Universität Bremen*

Modeling and System Identification

— Skript zur Vorlesung —

Jürgen Pannek

Dynamics in Logistics
Fachbereich 04: Produktionstechnik
Universität Bremen

*EXZELLENT.

Foreword

This script originates from a correspondent lecture held during the summer term 2014 at the University of Bremen. The central aims of the lecture are the introduction of system identification techniques for both static and dynamic systems. In particular, we discuss

- Basic stochastic definitions,
- Least square estimator,
- Maximum likelihood and Bayes estimator,
- Dynamic system models, and
- Kalman filtering.

Such models are used, e.g., for the control, state estimation or prediction of logistic, mechatronic or economic systems. At the end of the lecture, students should understand the concepts, know basic formulas, be able to comprehend and interpret input and output of the methods and to make a suitable choice between the presented methods.

Parts of the scripts are based on the script of Prof. Schoukens [**?**] as well as the books [**?**, **?**], which will be used without further notice. Additional useful information may be found in [**?**, **?**, **?**].

Contents

Chapter 1

Introduction

Within this chapter, we give a brief and intuitive introduction to identification theory. To this end, we use a simple example to illustrate pitfalls associated with a model built from noisy measurements. Additionally, we give an overview of the identification process and introduce a statistical characterization of the parameters.

1.1 Identifying identification

What is identification and how can we define it? Intuitively, we all experienced identification by learning to control our actions using prediction of the effect of these actions. These predictions are based on a model, which we fit to reality using our past experiences. This can be very simple, i.e. "I push a ball, then it rolls", but may also culminate in the design of very complicated systems such as airplanes, mobile phones or weather forecast. What we also see here is that we always focus on those aspects we are interested in and do not try to describe all of reality. Hence, the problem is split into two parts,

- the system, which describes what we are interested in, and
- the environment, which contains everything else.

Interactions between system and environment are described by inputs and outputs.

Here, we will particularly focus on the quantitative analysis, i.e. we will not consider qualitative results such as "a ball will roll downhill". Instead, we will utilize laws and models, which may, e.g., be given by physics or a supply chain, and describes at least some part of our impression of reality. Fitting these laws or models to our observations creates new problems:

- For one, we consider noisy measurements. In this context, noisy means that if we take a measurement, e.g. length, weight, time etc., then errors occur since the instruments we use are not perfect.
- And secondly, our laws and models are imperfect as reality is far more complex than the rules we apply. They also show a stochastic behavior, which makes it impossible to predict exactly their output.

To still identify the system, we split the model into a deterministic and a stochastic part. The deterministic aspects are captured by the mathematical system model. These are complemented by the stochastic behavior, which modeled as a noise distortion. Hence, the aim of identification theory is the following:

> Identification theory provides a systematic approach to fit the mathematical model to the deterministic part as well as possible, and to eliminate the noise distortions as much as possible.

Before defining the terms system and goodness of fit, we motivate and illustrate many of the aspects and problems in identification theory by a simple example.

1.2 A simple example

Using two electric circuits as shown in Figure 1.1, we pass a constant but unknown current through the resistor. The voltage u across the resistor and the current i through it are measured using a voltmeter and an ampere meter, where the input impedance of the voltmeter is chosen large compared with the unknown resistor to ensure that all the measured current passes through the resistor.

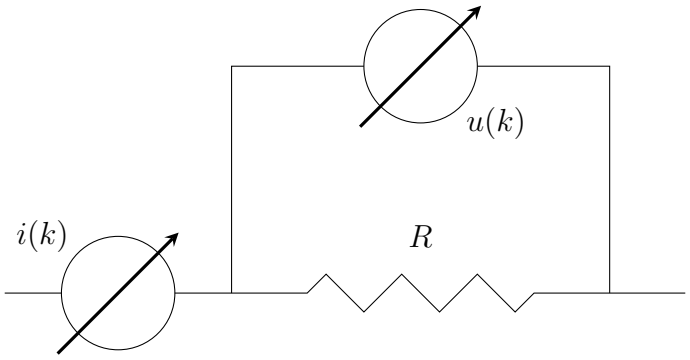
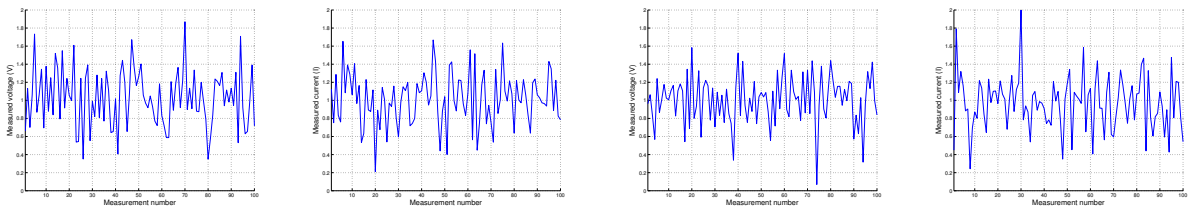


Figure 1.1: Measurement of a resistor.

Two sets of measurements $u(k)$, $i(k)$ with $k = 1, 2, \ldots, N$ are taken and called group A and B, cf. Figure 1.2, and the resistances $r(k)$, $k = 1, 2, \ldots, N$ are computed via $r(k) = u(k)/i(k)$, see Figure 1.3 for respective results.



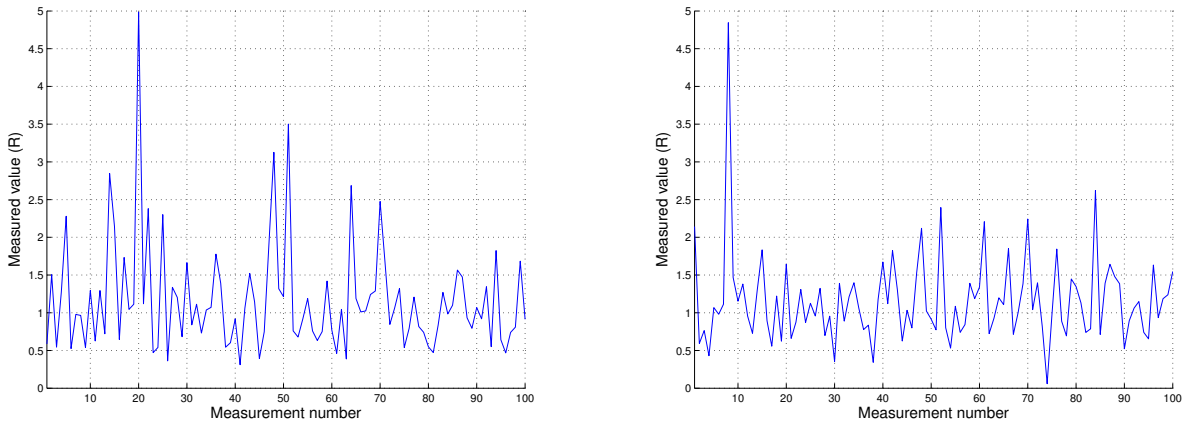
(a) Group A of measurements

(b) Group B of measurements

Figure 1.2: Measurement values for two groups

Since the measurements are very noise, different estimators can be analyzed:

$$\hat{R}_{\mathrm{SA}}(N) = \frac{1}{N} \sum_{k=1}^{N} \frac{u(k)}{i(k)} \tag{1.1}$$



(a) Computed resistance from group A

(b) Computed resistance from group B

Figure 1.3: Computed resistances from measurement groups

$$\hat{R}_{\mathrm{EV}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N} u(k)}{\frac{1}{N}\sum_{k=1}^{N} i(k)} \tag{1.2}$$

$$\hat{R}_{\mathrm{LS}}(N) = \operatorname*{argmin}_{R\in\mathbb{R}} \sum_{k=1}^{N} (R\cdot i(k) - u(k))^2 \tag{1.3}$$

Within these estimators, N indicates the number of used measurements. Note that the three estimators result in the same estimate on noiseless data. The first estimator, the **S**imple **A**pproach, averages the quotients of voltage and current measurements. The second estimator, the **E**rror-in-**V**ariable approach, averages the voltages and currents first and then computes the quotient. And the last estimator, the **L**east **S**quare approach, computes the minimal distance linear function between voltage and current pairs in the 2-norm.

To compute the latter, we set $f(R) := \sum_{k=1}^{N} (R\cdot i(k) - u(k))^2$ and minimize it, i.e., we are looking for a value R such that

$$\frac{\partial f(R)}{\partial R} = 0.$$

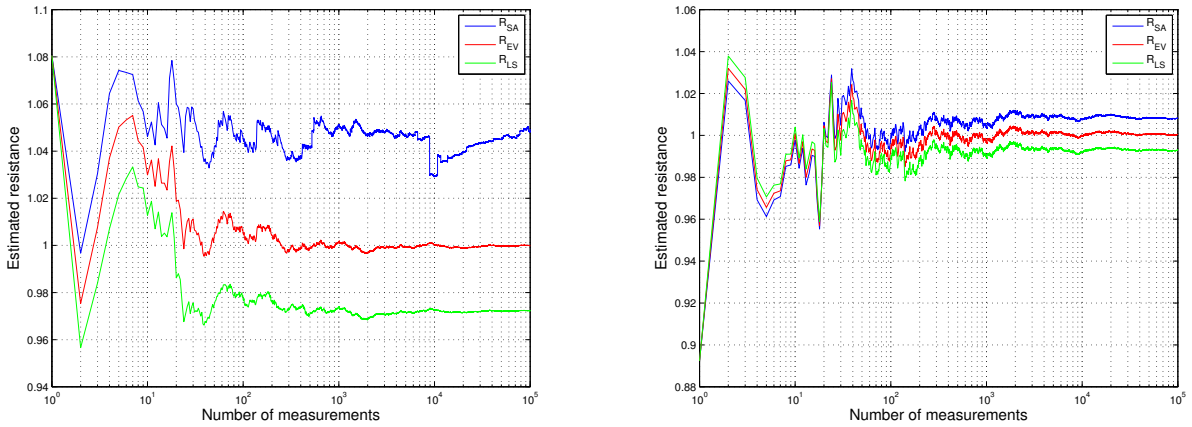
From the definition of $f(R)$, we obtain

$$\begin{aligned}\frac{\partial f(R)}{\partial R} &= \sum_{k=1}^{N} 2\cdot(R\cdot i(k) - u(k))\cdot i(k)\\ &= 2\cdot R\cdot\sum_{k=1}^{N} i(k)^2 - 2\cdot\sum_{k=1}^{N} u(k)\cdot i(k).\end{aligned}$$

Hence, we have

$$\hat{R}_{\mathrm{LS}}(N) = \frac{\sum_{k=1}^{N} u(k) \cdot i(k)}{\sum_{k=1}^{N} i(k)^2}. \tag{1.4}$$

Utilizing these estimation formulas, we can compute the estimated resistances as displayed in Figure 1.3. From this figure, we can make several observations:



(a) Estimated resistance from group A

(b) Estimated resistance from group B

Figure 1.4: Estimated resistances from measurement groups with $\hat{R}_{\mathrm{SA}}$ in blue, $\hat{R}_{\mathrm{EV}}$ in red and $\hat{R}_{\mathrm{LS}}$ in green.

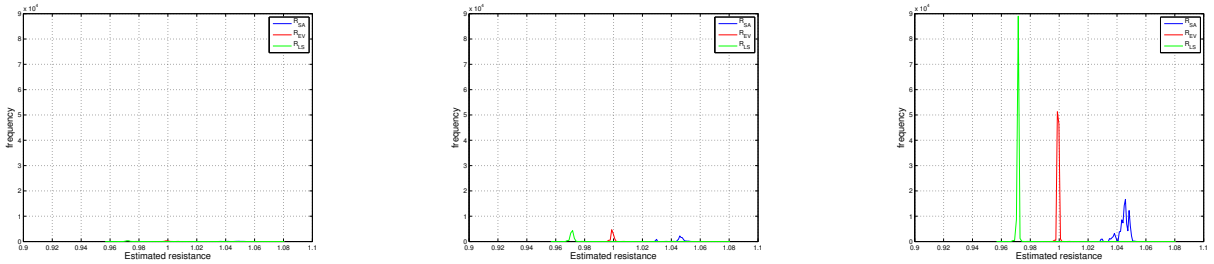
1. All estimators have large variations for small values of N, and – except for $\hat{R}_{\mathrm{SA}}$ from group A – seem to converge to an asymptotic value for large values of N. This corresponds to the intuitively expected behavior: if a large number of data points are processed, then we should be able to eliminate the noise influence due to the averaging effect.

2. The asymptotic values of the estimators depend on the kind of averaging technique that is used. this shows that there is a serious problem: at least two out of the three methods converge to a wrong value. It is not even certain that any one of the estimators is doing well. This is quite catastrophic: even an infinite amount of measurements does not guarantee that the exact value is found.

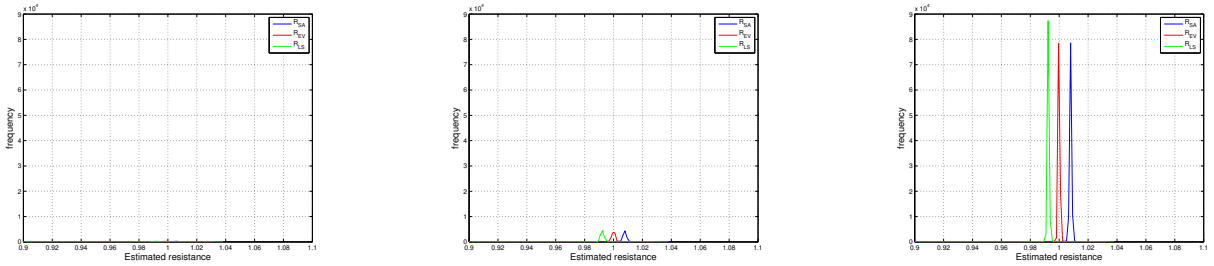
3. The $\hat{R}_{\mathrm{SA}}$ from group A behaves very strangely. Instead of converging to a fixed value, it jumps irregularly up and down.

These observations clearly indicate that a good theory is needed to explain and understand the behavior of candidate estimators. This will allow us to make a sound selection out of many possibilities and to indicate in advance if a method is prone to serious shortcomings before running expensive experiments.

To gain more insight, we can plot approximations of the probability density functions based on the data, cf. Figure 1.5. From this figure, we observe the following:



(a) Observed probability density functions for group A



(b) Observed probability density functions for group B

Figure 1.5: Observed probability density functions for groups. From left to right $N = 1000$, $N = 10000$ and $N = 100000$ with $\hat{R}_{\mathrm{SA}}$ in blue, $\hat{R}_{\mathrm{EV}}$ in red and $\hat{R}_{\mathrm{LS}}$ in green.

1. For small values of N, the estimates are widely scattered. As the number of processed measurements increases, the probability density function becomes more concentrated.

2. The estimates $\hat{R}_{\mathrm{LS}}$ are less scattered than $\hat{R}_{\mathrm{EV}}$ and $\hat{R}_{\mathrm{SA}}$, and odd behavior for $\hat{R}_{\mathrm{SA}}$ in group A appears again. The distribution of this estimate does not contract to a single value for growing values of N for group A, while it does for group B.

