\mathrm{rank}(\mathrm{BAB}) = \mathrm{rank}(\mathrm{A}).$$

1.3. (a) Let columns of matrix $\mathrm{A} \in \mathbb{R}^{m \times n}$ with $m > n$ be linearly independent ($\mathrm{rank}(\mathrm{A}) = n$). Show that matrix $\mathrm{A}^T\mathrm{A}$ is invertible.

(b) Show that equation (1.5) is correct.

1.4. Let matrices $\mathrm{A} > 0$ and $\mathrm{B} > 0$. Show that matrix $\mathrm{A}+\mathrm{B}$ is invertible.

1.5. With $\mathrm{A} = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}$, compute $\mathrm{A}^{\frac{1}{2}}$.

1.6. Find all matrices $\mathrm{B} \in \mathbb{R}^{2\times 2}$ such that $\mathrm{BB} = \mathrm{I}$. Which of the obtained matrices are symmetric and positive definite?

1.7. A user is inside an equilateral triangle building. The length of each side of the triangle is $6l$. The user has distance measurements from each side, these measurements are l, $2l$ and $3l$. What is the least squares estimate of the user's position?

1.8. Let $\mathrm{A} \in \mathbb{R}^{p\times n}$ and $b \in \mathbb{R}^p$ be constants and $\mathbf{x}$ a random variable such that $\mathrm{V}(\mathbf{x}) = \Sigma$. Compute

$$\mathrm{V}(\mathrm{A}\mathbf{x} + b)$$

1.9. Consider the system $\mathbf{y} = \mathrm{H}x + \varepsilon$, where $\varepsilon \sim \mathrm{N}(0, \sigma^2\mathrm{I})$. Compute the distribution of the estimator

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

Is the estimator unbiased ($\mathrm{E}(x - \hat{\mathbf{x}}) = 0$)?

1.10. Give the equation of the ellipse shown on the right of Figure 1.2. (hint. Matlab: `chi2inv(0.68,2)` ≈ 2.279)

1.11. Let

$$\mathbf{x} \sim \mathrm{N}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}\right) \quad \text{and} \quad a = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Compute the probability $\mathrm{P}\left(a^T\mathbf{x} \le 0\right)$.

1.12. Let $\mathbf{z} \sim \chi^2(n,\lambda)$. Compute $\mathrm{E}(\mathbf{z})$.

1.13. Let $\mathbf{x} \sim \mathrm{N}(\bar{x},\Sigma_x)$ and $\mathbf{y} \sim \mathrm{N}(\bar{y},\Sigma_y)$ be independent random variables, with $\Sigma_x > 0$ and $\Sigma_y > 0$. Show that

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}, \begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_y \end{bmatrix}\right).$$

tip: If A and C are square matrices, then $\det\left(\begin{bmatrix} \mathrm{A} & \mathrm{B} \\ 0 & \mathrm{C} \end{bmatrix}\right) = \det(\mathrm{A})\det(\mathrm{C})$.

1.14. Prove Equation (1.10) starting from Definition 7.

1.15. Derive the direction cosine matrix (1.13).

1.16. Let $\mathbf{p}$ be rotation vector between frames $A1$ and $A2$. Show that $\mathbf{p}^{A1} = \mathbf{p}^{A2}$. If $\mathrm{C}_{A2}^{A1} \neq \mathrm{I}$, can you find $\mathbf{q} \neq \alpha\mathbf{p},\ \alpha \in \mathbb{R}$ which satisfies $\mathbf{q}^{A1} = \mathbf{q}^{A2}$?

1.17. Try formula (1.17) with vectors $\mathbf{p}_1 = \begin{bmatrix} 0 & 0 & \theta \end{bmatrix}^T$ ja $\mathbf{p}_2 = \begin{bmatrix} 0 & 0 & -\theta \end{bmatrix}^T$

1.18. A hiker walks 5 km south, 5 km east and 5 km north, and ends up at the initial location. What was the origin of his journey? Express the movement using direction cosine matrices, and try to find those locations where $\mathrm{C}_2^1\mathrm{C}_3^2\mathrm{C}_4^3 = \mathrm{I}$.

1.19. A maglev train runs from Helsinki to Rovaniemi at 300 km/h. How large lateral force is needed to keep the train in north directed course?
What is the angle between Earth's rotation axis and the direction of the train with respect to time?
Assume a spherical Earth with radius R, train mass M and simplify the coordinates as:
Helsinki: $60^o10'00''$ north latitude $25^o0'00''$ east longitude
Rovaniemi: $66^o30'00''$ north latitude $25^o0'00''$ east longitude

1.20. Let a mobile phone coordinate system (B) be defined as follows: x-axis points to direction given by numbers 9-6-3, y-axis points to direction given by numbers 9-8-7, and z-direction is the cross product of x-axis and y-axis (from the screen to viewer of the screen). Rotate the phone -90 degrees around phone's y-axis. Next, rotate phone 90 degrees around x-axis. Finally, rotate phone 90 degrees around y-axis. Do the same mathematically, i.e. write direction cosine matrices. What happens if you change the order of the rotations? What if you do the rotations with respect to a non-rotating (e.g. fixed to your desktop) coordinate system?

1.21. Show that

$$\dot{\mathrm{C}}_{A2}^{A1} = \mathrm{C}_{A2}^{A1}(\mathbf{w}\times).$$

Hint: (1.17), (1.21) and $\mathbf{u}^T(\mathbf{u}\times) = \ldots$.

Chapter 2

Static positioning

NIILO SIROLA

Positioning (or localization) means estimation of a receiver's coordinates and possibly other interesting quantities (velocity, orientation, clock bias, etc). Many kinds of measurements coming from different sources can be used, and to solve the problem we first form a *mathematical model* to describe the measurements, and then apply suitable mathematical machinery to the model. No model describes reality perfectly, and often even if we knew the phenomenon quite accurately we may still decide to use some simpler model that fits better to the mathematical method we want to (or know how to) use for the problem.

Static positioning means computing a single location estimate from several simultaneously taken measurements, independent of previous or future measurements or estimates. The more general time series solution will be tackled later in Chapter 3; Depending on the choice of measurement model, the static positioning problem can be formulated as solving a nonlinear system of equations either in the sense of least squares (Section 2.3), in closed form (Section 2.2), or with respect to likelihoods (Section 2.4) or probability (Section 2.5).

Most of the methods needed for static positioning come from estimation theory, see e.g. [30, 31].

2.1 Measurement equations

Usually, not all details of the measurement process are known exactly, or it is not sensible to include them all in the model. Simple models often are sufficiently accurate without making computations too complicated. For instance, measurement error results from several different factors that can have complex mutual dependencies, but still modelling the error with one normally distributed term often works surprisingly well.

The measurement model can be written as a probability density function or in equation form. We start from the equation form, which is simpler but has some limitations (e.g. all measurements have to be numerical and continuous). Measurements are thought to consist of a "true value" and an error:

$$\text{measurement} = \text{theoretical value at location } \mathbf{x} + \text{error}.$$

All the measured values are collected into a vector $\mathbf{y}$ and the system of equations is written in the form

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \tag{2.1}$$

where the measurement function $\mathbf{h}$ is a known vector valued function and $\mathbf{v}$ is the measurement error vector, whose probability distribution is assumed known.*

When measurements have been made and the measurement equations known, we search for a location/state estimate $\hat{\mathbf{x}}$ that fits the measurements best. There are different solutions and solution methods depending on what we mean by the words "fit" and "best". The solution is not always unique, that is, more than one location candidate may fit the measurements equally well.

Example 14 (Measurement equations)**.** *Range measurement to a station located at* $\mathbf{s}$ *can be written as*

$$r = \|\mathbf{s} - \mathbf{x}\| + v,$$

where r is the measured range, $\mathbf{x}$ is location and $\|\mathbf{s} - \mathbf{x}\|$ ($= h(\mathbf{x})$) is the true range.

As another example, GPS pseudorange measurement can be written as

$$\rho = \|\mathbf{s} - \mathbf{x}\| + b + v,$$

where b is the additional error in meters caused by the receiver's clock bias. Because the clock bias b is also an unknown to be solved in the same way as location, we define a 4-dimensional state vector such that the first three state components, denoted by $\mathbf{x}_{1:3}$, contain location and the remaining component x_4 is the clock bias. If there are measurements to several different satellites/stations, then the measurement equations (2.1) can be written in vector form

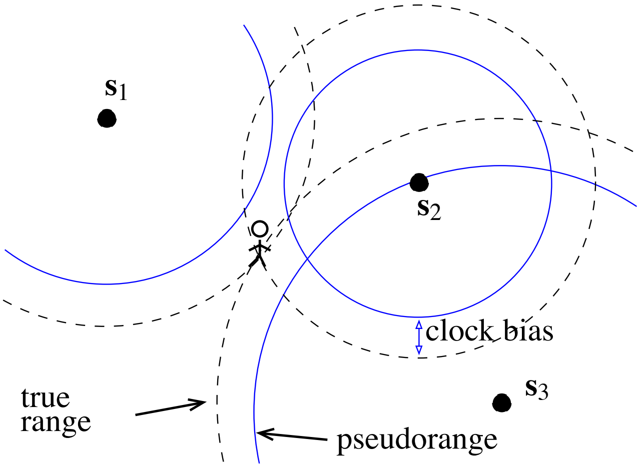


Figure 2.1: *Positioning with pseudoranges. Clock bias shows up as an error with same magnitude in all pseudoranges.*

$$\begin{bmatrix} \rho_1 \\ \vdots \\ \rho_n \end{bmatrix} = h(\mathbf{x}) + \mathbf{v} = \begin{bmatrix} \|\mathbf{s}_1 - \mathbf{x}_{1:3}\| + x_4 \\ \vdots \\ \|\mathbf{s}_n - \mathbf{x}_{1:3}\| + x_4 \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}.$$

[*] We do not go into the determination of real-life measurement error distributions based on the properties of the transmission path, transmitter and receiver or based on empirical measurement data. Just assume for now that the measurement error distribution or at least its variance has been given (or guessed) beforehand.

2.2 Closed form solutions

Closed form solution, or direct solution, is a general name for various non-iterative algorithms, where an initial guess is not needed to start the iteration. The advantage is that termination conditions are not needed for iteration, and there is no fear of converging to a wrong solution. If the problem has several solutions, the closed form solution gives them all. There is no general closed form solution capable of digesting any kind of measurement, it has to be derived separately for each measurement model, e.g. [2, 7, 17, 23].

In geometric solutions, every measurement is thought to define a surface in the parameter space on which the measurement equation is fulfilled. For instance, a range measurement defines a spherical surface in three-dimensional location-space, a range difference measurement defines a hyperboloid, etc. The problem's solutions are all the points where all measurement equations are satisfied, i.e. where all measurement surfaces intersect. Another approach, of which the Bancroft method is shown below as an example, is algebraic where different kinds of calculation tricks are used to find a special solution for the measurement equations.

Closed form solutions are useful as initial guesses for iterative methods, but otherwise mostly in theoretical research and problem visualization, because it is difficult to account for measurement errors with them.

Example 15 (Bancroft method [2]). *If all the measurements are pseudorange measurements (for instance if only GPS-measurements are used), then a least squares solution*[*] *can be computed in closed form as follows. Start from the system of measurement equations*

$$\begin{gathered}\|\mathbf{s}_1 - \mathbf{x}\| + b = y_1 \\ \vdots \\ \|\mathbf{s}_n - \mathbf{x}\| + b = y_n.\end{gathered}$$

Trick 1: Move the b's to the right hand side of the equations and square both sides of the equations. We then get n equations of the form:

$$\begin{gathered}\|\mathbf{s}_i - \mathbf{x}\|^2 = (y_i - b)^2 \\ \Leftrightarrow \|\mathbf{s}_i\|^2 - 2\mathbf{s}_i^T\mathbf{x} + \|\mathbf{x}\|^2 = y_i^2 - 2y_ib + b^2.\end{gathered}$$

Remark. Squaring both sides of an equation can cause the solution to have several branches that have to be treated separately. In this case, because norm must be non-negative, all solutions with $y_i - b < 0$ *are invalid.*

Trick 2: Collecting the squared terms into a new variable $\lambda = \|\mathbf{x}\|^2 - b^2$*, we get a linear equation whose variables are* $\mathbf{x}$*, b and* λ*:*

$$2\mathbf{s}_i^T\mathbf{x} - 2y_ib = \lambda + \|\mathbf{s}_i\|^2 - y_i^2.$$

[*] For over-determined systems, the solution is not the same as obtained with iterative methods, because the methods minimize slightly different functions.

Collect all the equations into a linear system and solve for $\mathbf{x}$ *and b in the least squares sense:*

$$\begin{bmatrix} 2\mathbf{s}_1^T & -2y_1 \\ & \vdots \\ 2\mathbf{s}_n^T & -2y_n \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \lambda + \begin{bmatrix} \|\mathbf{s}_1\|^2 - y_1^2 \\ \vdots \\ \|\mathbf{s}_n\|^2 - y_n^2 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x} \\ b \end{bmatrix} = \underbrace{\begin{bmatrix} 2\mathbf{s}_1^T & -2y_1 \\ & \vdots \\ 2\mathbf{s}_n^T & -2y_n \end{bmatrix}^{\dagger} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}}_{=\mathbf{p}} \lambda + \underbrace{\begin{bmatrix} 2\mathbf{s}_1^T & -2y_1 \\ & \vdots \\ 2\mathbf{s}_n^T & -2y_n \end{bmatrix}^{\dagger} \begin{bmatrix} \|\mathbf{s}_1\|^2 - y_1^2 \\ \vdots \\ \|\mathbf{s}_n\|^2 - y_n^2 \end{bmatrix}}_{=\mathbf{q}}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{x} \\ b \end{bmatrix} = \mathbf{p}\lambda + \mathbf{q} = \begin{bmatrix} \mathbf{d} \\ f \end{bmatrix} \lambda + \begin{bmatrix} \mathbf{e} \\ g \end{bmatrix}$$

where $A^{\dagger} = (A^T A)^{-1} A^T$ *and vectors* $\mathbf{p}$ *and* $\mathbf{q}$ *can be computed from known quantities. The original system of equations is thus fulfilled (in a least squares sense) if and only if*

$$\begin{aligned} \mathbf{x} &= \mathbf{d}\lambda + \mathbf{e} \\ b &= f\lambda + g. \end{aligned} \tag{2.2}$$

Substitute these back into the definition of λ*:*

$$\lambda = \|\mathbf{x}\|^2 - b^2 = \|\mathbf{d}\lambda + \mathbf{e}\|^2 - (f\lambda + g)^2,$$

expand square terms and rearrange into

$$(\|\mathbf{d}\|^2 - f^2)\lambda^2 + (2\mathbf{d}^T\mathbf{e} - 2fg - 1)\lambda + \|\mathbf{e}\|^2 - g^2 = 0$$

which is a second degree polynomial with respect to λ *and all other terms are known. Solve roots and compute the corresponding* $\mathbf{x}$ *and b by substituting the roots back into formula (2.2). Although there sometimes are two feasible solution candidates, GPS satellite geometry is such that the other solution is in outer space and can be neglected.*[*]

2.3 Iterative least squares

Write the system of measurement equations (2.1) again with the help of the *residual*:

$$\mathbf{p}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) - \mathbf{y}.$$

The residual describes the incompatibility between measurements and an assumed location $\mathbf{x}$. If measurements are error-free, then the residual goes to zero when $\mathbf{x}$ is the true location (but possibly also in other points). The idea of the least squares method is to determine a location estimate $\hat{\mathbf{x}}$ that minimizes the expression $\|\mathbf{p}\|^2$ that is equivalent to $\mathbf{p}(\mathbf{x})^T\mathbf{p}(\mathbf{x})$ or $\sum p_i(\mathbf{x})^2$.