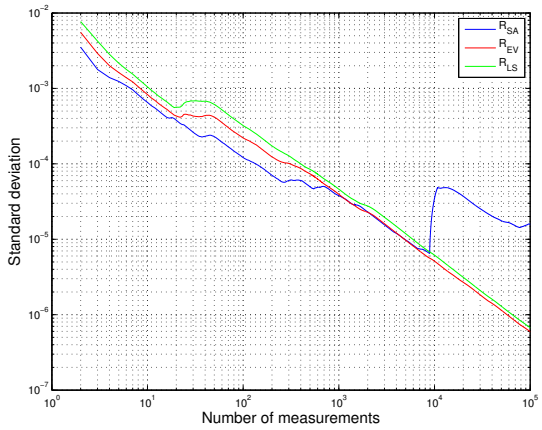
3. It is clearly visible that the distributions are concentrated around different values.

The choice of the estimator is not yet clear. And additionally, there seems to be a major problem with the measurements of group A, which was observed via $\hat{R}_{\mathrm{SA}}$. In order to quantify the scattering of the estimates, in particular of $\hat{R}_{\mathrm{SA}}$, the standard deviation can be calculated, cf. Figure 1.6. Here, we observe that the standard deviation decreases monotonically with N – except for $\hat{R}_{\mathrm{SA}}$ of group A. Moreover, the decrease is proportional to $1/\sqrt{N}$, which is the rule of thumb for the uncertainty on an averaged quantity obtained from independent measurements. Additionally, the uncertainty depends on the estimator.

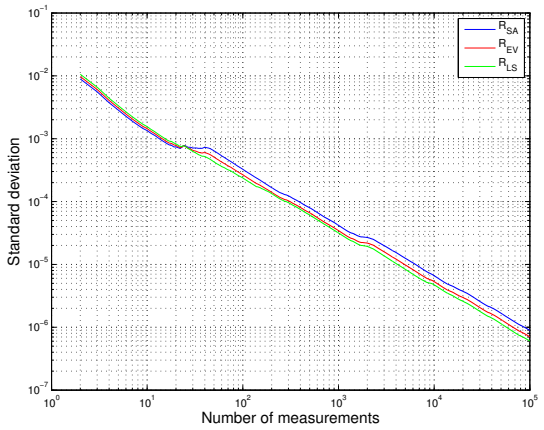
Regarding the strange behavior of $\hat{R}_{\mathrm{SA}}$ of group A, we reconsider the measurement data displayed in Figure 1.2 and compute respective histograms, cf. Figure 1.7. Due to possibly occuring zero values for the current in group A, we obtain a drastic increase in the estimation using the simple approach. This is due to a division by (almost) zero. In group B, such a case does not exist.

This example shows that there is a clear need for methods, which can generate and select between different estimators. We also like to note that although the noise on the measurements in group A is completely different distributed, the resulting estimation, e.g. by $\hat{R}_{\mathrm{EV}}$ and $\hat{R}_{\mathrm{LS}}$, seem to be the same as in group B.

Before coming to a structured approach of identifying a process in Chapter 2, we first need to introduce some notation and make us familiar with basic definitions.

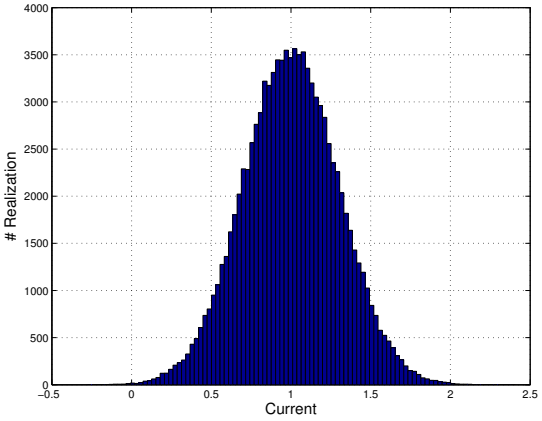


(a) Observed standard deviation for group A

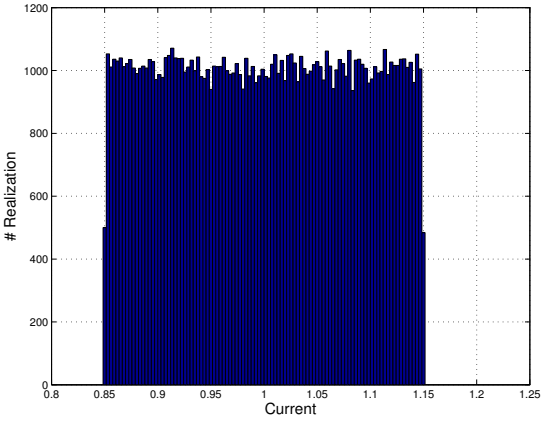


(b) Observed standard deviation for group B

Figure 1.6: Observed standard deviation for groups. From left to right $N = 1000$, $N = 10000$ and $N = 100000$ with $\hat{R}_{\mathrm{SA}}$ in blue, $\hat{R}_{\mathrm{EV}}$ in red and $\hat{R}_{\mathrm{LS}}$ in green.



(a) Histogram for $i(\cdot)$ for group A



(b) Histogram for $i(\cdot)$ for group B

Figure 1.7: Comparison of histograms for the current $i(\cdot)$

1.3 Basic definitions

In order to analyze estimators, we first need to classify them. As we have seen in the previous sections, estimators are obtained as functions of a finite number of noisy measurements. Hence, they are stochastic variables, just as the noisy measurements are. To characterize a stochastic variable completely, we require the respective probability density function. In practice, however, it is very hard to derive that function. Yet, the behavior of the estimates can be described by a few numbers, i.e. the mean value and the covariance, which may be seen as the location and dispersion of the estimate.

To formally introduce these numbers, we first require the notion of a probability space:

Definition 1.1 (Probability space)
Consider a set Ω, a set of subsets $\mathcal{F} \subseteq 2^{\Omega}$ and a function $\mathrm{P} : \mathcal{F} \to [0,1]$. Then, we call the triple $(\Omega, \mathcal{F}, \mathrm{P})$ a probability space if

- the sample space Ω is a non–empty set,
- the σ–algebra $\mathcal{F}$ of events satisfies
  - $\mathcal{F}$ contains the empty set, i.e.
$$\emptyset \in \mathcal{F},$$
  - $\mathcal{F}$ is closed under complements, i.e.
$$A \in \mathcal{F} \implies \Omega \setminus A \in \mathcal{F},$$
  - $\mathcal{F}$ is closed under countable unions, i.e.
$$A_i \in \mathcal{F}\ \forall i \in \{1,2,\ldots,k\}, k < \infty \implies \bigcup_{i \in \{1,2,\ldots,k\}} A_i \in \mathcal{F}$$
- the probability measure P satisfies
  - P is countably additive, i.e.
$$A_i \in \mathcal{F}\ \forall i \in \{1,2,\ldots,k\}, k < \infty \text{ with } A_i \cap A_j = \emptyset\ \forall i,j \in \{1,2,\ldots,k\}, i \neq j$$
$$\implies \mathrm{P}\left(\bigcup_{i \in \{1,2,\ldots,k\}} A_i \right) = \sum_{i \in \{1,2,\ldots,k\}} \mathrm{P}\left(A_i\right),$$
  - the measure of the sample space Ω is one, i.e.
$$\mathrm{P}\left(\Omega\right) = 1.$$

In short, a probability space is a measure space, but with the additional property that the measure of the whole space is equal to one. Secondly, we require so called random variables:

Definition 1.2 (Random variable)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ and a measurable space E with σ–algebra $\mathcal{E}$ of E. Then we call a function $X : \Omega \to E$ a random variable if

$$\forall B \in \mathcal{E} :\ X^{-1}(B) \in \mathcal{F}, \quad \text{where } X^{-1}(B) := \left\{\omega \in \Omega \mid X(\omega) \in B\right\}.$$

Hence, a random variable is a function, which allows us to use a more comfortable description of properties or measurements of a sample, i.e. if B is an interval $[a,b]$ or the property "lottery player", then we identify the corresponding event $X^{-1}(B)$ in the σ–algebra $\mathcal{F}$.

Now, we can introduce the expected value, sometimes also called mean, first moment or expectation):

Definition 1.3 (Expected value or mean)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ and a random variable X defined on that triple. Then, the expected value $\mathrm{E}(X)$ or mean of X is defined as the Lebesgue integral

$$\mathrm{E}(X) := \int_{\Omega} X \, d\mathrm{P} = \int_{\Omega} X(\omega) \, d\mathrm{P}(\omega) \tag{1.5}$$

whenever the integral exists.

Note that since the integral may not converge absolutely, not all random variables have a finite expected value, and for some it is not defined at all (e.g., Cauchy distribution).

In order to define the second important number, the covariance, we first introduce the notion of moments:

Definition 1.4 (Moment)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$, a natural number $n \in \mathbb{N}$ and a random variable X defined on that triple. Then, the n–th moment is given by

$$m_n := \mathrm{E}(X^n). \tag{1.6}$$

Hence, the mean is also the first moment. Regarding the covariance, we require the second moment to describe, how much two random variables in one probability space change together, i.e. what the nature of their connection and how strong this connection is:

Definition 1.5 (Covariance)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ and two random variables X and Y defined on that triple. Then, the covariance $\mathrm{Cov}(X, Y)$ is defined as

$$\mathrm{Cov}(X, Y) := \mathrm{E}((X - \mathrm{E}(X))(Y - \mathrm{E}(Y))) \tag{1.7}$$

whenever the second moments of X and Y exist.
If $X = Y$, then covariance is called variance and we obtain $\mathrm{Cov}(X, X) = \sigma^2(X)$.

Higher moments describe the skewness and kurtosis of the probability function P, which can be interpreted as a deviation measure from the normal distribution and a deviation measure from a symmetric distribution respectively.

The following notion of a so called probability density function uses the nice property of a random variable to be a transformation to an easily interpretable space. I.e., it describes the relative likelihood for this random variable to take on a given value (evaluated in the image space of the random variable):

Definition 1.6 (Probability density function)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ and a random variable $X : \Omega \to E$ defined on that triple, where the set E equipped with measure μ and $\mathcal{E}$ is a σ–algebra of E. Then, any measurable

function $f : \mathcal{E} \to \mathbb{R}_0^+$, which satisfies

$$\Pr(X \in B) \left(= \int\limits_{X^{-1}(B)} d\mathrm{P} \right) = \int\limits_{B} f \, d\mu \tag{1.8}$$

for any measurable set $B \in \mathcal{E}$ is called a probability density function.

Last, we require some convergence concepts to formally describe what we observed in Figures 1.4 and 1.5. There are several several convergence concepts for different purposes: Some of these concepts are stronger, i.e. exhibit more requirements. The advantage of a strong concept is that, if a convergence can be shown for a method using the strong concept, then we also obtain convergence in the weak one. A schematic illustration of the convergence concepts we consider here is given in Figure 1.8, and their relation is shown in Figure 1.9.

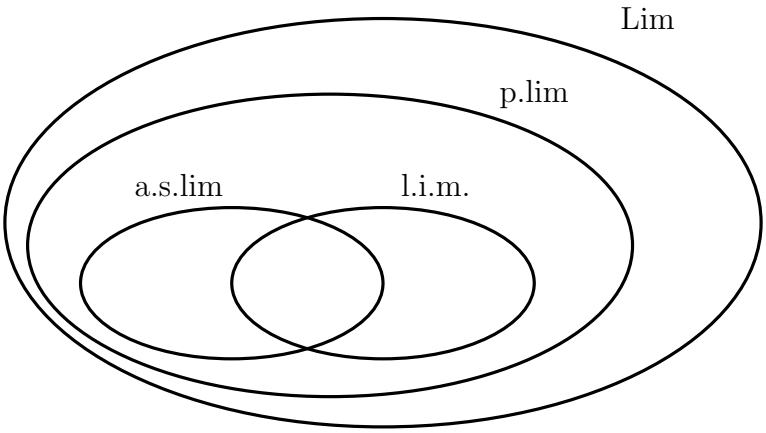


Figure 1.8: Schematic illustration of the convergence areas for stochastic limits.

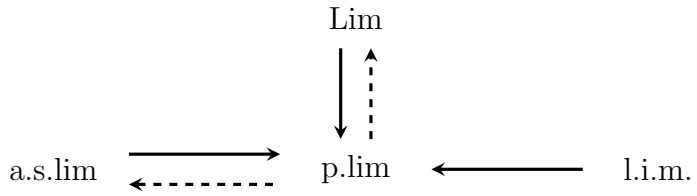


Figure 1.9: Inclusions between stochastic limits.

Convergence in distribution is the most weak concept, but it is suffers from a major disadvantage: It is very hard, if not impossible, to show that the required conditions hold:

Definition 1.7 (Convergence in distribution)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$, a measurement vector $z \in \mathbb{R}^N$ and a sequence of random variables $X(N)$, $N \in \mathbb{N}$ and a random variable X, both defined on that triple. The respective probability distribution functions are denoted by f_N and f. Then, we call $X(N)$ **to converge to X in distribution** if

- $\lim_{N\to\infty} f_N(z,(X(N))(\omega)) \to f\,(z,X(\omega))\ \forall\omega\in\Omega$ where f is continuous.

For short, we write $\operatorname*{Lim}_{N\to\infty} X(N) = X$.

The reason for this problem lies, among other problems, in the non–uniqueness of the probability density function, cf. Definition 1.6. Hence, we would have to find a suitable probability density function across the sequence of random variables.

Incorporating the probability function, we obtain a more strict and more easily provable convergence criterion:

Definition 1.8 (Convergence in probability)
Consider a probability space $(\Omega,\mathcal{F},\mathrm{P})$ and a sequence of random variables $X(N)$, $N\in\mathbb{N}$ and a random variable X, both defined on that triple. Then, we call $X(N)$ **to converge to** X **in probability** if

- $\forall\varepsilon,\delta>0:\ \exists N_0\in\mathbb{N}:\ \mathrm{P}\left(|X(N)-X|\leq\varepsilon\right)>1-\delta\ \ \forall N>N_0$.

For short, we write $\operatorname*{p.lim}_{N\to\infty} X(N) = X$.

Using convergence in probability, we need to show existence of bounds N_0 for all pairs ε,δ. Although this is a tricky task, it may be solved using knowledge of the probability function P and of the random variables, which are also functions, cf. Definition 1.2. Hence, this may also be difficult.

Neglecting the probability function P, i.e. impose more restrictions, we can solely focus on the random variables:

Definition 1.9 (Convergence with probability 1)
Consider a probability space $(\Omega,\mathcal{F},\mathrm{P})$ and a sequence of random variables $X(N)$, $N\in\mathbb{N}$ and a random variable X, both defined on that triple. Then, we call $X(N)$ **to converge to** X **with probability** 1 if

- $\lim_{N\to\infty} (X(N))(\omega) = X(\omega)$ for almost all $\omega\in\Omega$.

For short, we write $\operatorname*{a.s.lim}_{N\to\infty} X(N) = X$ or $\mathrm{P}\left(\lim_{N\to\infty} X(N) = X\right) = 1$.