[*] Unless the application is e.g. space rocket positioning…

If there are no measurement errors, then the residual goes to zero in true location $\bar{\mathbf{x}}$:

$$\mathbf{p}(\bar{\mathbf{x}}) = \mathbf{0},$$

in which case the solution(s) could be found also by directly solving the system of measurement equations (see Section 2.2). Usually, the measurements contain some error, and the residual does not necessarily have zeros at all. Then the location estimate is obtained by solving the minimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{p}(\mathbf{x})^T \mathbf{p}(\mathbf{x}). \tag{2.3}$$

Nonlinear optimization is a demanding problem and there is no general way to solve it analytically.

If the function to be minimized is well-behaved and there is an initial guess "close enough" to the minimum, then we can use an iterative optimization method. The idea is to start from an initial guess $\mathbf{x}_0$ and then compute refined estimates $\mathbf{x}_1$,$\mathbf{x}_2$, etc., until the solution does not change anymore. In Gauss-Newton method [12, section 4.2.1], [16, p. 93], [26, section 9.6] (which in positioning context is usually called iterative least squares method) the residual *is linearized* in the neighborhood of $\mathbf{x}_k$ using the first order Taylor expansion $\mathbf{p}(\mathbf{x}_k + \Delta\mathbf{x}_k) \approx \mathbf{p}(\mathbf{x}_k) + \mathbf{p}'(\mathbf{x}_k)\Delta\mathbf{x}_k$ and a step $\Delta\mathbf{x}_k$ is sought such that the linearized residual goes to zero:

$$\mathbf{p}(\mathbf{x}_{k+1}) = \mathbf{p}(\mathbf{x}_k + \Delta\mathbf{x}_k) \approx \mathbf{p}(\mathbf{x}_k) + \mathbf{p}'(\mathbf{x}_k)\Delta\mathbf{x}_k = \mathbf{0}.$$

Denoting the residual's derivative matrix, or the *Jacobian matrix* , with $\mathrm{J}_k = \mathbf{p}'(\mathbf{x}_k) = \mathbf{h}'(\mathbf{x}_k)$, the equation can be written in the form

$$\mathrm{J}_k \Delta\mathbf{x}_k = -\mathbf{p}(\mathbf{x}_k).$$

As long as there are enough independent measurements, this is an overdetermined system of linear equations, whose least squares solution (1.6) is

$$\Delta\mathbf{x}_k = -(\mathrm{J}_k^T \mathrm{J}_k)^{-1} \mathrm{J}_k^T \mathbf{p}(\mathbf{x}_k).$$

Thus, a better estimate for the minimizer is obtained with the iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathrm{J}_k^T \mathrm{J}_k)^{-1} \mathrm{J}_k^T \mathbf{p}(\mathbf{x}_k). \tag{2.4}$$

For algorithm implementation, the initial guess $\mathbf{x}_0$ is needed, and starting from that, the iteration is repeated until the solution has converged to some point $\hat{\mathbf{x}}$ (or the maximum number of iterations has been reached). It can be shown that the iteration converges if the initial point is "close enough" to the minimum and the second derivative of the residual is "small enough" [16]. When using just satellite measurements, the center of the Earth usually works as an initial guess when no better information is available, because the measurement equations are nearly linear.

If some measurements are more accurate than others, the algorithm can further be improved by using a weight matrix that forces good-quality measurement equations to have more influence

Algorithm 1 Gauss-Newton method (Iterative Least Squares)

```
1. Choose initial guess $\mathbf{x}_0$ and stopping tolerance $\delta$. Set $k = 0$.
2. Compute $\mathrm{J}_k = \mathbf{h}'(\mathbf{x}_k)$.
3. Step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$, where $\Delta\mathbf{x}_k = -\mathrm{J}_k \backslash (\mathbf{h}(\mathbf{x}_k) - \mathbf{y})$
4. If stopping condition $\|\Delta\mathbf{x}_k\| < \delta$ is not satisfied, then increase $k$ and repeat from Step 2.
```

than poor-quality equations. When the inverse of the measurement covariance matrix is used as a weight matrix the problem to be solved is thus

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{p}(\mathbf{x})^T \Sigma^{-1} \mathbf{p}(\mathbf{x}). \tag{2.5}$$

This produces the minimum variance estimate (proved in [20, chapter 6.A.1]), and in the case of normally distributed errors, also the maximum likelihood estimate (see Example 17 on page 32).

Algorithm 2 Weighted Gauss-Newton method (Iterative Weighted Least Squares)

```
1. Choose initial guess $\mathbf{x}_0$ and stopping tolerance $\delta$. Additionally, measurement covariance
   matrix $\Sigma = \mathrm{cov}(\mathbf{v})$ is required. Set $k = 0$.
2. Compute $\mathrm{J}_k = \mathbf{h}'(\mathbf{x}_k)$
3. Step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$, where $\Delta\mathbf{x}_k = -(\Sigma^{-\frac{1}{2}}\mathrm{J}_k) \backslash \left(\Sigma^{-\frac{1}{2}} (\mathbf{h}(\mathbf{x}_k) - \mathbf{y})\right)$
4. If stopping condition $\|\Delta\mathbf{x}_k\| < \delta$ is not satisfied, increase $k$ and repeat from item 2.
```

Example 16 (Jacobian matrix computation)**.** *The system of measurement equations for range measurements to three stations in locations* $\mathbf{s}_1$*,* $\mathbf{s}_2$ *and* $\mathbf{s}_3$ *is*

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \|\mathbf{s}_1 - \mathbf{x}\| \\ \|\mathbf{s}_2 - \mathbf{x}\| \\ \|\mathbf{s}_3 - \mathbf{x}\| \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}.$$

The Jacobian matrix required in Gauss-Newton algorithm is

$$\mathrm{J}(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \mathbf{h}(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \|\mathbf{s}_1 - \mathbf{x}\| \\ \|\mathbf{s}_2 - \mathbf{x}\| \\ \|\mathbf{s}_3 - \mathbf{x}\| \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}} \|\mathbf{s}_1 - \mathbf{x}\| \\ \frac{\partial}{\partial \mathbf{x}} \|\mathbf{s}_2 - \mathbf{x}\| \\ \frac{\partial}{\partial \mathbf{x}} \|\mathbf{s}_3 - \mathbf{x}\| \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{(\mathbf{s}_1 - \mathbf{x})^T}{\|\mathbf{s}_1 - \mathbf{x}\|} \\ -\frac{(\mathbf{s}_2 - \mathbf{x})^T}{\|\mathbf{s}_2 - \mathbf{x}\|} \\ -\frac{(\mathbf{s}_3 - \mathbf{x})^T}{\|\mathbf{s}_3 - \mathbf{x}\|} \end{bmatrix}.$$

In this case, each row of the Jacobian matrix is the transpose of a unit vector pointing from the presumed position $\mathbf{x}$ *to a station.*

2.3.1 Error analysis and sensitivity

In addition to location or state estimates, we are interested in the accuracy and reliability of the estimate. Measurements always contain errors, so after computing the location estimate an essential question is how close to the true location the estimate can be promised to be. Before actual error analysis, it is useful to classify different kinds of errors and to specify which of them we even try to consider.

- **modelled errors: caused by measurement errors and measurement geometry**
- modeling errors: wrong assumptions/guesses about errors' distributions or their dependencies, measurement resolution or detection threshold, incorrect values of physical constants, etc.
- numerical errors: caused by limited accuracy in computations (and/or careless programming)
- approximation, truncation and sampling errors: often intentional in order to speed up the algorithm
- others: environmental factors, equipment misuse, etc.

In this section, we only consider modelled errors, i.e. the errors that our chosen measurement model says will be there. The obtained error bound and estimates strictly speaking hold only if there are no other kinds of errors, but all the models are correct, random errors follow their assumed distributions, and computations are done without significant numerical errors. Error bounds computed this way have to be carefully interpreted along the lines of: "If the models were correct, then...".

The error of the estimation process can be characterized in different ways:

- Error propagation: Given the distribution (or at least the covariance) of measurement errors (=input), compute the distribution (covariance) of the solution (=output)
- Confidence intervals: Compute the probability that the true location is inside a given area, or determine an area that includes the true position with e.g. 95% probability
- Sensitivity analysis: How much do small changes in measurements change the solution. If for measurements $\mathbf{y}$ we get estimate $\hat{\mathbf{x}}$, how much does the estimate change if the measurement is $\widetilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y}$, where the change $\Delta\mathbf{y}$ is assumed to be relatively small?

2.3.2 Least squares error analysis

Let $\bar{\mathbf{x}}$ be the true location, $\bar{\mathbf{y}}$ the error-free measurements and $\mathbf{v}$ the realized measurement error. If the least squares method converges to solution $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{u}$, then the iteration step (2.4) length at this point is zero, i.e.:

$$(\mathrm{J}^T\mathrm{J})^{-1}\mathrm{J}^T(\mathbf{h}(\bar{\mathbf{x}} + \mathbf{u}) - (\bar{\mathbf{y}} + \mathbf{v})) = \mathbf{0}.$$

The Jacobian matrix J is a function of $\mathbf{x}$, but if the position error $\mathbf{u}$ is assumed small, J is almost constant in the neighborhood and measurement equations can be linearized $\mathbf{h}(\hat{\mathbf{x}}) \approx \mathbf{h}(\bar{\mathbf{x}}) + \mathrm{J}\mathbf{u}$. We get

$$\begin{aligned}(\mathrm{J}^T\mathrm{J})^{-1}\mathrm{J}^T(\underbrace{\mathbf{h}(\bar{\mathbf{x}})}_{=\bar{\mathbf{y}}} + \mathrm{J}\mathbf{u} - (\bar{\mathbf{y}} + \mathbf{v})) &= \mathbf{0}\\ (\mathrm{J}^T\mathrm{J})^{-1}\mathrm{J}^T\mathrm{J}\mathbf{u} - (\mathrm{J}^T\mathrm{J})^{-1}\mathrm{J}^T\mathbf{v} &= \mathbf{0}\\ \mathbf{u} &= (\mathrm{J}^T\mathrm{J})^{-1}\mathrm{J}^T\mathbf{v}.\end{aligned}$$

If the measurement errors are normally distributed, $\mathbf{v} \sim \mathrm{N}(\mathbf{0}, \Sigma)$, then it follows from Theorem 4 (p. 10) that

$$\mathbf{u} \sim \mathrm{N}\left(\mathbf{0}, (\mathrm{J}^T\mathrm{J})^{-1}\mathrm{J}^T\Sigma\mathrm{J}(\mathrm{J}^T\mathrm{J})^{-1}\right). \tag{2.6}$$

In the special case where measurement errors are distributed independently and identically, i.e. $\Sigma = \sigma^2 I$, the location error simplifies to $\mathbf{u} \sim \mathrm{N}\left(\mathbf{0}, \sigma^2(\mathrm{J}^T\mathrm{J})^{-1}\right)$. Then the covariance of estimation error derived from measurement errors is simply the product of the variance σ^2 of measurement errors and matrix $(\mathrm{J}^T\mathrm{J})^{-1}$ which depends on measurement geometry. In the context of GPS, this matrix is called DOP matrix (Dilution of Precision) [20, sections 6.A.1–2], from which different DOP-numbers can be computed as follows:

$$\begin{aligned}\text{Global/Geometric Dilution of Precision: } GDOP &= \sqrt{\mathrm{tr}(\mathrm{J}^T\mathrm{J})^{-1}}\\ \text{Position Dilution of Precision: } PDOP &= \sqrt{\mathrm{tr}[(\mathrm{J}^T\mathrm{J})^{-1}]_{(1:3,1:3)}}\\ \text{Horizontal Dilution of Precision: } HDOP &= \sqrt{\mathrm{tr}[(\mathrm{J}^T\mathrm{J})^{-1}]_{(1:2,1:2)}}\\ \text{Vertical Dilution of Precision: } VDOP &= \sqrt{[(\mathrm{J}^T\mathrm{J})^{-1}]_{(3,3)}}\\ \text{Time Dilution of Precision: } TDOP &= \sqrt{[(\mathrm{J}^T\mathrm{J})^{-1}]_{(4,4)}}\end{aligned}$$

DOP figures are useful because they summarize the quality of the satellite geometry by a single number. For example, the position error standard deviation can be estimated simply by multiplying measurement error deviation with the PDOP figure. If measurements have different deviations, however, then the error formula (2.6) does not simplify into DOP form, see Computer Exercise 2.5.

Let us emphasize that in the derivation of formula (2.6), the error $\mathbf{u}$ was assumed small so that the measurement function $\mathbf{h}(\mathbf{x})$ could be linearized in the neighborhood of the estimate. The more nonlinear, or "curved", the measurement equations are, the poorer the estimate for location error obtained this way is. (see Exercise 2.17)

Remarks:

1. The Weighted Least Squares method gives optimal results (in variance sense) as long as the measurement covariance is known (that is, if measurement errors follow the presumed distribution). The least squares criterion is sensitive to large measurement errors, called "outliers", which should be detected and removed.

2. If the measurement equation is strongly nonlinear, then the Gauss-Newton method can behave poorly and some other optimization method such as line search or Levenberg-Marquardt can be better. If the measurement equation has discontinuities, then it usually has to be either approximated with some continuous function or the residual minimized with some Monte Carlo type algorithm. [26]

3. In addition to measurement equations, inequality constraints (e.g. mobile phone network cell sector boundary) can also be taken into the measurement model. Then the problem becomes a constrained minimization problem, for which there are special techniques.

2.4 Maximum likelihood

The conditional probability density function is a more general form of measurement model. It is denoted with $p(\mathbf{y} \mid \mathbf{x})$, which can be read "the probability that $\mathbf{y}$ is measured in location $\mathbf{x}$" (see Section 1.3.3). When $\mathbf{x}$ is considered as the variable and measurements $\mathbf{y}$ as a constant in the expression, the function is then called the *likelihood function* (or simply the likelihood) and it is interpreted such that the larger the value of likelihood with measurements $\mathbf{y}$ and in location $\mathbf{x}$, the better the location in question "fits" the measurements. The likelihood function is sometimes denoted $L(\mathbf{x}|\mathbf{y})$ to emphasize that $\mathbf{x}$ is considered a variable. The joint likelihood of independent measurements is obtained by multiplying the individual measurements' likelihoods together.

If the measurement model is given in equation form $\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$, then it is easily written in likelihood form because $\mathbf{v} = \mathbf{y} - \mathbf{h}(\mathbf{x})$ and the distribution of $\mathbf{v}$ is known, so

$$p(\mathbf{y} \mid \mathbf{x}) = p_{\mathbf{v}}(\mathbf{y} - \mathbf{h}(\mathbf{x})),$$

where $p_{\mathbf{v}}$ is the known error distribution. Converting a likelihood function into a measurement equation is however possible only in special cases. Maximum likelihood as a method is therefore more generally usable than the ones previously presented.

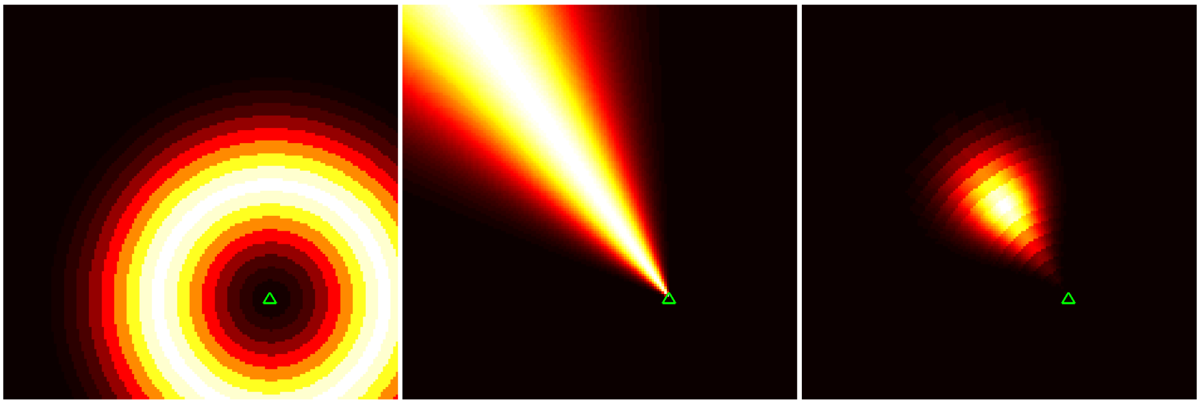


Figure 2.2: Examples of measurement likelihood functions. 1) range from a base station, 2) angle of arrival from a station, 3) the range and angle together (the product of likelihoods)

When the measurement model is in likelihood form and we obtain a measurement, the solution $\hat{\mathbf{x}}$ is sought such that its likelihood given the measurements is as large as possible: $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})$. To accomplish this, we can use nonlinear optimization methods, for instance the Gauss-Newton method.

Example 17. *Consider a system of n measurements*

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

where the error $\mathbf{v}$ *is normally distributed,* $\mathbf{v} \sim \mathrm{N}(0, \Sigma)$. *Remember that the density function of a multidimensional normal distribution is*

$$p_{\mathbf{v}}(\mathbf{z}) = C \exp\left(-\frac{1}{2}\mathbf{z}^T\Sigma^{-1}\mathbf{z}\right)$$

(the scaling constant $C = (2\pi)^{-n/2}\det(\Sigma)^{-1/2}$ *is not relevant here). Now the likelihood function is*

$$p(\mathbf{y} \mid \mathbf{x}) = p_{\mathbf{v}}(\mathbf{y} - \mathbf{h}(\mathbf{x})) = C \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}))^T\Sigma^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}))\right).$$

Because the covariance matrix Σ *is positive definite, the exponential function's argument is always non-positive. Therefore the solution with largest likelihood is found by minimizing the expression* $(\mathbf{y} - \mathbf{h}(\mathbf{x}))^T\Sigma^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}))$, *which is exactly the same as the weighted least squares method's cost function with weight matrix* Σ^{-1}.

Thus, for normally distributed errors, the maximum likelihood and weighted least squares method are identical if the inverse of the measurement covariance matrix is used as the weight matrix.

2.5 Bayesian solution

The likelihood function $p(\mathbf{y} \mid \mathbf{x})$ does not have a concrete interpretation as a probability density. A more interesting quantity would be the probability density of the state (i.e. location) conditioned on obtained measurements $p(\mathbf{x} \mid \mathbf{y})$. This quantity can be computed using Bayes' formula:

$$p(\mathbf{x} \mid \mathbf{y}) = \frac{p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})}{\int p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})d\mathbf{x}} \tag{2.7}$$

where $p(\mathbf{y} \mid \mathbf{x})$ is the likelihood function of the measurement model and $p(\mathbf{x})$ is *prior distribution*, independent of measurements, that gives an "educated guess" of possible values of $\mathbf{x}$ in the form of a density function[*]. When the obtained measurement $\mathbf{y}$ is substituted into Bayes' formula, the result is the *posterior distribution* i.e. the conditional probability distribution for state $\mathbf{x}$.

[*] If we know nothing beforehand about $\mathbf{x}$, the prior can be chosen to have uniform density.

Bayes' formula gives a probability distribution $p(\mathbf{x} \mid \mathbf{y})$, and this is more informative than just the state estimate $\hat{\mathbf{x}}$. The distribution contains all information we have on the state, based on prior information and measurement. From the posterior distribution we can, when needed, compute expectation or most probable estimate, covariance, confidence intervals, etc. and it can be used as prior information for the solution of the next time instant .

If densities are more complicated, then the posterior density cannot usually be found analytically, and we resort to Monte Carlo integration, which works for arbitrary functions. More about this in Section 3.5.

Example 18. *Consider the one-dimensional example where we measure the location of an object in a 10 meter long pipe. Let the measurement error distribution be* $\mathrm{N}(0, 3^2)$.

The likelihood function is now the normal density function (the constant coefficient is not relevant)

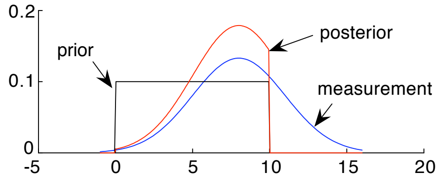
$$p(y \mid x) = C \exp\left(-\frac{(x-y)^2}{2 \cdot 3^2}\right)$$

and a uniform distribution inside the pipe is used as prior:

$$p(x) = \begin{cases} \frac{1}{10} & \mathit{if}\, 0 \le x \le 10 \\ 0 & \mathit{otherwise} \end{cases}$$

If measurement is, say $y = 8$ *meters, then using Bayes formula (2.7) we get*

$$\begin{aligned} p(x \mid y) &= \frac{p(x)p(8 \mid x)}{\int p(x)p(8 \mid x)dx} = \frac{p(x)C\exp\left(-\frac{(x-8)^2}{2\cdot 3^2}\right)}{\int p(x)C\exp\left(-\frac{(x-8)^2}{2\cdot 3^2}\right)dx} \\ &= \begin{cases} \dfrac{\exp\left(-\frac{(x-8)^2}{18}\right)}{\int_0^{10}\exp\left(-\frac{(x-8)^2}{18}\right)dx} & \mathit{if}\, 0 \le x \le 10 \\ 0 & \mathit{otherwise} \end{cases} \\ &= \begin{cases} 0.1788\exp\left(-\frac{(x-8)^2}{18}\right) & \mathit{if}\, 0 \le x \le 10 \\ 0 & \mathit{otherwise} \end{cases}. \end{aligned}$$



Prior information was used here to restrict the estimate to be inside the allowed area. If the rough location of the object in the pipe had been known more precisely, this information could also have been used as prior.

To the best of our knowledge, the obtained posterior distribution $p(x|y)$ *now describes the object's location probability in different parts of the pipe. From the posterior we can compute any estimates required, such as expectation* $\mu = \int xp(x|y)dx \approx 6.76$ *and variance* $\sigma^2 = \int (x-\mu)^2 p(x|y)dx \approx 2.12^2$. *Compared to measurement* $y = 8$, *the use of prior information thus moved the estimate a little more to the center and reduced estimate's* *standard deviation from 3 meters to 2.12 meters. If we were to get further independent measurements (assuming the object is not moving), then the currently solved posterior can be used as the new prior, thus using all the information from all the measurements optimally.*

2.5.1 Bayesian error analysis

It was previously mentioned that a Bayesian location solution produces a probability density function of the location, from which we can compute both location estimate and its expected mean square error

$$MSE = \int (\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T p(\mathbf{x} \mid \mathbf{y}) d\mathbf{x}$$

where the integral is usually computed numerically [22]. If the estimate $\hat{\mathbf{x}}$ is the expectation $E(\mathbf{x} \mid \mathbf{y})$, then the mean square error is by definition the posterior covariance.

Example 19. *Here we show that for normally distributed measurements, the error covariance is (nearly) the same as obtained in least squares method error analysis. Let measurement errors' distribution be* $\mathbf{v} \sim \mathrm{N}(\mathbf{0}, \sigma^2 \mathrm{I})$*, and prior* $p(\mathbf{x})$ *be uniform distributed*[*]*. Now the likelihood function is*

$$p(\mathbf{y} \mid \mathbf{x}) = C \exp\left(-\frac{(\mathbf{h}(\mathbf{x})-\mathbf{y})^T(\mathbf{h}(\mathbf{x})-\mathbf{y})}{2\sigma^2}\right)$$

and thus

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{y}) &= \frac{p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})}{\int p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) d\mathbf{x}} = \frac{p(\mathbf{y} \mid \mathbf{x})}{\int p(\mathbf{y} \mid \mathbf{x}) d\mathbf{x}} \\ &= \frac{C \exp\left(-\frac{(\mathbf{h}(\mathbf{x})-\mathbf{y})^T(\mathbf{h}(\mathbf{x})-\mathbf{y})}{2\sigma^2}\right)}{\int C \exp\left(-\frac{(\mathbf{h}(\mathbf{x})-\mathbf{y})^T(\mathbf{h}(\mathbf{x})-\mathbf{y})}{2\sigma^2}\right) d\mathbf{x}} \\ &= C_2 \exp\left(-\frac{(\mathbf{h}(\mathbf{x})-\mathbf{y})^T(\mathbf{h}(\mathbf{x})-\mathbf{y})}{2\sigma^2}\right). \end{aligned}$$

If we use the same linearization in the neighborhood of true location as in least squares analysis, $\mathbf{h}(\mathbf{x}) \approx \mathbf{h}(\bar{\mathbf{x}}) + \mathrm{J}(\bar{\mathbf{x}} - \mathbf{x}) = \bar{\mathbf{y}} + \mathrm{J}\Delta\mathbf{x} = \mathbf{y} - \Delta\mathbf{y} + \mathrm{J}\Delta\mathbf{x}$*, then we get*

$$p(\mathbf{x} \mid \mathbf{y}) \approx C_2 \exp\left(-\frac{(\mathrm{J}\Delta\mathbf{x} - \Delta\mathbf{y})^T(\mathrm{J}\Delta\mathbf{x} - \Delta\mathbf{y})}{2\sigma^2}\right),$$

[*] The prior is often chosen to be a uniform density in whole space, which is a contradiction because a probability density function's integral should equal to 1. In practice, the uniformly distributed prior cancels out from the computations, and it can be thought that we used prior that is uniformly distributed in an area "large enough" and that outside of this area there remains so little probability mass that it does not have any significant effect on the results. Another approach is to use a Gaussian prior with covariance $\lambda \mathrm{I}$ and let $\lambda \to \infty$

which, unlike the corresponding least squares variance, depends on the realized measurement error $\Delta\mathbf{y}$. *For illustrative purposes, see what happens when* $\Delta\mathbf{y} = \mathbf{0}$ *(reasonable because the measurement errors are zero centered):*

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{y})|_{\Delta\mathbf{y}=\mathbf{0}} &\approx C_2 \exp\left(-\frac{1}{2}\frac{\Delta\mathbf{x}^T \mathrm{J}^T \mathrm{J} \Delta\mathbf{x}}{\sigma^2}\right) \\ &= C_2 \exp\left(-\frac{1}{2}\Delta\mathbf{x}^T \left[\sigma^2(\mathrm{J}^T\mathrm{J})^{-1}\right]^{-1} \Delta\mathbf{x}\right) \end{aligned}$$

which is the density function of the distribution $\Delta\mathbf{x} \sim \mathrm{N}(\mathbf{0},\ \sigma^2(\mathrm{J}^T\mathrm{J})^{-1})$ *i.e. the same result which was obtained in previous section for least squares method. The Bayesian error estimates in this case thus fluctuate around the one given by least squares, depending on the realized measurements.*

2.6 Confidence intervals

In one-dimensional case, confidence interval means an interval inside of which the estimated quantity is with e.g. 99% probability.[*] Confidence interval is usually centered around the estimate, but nothing prevents setting it asymmetrically if needed. In multidimensional case, we usually use confidence spheres or ellipsoids centered around the estimate, but again nothing stops us from using area of any shape, as long as it contains the desired fraction of the distribution.

If the distribution is known, then the probability of being inside a certain interval is computed by integrating the distribution over the desired interval:

$$P(a \le x \le b \mid y) = \int_a^b p(x \mid y)dx.$$

In the case of Gaussian distribution, we can use the precomputed values of the cumulative distribution function and its inverse function, and get the familiar rules of thumb: inside $\pm 2\sigma$ is 95% of the probability, inside $\pm 3\sigma$ is 99.7% etc.

Even when the distribution of the estimated quantity is not known, its variance can usually be approximated. Confidence intervals derived from the normal distribution do not hold in general, but we can apply Chebyshev's inequality, which gives a lower bound for how much probability mass lies outside a certain interval when the distribution variance is σ^2:

$$P(|x - \hat{x}| \ge \lambda\sigma) \le \frac{1}{\lambda^2}. \tag{2.8}$$

[*] This is the Bayesian interpretation which we prefer over the traditional one in this context. Basically the question is how randomness should be modeled mathematically. According to the frequentistic interpretation, probability means the proportion of successes if a test is repeated "many" times. In the Bayesian interpretation probability is a subjective measure that tells us how much we know (or think we know) about something.

The difference in the views is seen e.g. in the interpretation of the 99% confidence interval. In the frequentistic sense the true location is a fixed value and in 99% of the test repetitions the confidence interval that we compute will contain the true location. The Bayesian interpretation is slightly more intuitive: we compute the confidence interval just once, and the true location is considered to be a random variable whose probability distribution has 99% of its probability lying inside the confidence interval.

For example, according to this formula, the probability mass lying over three sigmas away from the average can be at most 1/9, which means that whatever the distribution is, there is always at least $8/9 \approx 89\%$ of the probability inside the interval $\hat{x} \pm 3\sigma$.

The corresponding n-dimensional Chebyshev inequality is

$$P\left(\|\mathbf{x}-\hat{\mathbf{x}}\| \geq \lambda\sqrt{\mathrm{trace}(\Sigma)}\right) \leq \frac{1}{\lambda^2}$$

where Σ is the covariance matrix of the distribution. If variables have large correlations or they have variances with different orders of magnitude, then the less simplified formula

$$P\left(\sqrt{(\mathbf{x}-\hat{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\hat{\mathbf{x}})} \geq \lambda\sqrt{n}\right) \leq \frac{1}{\lambda^2} \tag{2.9}$$

gives a significantly tighter limit.

Formula (2.9) can also be used for defining the equation for confidence ellipsoid. For instance, the 95% confidence ellipsoid results from choosing on the right hand side of the equation $1/\lambda^2 = 0.05$, meaning that at most 5% of the probability is outside the ellipsoid. Thus, $\lambda = \sqrt{1/0.05}$, and the ellipsoid

$$E_{95\%} = \left\{\mathbf{x} \mid (\mathbf{x}-\hat{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\hat{\mathbf{x}}) < \frac{n}{0.05}\right\}$$

contains at least 95% of the probability mass (compare with Exercise 1.10 on page 20).

Conclusion and observations

- The static positioning problem can be formulated as either solving a nonlinear system of equations or seeking maximum likelihood or the maximum or mean of the Bayesian posterior. Problems are solved either with standard optimization methods or with numerical integration [22], which we return to in Section 3.5.
- If measurement equations are nearly linear such that $\mathrm{J}(\mathbf{x}) = \mathbf{h}'(\mathbf{x})$ is nearly constant near the solution (then it can be denoted only by J) and measurement errors are normally distributed or can be modeled as such, then the least squares estimate and Bayesian mean estimate are practically identical.
- Closed form solutions are harder to derive, and usually more sensitive to measurement errors. They are however sometimes useful for providing an initial guess or all the possible solutions for the problem.