For the convergence with probability 1 concept, we still need to check the criterion for almost all $\omega\in\Omega$, which can be done by exploiting properties like continuity etc. of the random variables. Hence, this concept is appropriate for our forthcoming analyses.

Another nice concept is based on distinct properties of the random variables, i.e. of its first and second moment:

Definition 1.10 (Mean square convergence)
Consider a probability space $(\Omega,\mathcal{F},\mathrm{P})$ and a sequence of random variables $X(N)$, $N\in\mathbb{N}$ and a random variable X, both defined on that triple. Then, we call $X(N)$ **to converge to** X **in mean square** if

- $\mathrm{E}\left(|X|^2\right) < \infty$,
- $\mathrm{E}\left(|X(N)|^2\right) < \infty$ for all $N \in \mathbb{N}$, and
- $\lim\limits_{N\to\infty} \mathrm{E}\left(|X(N) - X|^2\right) = 0$.

For short, we write $\underset{N\to\infty}{\text{l.i.m.}}\, X(N) = X$.

Again, this a checkable concept, which we will consider within the identification process.

Chapter 2

Structure of the identification process

Within the last chapter, we introduce the notions from stochastic analysis, which we require to study the modeling and identification process. Within the current chapter, we will first focus on the properties which we are looking for in an estimator. Exemplary, we will check these properties for one of the estimators given in the circuit example from the previous Chapter 1. Based on the described properties, we then discuss the general design sequence of a system identification.

2.1 Properties of estimators

Regarding properties, we start of with the claim that a good estimation of a system should exhibit the same characteristics, i.e. the same probability density function. Since the probability density function completely defines the properties of a system, such an estimation would do this as well. Unfortunately, as discussed in the context of Definition 1.7, without additional conditions it is very hard to show the respective convergence in distribution. But we also learned that certain properties of the expectation value are sufficient to guarantee mean square convergence, cf. Definition 1.10, which is in turn sufficient for convergence in distribution — the property we like to have.

Hence, our first demand for an estimator is that it reflects an identical expectation value.

Definition 2.1 (Unbiased estimator)
Suppose a probability space $(\Omega, \mathcal{F}, \mathrm{P})$, a measurable space E with σ–algebra $\mathcal{E}$ of E and an estimator (random variable) $\hat{\theta} : \Omega \to E$ for the parameter $\theta \in E$ to be given. If

$$\mathrm{E}\left(\hat{\theta}\right) = \theta \qquad \forall \theta \in E \tag{2.1}$$

holds true, then we call the estimator $\hat{\theta}$ unbiased. If

$$\lim_{N\to\infty} \mathrm{E}\left(\hat{\theta}(N)\right) = \theta \qquad \forall \theta \in E \tag{2.2}$$

holds, then we call the estimator $\hat{\theta}$ asymptotically unbiased. Otherwise, it is called biased.

Note that, if the estimator is unbiased, its mean converges towards the mean of the model or model parameters. Yet, since we design the model to represent only a certain part of reality, the model is typically not exact. Hence, the „ideal“ situation is not realistic and we have to

think about generalizations. One possibility is to suppose that we evaluate the estimator in a noiseless situation to obtain an approximation. Then, these reference values are compared to results with noise. The final step is to eliminate the influence of the disturbance such that the estimator converges to its reference.

Unfortunately, it is very difficult if not impossible to find the expected value by analytical means. And for some probability density functions, the expected value does not exist. And last, we may face the problem that while the expected values are identical, i.e. the estimator is unbiased according to Definition 2.1, the probability density functions are very different and coincide only in the expected value. If such an estimator were used, the outcome of a system may be very different from the real one. To avoid such a problem, we introduce the concept of consistency:

Definition 2.2 (Weak and strong consistency)
Suppose an estimator $\hat{\theta}$ and parameters θ to be given. If $\hat{\theta}$ converges in probability to θ,

$$\operatorname*{p.lim}_{N\to\infty} \hat{\theta}(N) = \theta, \tag{2.3}$$

then the estimator $\hat{\theta}$ is called weakly consistent.
If $\hat{\theta}$ converges almost surely to θ,

$$\operatorname*{a.s.lim}_{N\to\infty} \hat{\theta}(N) = \theta, \tag{2.4}$$

then the estimator $\hat{\theta}$ is called strongly consistent.

The advantage of this concept is that we can prove consistency much easier than unbiasedness. Since the limit operator may be interchanged with a continuous function ($\operatorname{p.lim} f(x) = f(\operatorname{p.lim}(x))$) if both limits exist, the consistency idea also exhibits nice calculation properties.

Apart from unbiasedness and consistency, we are also interested in obtaining an estimator, which shows minimal errors only. In particular, we want to minimize the scatter range of the estimator around its limiting value. That gives us the concept of efficiency:

Definition 2.3 (Efficiency)
Suppose an unbiased estimator $\hat{\theta}$ of parameter θ to be given. If for any unbiased estimator $\hat{\theta}_1$ of parameter θ the inequality

$$\operatorname{Cov}\left(\hat{\theta},\hat{\theta}\right) \leq \operatorname{Cov}\left(\hat{\theta}_1,\hat{\theta}_1\right), \tag{2.5}$$

then the estimator $\hat{\theta}$ is called efficient.

Since we can rely on a finite number of noisy measurements only, it is clear that there are limits on the accuracy and precision that can be reached by the estimator. The connection between measurements and accuracy is given by the so called Cramer-Rao rule:

Theorem 2.4 (Cramer-Rao rule)
Consider a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ *and a random variable* $X : \Omega \to E$ *defined on that triple, where the set* E *equipped with measure* μ *and* $\mathcal{E}$ *is a* σ*–algebra of* E*. Let* $f(z,\theta)$ *be the*

probability density function of the measurements $z \in \mathbb{R}^N$. Assume that $f(z,\theta)$ and its first and second derivatives w.r.t. θ exist for all θ and that the boundaries of the domain of $f(z,\theta)$ w.r.t. z are independent of θ. Then, the Cramer–Rao lower bound on the mean square error of any estimator $\hat{G}(z)$ of the function $G(\theta) \in \mathbb{C}^r$ is

$$\mathrm{MSE}\left(\hat{G}(\hat{\theta}(z))\right) \geq \left(\frac{\partial G(\theta)}{\partial \theta} + \frac{\partial b_G}{\partial \theta}\right) \mathrm{Fi}(\theta)^{+} \left(\frac{\partial G(\theta)}{\partial \theta} + \frac{\partial b_G}{\partial \theta}\right)^{H} + b_g b_g^H \tag{2.6}$$

where b_G denotes the expected value bias given by

$$b_G := \mathrm{E}\left(\hat{G}(\hat{\theta}(z))\right) - G(\theta)$$

and $\mathrm{Fi}(\theta)$ represents the Fisher information matrix of the parameters θ

$$\mathrm{Fi}(\theta) := \mathrm{E}\left(\left(\frac{\partial \ln f(z,\theta)}{\partial \theta}\right)^{\top} \left(\frac{\partial \ln f(z,\theta)}{\partial \theta}\right)\right) = -\mathrm{E}\left(\frac{\partial^2 \ln f(z,\theta)}{\partial \theta^2}\right).$$

We like to stress that the Cramer–Rao rule requires knowledge of the true parameter θ, which may not be at hand. An approximation can still be calculated by replacing θ in (2.6) by its estimated value $\hat{\theta}$. Similarly, the probability density function $f(z,\theta)$ can be approximated using available measurements z only.

The Cramer–Rao rule gives us a very simple way to check efficiency:

Corollary 2.5 (Efficiency)
If a given estimator $\hat{\theta}$ reaches the Cramer-Rao bound (2.6)*, then it is efficient.*

Remark 2.6 (Special cases)
There are a few special cases we like to point out:

1. *Inequality* (2.6) *becomes an equality if and only if there exists a matrix Γ such that*

$$\hat{G}(\hat{\theta}(z)) - \mathrm{E}\left(\hat{G}(\hat{\theta}(z))\right) = \Gamma \left(\frac{\partial \ln f(z,\theta)}{\partial \theta}\right)^{\top}.$$

2. *If $G(\theta) = \theta$, $b_G = 0$ and $\mathrm{Fi}(\theta)$ is regular, then we obtain the Cramer–Rao lower bound for unbiased estimators*

$$\mathrm{Cov}\left(\hat{\theta}(z), \hat{\theta}(z)\right) \geq \mathrm{Fi}(\theta)^{-1}.$$

3. *If $G(\theta) = \theta$, $b_G \neq 0$ and $\mathrm{Fi}(\theta)$ is regular, then we find the Cramer–Rao lower bound on the mean square error of biased estimators*

$$\mathrm{MSE}\left(\hat{\theta}(z)\right) \geq \left(Id + \frac{\partial b_G}{\partial \theta}\right) \mathrm{Fi}(\theta)^{-1} \left(Id + \frac{\partial b_G}{\partial \theta}\right)^{\top} + b_G b_G^{\top}.$$

2.2 Exemplary analysis

Recall the simple resistance example from Section 1.2. There, we have already seen that the simple approach estimator R_{SA} does not reveal good results, cf. Figures 1.4 and 1.6. To keep the analysis simple, we consider the following assumption:

Assumption 2.7 (Noise)
The measurements are disturbed by additive random variables, i.e.

$$i(k) = i_0 + X_i(k) \qquad \text{and} \qquad u(k) = u_0 + X_u(k) \tag{2.7}$$

with the properties that

- each random variable has zero mean and variance σ_u^2, σ_i^2,
- each random variable is independently and identically distributed (iid),
- each random variable exhibits a symmetric distribution, and
- the random variables are mutually independent.

2.2.1 Unbiasedness

Analysis of $\hat{\theta}_{\mathbf{EV}} = R_{\mathbf{EV}}$:

Using the model (2.7) within formula (1.2) we directly see

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{EV}}\right) = \lim_{N\to\infty} \hat{\theta}_{\mathrm{EV}}(N) = \lim_{N\to\infty} \frac{\frac{1}{N}\sum_{k=1}^{N} u(k)}{\frac{1}{N}\sum_{k=1}^{N} i(k)} = \lim_{N\to\infty} \frac{\frac{1}{N}\sum_{k=1}^{N} u_0 + X_u(k)}{\frac{1}{N}\sum_{k=1}^{N} i_0 + X_i(k)}$$

$$= \lim_{N\to\infty} \frac{u_0 + \frac{1}{N}\sum_{k=1}^{N} X_u(k)}{i_0 + \frac{1}{N}\sum_{k=1}^{N} X_i(k)} = \frac{u_0 + \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_u(k)}{i_0 + \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_i(k)}.$$

Now, we can apply the zero mean and iid property of X_u and X_i, that is

$$\mathrm{E}\left(X_u\right) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_u(k) = 0, \qquad \mathrm{E}\left(X_i\right) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_i(k) = 0. \tag{2.8}$$

Hence, we obtain

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{EV}}\right) = \frac{u_0 + \overbrace{\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_u(k)}^{=0}}{i_0 + \underbrace{\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_i(k)}_{=0}} = \frac{u_0}{i_0} = R_0 \tag{2.9}$$

which shows that the error-in-variable estimator is unbiased.

Analysis of $\hat{\theta}_{\mathbf{LS}} = R_{\mathbf{LS}}$:

Again, we first compute the expectation value of the estimator. Here, we obtain

$$\begin{aligned}
\mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right) &= \lim_{N\to\infty} \hat{\theta}_{\mathrm{LS}}(N) = \lim_{N\to\infty} \frac{\sum_{k=1}^{N} u(k)\cdot i(k)}{\sum_{k=1}^{N} i(k)^2} \\
&= \frac{\lim_{N\to\infty} \sum_{k=1}^{N} (u_0 + X_u(k))\cdot(i_0 + X_i(k))}{\lim_{N\to\infty} \sum_{k=1}^{N} (i_0 + X_i(k))^2} \\
&= \frac{\lim_{N\to\infty} u_0 i_0 + \frac{u_0}{N}\sum_{k=1}^{N} X_i(k) + \frac{i_0}{N}\sum_{k=1}^{N} X_u(k) + \frac{1}{N}\sum_{k=1}^{N} X_u(k)X_i(k)}{\lim_{N\to\infty} i_0^2 + \frac{2i_0}{N}\sum_{k=1}^{N} X_i(k) + \frac{1}{N}\sum_{k=1}^{N} X_i(k)^2}
\end{aligned}$$

Now, from Assumption 2.7 we use the zero mean and iid property properties (2.8) of the random variables X_u and X_i as well as their mutually independency and the variance assumption, i.e.

$$\forall k = 1,\ldots,N:\ X_u(k)X_i(k) = 0, \qquad \sigma^2(X_i) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} X_i(k)^2 = \sigma_i^2 \tag{2.10}$$

to obtain

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right) = \frac{u_0 i_0}{i_0^2 + \sigma_i^2} = \frac{R_0}{1 + \sigma_i^2/i_0^2}.$$

Hence, the least square estimator will always underestimate the magnitude of the value it is supposed to approximate. Note that the noise is removed from the nominator, but is always present in the denominator. Utilizing Definition 2.1, the least square estimator is biased. The bias depends on the *signal-to-noise ration (SNR)* of the measurements i_0/σ_i.

Analysis of $\hat{\theta}_{\mathbf{SA}} = R_{\mathbf{SA}}$:

If we take a closer look at the simple approach estimator $\hat{\theta}_{\mathrm{SA}}$ and incorporate the structural assumptions (2.7), we obtain

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) = \lim_{N\to\infty} \hat{\theta}_{\mathrm{SA}}(N) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \frac{u(k)}{i(k)} = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \frac{u_0 + X_u(k)}{i_0 + X_i(k)}$$

Rewriting this equation, we have

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) = \lim_{N\to\infty} \frac{1}{N}\frac{u_0}{i_0}\sum_{k=1}^{N} \frac{1 + X_u(k)/u_0}{1 + X_i(k)/i_0} = \lim_{N\to\infty} R_0 \frac{1}{N}\left(\sum_{k=1}^{N} \frac{1}{1 + X_i(k)/i_0} + \sum_{k=1}^{N} \frac{X_u(k)/u_0}{1 + X_i(k)}\right)$$

Applying the Taylor series

$$\frac{1}{1 + X_i(k)/i_0} = \sum_{j=0}^{\infty} (-1)^j \left(\frac{X_i(k)}{i_0}\right)^j$$

we obtain

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) = \lim_{N\to\infty} R_0 \frac{1}{N}\left(\sum_{k=1}^{N}\sum_{j=0}^{\infty}(-1)^j\left(\frac{X_i(k)}{i_0}\right)^j + \sum_{k=1}^{N}\sum_{j=0}^{\infty}(-1)^j\left(\frac{X_i(k)}{i_0}\right)^j X_u(k)\right)$$

Applying the mutually independent property (2.8), we can shorten this expression to

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) = \lim_{N\to\infty} R_0 \frac{1}{N}\left(\sum_{k=1}^{N}\sum_{j=0}^{\infty}(-1)^j\left(\frac{X_i(k)}{i_0}\right)^j + \sum_{k=1}^{N} 1\cdot X_u(k)\right)$$

which can again be shortened using the zero mean property of X_u displayed in (2.8) to

$$\begin{aligned}\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) &= \lim_{N\to\infty} R_0 \frac{1}{N}\sum_{k=1}^{N}\sum_{j=0}^{\infty}(-1)^j\left(\frac{X_i(k)}{i_0}\right)^j = \lim_{N\to\infty} R_0 \frac{1}{N}\sum_{k=1}^{N}\left(1+\sum_{j=1}^{\infty}(-1)^j\left(\frac{X_i(k)}{i_0}\right)^j\right)\\ &= \lim_{N\to\infty} R_0\left(1+\frac{1}{N}\sum_{j=1}^{\infty}\sum_{k=1}^{N}(-1)^j\left(\frac{X_i(k)}{i_0}\right)^j\right).\end{aligned}$$

Since X_i is symmetric due to Assumption 2.7, we have $\sum_{k=1}^{N}(-1)^j\left(X_i(k)/i_0\right)^j = 0$ for odd numbers j. Therefore, the limiting value of the expectation value is

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) = R_0\left(1+\lim_{N\to\infty}\left(\frac{1}{N}\sum_{j=1}^{\infty}\sum_{k=1}^{N}\left(\frac{X_i(k)}{i_0}\right)^{2j}\right)\right) = R_0\left(1+\sum_{j=1}^{\infty}\frac{m_j^{2j}(X_i(k))}{i_0^{2j}}\right)$$

where we used the moments of $X_i(k)$ according to Definition 1.4. For small disturbances $|X_i(k)/i_0| < 1$, we can neglect the moments of order 4 and higher and finally obtain

$$\mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right) = R_0\left(1+\frac{\sigma^2\left(X_i\right)}{i_0^2}\right) = R_0\left(1+\frac{\sigma_i^2}{i_0^2}\right).$$

Similar to the least square estimator $\hat{\theta}_{\mathrm{LS}} = R_{\mathrm{LS}}$, the analysis shows that the estimator converges to a value larger than the desired one. Note that the problem for group A here is that the Taylor series expansion cannot be performed.

2.2.2 Consistency

Note that we have already done these computations during the computation of the expected values since we have been using the concept of convergence with probability 1, which is a stronger concept than convergence in probability. In particular, for the error–in–variables approach we have

$$\operatorname*{p.lim}_{N\to\infty}\hat{\theta}_{\mathrm{EV}} = \operatorname*{p.lim}_{N\to\infty}\frac{\frac{1}{N}\sum_{k=1}^{N}u(k)}{\frac{1}{N}\sum_{k=1}^{N}i(k)} = \frac{\operatorname*{p.lim}_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}u_0+X_u(k)}{\operatorname*{p.lim}_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}i_0+X_i(k)} = \frac{u_0}{i_0} = R_0.$$

Hence, $\hat{\theta}_{\mathrm{EV}}$ is a weakly consistent estimator. As we can see, it is much easier to check consistency than (asymptotic) unbiasedness.

Additionally, the concept is typically used based on a cost function interpretation of an estimator. Hence, it reveals an insight into the anticipated errors.

Interpretation of $\hat{\theta}_{\mathbf{LS}} = R_{\mathbf{LS}}$:

Consider, for example, the least square estimator $\hat{\theta}_{\mathrm{LS}}$ from (1.3). The idea is to **minimize the equation errors** $e(k) := u(k) - Ri(k)$ in the model equation in a least square sence. This gives us the cost function

$$V_{\mathrm{LS}}(N) := \sum_{k=1}^{N} e^2(k, R)$$

and allows us to restate the estimation problem via

$$\hat{\theta}_{\mathrm{LS}}(N) = \operatorname*{argmin}_{R \in \mathbb{R}} V_{\mathrm{LS}}(N, R).$$

Interpretation of $\hat{\theta}_{\mathbf{SA}} = R_{\mathbf{SA}}$:

Similarly, the simple approach estimator can be rewritten using the resistance estimates $R(k) := u(k)/i(k)$ defining the cost function $V_{\mathrm{SA}}(N, R) := \sum_{k=1}^{N} \left(R(k) - R\right)^2$ and the estimation problem

$$\hat{\theta}_{\mathrm{SA}}(N) = \operatorname*{argmin}_{R \in \mathbb{R}} V_{\mathrm{SA}}(N, R).$$

The difference compared to the least square estimator $\hat{\theta}_{\mathrm{LS}}$ is that here, we do not consider a model of the problem, which is subject to disturbances. Instead, the **measurement quotient** $R(k)$ is **considered as disturbed**.

Interpretation of $\hat{\theta}_{\mathbf{EV}} = R_{\mathbf{EV}}$:

Last, the error–in–variable approach **assumes that each variable by itself is disturbed**, not the quotient as for the simple approach. Here, the cost function is defined via

$$V_{\mathrm{EV}}(N, R, i_p, u_p) := \sum_{k=1}^{N} \left(u(k) - u_p\right)^2 + \sum_{k=1}^{N} \left(i(k) - i_p\right)^2,$$

which gives us the estimation problem

$$\begin{aligned} \hat{\theta}_{\mathrm{EV}}(N) = & \operatorname*{argmin}_{R, i_p, u_p, N} V_{\mathrm{EV}}(N, R, i_p, u_p) \\ & \text{subject to } u_p = R i_p. \end{aligned}$$

Note that one typically considers a quadratic cost criterion, but basically one is free to choose any other costs as well. The advantage of the quadratic cost is the simplicity in minimization. Moreover, one can show that normally distributed disturbing noise leads to a quadratic criterion.

2.2.3 Efficiency

In order to analyze efficiency of an estimator, we need to calculate the second moment of it. Alternatively, as we have seen in Corollary 2.5, the Cramer–Rao rule (2.5) can be evaluated. For the present example, the probability density functions are not known exactly and may only be approximated using respective measurements. For this reason, we focus on a manual calculation of the variance of the estimators.

Analysis of $\hat{\theta}_{\mathbf{EV}} = R_{\mathbf{EV}}$:

Regarding the variance, we apply Definition 1.5, that is

$$\sigma^2\left(\hat{\theta}_{\mathrm{EV}}\right) = \mathrm{E}\left(\left(\hat{\theta}_{\mathrm{EV}} - \mathrm{E}\left(\hat{\theta}_{\mathrm{EV}}\right)\right)^2\right).$$

To compute this value, we reconsider $\hat{\theta}_{\mathrm{EV}}$ and — since we are interested in the second moment only — neglect all second order contributions within such as X_i^2 or $X_u X_i$ in this term, i.e.

$$\begin{aligned}
\hat{\theta}_{\mathrm{EV}} &= \frac{u_0 + \frac{1}{N}\sum_{k=1}^{N} X_u(k)}{i_0 + \frac{1}{N}\sum_{k=1}^{N} X_i(k)} = \frac{u_0 + \frac{1}{N}\sum_{k=1}^{N} X_u(k)}{i_0 + \frac{1}{N}\sum_{k=1}^{N} X_i(k)} \cdot \frac{i_0 - \frac{1}{N}\sum_{k=1}^{N} X_i(k)}{i_0 - \frac{1}{N}\sum_{k=1}^{N} X_i(k)} \\
&\overset{\text{neglect 2nd order}}{\approx} \frac{u_0 i_0 + \frac{i_0}{N}\sum_{k=1}^{N} X_u(k) - \frac{u_0}{N}\sum_{k=1}^{N} X_i(k)}{i_0^2} \\
&= \frac{u_0}{i_0} + \frac{1}{i_0 N}\sum_{k=1}^{N} X_u(k) - \frac{u_0}{i_0^2 N}\sum_{k=1}^{N} X_i(k) \\
&= R_0\left(1 + \frac{1}{N}\sum_{k=1}^{N}\frac{X_u(k)}{u_0} - \frac{1}{N}\sum_{k=1}^{N}\frac{X_i(k)}{i_0}\right).
\end{aligned}$$

Hence, we obtain

$$\begin{aligned}
\sigma^2\left(\hat{\theta}_{\mathrm{EV}}\right) = \mathrm{E}\left(\left(\hat{\theta}_{\mathrm{EV}} - R_0\right)^2\right) &= \mathrm{E}\left(\left(R_0\left(\frac{1}{N}\sum_{k=1}^{N}\frac{X_u(k)}{u_0} - \frac{1}{N}\sum_{k=1}^{N}\frac{X_i(k)}{i_0}\right)\right)^2\right) \\
&\overset{\text{mutually ind.}}{=} \mathrm{E}\left(\frac{R_0^2}{N^2}\sum_{k=1}^{N}\frac{X_u(k)^2}{u_0^2} + \frac{R_0^2}{N^2}\sum_{k=1}^{N}\frac{X_i(k)^2}{i_0^2}\right) \\
&\overset{\text{linearity}}{=} \frac{R_0^2}{N^2}\left(\mathrm{E}\left(\sum_{k=1}^{N}\frac{X_u(k)^2}{u_0^2}\right) + \mathrm{E}\left(\sum_{k=1}^{N}\frac{X_i(k)^2}{i_0^2}\right)\right) \\
&= \frac{R_0^2}{N^2}\left(\frac{\sigma^2(X_u)}{u_0^2} + \frac{\sigma^2(X_i)}{i_0^2}\right) = \frac{R_0^2}{N^2}\left(\frac{\sigma_u^2}{u_0^2} + \frac{\sigma_i^2}{i_0^2}\right)
\end{aligned}$$

Analysis of $\hat{\theta}_{\mathbf{LS}} = R_{\mathbf{LS}}$:

Considering the variance, Definition 1.5 reveals

$$\sigma^2\left(\hat{\theta}_{\mathrm{LS}}\right) = \mathrm{E}\left(\left(\hat{\theta}_{\mathrm{LS}} - \mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right)\right)^2\right).$$

Similar to the estimator $\hat{\theta}_{\mathrm{EV}}$, we first reconsider $\hat{\theta}_{\mathrm{LS}}$ and approximate it using only zero and first order terms:

$$\hat{\theta}_{\mathrm{LS}} \approx \frac{u_0 i_0 + \frac{u_0}{N}\sum_{k=1}^{N} X_i(k) + \frac{i_0}{N}\sum_{k=1}^{N} X_u(k)}{i_0^2 + \frac{2i_0}{N}\sum_{k=1}^{N} X_i(k)} \cdot \frac{i_0^2 - \frac{2i_0}{N}\sum_{k=1}^{N} X_i(k)}{i_0^2 - \frac{2i_0}{N}\sum_{k=1}^{N} X_i(k)}$$

$$\overset{\text{neglect 2nd order}}{=} \frac{u_0 i_0^3 + \frac{u_0 i_0^2}{N}\sum_{k=1}^{N} X_i(k) + \frac{i_0^3}{N}\sum_{k=1}^{N} X_u(k) - \frac{2u_0 i_0^2}{N}\sum_{k=1}^{N} X_i(k)}{i_0^4}$$

$$= \frac{u_0}{i_0} + \frac{1}{N i_0}\sum_{k=1}^{N} X_u(k) - \frac{u_0}{N i_0^2}\sum_{k=1}^{N} X_i(k)$$

$$= R_0\left(1 + \frac{1}{N}\sum_{k=1}^{N}\frac{X_u(k)}{u_0} - \frac{1}{N}\sum_{k=1}^{N}\frac{X_i(k)}{i_0}\right)$$

This expression is identical to $\hat{\theta}_{\mathrm{EV}}$, and we therefore obtain

$$\sigma^2\left(\hat{\theta}_{\mathrm{LS}}\right) = \sigma^2\left(\hat{\theta}_{\mathrm{EV}}\right) = \frac{R_0^2}{N^2}\left(\frac{\sigma_u^2}{u_0^2} + \frac{\sigma_i^2}{i_0^2}\right).$$

Analysis of $\hat{\theta}_{\mathbf{SA}} = R_{\mathbf{SA}}$:

Computing the variance via Definition 1.5, that is

$$\sigma^2\left(\hat{\theta}_{\mathrm{SA}}\right) = \mathrm{E}\left(\left(\hat{\theta}_{\mathrm{SA}} - \mathrm{E}\left(\hat{\theta}_{\mathrm{SA}}\right)\right)^2\right).$$

Similar to the estimator $\hat{\theta}_{\mathrm{EV}}$, we first reconsider $\hat{\theta}_{\mathrm{SA}}$ and approximate it using only zero and first order terms:

$$\hat{\theta}_{\mathrm{SA}} = \frac{1}{N}\sum_{k=1}^{N}\frac{u_0 + X_u(k)}{i_0 + X_i(k)} \cdot \frac{i_0 - X_i(k)}{i_0 - X_i(k)}$$

$$\overset{\text{neglect 2nd order}}{=} \frac{1}{N}\sum_{k=1}^{N}\frac{u_0 i_0 + i_0 X_u(k) - u_0 X_i(k)}{i_0^2}$$

$$= R_0\left(1 + \frac{1}{N}\sum_{k=1}^{N}\frac{X_u(k)}{u_0} - \frac{1}{N}\sum_{k=1}^{N}\frac{X_i(k)}{i_0}\right)$$

This expression is identical to $\hat{\theta}_{\mathrm{EV}}$ and $\sigma^2\left(\hat{\theta}_{\mathrm{EV}}\right)$, and we therefore obtain

$$\sigma^2\left(\hat{\theta}_{\mathrm{SA}}\right) = \sigma^2\left(\hat{\theta}_{\mathrm{EV}}\right) = \sigma^2\left(\hat{\theta}_{\mathrm{LS}}\right) = \frac{R_0^2}{N^2}\left(\frac{\sigma_u^2}{u_0^2} + \frac{\sigma_i^2}{i_0^2}\right).$$

2.2.4 Assessment

Unbiasedness

The simple approach estimator $\hat{\theta}_{\mathrm{SA}}$ continuously overestimates the true value of the parameter, hence it is biased. The error–in–variable estimator $\hat{\theta}_{\mathrm{EV}}$ approximate the true value of the paramter and is therefore unbiased. Last, the least square estimator $\hat{\theta}_{\mathrm{LS}}$ underestimates the true value of the parameter. Like $\hat{\theta}_{\mathrm{SA}}$, it is biased.
Hence, from a bias point of view, the error–in–variable estimator is preferable.

Consistency

Within our analysis, we found that all three estimators are converging in distribution. Therefore, the also converge in probability and are consistent.

Efficiency

Last, we have seen, all three estimators show the same second moment. Hence, none of them is more efficient.

Concluding, the error–in–variable estimator is the choice at hand given the presented alternatives, since it performs as well as the other estimators in all three categories and gives better results in terms of bias.

2.3 Basic design of estimators

Each identification process consists of a series of basic steps:

1. Collect information on the system
2. Select a model to represent the system
3. Choose an optimization criterion
4. Fit the model parameters to the measurements accordingly
5. Validate the computed model

Note that some of the steps may be hidden from the user or selected without being aware of a choice, which may result in suboptimal or even poor performance.

2.3.1 Step 1: Gathering information

In order to identify a process, we first need to build a model of that part of the system, which we are interested in. To this end, we need to gather information about the process. This step can be done either by observing natural fluctuations, but it is by far more efficient to set up dedicated experiments that actively excite the system via known inputs. While a good example of the first are default fluctuations in demand for a supply chain, the latter can be interpreted as a stress test of a supply chain by uncommon and/or extreme demands. Additionally, the controlled second approach allows for optimization of information gathering goals, such as minimum cost and time, measurement accuracy over a certain bandwidth or other possible aims. Note that the quality of the total identification process may heavily depend on these choices.