Exercises

2.1. The unknown to be solved is the two-dimensional position $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$. Write the measurement model when the measurement y is

(a) a noisy measurement of the coordinate x_2,

(b) a noisy range from a fixed station at $\mathbf{s} = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T$,

(c) a noisy angle to the station $\mathbf{s}$ so that zero angle points north, and the angle grows clockwise,

(d) the difference of noisy ranges to stations $\mathbf{s}^1$ and $\mathbf{s}^2$.

2.2. An ideal INS unit measures the velocity vector in B-coordinates and the rotation vector $\mathbf{p}$ defining the attitude of the B-frame with respect to the local L-frame. Write the measurement equation assuming $\mathbf{p}$ is error-free.

2.3. The unknown to be solved is the 2D position $\mathbf{x}$. The measurement model is

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} = \begin{bmatrix} \|[100\ \ 0]^T - \mathbf{x}\| \\ \|[0\ \ 50]^T - \mathbf{x}\| \end{bmatrix} + \mathbf{v}, \quad \text{where } \mathbf{v} \sim N(0, 10^2 I)$$

(a) Give the formulas for the residual $\mathbf{p}(\mathbf{x})$ and its derivative matrix $\mathrm{J} = \mathbf{p}'(\mathbf{x})$.

(b) Let the measurement be $\mathbf{y} = \begin{bmatrix} 108 & 46 \end{bmatrix}^T$. What is the first step of the Gauss-Newton method, if the starting point is $\mathbf{x} = [0\ 0]^T$?

2.4. Derive the iteration step of the Weighted Gauss-Newton method (Algorithm 2), that is, show that the step minimizes the cost $\mathbf{p}(\mathbf{x})^T \Sigma^{-1} \mathbf{p}(\mathbf{x})$.

2.5. Derive the distribution of the error of the Weighted Gauss-Newton method (Algorithm 2) similarly to Section 2.3.2. The weighting matrix is W and the measurement covariance Σ.

2.6. A radar measures the range to a target and the rate of the change of the range, i.e. the radial velocity. The antenna is installed on a rotating rod, so that its coordinates at time t are $\mathbf{s}(t) = [\cos t\ \ \sin t]^T$.
Write the measurement equation and its Jacobian matrix at time t with respect to the state vector $[r_1\ r_2\ r_1'\ r_2']^T$.
(Note: be careful to distinguish between differentation with respect to time and differentation with respect to state.)

2.7. Given a linear measurement model $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$ and a vector $\mathbf{x}_0$, derive an estimator that minimizes the formula $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \|\mathbf{x}_0 - \mathbf{x}\|^2$.

2.8. Derive a closed-form solution (similar to Example 15 on page 25) for the measurement system

$$\begin{aligned} \|\mathbf{s}_1 - \mathbf{x}\| &= y_1 \\ &\vdots \\ \|\mathbf{s}_n - \mathbf{x}\| &= y_n. \end{aligned}$$

Then apply your algorithm to the situation of Exercise 2.3.

2.9. With WLAN signals, for example, it is not often possible or reasonable to convert the measured signal strength into a range measurement. Instead, in *fingerprint methods*, the positioning system is first calibrated by choosing a set of locations $\mathbf{p}_i$, $i = 1\ldots n$, and measuring the signal strength of all stations $\mathbf{a}_i \in \mathbb{R}^m$ in each calibration point. After this, positioning is performed by comparing received measurements with the calibration database.

Write the measurement model assuming that measurement noise is i.i.d. $N(0, \sigma^2)$ and that the calibration database contains the accurate expected values of the measurements at each calibration point. Consider also, what kind of algorithm would be needed to solve position from your measurement model.

2.10. Write the likelihood functions $p(y \mid x)$ corresponding to the following measurement equations:

(a) $y = x^2 + v, \quad v \sim \text{Uniform}[-1, 1]$

(b) $y = h(x) + v, \quad v \sim \text{Uniform}[-1, 1]$

(c) $y = 1$ if $\|\mathbf{s} - \mathbf{x}\| \leq 100$, otherwise $y = 0$

(d) $y = 1$ if $\|\mathbf{s} - \mathbf{x}\| + v \leq 100, \ v \sim \text{N}(0, 1)$, otherwise $y = 0$

(e) $y = \|\mathbf{s} - \mathbf{x}\|^2, \quad \mathbf{s} \sim \text{N}(\hat{\mathbf{s}}, \text{I})$

Note that in (c) and (d), the measurement y is binary, so that its likelihood function consists of just the two values $p(0 \mid x)$ and $p(1 \mid x)$.

2.11. The measurement model is $y = \text{floor}(x + v)$, where $v \sim \text{N}(0, (1/2)^2)$ and $\lfloor \cdot \rfloor$ denotes the floor function.

(a) What is the likelihood function of the measurement? Sketch graphs of the likelihood as a function of y (with some fixed value of x), and as a function of x (with some fixed y).

(b) What is the likelihood of n independent measurements?

(c) (optional Matlab exercise) The values of five independent measurements are 0,-2,-1,0 and -1. Ignoring the floor function, the estimate of x would be the mean of the measurements, i.e. -0.8.

Plot the joint likelihood over some suitable interval, for example $[-2, 1]$, and estimate the maximum likelihood point from the graph.

2.12. Derive the Bayes formula (2.7) from the definition of conditional density (page 13).

2.13. When both the prior and likelihood are one-dimensional normal distributions, also the posterior is a normal distribution. Derive its parameters.

2.14. Let the density of the prior be $p(x) = \begin{cases} 1/2 & \text{when } -1 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$

What is the mean and variance of this distribution?

The measurement model is $p(y \mid x) = \begin{cases} (y+1)/2 & \text{when } x \geq 0 \text{ and } -1 \leq y \leq 1 \\ (1-y)/2 & \text{when } x < 0 \text{ and } -1 \leq y \leq 1 \\ 0 & \text{otherwise.} \end{cases}$

Work out what this means, e.g. how the value of x affects the distribution of measured y. Compute the posterior distribution, and based on that the Bayesian state estimate and its variance.

2.15. Consider the Chebyshev inequality (2.8).

(a) If x is distributed according to a *two-sided exponential distribution*, $p(x) = \frac{1}{2}e^{-|x|}$, its expectation value is 0 and standard deviation $\sqrt{2}$. Compute the 67% confidence interval $[-a,\ a]$ both accurately and using the Chebyshev inequality.

(b) Give an example of a distribution that satisfies the equality $P(|x-\hat{x}| \geq \sigma) = 1$.

Computer exercises

2.16. Continues Exercise 2.3:

(a) Program the Gauss-Newton method (Algorithm 1 on page 28), and run it using initial guess $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Use the stopping tolerance $\delta = 10^{-3}$. What solution did you end up and how many iteration steps were needed?

(b) Run algorithm again using initial guess $\mathbf{x}_0 = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$. What solution did you end up and how many iteration steps were needed?

(c) Draw the residual norm in area $x \in [-50, 150], y \in [-50, 150]$ with Matlab's `surf` or `contour` command. Estimate how close to the correct solution initial guess should be so that iteration finds correct solution.

2.17. Continues previous exercise.
Because all the measurements have the same variance σ^2, location error covariance can be estimated in the form $\sigma^2(J^TJ)^{-1}$.
Let true location be $\mathbf{x}_{\text{true}} = [0\ 0]^T$. Location estimate distribution can be approximated numerically with simulated measurements. Now $\mathbf{y} \sim \mathrm{N}([100\ 50], \sigma^2\mathrm{I})$.

(a) Compute location estimate a thousand times with different measurement realizations, and compute covariance of the obtained estimates and its error compared to computed covariance estimate $\sigma^2(J^TJ)^{-1}$ when $\sigma = 1$ and when $\sigma = 10$.

(b) Draw DOP values in area $x \in [-50, 150]$, $y \in [-50, 150]$ with Matlab's `surf` or `contour` command. It may be necessary to cut off too large values from the figure, e.g. to draw `min(DOP,10)`.

Chapter 3

Filtering

SIMO ALI-LÖYTTY

A filter uses all the earlier measurements in addition to the current ones for computing the current state estimate. A basic prerequisite for the use of earlier measurements in positioning is a dynamic *state model*

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \tag{3.1}$$

which describes how current state depends on the previous ones. We use the board term *state* instead of location, so that we can include for example velocities, clock biases, or other interesting quantities as unknowns. In equation (3.1) we use the following notations: $\mathbf{x}_k$ is the state, $\mathbf{f}_k$ is the state transfer function and $\mathbf{w}_k$ is the state model error. The subscript k refers to time instant t_k. In this chapter, the state $\mathbf{x}$ of an observable is interpreted as a stochastic process [1, 9, 13].

Definition 9 (Stochastic process)**.** *Let* $(\Omega, \mathcal{F}, \mathbb{P})$ *be a probability space and* T *a parameter set. Stochastic process is a mapping* $\mathbf{x} : \Omega \times T \to \mathbb{R}^n$*, such that for every fixed* $t \in T$, $\mathbf{x}(\cdot, t)$ *is a random variable, which is usually denoted* $\mathbf{x}_t$ *or* $\mathbf{x}(t)$*.*

Because state is a stochastic process, computation of the state estimate is done in two phases. At each time instant, we find the state distribution conditioned on measurements:

$$p_{\mathbf{x}_k|\mathbf{y}_{1:k}}(x_k|y_{1:k}).$$

Here the notation $y_{1:k}$ is used for all measurements up to current time instant. Measurements are also interpreted as realizations of random variables. Measurement equations are the same as in the static case (2.1):

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \tag{3.2}$$

where $\mathbf{y}_k$ are the measurements, $\mathbf{h}_k$ is measurement function and $\mathbf{v}_k$ measurement error. In addition to the state model and the measurement equations, we also need an initial state $\mathbf{x}_0$ for the computation of the conditional distribution. Once we have the conditional state distribution, we can compute an estimate that is optimal with respect to some chosen criterion. These optimal estimates are handled in Section 3.4. From the state distribution, we can compute also other interesting quantities such as the covariance matrix of the estimation error.

In order to do estimation within reasonable time, we must be able to compute the state's conditional distribution recursively instead of reusing the whole measurement history at each time instant. This can be done if we make some independence assumptions. The first assumption is that errors $\mathbf{w}_k$ and $\mathbf{v}_k$ are mutually independent and *white noise*, which means that the errors independent of past errors. For simplicity we also assume that the errors are zero-mean; if necessary we can always change the state transfer- and measurement functions so that the errors become zero-mean. Additionally we assume that errors $\mathbf{w}_k$ and $\mathbf{v}_k$ are independent of the initial state $\mathbf{x}_0$. With these assumptions the state's conditional distribution can be computed recursively. This is covered in Sections 3.2 and 3.4.

Computing the conditional state distribution in a time series, from which we obtain the desired estimate, is called *filtering*. Filtering has many advantages compared to a static location solution. First of all, we can use all the measurements nearly optimally independent of how many measurements we get at each time instant. This enhances positioning accuracy considerably compared to static positioning. Additionally, a location solution can also be computed even if we have not got any measurement at the concerned time instant, because with the help of the state model we can compute the distribution of the coming state without new measurements.

Filtering makes it easy to use possible other information in positioning such as map information and sector/maximum range information of the base stations. Also, different kinds of error models that take into account multipath effects or other faulty measurements can be used to make positioning more fault tolerant. On the other hand, filtering has disadvantages also, first of all in several cases computation of conditional distribution is not possible analytically and although the computation can be done numerically it typically takes many times more computation time than static positioning. Additionally, filtering needs an initial state, the state dynamics model, and the distributions of the errors, any of which we do not necessarily need in static positioning. Often these are not well known, and thus the model does not necessarily correspond to reality as well as in static positioning.

As previously mentioned, we often cannot solve the conditional state distribution analytically. However, if both the state model and the measurement equations are linear and the errors and the initial state are normally distributed, then the state conditional distribution can be computed recursively. The algorithm doing this is called the Kalman filter, named after Rudolf E. Kalman [14]. The Kalman filter also solves a wider class of problems, as it remains the best linear unbiased estimator (BLUE, Best Linear Unbiased Estimator) even when the distributions are non-Gaussian. The Kalman filter is presented in Section 3.2 from the BLUE point of view, and in Exercise 3.7 you are asked to confirm that it also computes the state's conditional distribution. There also exists numerous approximations of the Kalman filter in cases where the state model and/or the measurement equations are not linear; these non-linear Kalman filters are presented in Section 3.3.

The general Bayesian filtering problem is presented in Section 3.4. Numerical methods for solving the general Bayesian filtering problem are presented in Section 3.5.

3.1 Constant velocity model

The state dynamics model (3.1), like the measurement model, is dependent on the observed system. Clearly, for a stationary target, a suitable state model is different than for a cyclist weaving through a crowd. The problem is that we rarely know the correct state model in advance. This is why, for instance, the interactive multiple model method (IMM) uses a set of several measurement/state models from which the filter tries to choose the best during runtime [27]. Another approach to this problem is adaptive filter that try to estimate and update the state model parameters along with the state itself. It is good to remember that often we have to settle for broad generalizations in the selection of the models, because the purpose is to get models that work as good as possible and are simple enough. In this section, we introduce a simple but commonly used model called the *constant velocity model*.

The constant velocity model has 6 state variables, the three-dimensional location and three-dimensional velocity. The clock bias and its derivative required in satellite positioning can be included in the same manner, as is done for example in the books [5, 8, 15, 21].

We model the state as a stochastic differential equation

$$d\mathbf{x} = \mathrm{F}\mathbf{x}dt + \mathrm{G}d\beta, \tag{3.3}$$

where

$$\mathrm{F} = \begin{bmatrix} 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix} \quad \text{ja} \quad \mathrm{G} = \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix},$$

Here the velocity error is modeled as Brownian motion, denoted with β. The diffusion matrix for the Brown motion β is chosen to be a simple diagonal matrix $\mathrm{Q_c} = \sigma_\mathrm{c}^2\mathrm{I}$, where σ_c^2 describes velocity error in the axis directions. To be more precise, σ_c^2 describes how much variance needs to be added to the velocity error within one second in north-south, east-west and up-down directions.

The solution of the stochastic differential equation (3.3) can be discretized into a state model (see e.g.[19, s.171] or [9, s.200])

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1},$$

where

$$\Phi_{k-1} = e^{(t_k - t_{k-1})\mathrm{F}} = \begin{bmatrix} \mathrm{I} & \Delta t\mathrm{I} \\ 0 & \mathrm{I} \end{bmatrix}.$$

The term $\Delta t = t_k - t_{k-1}$ is the length of time step. State model error $\mathbf{w}_k$ is white zero-mean normally distributed noise, which is assumed independent of initial condition $\mathbf{x}_0$. The covariance matrix Q_k of the state model error is

$$\mathrm{Q}_k = \mathrm{V}(\mathbf{w}_{k-1}) = \int_{t_{k-1}}^{t_k} \Phi(t_k,t)\mathrm{G}\mathrm{Q_c}\mathrm{G}^T\Phi(t_k,t)^T dt = \begin{bmatrix} \frac{1}{3}\Delta t^3\mathrm{I} & \frac{1}{2}\Delta t^2\mathrm{I} \\ \frac{1}{2}\Delta t^2\mathrm{I} & \Delta t\mathrm{I} \end{bmatrix}\sigma_\mathrm{c}^2.$$

3.2 The Kalman filter

The Kalman filter solves the filtering problem where the state model and the measurement model (measurement equations) are linear. Then the state model is

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{3.4}$$

where $\mathbf{x}_k$ is the process state, Φ_k is the state transfer matrix and $\mathbf{w}_k$ is the state model error, which is assumed zero-mean white noise and independent of the errors in the measurements and the initial state. The covariance matrix of the state model error is denoted with Q_k. The measurement model is

$$\mathbf{y}_k = \mathrm{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{3.5}$$

where $\mathbf{y}_k$ are the measurements, H_k is the measurement matrix (or system matrix) and $\mathbf{v}_k$ is the measurement model error, which is assumed zero-centered white noise and independent of the errors in the measurements and the initial state. The covariance matrix of the measurement model error is denoted with R_k, which is assumed positive definite. In addition to the state model and the measurement model, the Kalman filter needs initial state $\mathbf{x}_0$. We denote the the expectation of the initial state with vector $\hat{\mathbf{x}}_0$ and the covariance matrix with constant matrix P_0, which is assumed positive definite. The hat notation refers to an estimator, which is a function of the measurements. In the case of $\hat{\mathbf{x}}_0$, the estimator is a constant random variable, because at time instant t_0 we do not yet have any measurements. Covariance matrix P_0 can also be interpreted as a mean square error (MSE) matrix $\mathrm{E}\left(\mathbf{e}_0\mathbf{e}_0^T\right)$, where $\mathbf{e}_0 = \mathbf{x}_0 - \hat{\mathbf{x}}_0$.

The purpose of the Kalman filter is to solve a best linear unbiased estimator for state $\mathbf{x}_k$. Because of the independence assumptions, this can be done recursively and then the estimator can be written as

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \mathrm{J}_{k-1} & \mathrm{K}_k \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{y}_k \end{bmatrix}, \tag{3.6}$$

where $\hat{\mathbf{x}}_{k-1}$ is the best linear unbiased estimator for state $\mathbf{x}_{k-1}$ at time instant t_{k-1}. The corresponding mean square error matrix is denoted P_{k-1}. The measurements at time instant t_k are denoted with random variable $\mathbf{y}_k$.

In the remainder of this section, we derive such matrices J_{k-1} and K_k that (3.6) becomes the best linear unbiased estimator, i.e. the Kalman filter. The best here means that the mean square error of the estimator $\mathrm{E}(\mathbf{e}_k^T\mathbf{e}_k) = \mathrm{tr}(\mathrm{E}(\mathbf{e}_k\mathbf{e}_k^T))$ would be as small as possible. This estimator is called the least squares estimator (MSE-estimator). Because the estimator is required to be unbiased, we get

$$\mathrm{E}(\mathbf{x}_k) = \mathrm{E}(\hat{\mathbf{x}}_k) = \begin{bmatrix} \mathrm{J}_{k-1} & \mathrm{K}_k \end{bmatrix} \begin{bmatrix} \mathrm{E}(\hat{\mathbf{x}}_{k-1}) \\ \mathrm{E}(\mathbf{y}_k) \end{bmatrix}.$$

Using the unbiasedness of the estimator $\hat{\mathbf{x}}_{k-1}$, the state model (3.4) and the measurement model (3.5) we get

$$\Phi_{k-1}\,\mathrm{E}(\mathbf{x}_{k-1}) = \mathrm{J}_{k-1}\,\mathrm{E}(\mathbf{x}_{k-1}) + \mathrm{K}_k\mathrm{H}_k\Phi_{k-1}\,\mathrm{E}(\mathbf{x}_{k-1}).$$

This equation needs to hold for any expectation $\mathrm{E}(\mathbf{x}_{k-1})$, thus

$$\mathrm{J}_{k-1} = \Phi_{k-1} - \mathrm{K}_k\mathrm{H}_k\Phi_{k-1}. \tag{3.7}$$

The best linear unbiased estimator of state $\mathbf{x}_k$ which does not use measurements at the current time instant t_k is called the *prior estimator*. In other words, the prior estimator is the same form as the estimator in formula (3.6), but with matrix K_k set to zero. Then the prior estimator, denoted by a minus superscript, is according to formula (3.7)

$$\hat{\mathbf{x}}_k^- = \Phi_{k-1}\hat{\mathbf{x}}_{k-1}. \tag{3.8}$$

The mean square error matrix of the prior estimator is thus

$$\begin{aligned}
\mathrm{P}_k^- &= \mathrm{E}\left((\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T\right) \\
&\overset{(3.4)}{=} \mathrm{E}\left((\Phi_{k-1}\mathbf{e}_{k-1} + \mathbf{w}_{k-1})(\Phi_{k-1}\mathbf{e}_{k-1} + \mathbf{w}_{k-1})^T\right) \\
&\overset{\text{indep.}}{=} \Phi_{k-1}\,\mathrm{E}\left(\mathbf{e}_{k-1}\mathbf{e}_{k-1}{}^T\right)\Phi_{k-1}^T + \mathrm{E}\left(\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T\right) \\
&= \Phi_{k-1}\mathrm{P}_{k-1}\Phi_{k-1}^T + \mathrm{Q}_{k-1}.
\end{aligned}$$

The estimator in formula (3.6) that also uses measurements at time instant t_k is called the *posterior estimator*. The error of the posterior estimator is

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k \\
&= \mathbf{x}_k - (\Phi_{k-1} - \mathrm{K}_k\mathrm{H}_k\Phi_{k-1})\hat{\mathbf{x}}_{k-1} - \mathrm{K}_k\mathbf{y}_k \\
&\overset{(3.5)}{=} (\mathrm{I} - \mathrm{K}_k\mathrm{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathrm{K}_k\mathbf{v}_k
\end{aligned} \tag{3.9}$$

According to the assumption, the error of the prior estimator $\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$ is independent of the measurement error $\mathbf{v}_k$, so that the mean square error matrix of the posterior estimator is

$$\begin{aligned}
\mathrm{P}_k &= \mathrm{E}\left(\mathbf{e}_k\mathbf{e}_k^T\right) \\
&\overset{(3.9)}{=} (\mathrm{I} - \mathrm{K}_k\mathrm{H}_k)\mathrm{P}_k^-(\mathrm{I} - \mathrm{K}_k\mathrm{H}_k)^T + \mathrm{K}_k\mathrm{R}_k\mathrm{K}_k^T \\
&\overset{\mathit{unbiased},\mathit{ex}.3.2}{=} \mathrm{P}_k^- - \mathrm{P}_k^-\mathrm{H}_k^T(\mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k)^{-1}\mathrm{H}_k\mathrm{P}_k^- \\
&\quad + (\mathrm{K}_k\mathrm{B} - \mathrm{P}_k^-\mathrm{H}_k^T\mathrm{B}^{-1})(\mathrm{K}_k\mathrm{B} - \mathrm{P}_k^-\mathrm{H}_k^T\mathrm{B}^{-1})^T,
\end{aligned} \tag{3.10}$$

where $\mathrm{B} = (\mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k)^{\frac{1}{2}}$. The trace of P_k is at minimum when the trace of the matrix

$$(\mathrm{K}_k\mathrm{B} - \mathrm{P}_k^-\mathrm{H}_k^T\mathrm{B}^{-1})(\mathrm{K}_k\mathrm{B} - \mathrm{P}_k^-\mathrm{H}_k^T\mathrm{B}^{-1})^T \tag{3.11}$$

is minimized, because the other terms in the sum do not depend on matrix K_k. The trace of matrix (3.11) is smallest possible when $\mathrm{K}_k\mathrm{B} - \mathrm{P}_k^-\mathrm{H}_k^T\mathrm{B}^{-1} = 0$ (Exercise 3.3), and the solution to this equation is

$$\mathrm{K}_k = \mathrm{P}_k^-\mathrm{H}_k^T(\mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k)^{-1}. \tag{3.12}$$

This matrix is called the Kalman gain. Combining the results, we get the Kalman filter estimator

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathrm{K}_k(\mathbf{y}_k - \mathrm{H}_k\hat{\mathbf{x}}_k^-), \tag{3.13}$$

where the prior estimator has been given in formula (3.8) and the Kalman gain in formula (3.12). The mean square error matrix of this estimator (3.10) can be written as

$$\mathrm{P}_k = (\mathrm{I} - \mathrm{K}_k \mathrm{H}_k)\mathrm{P}_k^-.$$

The Kalman filter has been given in compact form in Algorithm 3.

In formula (3.13), the difference between the measurement $\mathbf{y}_k$ and the predicted measurement $\hat{\mathbf{y}}_k = \mathrm{H}_k \hat{\mathbf{x}}_k^-$, i.e. $\mathbf{y}_k - \mathrm{H}_k \hat{\mathbf{x}}_k^-$ is called the *innovation*. The mean square error matrix of the measurement prediction (the covariance matrix of the innovation) is $\mathrm{H}_k \mathrm{P}_k^- \mathrm{H}_k^T + \mathrm{R}_k$ (Exercise. 3.1). The use of the innovation is almost the only way of monitoring the quality of the filter solution during runtime. Because the covariance matrix of the innovation is known, we can do statistical tests on whether the realized measurement is probable or not. Many *robust filters* do these kinds of tests and if the result is that the measurement is unlikely, it is either not used at all or it is given less weight than the "likely" measurements. Additionally, many *adaptive filters* use innovations to define or update filter parameters during runtime, for instance the covariances of the errors [3, Chapter 11].

Algorithm 3 The Kalman filter

```
• The state model:            $\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}$,   $V(\mathbf{w}_{k-1}) = \mathrm{Q}_{k-1}$
• The measurement model:      $\mathbf{y}_k = \mathrm{H}_k\mathbf{x}_k + \mathbf{v}_k$,          $V(\mathbf{v}_k) = \mathrm{R}_k$
• The measurements:           $y_{1:m} = \{y_1, y_2, \ldots, y_m\}$
• The initial state estimate and its MSE: $\hat{x}_0$ and $\mathrm{P}_0$

1. Set $k = 1$.

2.   • The prior estimate:      $\hat{x}_k^- = \Phi_{k-1}\hat{x}_{k-1}$
     • The prior MSE:           $\mathrm{P}_k^- = \Phi_{k-1}\mathrm{P}_{k-1}\Phi_{k-1}^T + \mathrm{Q}_{k-1}$
     • The Kalman gain:         $\mathrm{K}_k = \mathrm{P}_k^-\mathrm{H}_k^T(\mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k)^{-1}$
     • The posterior estimate:  $\hat{x}_k = \hat{x}_k^- + \mathrm{K}_k(y_k - \mathrm{H}_k\hat{x}_k^-)$
     • The posterior MSE:       $\mathrm{P}_k = (\mathrm{I} - \mathrm{K}_k\mathrm{H}_k)\mathrm{P}_k^-$

3. Stop if $k = m$, otherwise set $k = k + 1$ and get back to step 2.
```

Example 20 (A Kalman filter)**.** *Figure 3.1 shows an example of a situation where a two-dimensional constant velocity model has been used, see Section 3.1. Constant velocity model satisfies assumptions of Kalman filter. The parameters used in the example are as following:*

$$\begin{aligned}
\mathbf{x}_0 &= \begin{bmatrix} 0 & 0 & 5 & 0 \end{bmatrix}^T, \mathrm{P}_0 = \begin{bmatrix} 10^2\mathrm{I} & 0 \\ 0 & 3^2\mathrm{I} \end{bmatrix}, \\
\Phi &= \begin{bmatrix} \mathrm{I} & \mathrm{I} \\ 0 & \mathrm{I} \end{bmatrix}, \qquad \mathrm{Q} = \begin{bmatrix} \frac{1}{3}\mathrm{I} & \frac{1}{2}\mathrm{I} \\ \frac{1}{2}\mathrm{I} & \mathrm{I} \end{bmatrix}, \\
\mathrm{H} &= \begin{bmatrix} \mathrm{I} & 0 \end{bmatrix} \quad \text{and} \quad \mathrm{R} = \begin{bmatrix} 30^2 & 0 \\ 0 & 50^2 \end{bmatrix}.
\end{aligned} \tag{3.14}$$

The true route over one minute period has been drawn with a black dashed line. The positioning system gives location measurements according to the measurement model at one second intervals (red dots linked with narrow lines). The Kalman filter uses the locations as measurements, and the resulting state estimate is drawn with a continuous blue line.

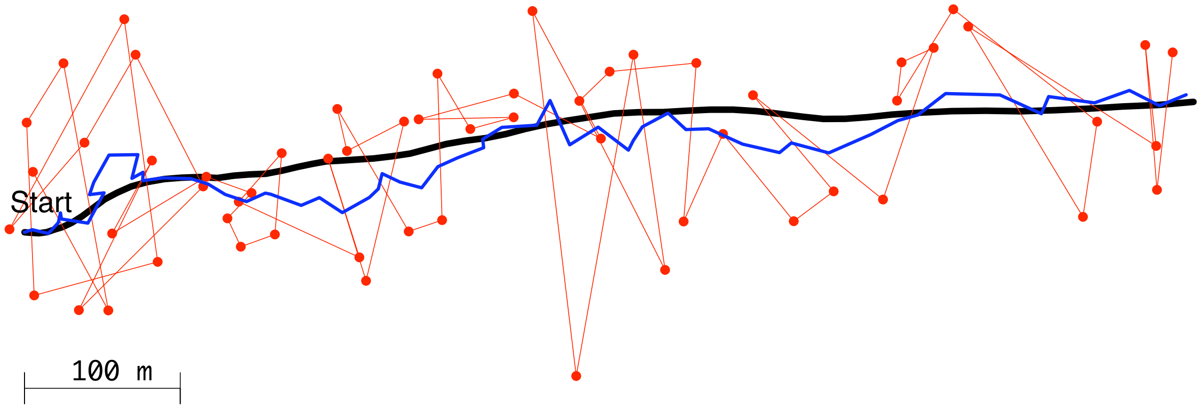


Figure 3.1: Example of a Kalman filter applied to two-dimensional location measurements.

3.3 Nonlinear Kalman filters

The Kalman filter is very popular for linear systems, because it is relatively easy to compute and requires few assumptions. For this reason, there are several methods of applying something similar to the Kalman filter also to nonlinear systems (3.1), (3.2). In this section, we introduce the basic ideas on which most Kalman filter extensions are based. First, we give a slightly generalized version of the BLU-estimator. Let

$$\mathrm{E}\left(\begin{bmatrix}\mathbf{x}_k\\ \mathbf{y}_k\end{bmatrix}\right)=\begin{bmatrix}\bar{x}_k\\ \bar{y}_k\end{bmatrix},\ \text{and}\ \mathrm{V}\left(\begin{bmatrix}\mathbf{x}_k\\ \mathbf{y}_k\end{bmatrix}\right)=\begin{bmatrix}\mathrm{P}_{xx_k} & \mathrm{P}_{xy_k}\\ \mathrm{P}_{yx_k} & \mathrm{P}_{yy_k}\end{bmatrix}.$$

Now the BLU-estimator of state $\mathbf{x}_k$ is [3]

$$\begin{aligned}\hat{\mathbf{x}}_k &= \bar{x}_k+\mathrm{P}_{xy_k}\mathrm{P}_{yy_k}^{-1}(\mathbf{y}_k-\bar{y}_k),\\ \mathrm{E}((\mathbf{x}_k-\hat{\mathbf{x}}_k)(\mathbf{x}_k-\hat{\mathbf{x}}_k)^T) &= \mathrm{P}_{xx_k}-\mathrm{P}_{xy_k}\mathrm{P}_{yy_k}^{-1}\mathrm{P}_{yx_k}.\end{aligned}\tag{3.15}$$

Nonlinear extensions of the Kalman filter try to solve the unknown quantities of equation (3.15), $\bar{x}_k, \bar{y}_k, \mathrm{P}_{xx_k}, \mathrm{P}_{xy_k}=\mathrm{P}_{yx_k}^T, \mathrm{P}_{yy_k}$ by using some approximation. For these kinds of algorithms we use a general name *nonlinear Kalman filter*, whose algorithm has been given on page 48. Next, we treat two commonly used nonlinear Kalman filters in more detail, the so-called extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [27].