2.3.2 Step 2: Selecting the model structure

The model structure is the most variable part of the identification. It not only depends on the problem of identification itself, but may be subject to the further use of the model. For example, an approximation of the elasticity of a wheel via a PDE may give a good dynamical model. Yet, if the model is to be used in a feedback loop, the required computing time to evaluate the model is larger than the sampling time of the loop. Hence, a coarser (or worse) model is necessary for the subsequent task. Keeping this in mind, we distinguish the following:

Parametric vs. nonparametric models

In a parametric model, the system is described by a small number of characteristic quantities. These quantities are called parameters of the model. Regarding our simple electrical circuit example, the expected value is one parameter of the model, the variance the second one. An alternative example is given by the transfer function, e.g. of a filter, which is described by its poles and zeros.

A nonparametric model is given by measurements of a system function at a large number of points. Reviewing the transfer function example, a description via an impulse response at a large number of points is such a characterization.

Note that is usually simpler to create a nonparametric model than a parametric one because the modeler needs less knowledge about the system itself in the first case. Yet, insight into the problem and concentration of information in a few characteristics is more substantial for parametric models and make the problem simulation faster.

White box vs. black box models

In a white box model, the internal functioning of the system is – at least to some degree – understood. In particular, skills of the experimenter as well as connections between components such as physical laws can be used, whose availability and applicability depend on such an insight. Here, a loudspeaker illustrates the need for extensive understanding of mechanical, electrical and acoustical phenomena in order to derive an appropriate model.

In contrast to the white box idea of using insight into the system, the black box approach uses a brute force modeling. To this end, a mathematical model is proposed, which allows the description of any observed input and output measurements, but may not even be connected to the real system. Regarding the loudspeaker, a high order transfer function may be used as such a model.

Again, the choice depends on the further aim. While the white box idea provides a better insight gain into the working principles of the system, the black box model may be sufficient for simulations/predictions. Note that it is typically a good idea to include as much knowledge as possible during modeling, yet that may not always be easy to accomplish. Analyzing a stable system for example, it is not simple to express this information if the polynomial coefficients are used as parameters of the model.

Linear vs. nonlinear models

In almost all cases, real life applications are nonlinear. Unfortunately, theory of nonlinear systems is quite involved and may be difficult to understand for a user unfamiliar with this theory. A nonlinear approach describes the system over its complete operating range and covers also rare and unusual phenomena.

Linear systems, on the other hand, are (almost) completely understood, nice to handle and can be evaluated quickly. Unfortunately, as stated above, real life is typically nonlinear. Therefore, linear systems commonly represent approximations of nonlinear systems within some region – assuming the region can be linearized. Within such a so called operating region, the linear part of the system can be regarded as dominant, i.e. the nonlinear part can be neglected without changing the behavior of the system.

Similar to the other choices, the scope of the problem is relevant to make an appropriate choice. For example, a nonlinear model is needed to describe the distortion of an amplifier, but a linear model is sufficient to represent its transfer characteristics if the operating range is small enough.

Linear-in-parameter vs. nonlinear-in-parameter models

The last choice has to be made between linear and nonlinear influence of parameters of the model. A model is called linear-in-parameter if there exists a linear relation between these parameters and the error that is minimized. Note that linear-in-parameters does not imply a linear model. For example, $e = y-(au^2+bu+c)$ is linear in a, b and c, but the model is nonlinear. Likewise, $e = y - (a + bj\omega)/(c + dj\omega)u$ is a linear model, but it is nonlinear-in-parameter in c and d.

The impact of this choice can be seem, e.g., for the least square estimator. If the model is linear-in-parameter, then the minimization problem of the least squares can be solved analytically, and does not require an iterative optimization method. Hence, the complexity of a linear-in-parameter model is much lower.

2.3.3 Step 3: Choose optimization criterion

After choosing a model, is must be matched to the available measurements of the process. To this end, one typically introduces a criterion, which measures the goodness of fit, i.e. the distance between the computed and the measured values. Note that the choice of this criterion is important regarding the outcome of the identification process as it determines the stochastic properties of the estimator. Regarding our simple resistance example, there are several choices which lead to estimators with different properties, cf. Section 2.2.

The cost criterion can be chosen arbitrarily. Yet, it typically resides on ad hoc intuitive insight. In the following Chapter 3, we provide a more systematic approach based on stochastic arguments to obtain such a criterion.

Remark 2.8
There exist test on the cost criterion to check – even before deriving the estimator – if the resulting estimator can be consistent. These are necessary conditions, which are outside the scope of this lecture.

2.3.4 Step 4: Fitting model parameters

In the ensuing step of fitting the parameters, the design work is done and the computations start. Within this step, numerical or symbolic methods are applied to solve the minimization problem arising from the cost criterion in Step 3 subject to the model chosen in Step 2 with respect to the measurement derived in Step 1. Although this step seems to be the essential one, we can already see that the most of the work is the design. This is due to the fact that nowadays, computing power is cheap and there exist a wide area of methods to solve certain problems. The actual art is to design the problem such that it is easily solvable but satisfies the constraints, which bound the model in its further use.

2.3.5 Step 5: Validating obtained model

In the final step, the validity of the obtained model shall be tested. Here, the following question are essential:

- Does the model describe the available data properly?
- Are there indications that some parts of the model are not well designed or flawed?

Note that, as mentioned before, the model with the smallest error is not always the preferred one in practice. Instead, a simpler model may be better suited if it describes the system within user-specified error bounds.

Within the validation process, errors should be separated into different classes such as un-modeled linear dynamics or nonlinearity distortions. Such information shall allow further improvements of the model if necessary. During the validation, the application should be kept in mind, i.e. conditions similar to reality are to be used. Note that extrapolation should be avoided as the errors of extrapolation increase drastically if many measurements are used, which is the typical case for estimator design.

Chapter 3

Least square estimation

Within this chapter, we pursue a systematic approach to the parameter estimation problem. In particular, we ask what criterion should be used to match the model to the data. To answer this question, we use a statistical approach to select a criterion to measure the quality of the resulting fit. After defining the problem at hand, we discuss two estimators here, the least square and the weighted least square estimator. Note that it is also possible to use other estimator types such as the least absolute values.

3.1 Problem definition

Let a input-output model be given by

$$y_0(k, \theta) = g\left(u_0(k), \theta\right) \tag{3.1}$$

where $k \in \mathbb{N}_0$ represents the measurement index, $y_0(k) \in \mathbb{Y} = \mathbb{R}^{n_y}$ the output, $u(k) \in \mathbb{U} = \mathbb{R}^{n_u}$ the input and $\theta \in \Theta = \mathbb{R}^{n_\theta}$ the true parameter vector.

The aim is to estimate the parameters from noisy observations at the output of the systems. To this end, we assume that the output is separated into a deterministic and a probabilistic part $y_0(\cdot)$ and $X_y(\cdot)$:

Assumption 3.1
Given an input output model, noise disturbances only occur within the output observations

$$y(k, X_y) = y_0(k) + X_y(k) \tag{3.2}$$

where $y(k, X_y)$ and $y_0(k)$ represent the modeled and nominal output and $X_y(k)$ denotes the random output variable.

To achieve the described goal, we minimize the errors

$$e(k, \theta) = z_k - y_0(k, \theta) \tag{3.3}$$

between the measured and the estimated/modeled values z_k and $y_0(k, \theta)$ respectively.
The quality of a fit can then be expressed via a cost criterion. One such criterion is given by the so called nonlinear least squares (NLS), which is derived from the minimization of the sum of squared values:

Definition 3.2 (Least Square estimator)
The least square estimator $\hat{\theta}_{\mathrm{NLS}}(N)$ is given by

$$\hat{\theta}_{\mathrm{NLS}}(N) = \underset{\theta}{\operatorname{argmin}}\, V_{\mathrm{NLS}}(N,\theta), \quad \text{with } V_{\mathrm{NLS}}(N,\theta) := \frac{1}{2}\sum_{k=1}^{N} e^2(k,\theta). \tag{3.4}$$

Alternatively, one may also use the sum of absolute values

$$\hat{\theta}_{\mathrm{NLA}}(N) = \underset{\theta}{\operatorname{argmin}}\, V_{\mathrm{NLA}}(N,\theta), \quad \text{with } V_{\mathrm{NLA}}(N,\theta) := \frac{1}{2}\sum_{k=1}^{N} |e(k,\theta)| \tag{3.5}$$

The least square estimator (3.4) is the most popular one. Yet, by choosing the cost function arbitrarily as we have done it here, it is not at all clear that the result is not necessarily optimal. Least squares, however, are strongly motivated by numerical aspects. This is due to the fact that minimizing a least squares cost function is usually less involved than alternative cost functions. Here, the quadratic nature can be exploited which reveal that the necessary first order conditions for an optimal are also sufficient. Still, we like to mention that the nonlinear least absolute values (3.5) are less sensitive to outliers in the data and may for this reason be interesting in certain applications as well.

As we have seen in Section 2.2, even within the class of least squares different estimators can be designed which lead to results with different properties. In context of an optimal outcome with respect to the properties presented in Chapter 2, it is important to see where the noise enters into the raw data. Thereafter, a cost function should be selected that explicitly accounts for these errors.

3.2 Linear least square

If the model is chosen to be linear-in-parameter θ, equations (3.1) and (3.3) simplify to

$$y_0(\theta) = K(u_0)\,\theta \tag{3.6}$$

with input/output matrix $K(u) \in \mathbb{R}^{N\times n_\theta}$, input vector $u_0 \in \mathbb{R}^N$ and output vector $y_0 \in \mathbb{R}^N$. Hence, the error can be rewritten as

$$e(\theta) = z - K(u_0)\,\theta \tag{3.7}$$

where $z \in \mathbb{R}^N$ represents the vector of measurements. The quality criterion $V_{\mathrm{NLS}}(N,\theta)$ reduces to a linear one given by

$$\begin{aligned} V_{\mathrm{LS}}(N,\theta) &:= \frac{1}{2} e(\theta)^\top e(\theta) = \frac{1}{2}\left(z - K(u_0)\,\theta\right)^\top \left(z - K(u_0)\,\theta\right) \\ &= \frac{1}{2}\sum_{k=1}^{N} \left(z_k - K(u_0(k))\,\theta\right)^2. \end{aligned} \tag{3.8}$$

Hence, we obtain the following:

Definition 3.3 (Linear least square estimation problem)
The linear least square estimate $\hat{\theta}_{\mathrm{LS}}(N)$ is given by

$$\hat{\theta}_{\mathrm{LS}}(N) = \underset{\theta}{\operatorname{argmin}} V_{\mathrm{LS}}(N, \theta) \tag{3.9}$$

with $V_{\mathrm{LS}}(N, \theta)$ according to (3.8).

Since V_{LS} is quadratic, we can compute the minimizer of this loss function explicitly via

$$\frac{\partial V_{\mathrm{LS}}(N, \theta)}{\partial \theta} = 0.$$

This gives us

$$0 = \frac{\partial V_{\mathrm{LS}}(N, \theta)}{\partial \theta} = e(\theta)^{\top} \frac{\partial e(\theta)}{\theta} = e(\theta)^{\top} (-K(u_0)) = -K(u_0)^{\top} e(\theta).$$

Hence, we have to solve the equation

$$-K(u_0)^{\top} (z - K(u_0)\,\theta) = 0$$

for θ which reveals the solution

$$\hat{\theta}_{\mathrm{LS}}(N) = \theta = \left(K(u_0)^{\top} K(u_0)\right)^{-1} K(u_0)^{\top} z.$$

Concluding, we have shown the following:

Theorem 3.4 (Solution of $\hat{\theta}_{\mathrm{LS}}$)
The solution to the linear least square estimation problem (3.9), (3.8)

$$\hat{\theta}_{LS}(N) = \underset{\theta}{\operatorname{argmin}} V_{LS}(N, \theta) \quad \textit{with} \quad V_{LS}(N, \theta) = \frac{1}{2} (z - K(u_0)\,\theta)^{\top} (z - K(u_0)\,\theta)$$

is given by

$$\hat{\theta}_{LS}(N) = \left(K(u_0)^{\top} K(u_0)\right)^{-1} K(u_0)^{\top} z. \tag{3.10}$$

Here, we like to note that one typically does not compute the least square estimator via formula (3.10), but instead solves the linear equation

$$\left(K(u_0)^{\top} K(u_0)\right) \hat{\theta}_{\mathrm{LS}}(N) = K(u_0)^{\top} z$$

and avoids inverting the matrix $K(u_0)^{\top} K(u_0)$. Unfortunately, the matrix $K(u_0)^{\top} K(u_0)$ causes numerical instability since eigenvalues are raised by the power of two. There are, however, ways to compute the solution of the linear least square estimation problem (3.9), (3.8) by other, more stable algorithms such as the QR decomposition.

In order to generate the matrix K, one has to reformulate the model of the problem (3.6) combined for the available inputs and outputs $u(k)$ and $y(k)$, $k = 1, \ldots, N$. Let us consider two examples:

Example 3.5
The simplest model is given by

$$y_0 = \theta,$$

which is independent from the input. Combining all available outputs $y(k)$, $k = 1, \ldots, N$ *this reads*

$$\begin{aligned} y_0(1) &= \theta \\ &\vdots \\ y_0(N) &= \theta \end{aligned}$$

and reveals the matrix

$$K = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Using formula (3.10) *we obtain the estimator*

$$\begin{aligned} \hat{\theta}_{LS}(N) &= \left(K^\top K\right)^{-1} K^\top z \\ &= \left((1, \ldots, 1) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right)^{-1} (1, \ldots, 1)\, z \\ &= (N)^{-1} (1, \ldots, 1)\, z = \frac{1}{N} \sum_{k=1}^{N} z(k). \end{aligned}$$

To illustrate the result, we chose measurements of the form

$$z = \theta \quad \textit{with} \quad \theta = 1 + 0.2 X_y,$$

where X_y *is normally independently distributed with mean* 0 *and standard deviation* 1, *i.e.* $X_y \in \mathcal{N}(0, 1)$ *and* $\theta \in \mathcal{N}(1, 0.2)$. *Considering* 100 *such measurements, we obtain the result display in Figure 3.1. The respective program is displayed in Program A.3.*

Example 3.6
Given the model

$$y = u_1 \theta_1 + u_2^2 \theta_2$$

we can combine inputs and outputs to obtain

$$\begin{aligned} y(1) &= u_1(1)\theta_1 + u_2^2(1)\theta_2 \\ &\vdots \\ y(N) &= u_1(N)\theta_1 + u_2^2(N)\theta_2. \end{aligned}$$

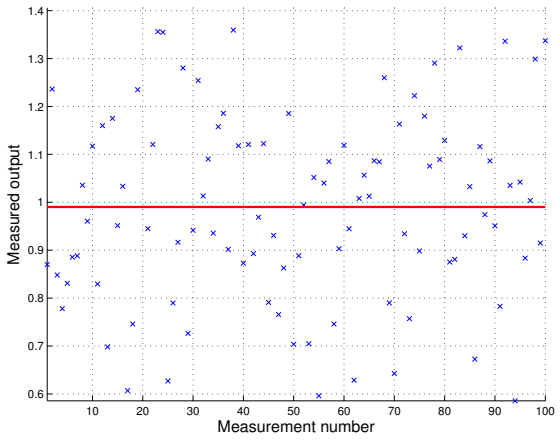


Figure 3.1: Sample measurements and estimation for Example 3.5

Hence, we have (3.6)*, i.e.*

$$y_0 = K(u_0)\,\theta$$

with

$$y_0 = \begin{pmatrix} y(1) \\ \vdots \\ y(N) \end{pmatrix}, \quad u_0 = \begin{pmatrix} u_1(1) \\ u_2(1) \\ \vdots \\ u_1(N) \\ u_2(N) \end{pmatrix}, \quad \textit{and} \quad K(u_0) = \begin{pmatrix} u_1(1) & u_2^2(1) \\ \vdots & \vdots \\ u_1(N) & u_2^2(N) \end{pmatrix}.$$

Now, we can apply formula (3.10) *to obtain the estimator*

$$\begin{aligned}
\hat{\theta}_{LS}(N) &= \left(K(u_0)^\top K(u_0)\right)^{-1} K(u_0)^\top z \\
&= \left(\begin{pmatrix} u_1(1) & u_2^2(1) \\ \vdots & \vdots \\ u_1(N) & u_2^2(N) \end{pmatrix}^\top \begin{pmatrix} u_1(1) & u_2^2(1) \\ \vdots & \vdots \\ u_1(N) & u_2^2(N) \end{pmatrix}\right)^{-1} \begin{pmatrix} u_1(1) & u_2^2(1) \\ \vdots & \vdots \\ u_1(N) & u_2^2(N) \end{pmatrix}^\top z \\
&= \left(\begin{pmatrix} u_1(1) & \dots & u_1(N) \\ u_2^2(1) & \dots & u_2^2(N) \end{pmatrix} \begin{pmatrix} u_1(1) & u_2^2(1) \\ \vdots & \vdots \\ u_1(N) & u_2^2(N) \end{pmatrix}\right)^{-1} \begin{pmatrix} u_1(1) & \dots & u_1(N) \\ u_2^2(1) & \dots & u_2^2(N) \end{pmatrix} z \\
&= \begin{pmatrix} \sum_{k=1}^{N} u_1^2(k) & \sum_{k=1}^{N} u_1(k)u_2^2(k) \\ \sum_{k=1}^{N} u_1(k)u_2^2(k) & \sum_{k=1}^{N} u_2^4(k) \end{pmatrix}^{-1} \begin{pmatrix} \sum_{k=1}^{N} u_1(k)z_k \\ \sum_{k=1}^{N} u_2^2(k)z_k \end{pmatrix}.
\end{aligned}$$

The estimator can be computed by solving the two–dimensional linear equation $A\hat{\theta}_{LS}(N) = b$

with

$$A = \begin{pmatrix} \sum_{k=1}^{N} u_1^2(k) & \sum_{k=1}^{N} u_1(k)u_2^2(k) \\ \sum_{k=1}^{N} u_1(k)u_2^2(k) & \sum_{k=1}^{N} u_2^4(k) \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} \sum_{k=1}^{N} u_1(k)z_k \\ \sum_{k=1}^{N} u_2^2(k)z_k \end{pmatrix}.$$

To illustrate the result, we chose N inputs

$$u_1(k) = 1 + \frac{k}{N-1},$$
$$u_2(k) = 2 + \frac{10k}{N-1},$$

which gives us u_0 and $K(u_0)$. Then, we generated measurements of the form

$$z = K(u_0)\,\theta$$

with

$$\theta_1 = 1 + 2X_{y,1}$$
$$\theta_2 = 2 + 1X_{y,2}$$

where $X_{y,1}, X_{y,2}$ are normally independently distributed with mean 0 and standard deviation 1, i.e. $\theta_1 \in \mathcal{N}(1,2)$ and $\theta_2 \in \mathcal{N}(2,1)$. Considering 100 such measurements, we obtain the result display in Figure 3.2. The respective program is displayed in Program A.4.

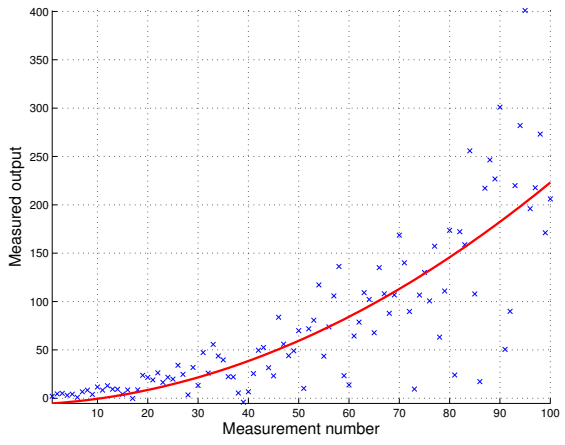


Figure 3.2: Sample measurements and estimation for Example 3.6

3.2.1 Properties of the linear least square estimator

Note that we did not formulate any assumptions on the behavior of the noise X_y to compute formula (3.10), but instead calculated it directly from the measurements and the model without

bothering about the noise behavior. However, in order to make statements about the properties of the estimator, it is necessary to give some specifications on the noise behavior.

The expected value of the estimator $\hat{\theta}_{\mathrm{LS}}$ regarding model outputs, i.e. by considering $z = y(X_y)$, can be computed via

$$\begin{aligned}
\mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right) &\overset{(3.10)}{=} \mathrm{E}\left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top y(X_y)\right) \\
&\overset{(3.2)}{=} \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top \mathrm{E}\left(y_0 + X_y\right) \\
&= \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top y_0 + \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top \mathrm{E}\left(X_y\right) \\
&\overset{(3.6)}{=} \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top K\left(u_0\right)\theta + \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top \mathrm{E}\left(X_y\right) \\
&= \theta + \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top \mathrm{E}\left(X_y\right).
\end{aligned}$$

Now, in order for $\hat{\theta}_{\mathrm{LS}}$ to be unbiased, we require $\mathrm{E}\left(X_y\right) = 0$.

Corollary 3.7 (Unbiasedness of $\hat{\theta}_{\mathrm{LS}}$)
If the model is linear-in-parameter and the probabilistic part of the output satisfies $\mathrm{E}\left(X_y\right) = 0$*, then the least square estimator* $\hat{\theta}_{LS}$ *is unbiased.*

The second interesting characteristic is the covariance matrix of the estimator $\hat{\theta}_{\mathrm{LS}}$. Here, we see the following:

$$\begin{aligned}
\mathrm{Cov}\left(\hat{\theta}_{\mathrm{LS}}, \hat{\theta}_{\mathrm{LS}}\right) &= \mathrm{E}\left(\left(\hat{\theta}_{\mathrm{LS}} - \mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right)\right)\left(\hat{\theta}_{\mathrm{LS}} - \mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right)\right)^\top\right) \\
&= \mathrm{E}\left(\left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top \mathrm{E}\left(X_y\right)\right)\left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top \mathrm{E}\left(X_y\right)\right)^\top\right) \\
&= \left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top\right) \mathrm{E}\left(X_y X_y^\top\right) \left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top\right)^\top \\
&= \left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top\right) \mathrm{Cov}\left(X_y, X_y\right) \left(\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top\right)^\top
\end{aligned}$$

Similar to Corollary 3.7, we can make the following conclusion regarding the covariance matrix of the estimator $\hat{\theta}_{\mathrm{LS}}$.

Corollary 3.8 (Covariance of $\hat{\theta}_{\mathrm{LS}}$)
Consider a linear-in-parameter model. If the disturbing noise X_y *is white and uncorrelated, i.e.* $\mathrm{Cov}\left(X_y, X_y\right) = \sigma^2\left(X_y\right) Id_{n_\theta}$*, then the covariance matrix of the estimator* $\hat{\theta}_{LS}$ *is given by*

$$\mathrm{Cov}\left(\hat{\theta}_{LS}, \hat{\theta}_{LS}\right) = \sigma^2\left(X_y\right)\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} \tag{3.11}$$

Else, the covariance matrix can be computed via

$$\mathrm{Cov}\left(\hat{\theta}_{LS}, \hat{\theta}_{LS}\right) = L\,\mathrm{Cov}\left(X_y, X_y\right) L^\top. \tag{3.12}$$

where $L := \left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} K\left(u_0\right)^\top$.

To illustrate this, let us reconsider Examples 3.5 and 3.6.

Example 3.9
Given Assumption 3.1, consider the model

$$y_0 = \theta$$

and suppose the noise X_y to be white and uncorrelated. Using $K = (1, \dots, 1)^\top$, we can evaluate (3.11) *and obtain*

$$\mathrm{Cov}\left(\hat{\theta}_{LS}, \hat{\theta}_{LS}\right) = \frac{1}{N}\sigma^2\left(X_y\right).$$

Example 3.10
Consider the model

$$y = u_1\theta_1 + u_2^2\theta_2$$

and again assume Assumption 3.1 to hold and the noise X_y to be white and uncorrelated, i.e. $\mathrm{Cov}\left(X_y, X_y\right) = \sigma^2\left(X_y\right) Id_{n_\theta}$. *Then, we obtain*

$$\begin{aligned}\mathrm{Cov}\left(\hat{\theta}_{LS}, \hat{\theta}_{LS}\right) &= \sigma^2\left(X_y\right)\left(K\left(u_0\right)^\top K\left(u_0\right)\right)^{-1} \\ &\overset{\text{Example 3.6}}{=} \sigma^2\left(X_y\right)\begin{pmatrix} \sum_{k=1}^{N} u_1^2(k) & \sum_{k=1}^{N} u_1(k)u_2^2(k) \\ \sum_{k=1}^{N} u_1(k)u_2^2(k) & \sum_{k=1}^{N} u_2^4(k) \end{pmatrix}^{-1}.\end{aligned}$$

Here, we like to note that within the least square estimator (3.10)

$$\left(K\left(u_0\right)^\top K\left(u_0\right)\right)\hat{\theta}_{\mathrm{LS}}(N) = K\left(u_0\right)^\top z$$

the multiplication $K\left(u_0\right)^\top z$ includes an $N \times n_\theta$ and a $n_\theta \times 1$ matrix. To this sum, we can apply the central limit theorem we gives us that the estimator $\hat{\theta}_{\mathrm{LS}}$ asymptotically converges to a Gaussian distribution **even** if X_y is not Gaussian distributed, that is

$$\lim_{N\to\infty} \hat{\theta}_y = \mathcal{N}\left(\mathrm{E}\left(\hat{\theta}_{\mathrm{LS}}\right), \mathrm{Cov}\left(\hat{\theta}_{\mathrm{LS}}, \hat{\theta}_{\mathrm{LS}}\right)\right).$$

3.3 Weighted least square

So far, we have only been looking at equally weighted measurements in (3.4) (and (3.5)). However, it may be desirable to change this property, e.g. to suppress measurements with high uncertainty and to emphasize those with low uncertainty. To design such a weighting, the covariance matrix can be used.

In practice, it is not always clear what weighting should be used. Yet certain indicators can be used to improve the estimator. For example, if it is known that the model exhibits errors, then utilizing the covariance matrix may not be a good idea. Instead, the user may prefer to

put a dedicated wighting in order to keep the model errors small in some specific operation regions.

Definition 3.11 (Weighted Least Square estimator)
The weighted least square estimator $\hat{\theta}_{\text{WLS}}(N)$ is given by

$$\hat{\theta}_{\text{WLS}}(N) = \underset{\theta}{\operatorname{argmin}}\, V_{\text{WLS}}(N,\theta), \quad \text{with } V_{\text{WLS}}(N,\theta) := \frac{1}{2} e(\theta)^\top W e(\theta) \tag{3.13}$$

where $W \in \mathbb{R}^{N\times N}$ is symmetric and positive definite.

Again we can utilize the quadratic nature of V_{WLS} to compute the minimizer of this loss function explicitely via

$$\frac{\partial V_{\text{WLS}}(N,\theta)}{\partial \theta} = 0.$$

This gives us

$$0 = \frac{\partial V_{\text{WLS}}(N,\theta)}{\partial \theta} = e(\theta)^\top W^\top \frac{\partial e(\theta)}{\theta} = e(\theta)^\top W^\top (-K(u_0)) = -K(u_0)^\top W e(\theta).$$

Solve the equation

$$-K(u_0)^\top W (z - K(u_0)\,\theta) = 0$$

for θ reveals

$$\hat{\theta}_{\text{WLS}}(N) = \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W z.$$

Hence, we have shown the following:

Theorem 3.12 (Solution of $\hat{\theta}_{\text{WLS}}$)
The solution to the weighted linear least square estimation problem (3.13) *is given by*

$$\hat{\theta}_{WLS}(N) = \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W z. \tag{3.14}$$

Let us reconsider the more elaborate Example 3.6 for the weighted linear least square estimator:

Example 3.13
Recall the model

$$y = u_1\theta_1 + u_2^2\theta_2.$$

Combining inputs and outputs

$$\begin{aligned} y(1) &= u_1(1)\theta_1 + u_2^2(1)\theta_2 \\ &\vdots \\ y(N) &= u_1(N)\theta_1 + u_2^2(N)\theta_2. \end{aligned}$$

we have (3.6)

$$y_0 = K(u_0)\,\theta$$

with

$$y_0 = \begin{pmatrix} y(1) \\ \vdots \\ y(N) \end{pmatrix}, \quad u_0 = \begin{pmatrix} u_1(1) \\ u_2(1) \\ \vdots \\ u_1(N) \\ u_2(N) \end{pmatrix}, \quad \textit{and} \quad K(u_0) = \begin{pmatrix} u_1(1) & u_2^2(1) \\ \vdots & \vdots \\ u_1(N) & u_2^2(N) \end{pmatrix}.$$

Now, we choose

$$W = diag(0, \frac{1}{N-1}, \frac{2}{N-1}, \ldots, 1) \in \mathbb{R}^{N \times N},$$

i.e., measurements with larger index k are associated with higher weights.
To illustrate the difference between the $\hat{\theta}_{LS}$ and $\hat{\theta}_{WLS}$, we again chose N inputs

$$u_1(k) = 1 + \frac{k}{N-1},$$
$$u_2(k) = 2 + \frac{10k}{N-1},$$

which gives us u_0 and $K(u_0)$. Then, we generated measurements of the form

$$z = K(u_0)\,\theta$$

with

$$\theta_1 = 1 + 2X_{y,1}$$
$$\theta_2 = 2 + 1X_{y,2}$$

where $X_{y,1}, X_{y,2}$ are normally independently distributed with mean 0 and standard deviation 1, i.e. $\theta_1 \in \mathcal{N}(1,2)$ and $\theta_2 \in \mathcal{N}(2,1)$. Considering 100 such measurements, we obtain the result display in Figure 3.3. The respective program is displayed in Program A.5. Here, we see that the estimated curve deviates for measurements with small index k. This is to be expected since the respective weights are very small.