When using nonlinear Kalman filters, it is good to remember that they do not actually even approximate the optimal solution which is handled in Section 3.4. For this reason, EKF for instance may give completely false results like in the example on page 56. Nevertheless, in many practical application nonlinear Kalman filters work fairly well (see Exercise 3.14) and they often require significantly less computation than the numerical approximations of the general Bayesian filter, which are handled in Section 3.5. Because of these reasons, nonlinear Kalman filters are popular in engineering.

Algorithm 4 Nonlinear Kalman filter (EKF and UKF)

```
• The state model:           $\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$,    $\mathrm{V}(\mathbf{w}_{k-1}) = \mathrm{Q}_{k-1}$
• The measurement model:     $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$,    $\mathrm{V}(\mathbf{v}_k) = \mathrm{R}_k$
• The initial state estimate, its MSE and the measurements: $\hat{x}_0$, $\mathrm{P}_0$ and $y_{1:m} = \{y_1, y_2, \dots, y_m\}$
1. Set $k = 1$.
2.   • The prior estimate:      $\hat{x}_k^- = \bar{x}_k$
     • The prior MSE:           $\mathrm{P}_k^- = \mathrm{P}_{xx_k}$
     • The posterior estimate:  $\hat{x}_k = \hat{x}_k^- + \mathrm{P}_{xy_k}\mathrm{P}_{yy_k}^{-1}(y_k - \bar{y}_k)$
     • The posterior MSE:       $\mathrm{P}_k = \mathrm{P}_{xx_k} - \mathrm{P}_{xy_k}\mathrm{P}_{yy_k}^{-1}\mathrm{P}_{yx_k}$,
3. Stop, if $k = m$, otherwise set $k = k + 1$ and go back to step 2.
```

3.3.1 Extended Kalman filter

The extended Kalman filter is a nonlinear Kalman filter that is based on the first order Taylor approximations of the state model and the measurement model.

- The state model approximation:
  $\mathbf{x}_k \approx \mathbf{f}_{k-1}(\hat{x}_{k-1}) + \Phi_{k-1}(\mathbf{x}_{k-1} - \hat{x}_{k-1}) + \mathbf{w}_{k-1}$ where $\Phi_{k-1} = \mathbf{f}'_{k-1}(\hat{x}_{k-1})$ and
- The measurement model approximation:
  $\mathbf{y}_k \approx \mathbf{h}_k(\hat{x}_k^-) + \mathrm{H}_k(\mathbf{x}_k - \hat{x}_k^-) + \mathbf{v}_k$, where $\mathrm{H}_k = \mathbf{h}'_k(\hat{x}_k^-)$.

Then the unknown quantities in Algorithm 4 are easily computed:

$$\begin{bmatrix} \bar{x}_k \\ \bar{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{k-1}(\hat{x}_{k-1}) \\ \mathbf{h}_k(\hat{x}_k^-) \end{bmatrix}, \text{ and}$$

$$\begin{bmatrix} \mathrm{P}_{xx_k} & \mathrm{P}_{xy_k} \\ \mathrm{P}_{yx_k} & \mathrm{P}_{yy_k} \end{bmatrix} = \begin{bmatrix} \mathrm{P}_k^- & \mathrm{P}_k^- \mathrm{H}_k^T \\ \mathrm{H}_k \mathrm{P}_k^- & \mathrm{H}_k \mathrm{P}_k^- \mathrm{H}_k^T + \mathrm{R}_k \end{bmatrix},$$

where $\mathrm{P}_k^- = \Phi_{k-1}\mathrm{P}_{k-1}\Phi_{k-1}^T + \mathrm{Q}_{k-1}$. It is also possible to use a higher order Taylor approximation of the quantities listed above. On the other hand, higher order approximations often need more restrictive assumptions about the prior and posterior distributions, for instance normality, so that the quantities in formula (3.15) can be computed. Additionally, the analytical computation of even a first order Taylor approximation is difficult and in some cases impossible. This the reason for developing nonlinear Kalman filters that do not require the derivative of the state transfer function or the measurement function, such as the UKF in the next section.

3.3.2 Unscented Kalman filter

Unscented Kalman filter (UKF) is a derivative-free nonlinear Kalman filter based on a numerical integration method called the unscented transformation (UT). The UT approximates the expectation of function $\mathbf{f}(\mathbf{x})$ using the so-called σ-points $\{\chi_0, \ldots, \chi_{N-1}\}$ when the distribution of random variable $\mathbf{x}$ is known. This numerical integration method can be written as

$$\mathrm{E}(\mathbf{f}(\mathbf{x})) = \int \mathbf{f}(x)p(x)dx \approx \sum_{i=0}^{N-1} \omega_i \mathbf{f}(\chi_i), \tag{3.16}$$

where weights ω_i and sigma-points χ_i are selected such that the approximation is accurate for certain degree polynomials $\mathbf{f}(x)$ when random variable $\mathbf{x}$ is normally distributed.

Let n be the state dimension, $\hat{x}$ be the expectation of random variable $\mathbf{x}$ and the covariance matrix be P. Then the generally used σ-points selection contains $2n+1$ points which are

$$\begin{array}{ccc}
\text{Index } (i) & \text{Weight } (\omega_i) & \sigma\text{-point } (\chi_i) \\
\hline
0 & \frac{\kappa}{\kappa+n} & \hat{x} \\
1, \ldots, n & \frac{1}{2(\kappa+n)} & \hat{x} + \sqrt{\kappa+n}\sqrt{\mathrm{P}}e_i \\
n+1, \ldots, 2n & \frac{1}{2(\kappa+n)} & \hat{x} - \sqrt{\kappa+n}\sqrt{\mathrm{P}}e_{i-n},
\end{array} \tag{3.17}$$

where $\sqrt{\mathrm{P}}$ means such a matrix which has the property that $\sqrt{\mathrm{P}}\sqrt{\mathrm{P}}^T = \mathrm{P}$. Parameter $\kappa > -n$ is a freely selectable constant. With these σ-points and weights, the approximation (3.16) is exact if the random variable $\mathbf{x}$ is normally distributed and function $\mathbf{f}(x)$ is a third degree polynomial. In addition, the choice $\kappa = 3 - n$ minimizes the error of the integral approximation of a fourth degree polynomial and it is therefore commonly used, see Exercise 3.8.

Let now $\{\chi_{0_{k-1}}, \ldots, \chi_{(N-1)_{k-1}}\}$ be the σ-points generated based on the posterior distribution at time instant t_k. Then the unknown quantities of Algorithm 4 can be computed:

$$\begin{aligned}
\bar{x}_k &= \sum_{i_{k-1}=0}^{N-1} \omega_{i_{k-1}} \mathbf{f}_{k-1}(\chi_{i_{k-1}}), \quad \bar{y}_k = \sum_{i_{k-1}=0}^{N-1} \omega_{i_{k-1}} \mathbf{h}_k(\chi_{i_{k|k-1}}), \\
\mathrm{P}_{xx_k} &= \sum_{i_{k-1}=0}^{N-1} \omega_{i_{k-1}} (\mathbf{f}_{k-1}(\chi_{i_{k-1}}) - \bar{x}_k)(\mathbf{f}_{k-1}(\chi_{i_{k-1}}) - \bar{x}_k)^T + \mathrm{Q}_{k-1}, \\
\mathrm{P}_{xy_k} &= \sum_{i_{k-1}=0}^{N-1} \omega_{i_{k-1}} (\chi_{i_{k|k-1}} - \bar{x}_k)(\mathbf{h}_k(\chi_{i_{k|k-1}}) - \bar{y}_k)^T \text{ and} \\
\mathrm{P}_{yy_k} &= \sum_{i_{k-1}=0}^{N-1} \omega_{i_{k-1}} (\mathbf{h}_k(\chi_{i_{k|k-1}}) - \bar{y}_k)(\mathbf{h}_k(\chi_{i_{k|k-1}}) - \bar{y}_k)^T + \mathrm{R}_k,
\end{aligned}$$

where $\chi_{i_{k|k-1}} = \mathbf{f}_{k-1}(\chi_{i_{k-1}})$. Another way is to generate these σ-points of the prior distribution from a prior distribution whose expectation is $\bar{x}_k$ and whose covariance matrix is P_{xx_k}, then the results are a little different from the previous implementation.

3.4 Bayesian filter

In this section, we use the state model, the measurement model, and the initial state as well as the usual independence assumptions introduced at the beginning of the chapter on page 41. The aim in Bayesian filtering is to determine the state's conditional density function conditioned on the measurements,

$$p_{\mathbf{x}_k|\mathbf{y}_{1:k}}(x_k|y_{1:k}) \overset{\triangle}{=} p(x_k|y_{1:k}).$$

The abbreviated notation is used where it is easy to see from the context what the notation means. Assume that the state's conditional density function at the previous time instant is $p(x_{k-1}|y_{1:k-1})$. The conditional density function for the current instant conditioned on previous measurements, $p(x_k|y_{1:k-1})$, can be determined with the help of the Chapman-Kolmogorov equation

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}.$$

This distribution is called the prior distribution. The conditional density function $p(x_k|x_{k-1})$ can be derived from the state model

$$p(x_k|x_{k-1}) \overset{(3.1)}{=} p_{\mathbf{w}_{k-1}}(x_k - \mathbf{f}_{k-1}(x_{k-1})).$$

Now, by Bayes' rule and the independence assumptions of the errors, we can derive a formula for the current state's conditional density function conditioned on current and previous measurements:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{\int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k}.$$

This distribution is called the posterior distribution. The first term of the numerator

$$p(y_k|x_k) = p_{\mathbf{y}_k=y_k|\mathbf{x}_k}(y_k|x_k) \overset{(3.2)}{=} p_{\mathbf{v}_k}(y_k - \mathbf{h}_k(x_k))$$

is called the likelihood. Note that this is not a probability density function. The density function of the posterior distribution is proportional to the product of the prior distribution and the likelihood. The denominator is the normalization constant $p(y_k|y_{1:k-1})$, which is the conditional density function of the new measurement conditioned on previous measurements. From this the filtering can be continued recursively.

As already mentioned, in linear-Gaussian cases the Kalman filter is the exact solution for this filtering problem, see Exercise 3.7. Nevertheless, in the general case the posterior distributions cannot be analytically solved. Because of this, several numerical methods for the approximation of the posterior distribution have been developed. These methods are described in Section 3.5.

Optimal estimator

In practical applications, we usually are interested in finding an optimal state estimator and an estimate of the estimation error, rather than the full Bayesian posterior distribution. By "estimator" we mean a function whose arguments are the measurements and the initial state, and by "optimal" we mean that the estimator minimizes the expectation of some scalar cost function $L(\mathbf{x}_k - \hat{\mathbf{x}}_k)$, in short

$$\hat{\mathbf{x}}_k^{opt} = \operatorname{argmin}_{\hat{\mathbf{x}}_k} \mathrm{E}(L(\mathbf{x}_k - \hat{\mathbf{x}}_k)).$$

The cost function can be chosen with many different ways and thus we get different kinds of optimal estimators. It is usually assumed, as we shall, that the cost function is non-negative and that it has the value zero in the origin. Because

$$\mathrm{E}(L(\mathbf{x}_k - \hat{\mathbf{x}}_k)) = \mathrm{E}(\mathrm{E}(L(\mathbf{x}_k - \hat{\mathbf{x}}_k)|\mathbf{y}_{1:k})), \tag{3.18}$$

it suffices to find an estimator that minimizes the conditional expectation

$$\hat{\mathbf{x}}_k^{opt} = \operatorname{argmin}_{\hat{\mathbf{x}}_k} \mathrm{E}(L(\mathbf{x}_k - \hat{\mathbf{x}}_k)) = \operatorname{argmin}_{\hat{\mathbf{x}}_k} \mathrm{E}(L(\mathbf{x}_k - \hat{\mathbf{x}}_k)|\mathbf{y}_{1:k} = y_{1:k}), \tag{3.19}$$

for all values of the measurements $y_{1:k}$. A common estimator is the *least squares estimator*, that we get by minimizing the expectation value of the square of the error norm, that is,

$$L(\mathbf{x}_k - \hat{\mathbf{x}}_k) = \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|^2. \tag{3.20}$$

It can be shown that the estimator corresponding to the cost function (3.20) is the same as the expectation of the posterior

$$\hat{\mathbf{x}}_k^{\mathrm{MSE}} = \mathrm{E}(\mathbf{x}_k|\mathbf{y}_{1:k} = y_{1:k}),$$

(see Exercise 3.9). In fact, this same cost function is used for deriving the Kalman filter (Section 3.2). It is however good to remember that the Kalman filter minimizes the expectation (3.18) among linear estimators only, whereas the optimal estimators handled in this section minimize the conditional expectation (3.19) from the set of all estimators, not just linear ones.

Another common estimator is the *maximum a posteriori*, i.e. MAP estimator, then accordingly the estimator is

$$\hat{\mathbf{x}}_k^{\mathrm{MAP}} = \operatorname{argmax}_{x_k} p(x_k|y_{1:k}).$$

The MAPestimator minimizes the so-called *hit or miss* cost function $L = \lim_{\delta\downarrow 0} L_\delta$, where

$$L_\delta(\mathbf{x}_k - \hat{\mathbf{x}}_k) = \begin{cases} 1, & \|\mathbf{x}_k - \hat{\mathbf{x}}_k\| \geq \delta \\ 0, & \|\mathbf{x}_k - \hat{\mathbf{x}}_k\| < \delta \end{cases},$$

3.5 Numerical methods for filtering

This section presents some numerical methods for solving filtering problems. A hallmark for a good numerical approximation method is that there is a way to control the trade-off between accuracy and computational load. for example parameter N that can be increased in order to make the solution approach the correct one. In numerical Bayesian filtering, this parameter often is the number of the pairs $\{p_k^i(x), \omega_k^i\}$ $(i = 1, \ldots, N)$, with which we approximate the density function of the posterior distribution

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N} \omega_k^i p_k^i(x_k).$$

For instance, for the particle filter (Section 3.5.2) and for the point mass filter

$$p_k^i(x_k) = \delta(x_k - x_k^i),$$

where δ is the delta function (Dirac's delta measure). For the grid filter, the function $p_k^i(x_k)$ is the characteristic function of set A_i and for the Gaussian mixture filter $p_k^i(x_k)$ is the density function of a normal distribution.

The biggest difference between the particle filter and the point mass filter is that in the particle filter particles $\{x_k^i\}$ are chosen randomly whereas in the point mass filter they are deterministically set. The particle filter is based on Monte Carlo integration, described next.

3.5.1 Monte Carlo integration

Monte Carlo integration is based on the law of large numbers.