3.3.1 Properties of the weighted linear least square estimator

Turn toward the bias of the weighted linear least square estimator, we can utilize $z = y(X_y)$ to compute

$$\begin{aligned} \mathrm{E}\left(\hat{\theta}_{\mathrm{WLS}}\right) &\overset{(3.14)}{=} \mathrm{E}\left(\left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W y(X_y)\right) \\ &\overset{(3.2)}{=} \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W \mathrm{E}(y_0 + X_y) \\ &= \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W y_0 + \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W \mathrm{E}(X_y) \end{aligned}$$

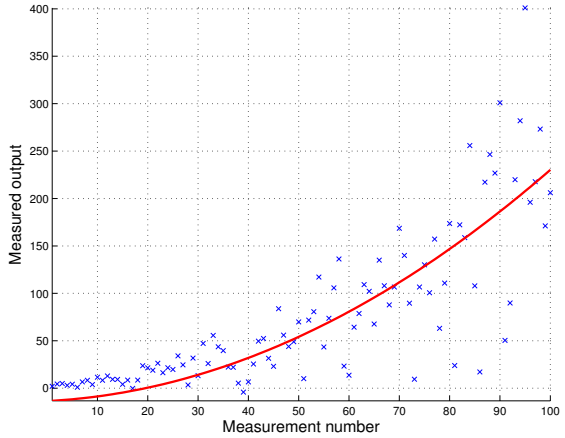


Figure 3.3: Sample measurements and estimation for Example 3.13

$$\overset{(3.6)}{=} \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W K(u_0)\,\theta + \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W \mathrm{E}(X_y)$$
$$= \theta + \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W \mathrm{E}(X_y).$$

Now, in order for $\hat{\theta}_{\mathrm{WLS}}$ to be unbiased, we require $\mathrm{E}(X_y) = 0$.

Corollary 3.14 (Unbiasedness of $\hat{\theta}_{\mathrm{WLS}}$)
If the model is linear-in-parameter and the probabilistic part of the output satisfies $\mathrm{E}(X_y) = 0$, *then the least square estimator* $\hat{\theta}_{WLS}$ *is unbiased.*

Similarly, we can compute the covariance matrix of the estimator $\hat{\theta}_{\mathrm{WLS}}$ using the arguments from the unweighted case. Here, we use the abbreviation $K := K(u_0)$.

$$\mathrm{Cov}\left(\hat{\theta}_{\mathrm{WLS}}, \hat{\theta}_{\mathrm{WLS}}\right) = \mathrm{E}\left(\left(\hat{\theta}_{\mathrm{WLS}} - \mathrm{E}(\theta)\right)\left(\hat{\theta}_{\mathrm{WLS}} - \mathrm{E}(\theta)\right)^\top\right)$$
$$\overset{(3.14)}{=} \mathrm{E}\left(\left(\left(K^\top W K\right)^{-1} K^\top W X_y\right)\left(\left(K^\top W K\right)^{-1} K^\top W X_y\right)^\top\right)$$
$$= \left(\left(K^\top W K\right)^{-1} K^\top W\right) \mathrm{E}\left(X_y X_y^\top\right) \left(\left(K^\top W K\right)^{-1} K^\top W\right)^\top$$
$$= \left(\left(K^\top W K\right)^{-1} K^\top W\right) \mathrm{Cov}(X_y, X_y) \left(\left(K^\top W K\right)^{-1} K^\top W\right)^\top$$

Hence, we can conclude the following about the covariance of $\hat{\theta}_{\mathrm{WLS}}$:

Corollary 3.15 (Covariance of $\hat{\theta}_{\mathrm{WLS}}$)
Consider a linear-in-parameter model. Then the covariance matrix of the estimator $\hat{\theta}_{WLS}$ *is given by*

$$\mathrm{Cov}\left(\hat{\theta}_{WLS}, \hat{\theta}_{WLS}\right) = L\,\mathrm{Cov}(X_y, X_y)\,L^\top \tag{3.15}$$

where $L := \left(K(u_0)^\top W K(u_0)\right)^{-1} K(u_0)^\top W$.

This result allows for a very interesting conclusion shown in [**?**], namely that the covariance matrix can be minimized if the weight is chosen as the inverse of the covariance matrix of the random variable X_y, that is $W = \mathrm{Cov}\left(X_y, X_y\right)^{-1}$.

Corollary 3.16 (Minimal covariance of $\hat{\theta}_{\mathrm{WLS}}$)
Consider a linear-in-parameter model. If the weighting matrix of the weighted linear least square estimator $\hat{\theta}_{WLS}$ is chosen as $W = \mathrm{Cov}\left(X_y, X_y\right)^{-1}$, then the covariance matrix of $\hat{\theta}_{WLS}$ is minimal and given by

$$\mathrm{Cov}\left(\hat{\theta}_{WLS}, \hat{\theta}_{WLS}\right) = \left(K\left(u_0\right)^{\top} W K\left(u_0\right)\right)^{-1}. \tag{3.16}$$

Chapter 4

Maximum likelihood and Bayes estimator

As we have seen in the previous Section 3.3, the covariance matrix of the noise may be used as weighting matrix to incorporate prior knowledge about the noise of the measurements. Yet, a full stochastic characterization requires the probability density function of the noise distortions. Given such a knowledge, it may be possible to obtain results, which are even better than those of the weighted linear least square estimator. The maximum likelihood estimator offers a theoretical framework to incorporate the knowledge about the distribution of the noise distortions in the estimator.

The Bayes estimator extends the maximum likelihood estimator by incorporating knowledge on the probability distribution function of the parameter itself. Hence, if such information is at hand, the Bayes estimator supersedes the maximum likelihood estimator. Unfortunately, in practical applications the probability density function of the parameter is hardly ever known, which renders the estimator to be impractical.

4.1 Maximum likelihood estimator

The probability density function f_{X_y} of the noise determines the conditional probability density function $f(y_0 \mid \theta)$ of the model of the measurements stated in the previous Chapter 3

$$y_0(k) = g\left(u_0(k), \theta\right) \tag{3.1}$$

describing the system and the inputs that excite the system. Similarly, Assumption 3.1 shall hold, i.e. the noise enters the model additively

$$y(k, X_y) = y_0(k) + X_y(k) \tag{3.2}$$

where $y(k, X_y)$ and $y_0(k)$ represent the modeled and nominal output and $X_y(k)$ denotes the random output variable. Then, the likelihood function becomes

$$f(y(k, X_y) \mid u_0, \theta) := f_{X_y}\left(y(k, X_y) - g\left(u_0(k), \theta\right)\right). \tag{4.1}$$

The maximum likelihood procedure now consists of two steps:

Algorithm 4.1 (Maximum likelihood procedure)
Input: Probability density function f_{X_y} and measurements z

- Plug actual measurements z into (4.1) for variable $y(k, X_y)$.
- Consider θ as the free variable and maximize the conditional probability density function

Output: Maximum likelihood estimate

$$\hat{\theta}_{\mathrm{ML}}(N) = \underset{\theta}{\operatorname{argmax}} f(z \mid u_0, \theta) \tag{4.2}$$

At first sight, this algorithm may appear simple, and indeed its applications is easy:

Example 4.2
Consider again the simplest model displayed in Example 3.5

$$y_0 = g(u_0, \theta) = \theta,$$

which is independent of the input. Assume that f_{X_y} is normal with zero mean and variance $\sigma^2(X_y)$, i.e.

$$f_{X_y}(x) = \frac{1}{\sqrt{2\pi\sigma^2(X_y)}} e^{-\frac{x^2}{2\sigma^2(X_y)}}.$$

Then, for each measurement z we obtain the conditional probability density function

$$\begin{aligned} f(z \mid u_0, \theta) &= f_{X_y}(z - g(u_0(k), \theta)) = f_{X_y}(z - \theta) \\ &= \frac{1}{\sqrt{2\pi\sigma^2(X_y)}} e^{-\frac{(z-\theta)^2}{2\sigma^2(X_y)}}. \end{aligned}$$

To maximize this expression, we have to minimize the exponent $\frac{(z-\theta)^2}{2\sigma^2(X_y)}$. Since $\sigma^2(X_y)$ is constant, this results in $\hat{\theta}_{ML} = z$.

Only one measurement is a very small batch, but we can see that the maximum likelihood estimator correctly identifies the measurement value. Next, we incorporate a whole series of measurements:

Example 4.3
Again we consider the model

$$y_0 = g(u_0, \theta) = \theta,$$

which is independent of the input and assume that f_{X_y} is normal with zero mean and variance $\sigma^2(X_y)$. Incorporating multiple independent measurements $z_1, \ldots, z_N$, the likelihood function is

$$\begin{aligned} f(z \mid u_0, \theta) &= f(z_1 \mid u_0, \theta) \cdot \ldots \cdot f(z_N \mid u_0, \theta) \\ &= f_{X_y}(z_1 - g(u_0(1), \theta)) \cdot \ldots \cdot f_{X_y}(z_N - g(u_0(N), \theta)) \\ &= f_{X_y}(z_1 - \theta) \cdot \ldots \cdot f_{X_y}(z_N - \theta). \end{aligned}$$

Hence, we obtain

$$f(z \mid u_0, \theta) = \left(\frac{1}{\sqrt{2\pi\sigma^2 (X_y)}} \right)^N e^{-\frac{1}{2\sigma^2(X_y)} \sum_{k=1}^{N} (z_k - \theta)^2}$$

and the minimizer becomes

$$\hat{\theta}_{ML}(N) = \frac{1}{N} \sum_{k=1}^{N} z_k$$

Note that in the previous example, the maximum likelihood estimator and the (weighted) least square estimator coincide. This is only the case for normally distributed errors.

Corollary 4.4
If $f_{X_y} \in \mathcal{N}(0, \sigma^2(X_y))$, then the $\hat{\theta}_{ML}(N) = \hat{\theta}_{LS}(N)$.

In general, we have the following:

Corollary 4.5
If f_{X_y} is normal and measurements z_k are iid, then the Maximum likelihood estimator problem is a linear least square problem.

4.1.1 Properties for normally independent distributions

Now that we have seen how the maximum likelihood estimator can be computed, let us consider some of its properties such as expected values of the estimated mean and the estimated variance. Here, we will focus on normally independent distributions. The computations, however, can also be performed for other distributions. In the upcoming Section 4.1.2, we provide some general statements on the maximum likelihood estimator.

Consider N samples z_k, $k = 1, \ldots, N$, which are normally independently distributed with mean μ and standard deviation σ. Then the likelihood function becomes

$$f(z \mid u_0, \theta) = f(z_1 \mid u_0, \theta) \cdot \ldots \cdot f(z_N \mid u_0, \theta) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{1}{2\sigma_y^2} \sum_{k=1}^{N} (z_k - \mu)^2}$$

and the loglikelihood function is

$$\ln f(z \mid u_0, \theta) = -\frac{N}{2} \ln\left(2\pi\sigma_y^2\right) - \frac{1}{2\sigma_y^2} \sum_{k=1}^{N} (z_k - \mu)^2 .$$

Then, we can compute the derivatives with respect to μ and σ^2, which gives us the necessary first order optimality conditions

$$\frac{\partial}{\partial \mu} \ln f(z \mid u_0, \theta) = \frac{1}{\sigma_y^2} \sum_{k=1}^{N} (z_k - \mu) \stackrel{!}{=} 0$$

$$\frac{\partial}{\partial \sigma_y^2} \ln f(z \mid u_0, \theta) = -\frac{N}{2\sigma_y^2} + \frac{1}{2\sigma_y^4} \sum_{k=1}^{N} (z_k - \mu)^2 \stackrel{!}{=} 0.$$

Solving these equations reveals

$$\mu\left(\hat{\theta}_{\mathrm{ML}}\right) = \frac{1}{N} \sum_{k=1}^{N} z_k \tag{4.3}$$

$$\sigma_y^2\left(\hat{\theta}_{\mathrm{ML}}\right) = \frac{1}{N} \sum_{k=1}^{N} \left(z_k - \mu\left(\hat{\theta}_{\mathrm{ML}}\right)\right)^2. \tag{4.4}$$

From (4.3), we directly obtain

$$\mathrm{E}\left(\mu\left(\hat{\theta}_{\mathrm{ML}}\right)\right) = \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}\left(z_k\right) = \mu,$$

which shows that the mean of the maximum likelihood estimator is unbiased.

Corollary 4.6 (Unbiasedness of $\hat{\theta}_{\mathrm{ML}}$)
If the output of a system is normally independently distributed, then the maximum likelihood estimator $\hat{\theta}_{ML}$ given by

$$\mathrm{E}\left(\hat{\theta}_{ML}\right) = \mu\left(\hat{\theta}_{ML}\right) = \frac{1}{N} \sum_{k=1}^{N} z_k \tag{4.3}$$

is unbiased.

Moreover, we can utilize (4.4) to see

$$\begin{aligned}
\mathrm{E}\left(\sigma_y^2\left(\hat{\theta}_{\mathrm{ML}}\right)\right) &= \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}\left(\left(z_k - \mu\left(\hat{\theta}_{\mathrm{ML}}\right)\right)^2\right) \\
&= \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}\left(\left((z_k - \mu) - \left(\mu\left(\hat{\theta}_{\mathrm{ML}}\right) - \mu\right)\right)^2\right) \\
&= \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}\left((z_k - \mu)^2 - 2\left(z_k - \mu\right)\left(\mu\left(\hat{\theta}_{\mathrm{ML}}\right) - \mu\right) + \left(\mu\left(\hat{\theta}_{\mathrm{ML}}\right) - \mu\right)^2\right) \\
&= \frac{1}{N} \sum_{k=1}^{N} \Big(\mathrm{E}\left((z_k - \mu)^2\right) - 2\mathrm{E}\left(\left(z_k - \mu\right)\left(\mu\left(\hat{\theta}_{\mathrm{ML}}\right) - \mu\right)\right) \\
&\qquad + \mathrm{E}\left(\left(\mu\left(\hat{\theta}_{\mathrm{ML}}\right) - \mu\right)^2\right)\Big) \\
&= \frac{1}{N} \sum_{k=1}^{N} \left(\sigma_y^2 - 2\mathrm{E}\left(\left(z_k - \mu\right)\left(\frac{1}{N}\sum_{k=1}^{N} z_k - \mu\right)\right) + \mathrm{E}\left(\left(\frac{1}{N}\sum_{k=1}^{N} z_k - \mu\right)^2\right)\right) \\
&= \frac{1}{N} \sum_{k=1}^{N} \left(\sigma_y^2 - 2\frac{1}{N}\mathrm{E}\left(\left(z_k - \mu\right)^2\right) + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathrm{E}\left(\left(z_i - \mu\right)\left(z_j - \mu\right)\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_{k=1}^{N}\left(\sigma_y^2 - 2\frac{1}{N}\sigma_y^2 + \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{E}\left((z_j-\mu)^2\right)\right)\\
&= \frac{1}{N}\sum_{k=1}^{N}\left(\sigma_y^2 - 2\frac{1}{N}\sigma_y^2 + \frac{1}{N}\sigma_y^2\right)\\
&= \frac{\sigma_y^2}{N}\sum_{k=1}^{N}\left(1-\frac{2}{N}+\frac{1}{N}\right) = \sigma_y^2\left(1-\frac{1}{N}\right).
\end{aligned}
$$

Hence, the variance of the maximum likelihood estimator is biased.