Theorem 7 (Strong law of large numbers). *Let $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$ be independent identically distributed random variables. Then*[*]

$$\mathbb{P}\left(\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i = \mu\right) = 1$$

if and only if the distribution has an expectation value and $\mathrm{E}(\mathbf{z}) = \mu$.

Write the integral to be computed in the form

$$I = \int g(z)dz = \int \frac{g(z)}{f(z)} f(z)dz \overset{\triangle}{=} \int h(z)f(z)dz, \tag{3.21}$$

where $f(z) > 0$ is the density function of a random variable $\mathbf{z}$. The random variable $\mathbf{z}$ can be chosen almost freely, so it is good to pick one from which it is convenient to draw random samples. Then according to the Strong law of large numbers (above), the sample mean

$$\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{z}_i) \tag{3.22}$$

[*] Then it is said that the sample mean *converges almost surely*. From almost certain convergence follows *convergence in the probability sense*, from which furthermore follows *convergence in the distribution sense*.

converges almost surely towards integral I (3.21). If $\mathrm{E}(h(z)^2)$ exists, then the variance of random variable $\bar{h}_n$ can be approximated

$$\mathrm{V}(\bar{h}_n) = \frac{1}{n}\int [h(z) - \mathrm{E}(h(z))]^2 f(z)dz \approx \frac{1}{n^2}\sum_{i=1}^{n}\left(h(\mathbf{z}_i) - \bar{h}_n\right)^2 \triangleq \sigma^2_{\bar{h}_n}. \tag{3.23}$$

Additionally, it can be shown that the quotient (compare to the central limit theorem)

$$\frac{\bar{h}_n - \mathrm{E}(h(z))}{\sqrt{\sigma^2_{\bar{h}_n}}}$$

approaches in a distribution sense the distribution of the random variable $\mathrm{N}(0,1)$.

Example 21 (Monte Carlo integration)**.** *Let a random variable be* $\mathbf{z} \sim \mathrm{N}(0,1)$ *and*

$$g(z) = \begin{cases} \frac{3}{4}(1-z^2), & |z| \leq 1 \\ 0, & |z| > 1 \end{cases}$$

When the number of the simulated random variables is $n = 10^4$*, one realization of Monte Carlo approximation of integral* $\int g(z)dz = 1$ *is about* 1.005*, equation (3.22) and the approximation of the variance of this approximation is about* $3.045 \cdot 10^{-5}$ *(3.23). The density function of the normal distribution corresponding to these values is visualized on the left in Figure 3.2. In the background of the density function there has been drawn a histogram of a thousand such Monte Carlo simulations. The histogram resembles a normal distribution whose expectation is one, just like in theory. In Figure 3.2 on the right is a visualization of the error of the Monte Carlo integration compared to the number* n *of random draws used. The line in the plot reflects the theoretic convergence rate* $O(n^{-\frac{1}{2}})$*.*

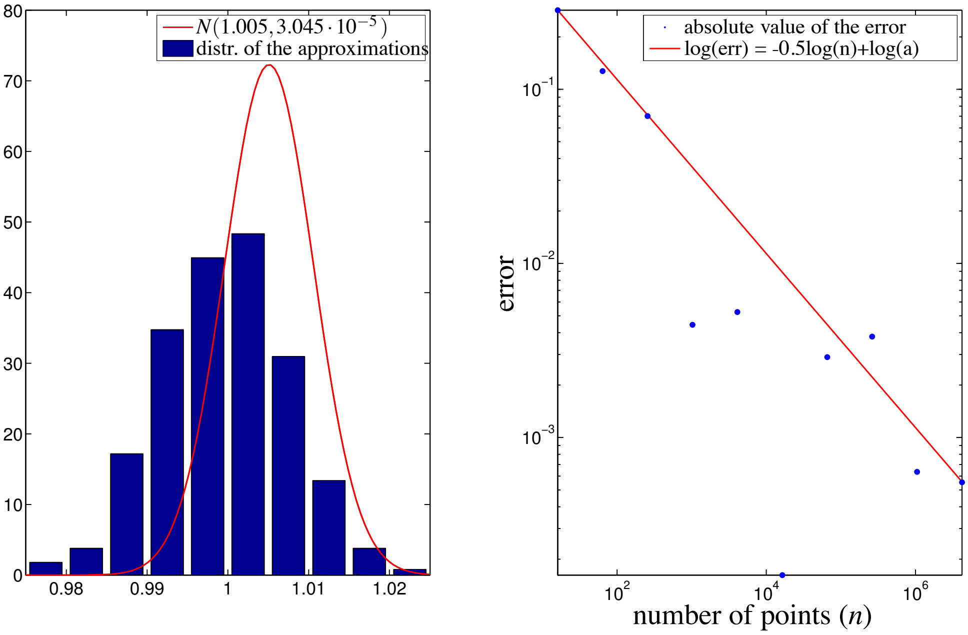


Figure 3.2: Results of the Monte Carlo simulation.

3.5.2 Particle filter

In this section, we describe different variants of particle filters [27]. A particle filter is based on the Monte Carlo method and it approximates prior and posterior distributions using samples, called particle clouds, of the distributions in question. One particle filter is given as the Algorithm 5. In fact, the algorithm in question can be considered as an algorithm for a family of particle filters, because by choosing different model parameters we get different kinds of particle filters. These choices are discussed in the next section. Additionally, it is good to remember that this is not the most general formulation for particle filter but there exists even more general or in some way differing particle filters.

Algorithm 5 Particle filter

- The state model: $\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$, $p_{\mathbf{w}_{k-1}}(w)$
- The measurement model: $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$, $p_{\mathbf{v}_k}(v)$
- The initial state and the measurements: $p_{\mathbf{x}_0}(x)$ and $y_{1:m} = \{y_1, y_2, \ldots, y_m\}$
- The number of particles and the proposal distributions: N and $g(x_k|x_{k-1}^i, y_k)$

In this algorithm $i = 1, \ldots, N$.

1. - Simulate samples x_0^i from distribution $p_{\mathbf{x}_0}(x)$
   - Set $\omega_0^i = \frac{1}{N}$ and $k = 1$
2. - Simulate particles x_k^i from distributions $g(x_k|x_{k-1}^i, y_k)$
   - Set $\omega_k^i = \omega_{k-1}^i \frac{p_{\mathbf{v}_k}(y_k - h_k(x_k^i)) p_{\mathbf{w}_{k-1}}(x_k^i - f_{k-1}(x_{k-1}^i))}{g(x_k^i|x_{k-1}^i, y_k)}$
3. Normalization: $\omega_k^i = \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^i}$
4. Compute point estimates, e.g.:
   - The posterior expectation: $\hat{x}_k \approx \sum_{i=1}^N \omega_k^i x_k^i$
   - The covariance matrix of the posterior: $\mathrm{P}_k \approx \sum_{i=1}^N \omega_k^i \left(x_k^i - \hat{x}_k\right) \left(x_k^i - \hat{x}_k\right)^T$
5. Resample when needed:
   - Simulate particles x_k^i from distribution $\sum_i^k \omega_k^i \delta(x_k - x_k^i)$
   - Set $\omega_k^i = \frac{1}{N}$
6. Stop, if $k = m$, otherwise set $k = k + 1$ and go back to Step 2.

Sequential importance sampling (SIS) is a simple particle filter and it is obtained from Algorithm 5 when the resampling (Step 5) is left out. However, this eventually causes all the weight to accumulate in just a few particles. Then all the other weights are practically zero, and a lot of computation capacity has to be spent without having any effect on the approximation of the posterior distribution. Because of this, in the SIR filter (sampling importance resampling) the resampling is done at every step. Additionally, prior distributions $p(x_k|x_{k-1}^i)$ are used as proposal distributions $g(x_k|x_{k-1}^i, y_k)$, so that the formula of the weights in step 2 of the algorithm simplifies into $\omega_k^i = p_{\mathbf{v}_k}(y_k - h_k(x_k^i))$.

In the SIS filter, resampling is not done at all, whereas in the SIR filter it is done at every time instant. Generally speaking, neither of these is the optimal way to operate. Different heuristics have been developed to decide whether it is worth resampling or not. One of these ways is to compute an approximation for the number of effective samples

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^{N} (\omega_k^i)^2},$$

and resample if it is smaller than a certain threshold value.

The resampling can be done in many ways. Algorithm 6 introduces *systematic resampling*. A computationally heavier resampling algorithm (so-called multinomial resampling) is obtained when in the systematic resampling the algorithm comparison points $z_i \sim \text{Uniform}(0, 1]$ are simulated each time. Systematic resampling takes one sample from each interval $\left(\frac{i-1}{N}, \frac{i}{N}\right]$. This guarantees that if the weight of a particle is $\omega_k^j \geq \frac{l}{N}$, where l is a natural number, then the corresponding particle appears at least l times in the resampled particle set. This also holds for the so-called stratified resampling where one comparison point is simulated from each interval $\left(\frac{i-1}{N}, \frac{i}{N}\right]$ and otherwise it functions the same as Algorithm 6.

Algorithm 6 Systematic resampling

```
• Particles and weights $\{x_k^i, \omega_k^i\}$, where $i = 1, \ldots, N$.

1. Simulate the starting point: $z_1 \sim \text{Uniform}(0, \frac{1}{N}]$ and set $i = 1$.

2. Compute current comparison point $z_i = z_1 + (i-1)\frac{1}{N}$.

3. Set $x_k^i = x_k^j$, where $j$ is set in such a way that $\sum_{l=1}^{j-1} \omega_k^l < z_i \leq \sum_{l=1}^{j} \omega_k^l$.

4. If $i = N$ set $\omega_k^i = \frac{1}{N}$ for all $i$ and stop, otherwise set $i = i + 1$ and go back to Step 2.
```

The proposal distribution $g(x_k|x_{k-1}^i, y_k)$ was mentioned earlier in connection with the SIR filter, which uses prior distributions $p(x_k|x_{k-1}^i)$ as proposal distributions. Prior distribution is a popular choice for the proposal distribution, although they are not the optimal choice when an effective sample size is considered. The optimal choice would be to use posterior distributions $p(x_k|x_{k-1}^i, y_k)$ as proposal distributions. However, usually they are not known*, and then we often end up using either the prior distributions or the approximations of the posterior distributions which can be computed for example with the nonlinear Kalman filter (Section 3.3).

Example 22 (PF and EKF)**.** *Figure 3.3 shows a positioning example using particle filter (PF) and extended Kalman filter (EKF). The system state is four-dimensional, two location coordinates and two velocity coordinates, and state model parameters* Φ *and* Q *are the same as in the Kalman filter example (3.14). A measurement is taken at one second intervals as a*

*Because they are the ones we are seeking.

range measurement from two base stations, which have also been drawn in the figure. Then the measurement model is

$$\mathbf{y}_k = \begin{bmatrix} \|\mathbf{x}_{bs_1} - \mathbf{x}_k\| \\ \|\mathbf{x}_{bs_2} - \mathbf{x}_k\| \end{bmatrix} + \mathbf{v}_k,$$

where $\mathbf{x}_{bs_1} = \begin{bmatrix} -250 \\ 0 \end{bmatrix}$, $\mathbf{x}_{bs_2} = \begin{bmatrix} 250 \\ 0 \end{bmatrix}$ *and* $\mathbf{v}_k \sim \mathrm{N}(0, 10^2 \mathrm{I})$. *The initial state of the system is*

$$\mathbf{x}_0 \sim \mathrm{N}\left(\begin{bmatrix} -150 \\ -30 \\ 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 20^2 & 0 & 0 & 0 \\ 0 & 20^2 & 0 & 0 \\ 0 & 0 & 5^2 & 0 \\ 0 & 0 & 0 & 5^2 \end{bmatrix} \right).$$

The location estimates given by the filters have been drawn in the figure, which are in this case approximations of the expectations of the posterior distributions. At time instants $\{0, 10, 20, 30, 40\}$ *the ellipses corresponding to the covariance matrices given by the EKF have been drawn. The ellipses are defined in such a way that if the posterior distributions were normal with parameters given by the EKF, then the ellipse would represent the level curve of the density function of the posterior distribution such that with about* 68% *probability the state is inside the ellipse.*

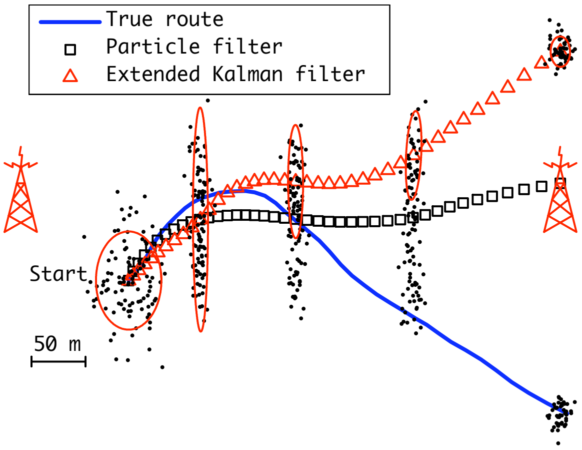


Figure 3.3: *Figure of the situation of the example, more detailed explanation is found in the text.*

The particle filter used is the SIR filter (p. 54) with a million particles. The particle clouds have been visualized at time instants $\{0, 10, 20, 30, 40\}$ *with a hundred particles, which have been got from the original set of one million particles by using systematic resampling (algorithm 6). Because the number of particles is relatively large, it can be thought that the particle filter represents some kind of reference filter in the example.*

From the figure we see that at the last time instant ($t = 40$)*, the posterior distribution clearly has two separate peaks. The EKF has started following one of these peaks, which is very characteristic for the EKF. In this case the EKF has “chosen” the wrong peak and then we say that the EKF has gone astray. Although the EKF gives completely false results, the covariance matrix shows that the EKF trusts very much the estimate in question. After this, the EKF does not easily “find” the correct location even when measurements from additional sources become available and the bimodal (two-peaked) posterior becomes unimodal. As a rule of thumb, we can say that it is worth applying the EKF to the system if the posterior distributions resemble normal distribution i.e. they are unimodal and the tails of the distribution rapidly decay towards zero, but in this kind of two-peaked case applying EKF starts to be quite risky. This bimodal situation would completely change if there were range* *measurements from one more base station or if the route did not go near the line connecting the base stations.*

Exercises

3.1. Show that the mean square error matrix of prediction $\hat{\mathbf{y}}_k = \mathrm{H}_k\hat{\mathbf{x}}_k^-$ of measurement $\mathbf{y}_k = \mathrm{H}_k\mathbf{x}_k + \mathbf{v}_k$ is $\mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k$.

3.2. Assume that matrix R is symmetric positive definite and matrix P is symmetric positive semi-definite. Show that

$$(\mathrm{I}-\mathrm{KH})\mathrm{P}(\mathrm{I}-\mathrm{KH})^T + \mathrm{KRK}^T = \mathrm{P} - \mathrm{AA}^T + (\mathrm{KB}-\mathrm{A})(\mathrm{KB}-\mathrm{A})^T,$$

where $\mathrm{B} = (\mathrm{HPH}^T + \mathrm{R})^{\frac{1}{2}}$ and $\mathrm{A} = \mathrm{PH}^T\mathrm{B}^{-1}$.

3.3. Let A be an arbitrary real matrix. Show that the zero matrix is the unique solution for the problem $\mathrm{argmin}_{\mathrm{A}}\,\mathrm{tr}(\mathrm{AA}^T)$.

3.4. Let prior distribution of the position $\mathbf{x} \sim \mathrm{N}(6,8)$ and measurement model $\mathbf{y} = 2\mathbf{x} + 4\mathbf{v}$, where random variable $\mathbf{v} \sim \mathrm{N}(0,1)$ is independent of the prior distribution. What is the conditional distribution of the position, when measurement $\mathbf{y} = 14$? Here prior and measurement are given at the same instant.

3.5. Let prior distribution of the position

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} 10 \\ 24 \end{bmatrix}, \begin{bmatrix} 60 & 0 \\ 0 & 20 \end{bmatrix}\right)$$

and measurement model $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{v}$, where random variable $\mathbf{v} \sim \mathrm{N}(0,10)$ is independent of the prior distribution. What is the conditional distribution of the position, when measurement $\mathbf{y} = 43$? Here prior and measurement are given at the same instant.

3.6. For a partitioned matrix it holds that (check it)

$$\begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{C} & \mathrm{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathrm{A}-\mathrm{BD}^{-1}\mathrm{C})^{-1} & -\mathrm{A}^{-1}\mathrm{B}(\mathrm{D}-\mathrm{CA}^{-1}\mathrm{B})^{-1} \\ -\mathrm{D}^{-1}\mathrm{C}(\mathrm{A}-\mathrm{BD}^{-1}\mathrm{C})^{-1} & (\mathrm{D}-\mathrm{CA}^{-1}\mathrm{B})^{-1} \end{bmatrix},$$

if the required inverse matrices exist. On the other hand, we know that the inverse matrix of a symmetric matrix is symmetric. Applying this information to matrix

$$\begin{bmatrix} \mathrm{P}^{-1} & \mathrm{H}^T \\ \mathrm{H} & -\mathrm{R} \end{bmatrix}$$

may be of help in the next task. Additionally, let A and B be symmetric positive definite matrices. Now, if $\mathrm{A} \leq \mathrm{B}$ then $\mathrm{B}^{-1} \leq \mathrm{A}^{-1}$ [11, Corollary 3.3.4.].

(a) Show that $(\mathrm{P}^{-1}+\mathrm{H}^T\mathrm{R}^{-1}\mathrm{H})^{-1} = \mathrm{P}-\mathrm{P}\mathrm{H}^T(\mathrm{H}\mathrm{P}\mathrm{H}^T+\mathrm{R})^{-1}\mathrm{H}\mathrm{P}$, when $\mathrm{P}>0$ and $\mathrm{R}>0$.

(b) Show that in linear Gaussian case posterior covariance matrix is "increasing". That is, show that if $0<\mathrm{P}_1\leq\mathrm{P}_2$ then $0<\mathrm{P}_1^+\leq\mathrm{P}_2^+$, where $\mathrm{P}_i^+=(\mathrm{I}-\mathrm{K}\mathrm{H})\mathrm{P}_i$ and $\mathrm{K}=\mathrm{P}\mathrm{H}^T(\mathrm{H}\mathrm{P}\mathrm{H}^T+\mathrm{R})^{-1}$.

3.7. (a) Let $\mathbf{x}_{k-1}\sim\mathrm{N}(\hat{x}_{k-1},\mathrm{P}_{k-1})$, $\mathbf{w}_{k-1}\sim\mathrm{N}(0,\mathrm{Q}_{k-1})$ and $\mathbf{v}_k\sim\mathrm{N}(0,\mathrm{R}_k)$ be independent random variables. What is the distribution of the random variable

$$\begin{bmatrix}\mathbf{x}_k\\ \mathbf{y}_k\end{bmatrix}=\begin{bmatrix}\Phi_{k-1} & \mathrm{I} & 0\\ \mathrm{H}_k\Phi_{k-1} & \mathrm{H}_k & \mathrm{I}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{k-1}\\ \mathbf{w}_{k-1}\\ \mathbf{v}_k\end{bmatrix}?$$

(b) Let

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\sim\mathrm{N}\left(\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix},\begin{bmatrix}\Sigma_{xx} & \Sigma_{xy}\\ \Sigma_{yx} & \Sigma_{yy}\end{bmatrix}\right),$$

show that $\mathbf{x}|\mathbf{y}=y\sim\mathrm{N}(\bar{x}+\Sigma_{xy}\Sigma_{yy}^{-1}(y-\bar{y}),\Sigma_{xx}-\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})$.

(c) Apply the result of part (b) to the distribution from part (a), and compare the obtained result with the BLU-estimator introduced in Section 3.2.

3.8. Show that in one-dimensional case, the unscented transformation using the points given in formula (3.17) make the approximation (3.16) exact for a third degree polynomial. Naturally, here $p(x)$ is the density function of distribution $\mathrm{N}(\mu,\sigma^2)$. On what grounds is the choice $\kappa=2$ reasonable? (Hint: $\int x^4p(x)dx=\mu^4+6\mu^2\sigma^2+3\sigma^4$)

3.9. Show that the posterior expectation minimizes the expectation $\mathrm{E}(\|\mathbf{x}_k-\hat{\mathbf{x}}_k\|^2 \mid \| \mathbf{y}_{1:k}=y_{1:k})$.

3.10. Let prior distribution of the position $\mathbf{x}\sim\mathrm{Uniform}(0,3\pi)$. The measurement model is $\mathbf{y}=\sin(\mathbf{x})+\mathbf{v}$, where random variable $\mathbf{v}\sim\mathrm{Uniform}(-\frac{1}{2},\frac{1}{2})$ is independent of the prior distribution. What is the probability density function (pdf) of the posterior distribution, when measurement $y=\frac{1}{2}$?

3.11. Let pdf of the random variable $\mathbf{w}$ be $p_{\mathbf{w}}(w)=\mathrm{N}_{\mathrm{Q}}^{0}(w)$ and pdf of the random variable $\mathbf{x}$

$$p_{\mathbf{x}}(x)=\alpha\mathrm{N}_{\Sigma}^{\mu_1}(x)+(1-\alpha)\mathrm{N}_{\Sigma}^{\mu_2}(x),$$

where $0\leq\alpha\leq1$ and

$$\mathrm{N}_{\Sigma}^{\mu}(x)=\frac{\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)}{\sqrt{\det(2\pi\Sigma)}}.$$

The random variables $\mathbf{w}$ and $\mathbf{x}$ are independent. Let the random variable $\mathbf{y}=\mathrm{F}\mathbf{x}+\mathbf{w}$, where F is a constant matrix.

(a) Calculate $\mathrm{E}(\mathbf{y})$.

(b) Calculate pdf of the random variable $\mathbf{y}$.

Lemma: If $\Sigma_1, \Sigma_2 > 0$ then $\mathrm{N}^{\mu}_{\Sigma_1}(x)\,\mathrm{N}^{\mathrm{H}x}_{\Sigma_2}(y) = \mathrm{N}^{\bar{\mu}}_{\Sigma_3}(x)\,\mathrm{N}^{\mathrm{H}\mu}_{\Sigma_4}(y)$, where $\bar{\mu} = \mu + \mathrm{K}(y - \mathrm{H}\mu)$, $\Sigma_3 = (\mathrm{I} - \mathrm{KH})\Sigma_1$, $\mathrm{K} = \Sigma_1 \mathrm{H}^T \Sigma_4^{-1}$, $\Sigma_4 = \mathrm{H}\Sigma_1\mathrm{H}^T + \Sigma_2$ and $\mathrm{N}^{\mu}_{\Sigma}(x) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)}{\sqrt{\det(2\pi\Sigma)}}$.

Computer exercises

3.12. Implement the Kalman filter (KF), whose specification is

```
%  [X,P] = kalman(x0,P0,F,Q,Y,H,R)
%
%  x_{k+1} = Fx_{k}+w
%    y_{k} = Hx_{k}+v
%
% IN: x0 = initial state at time 0 (R^n)
%     P0 = initial covariance matrix
%     F  = state transformation matrix
%     Q  = covariance matrix of process noise (w)
%     Y  = measurements Y=[y_1,...,y_k](R^(m \times k),
%     H  = ''measurement model''
%     R  = covariance matrix of measurements errors (v)
%
%OUT:    X  = [x_0,x_1,...,x_k], where x_i is estimate at time i.
% P(:,:,i)  = MSE matrix at time i

function [X,P] = kalman(x0,P0,F,Q,Y,H,R)
```

There are data in web page `http://www.students.tut.fi/~aliloytt/menetelmat/harkat.html` that you can test your filter. Visualize these tests.

3.13. Compute by using Monte Carlo integration the expression

$$I = \frac{\int_A \frac{x}{2\pi} \exp\left(-\frac{1}{2}x^T x\right) dx}{\int_A \frac{1}{2\pi} \exp\left(-\frac{1}{2}x^T x\right) dx},$$

where $A = [0,2] \times [0,2]$. Additionally, study the convergence of Monte Carlo simulations towards the exact value

$$I = \frac{1 - \exp(-2)}{\sqrt{2\pi}(\Phi(0) - \Phi(-2))} \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

where Φ is the cumulative function of distribution $\mathrm{N}(0,1)$. The integral in question represents the expectation of a posterior distribution whose prior distribution is $\mathrm{N}(0,\mathrm{I})$ and the measurement is the information that the state is inside the area A.

3.14. Implement two filters of this group {EKF, UKF, PF}, whose specifications is

```
%  [X,P,...] = filter(x0,P0,F,Q,Y,S,R)
%
%  x_{k+1} = Fx_{k}+w
%    y_{k} = h(x_{k})+v,
% where (h(x_{k}))_i is range measurement from base station s_i
%
% IN: x0 = initial state at time 0 (R^n)
%     P0 = initial covariance matrix
%     F  = state transformation matrix
%     Q  = covariance matrix of process noise (w)
%     Y  = measurements Y=[y_1,...,y_k](R^(m \times k),
%     S  = coordinates of base stations  [s_1,...,s_l]
%     R  = covariance matrix of measurements errors (v)
%
%OUT:    X  = [x_0,x_1,...,x_k], where x_i is estimate at time i.
% P(:,:,i)  = MSE matrix at time i

function [X,P,...] = filter(x0,P0,F,Q,Y,S,R)
```

There are data in web page `http://www.students.tut.fi/~aliloytt/menetelmat/harkat.html` that you can test your filter. Visualize these tests.

Bibliography

[1] ASH, R. B., AND GARDNER, M. F. *Topics in Stochastic Processes*, vol. 27 of *Probability and Mathematical Statistics*. Academic Press, 1975.

[2] BANCROFT, S. An algebraic solution of the GPS equations. *IEEE Transactions on Aerospace and Electronic Systems 21*, 7 (1986), 56–59. `http://ieeexplore.ieee.org/`.

[3] BAR-SHALOM, Y., LI, R. X., AND KIRUBARAJAN, T. *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.

[4] BORTZ, J. A new mathematical formulation for strapdown inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems 7*, 1 (1971), 61–66. `http://ieeexplore.ieee.org/`.

[5] BROWN, R. G. *Introduction to Random Signal Analysis and Kalman Filtering*. John Wiley & Sons, 1983.

[6] DAM, E. B., KOCH, M., AND LILLHOLM, M. Quaternions, interpolation and animation. Technical Report DIKU-TR-98/5, University of Copenhagen, 1998. `http://www.itu.dk/people/erikdam/DOWNLOAD/98-5.pdf`.

[7] FANG, B. T. Simple solutions for hyperbolic and related position fixes. *IEEE Transactions on Aerospace and Electronic Systems 26*, 5 (1990), 748–753. `http://ieeexplore.ieee.org/`.

[8] FARRELL, J. A., AND BARTH, M. *The Global Positioning System & Inertial Navigation*. McGraw-Hill, 1999.

[9] JAZWINSKI, A. H. *Stochastic Processes and Filtering Theory*, vol. 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[10] JHS. Public government recommendation JHS 153: ETRS89 coordinates in finland, 2006. `http://docs.jhs-suositukset.fi/jhs-suositukset/JHS153/JHS153.pdf`.

[11] KALEVA, O. Matemaattinen tilastotiede. Hand-out, TUT, Institute of Mathematics, November 2005. `http://www.tut.fi/~kaleva/`.

[12] KALEVA, O. Matemaattinen optimointi 1. Hand-out, TUT, Institute of Mathematics, January 2006. `http://www.tut.fi/~kaleva/`.

[13] KALEVA, O. Stokastiset prosessit. Hand-out, TUT, Institute of Mathematics, January 2007. http://www.tut.fi/~kaleva/.

[14] KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82 (1960), 35–45.

[15] KAPLAN, E. D., Ed. *Understanding GPS: principles and applications*. Artech House, Norwood, 2005.

[16] KINCAID, D., AND CHENEY, W. *Numerical analysis*, second ed. Brooks/Cole Publishing Company, Pacific Grove, California, 1996.

[17] LEVA, J. L. An alternative closed-form solution to the GPS pseudo-range equations. *IEEE Transactions on Aerospace and Electronic Systems 32*, 4 (1996), 1430–1439. http://ieeexplore.ieee.org/.

[18] MÄKI-MARTTUNEN, T. Stokastiset ja tavalliset differentiaaliyhtälöt inertiapaikannuksessa. M.Sc. thesis, Tampere University of Technology, Dec 2008.

[19] MAYBECK, P. S. *Stochastic Models, Estimation, and Control*, vol. 141 of *Mathematics in Science and Engineering*. Academic Press, 1979.

[20] MISRA, P., AND ENGE, P. *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.

[21] PARKINSON, B., AND SPILKER, J., Eds. *Global Positioning System: Theory and Applications Volume I.* Charles Stark Draper Laboratory, Inc., Cambridge, 1996.

[22] PESONEN, H. Numerical integration in Bayesian positioning. M.Sc. thesis, Tampere University of Technology, June 2006. http://math.tut.fi/posgroup/pesonen_mscth.pdf.

[23] PHATAK, M., CHANSARKAR, M., AND KOHLI, S. Position fix from three GPS satellites and altitude: a direct method. *IEEE Transactions on Aerospace and Electronic Systems 36*, 1 (1999), 350–354. http://ieeexplore.ieee.org/.

[24] POHJOLAINEN, S. Matrix algebra 1. Lecture Notes, TUT, Institute of Mathematics, 2004. http://matpc41.ee.tut.fi/PNF:byName:/ML02/.

[25] POOLE, D. *Linear Algebra*, second ed. Thomsom Brooks/Cole, 2006.

[26] PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., AND VETTERLING, W. T. *Numerical recipes: the art of scientific computing*. Cambridge University Press, Cambridge, 1986. http://www.numerical-recipes.com/.

[27] RISTIC, B., ARULAMPALAM, S., AND GORDON, N. *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Artech House, Boston, London, 2004.

[28] SAVAGE, P. Strapdown inertial navigation integration algorithm design. *Journal of Guidance, Control and Dynamics 21*, 1-2 (1998).

[29] SAVAGE, P. G. Strapdown inertial navigation. Lecture Notes, 1997.

[30] Stark, H., and Woods, J. W. *Probability, random processes, and estimation theory for engineers*. Prentice Hall, Englewood Cliffs, 1994.

[31] Vauhkonen, M., Karjalainen, P., and Tossavainen, O.-P. Estimointiteoria. Lecture notes, University of Kuopio, 2006. `http://venda.uku.fi/studies/kurssit/ETE/`.

[32] WGS84. Department of defence world geodetic system 1984. Tech. rep., National Imagery and Mapping Agency, 2000. `ftp://164.214.2.65/pub/gig/tr8350.2/wgs84fin.pdf`.

[33] Williams, D. *Probability with Martingales*, sixth ed. Cambridge University Press, Cambridge University, 1991.