Corollary 4.7 (Covariance of $\hat{\theta}_{\mathrm{ML}}$)
If the output of a system is normally independently distributed, then the covariance matrix of the estimator $\hat{\theta}_{ML}$ given by

$$\mathrm{Cov}\left(\hat{\theta}_{ML},\hat{\theta}_{ML}\right) = \frac{1}{N}\sum_{k=1}^{N}\left(z_k - \mu\left(\hat{\theta}_{ML}\right)\right)^2 \tag{4.4}$$

is biased by a factor $1-1/k$.

As we can directly see, $\lim_{k\to\infty} 1-1/k = 1$ holds which gives us the following efficiency results:

Corollary 4.8 (Efficiency)
If the output of a system is normally independently distributed, then the covariance matrix of the estimator $\hat{\theta}_{ML}$ reaches the Cramer-Rao lower bound for $k\to\infty$.

From this analysis of the restricted normally independently distributed case, we now show some more general results.

4.1.2 General properties of the maximum likelihood estimator

In the literature, a series of important properties is tabled assuming well-defined experimental conditions. If these conditions are met, then the user knows in advance what properties the estimator will have without going through the complete development process. The results here are only stated. Full proofs can be found in, e.g., [**?**].

The following invariance principle is a very powerful tool. In particular, this principle allows us to condense the measurements, i.e. to lower the dimension of the measurement vector, without compromising the maximum likelihood property of the estimator. Additionally, transformations of estimators given by g can be analyzed easily.

Theorem 4.9 (Principle of invariance)
If $\hat{\theta}_{ML}$ is a maximum likelihood estimator of θ and $g:\mathbb{R}_\theta^n\to\mathbb{R}^{n_g}$ is a function with $n_g\le n_\theta<\infty$, then $\hat{\theta}_g = g(\hat{\theta}_{ML})$ is a maximum likelihood estimator of $g(\theta)$.

Regarding the properties we discussed in Chapter 2, one can show the consistency and efficiency of the maximum likelihood estimator.

Theorem 4.10 (Consistency)
If $\hat{\theta}_{ML}$ is a maximum likelihood estimator based on N iid random variables with n_θ independent of N, then $\hat{\theta}_{ML}(N)$ converges to y_0 almost surely, i.e.

$$\operatorname*{a.s.lim}_{N\to\infty} \hat{\theta}_{ML}(N) = \theta.$$

Theorem 4.11 (Asymptotic efficiency)
If $\hat{\theta}_{ML}(N)$ is a maximum likelihood estimator based on N iid random variables with n_θ independent of N, then $\hat{\theta}_{ML}(N)$ is asymptotically efficient, i.e. $\operatorname{Cov}\left(\hat{\theta}_{ML}(N), \hat{\theta}_{ML}(N)\right)$ *asymptotically reaches the Cramer–Rao lower bound.*

A last property, which we see here, is the so called asymptotic normality. The importance of this property is not only that it allows one to calculate uncertainty bounds on the estimates, but that it also guarantees that most of the probability mass gets more and more unimodally concentrated around its limiting value.

Theorem 4.12 (Asymptotic normality)
If $\hat{\theta}_{ML}(N)$ is a maximum likelihood estimator based on N iid random variables with n_θ independent of N, then $\hat{\theta}_{ML}(N)$ converges in law to a normal random variable.

4.2 Bayes estimator

In contrast to the maximum likelihood estimator, the Bayes estimator requires knowledge on the probability density function of both the noise on the measurements and the unknown parameters. The kernel of the Bayes estimator is the conditional probability density function of the unknown parameters θ with respect to the measurements z denoted by $f_\theta(\theta \mid u_0, z)$. This probability density function contains complete information about the parameters θ, given a set of inputs u_0 and respective measurements z. This makes it possible for the experimenter to determine the best estimate of θ for the given situation. To select this best value, it is necessary to lay down an objective criterion, i.e. the minimization of a risk function $C(\theta \mid \theta_0)$. The risk function then describes the cost of selecting the parameter θ if θ_0 is the true but unknown parameter. The estimated parameter $\hat{\theta}$ is found as the minimizer of the risk function weighted with the probability density function $f_\theta(\theta \mid u, z)$ over the range $\mathbb{D}$ of the parameter θ, i.e.

$$\hat{\theta}(N) = \operatorname*{argmin}_{\theta_0} \int_{\theta\in\mathbb{D}} C(\theta \mid \theta_0)\, f_\theta(\theta \mid u, z)\, d\theta. \tag{4.5}$$

If the cost criterion is chosen in the form

- $C(\theta \mid \theta_0) = |\theta - \theta_0|^2$ (which leads to the mean value) or
- $C(\theta \mid \theta_0) = |\theta - \theta_0|$ (results in the median, which is less sensitive to outliers since these contribute less to the second criterion compared to the first one),

then a closed solution of (4.5) is known. In contrast to this „minimum risk“ estimators, one may also choose the criterion

$$\hat{\theta}_{\mathrm{BA}}(N) = \underset{\theta}{\operatorname{argmax}}\, f_\theta(\theta \mid u, z), \tag{4.6}$$

which reveals the Bayes estimator. In practice, it is very difficult to select the best out of these variants.

Here, we study the Bayes estimator in more detail. To search for the maximizer of (4.6), the Bayes rule

$$f_\theta(\theta \mid u, z) = \frac{f(z \mid \theta, u) f_\theta(\theta)}{f(z)}$$

is applied. In order to maximize the right hand side of this equation, it is sufficient to maximize its numerator as the denominator is independent of θ. Hence, the solution is given by looking for the maximum of

$$f(z \mid \theta, u) f_\theta(\theta).$$

As we can already see, a lot of a priori information is required to use the Bayes estimator, i.e. $f(z \mid \theta, u)$, which is also used in the maximum likelihood estimator in (4.1), and $f_\theta(\theta)$. Note that in many problems the probability density function $f_\theta(\theta)$ is unavailable, which renders the Bayes estimator to be barely used in practice.

Example 4.13
Let us reconsider Example 3.5 with modifications from Example 4.2, i.e. the model is given by

$$y_0 = g(u_0, \theta) = \theta,$$

which is independent of the input and f_{X_y} is normal with zero mean and variance $\sigma^2(X_y)$, i.e.

$$f_{X_y}(x) = \frac{1}{\sqrt{2\pi\sigma^2(X_y)}} e^{-\frac{x^2}{2\sigma^2(X_y)}}.$$

Additionally, the probability density function of θ is given by its mean w and standard deviation σ_w, that is

$$f_\theta(x) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(x-w)^2}{2\sigma_w^2}}.$$

Since we have

$$f(z \mid \theta, u) = f(z \mid \theta) = f_{X_y}(X_y) = f_{X_y}(z - \theta),$$

the Bayes estimator is found by maximizing the expression

$$f(z \mid \theta, u) f_\theta(\theta) = \frac{1}{\sqrt{2\pi\sigma^2(X_y)}} e^{-\frac{(z-\theta)^2}{2\sigma^2(X_y)}} \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(\theta-w)^2}{2\sigma_w^2}}$$

with respect to θ and the estimate becomes

$$\hat{\theta}_{BA} = \frac{z/\sigma^2(X_y) + w/\sigma_w^2}{1/\sigma^2(X_y) + 1/\sigma_w^2}.$$

The example shows two things: For one, if the quality of the a priori information w is high compared with the quality of the measurements, that is $\sigma_w^2 \ll \sigma^2(X_y)$, then the estimate is determined mainly by the a priori information. Conversely, if $\sigma_w^2 \ll \sigma^2(X_y)$, then the estimate is dominated by the information gained by measurements.

Example 4.14
Example 4.13 can be extended in a manner similar to Example 4.3 by considering N independent measurements $z_1, \ldots, z_N$. Then, the Bayes estimator becomes

$$\hat{\theta}_{BA}(N) = \frac{\sum_{k=1}^{N} z_N/\sigma^2(X_y) + w/\sigma_w^2}{N/\sigma^2(X_y) + 1/\sigma_w^2}.$$

From this last example, we can make the following conclusion:

Corollary 4.15
If f_{X_y} and f_θ are normal and measurements z_k are iid, then the Bayes estimator problem is a linear least square problem.

Chapter 5

Dynamic system models

Within this chapter, we will shift our focus from static to dynamic models. Again, these model are subject to parameters, which we aim to estimate. There are different types of dynamic models, that is discrete and continuous time one. Here, we start off with the discrete time variant, which is characterized by the fact that inputs, outputs and measurements of the system are available at discrete time instances only. In contrast to that, continuous time models exhibit continuous data streams. In general, a dynamic system can be seen as a blackbox, which assigns an output sequence to a given input sequence, cf. Figure 5.1.

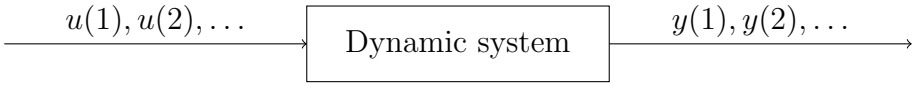


Figure 5.1: General structure of a dynamic system

We will particularly focus on so called LTI systems, that is linear time invariant systems. Moreover, we will give some basic fact on dynamic systems and show how the estimators discussed in the previous Chapters 3 and 4 can be applied.

5.1 Discrete time models

For the discrete time setting, the time as a variable is chosen to be an element of the integer number $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Hence, inputs and outputs of the model are functions $u : \mathbb{Z} \to \mathbb{U}$, $y : \mathbb{Z} \to \mathbb{Y}$. A respective model is then given by the following definition, which is closely connected to Figure 5.1.

Definition 5.1 (Discrete time Auto Regressive model with eXogenous input (ARX))
Consider sequences $(u(k))_{k=n-N,\ldots,n-1} \in \mathbb{U}^N$ and $(y(k))_{k=n-N,\ldots,n-1} \in \mathbb{Y}^N$ and the parameters $\theta = (\theta_u, \theta_y) \in \Theta$ to be given. Then we call

$$y(n) = g\left(u(n-1), \ldots, u(n-N), y(n-1), \ldots, y(n-N), \theta\right) \tag{5.1}$$

an auto regressive model with exogenous input or ARX model.

Note that in contrast to the static model we considered in the previous chapters, the parameters are now split over time. More compact, the system depends on its own past inputs and outputs.

Similar to Chapters 3 and 4, we focus on the linear case. In the context of dynamic system models, this leads to so called linear time invariant (LTI) systems.

Definition 5.2 (Discrete time Linear Time Invariant model)
Consider sequences $(u(k))_{k=n-N,\dots,n-1} \in \mathbb{U}^N$ and $(y(k))_{k=n-N,\dots,n-1} \in \mathbb{Y}^N$ and parameters $\theta = (\theta_u, \theta_y) \in \Theta$ to be given. Then we call

$$y(n) = \theta_u(1)u(n-1) + \dots + \theta_u(N)u(n-N) + \theta_y(1)y(n-1) + \dots + \theta_y(N)y(n-N) \quad (5.2)$$

a linear time invariant or LTI system.

Example 5.3
A very simple example of an LTI system is given by the model

$$y(n+1) = \theta y(n) + u(n)$$

For such a system, it is clear that if $\theta > 1$, then the system is unstable, i.e. $y(n) \to \pm\infty$ as $n \to \infty$. One possibility to circumvent this instability is to define the input as a feedforard control via

$$u(n) := \left(\frac{1}{2} - a\right) \frac{1}{2^n} y(0)$$

or via a feedback of the form

$$u(n) := \left(\frac{1}{2} - a\right) y(n).$$

The difference is that the feedback (or closed loop control) can compensate for unmodeled equation errors, the feedforward control however cannot.

Finding the parameters of dynamical systems is a crucial issue in engineering areas such as simulation or model based control. In particular, if the parameters $\theta = (\theta_u, \theta_y) \in \Theta$ are known, and if additionally future inputs $(u(k))_{k=n,\dots,n+k} \in \mathbb{U}^k$ are at hand, then the system can be simulated into the future. Given that the parameters are correct, the prediction will be correct as well. If the parameters are biased, then the prediction may diverge by the factor of $e^{kL \max\{\hat{\theta}-\theta\}}$, where L denotes the Lipschitz constant of g, i.e. $L = \|\theta\|_\infty$ in the LTI case.

5.2 Continuous time models

In continuous time, the corresponding time variable is an element of the real numbers $t \in \mathbb{R}$. Similarly, inputs and outputs are function $u : \mathbb{R} \to \mathbb{U}$ and $y : \mathbb{R} \to \mathbb{Y}$. In contrast to discrete time models, we do not consider a whole history of past data, but instead dig deeper regarding derivatives of g. To write the following definition of a continuous time ARX model more compactly, we abbreviate $y^k := \frac{\partial^k y}{\partial t^k}$ and $u^k := \frac{\partial^k u}{\partial t^k}$.

Definition 5.4 (Continuous time Auto Regressive model with eXogenous input (ARX))
Consider the derivative sequences $\left(u^k\right)_{k=0,\dots N-1}$ and $\left(y^k\right)_{k=0,\dots,N-1}$ and the parameters $\theta = (\theta_u, \theta_y) \in \Theta$ to be given. Then the differential equation

$$y^N(t) = g\left(u^0(t), \dots, u^{N-1}(t), y^0(t), \dots, y^{N-1}(t), \theta\right) \tag{5.3}$$

is called continuous time auto regressive model with exogenous input or ARX model.

A simple example of an ARX system is given by the following differential equation:

Example 5.5 (Logistic differential equation)
Consider the differential equation with initial conditions

$$y^1(t) = \theta_y(1)\left(1 - \frac{y^0(t)}{\theta_y(2)}\right) y^0(t), \qquad y^0(t_1) = y_{t_1}$$

with constants $\theta_y(1), \theta_y(2) > 0$ on the interval $t \in [t_1, t_2]$. This so called logistic differential equation describes the growth of a population. The growth rate $1 - \frac{y^0(t)}{\theta_y(2)}$ depends on the current size of the population $y^0(t)$. The parameters $\theta_y(1)$ and $\theta_y(2)$ describe the birth rate and the saturation level of the population.

Similar to the discrete time case, we can define a linear time invariant model as follows:

Definition 5.6 (Continuous time Linear Time Invariant model)
Consider the derivative sequences $\left(u^k\right)_{k=0,\dots N-1}$ and $\left(y^k\right)_{k=0,\dots,N-1}$ and the parameters $\theta = (\theta_u, \theta_y) \in \Theta$ to be given. Then the differential equation

$$y^N(t) = \theta_u(1)u^0(t) + \theta_u(N)u^{N-1}(t) + \theta_y(1)y^0(t) + \theta_y(N)y^{N-1}(t) \tag{5.4}$$

is called continuous time linear time invariant or LTI model.

We like to note that it is also possible to include a whole history of derivative data into the parametrization. The simulation of such a delayed differential equation model, however, is computationally much more involved than in the ordinary differential equation case we discuss here. Additionally, it is worth mentioning that also spatial derivatives can be included, which leads us to a partial differential equation formulation.

5.3 Basic facts about dynamic systems

Dynamic systems are categorized using different properties, which are displayed in Table 5.1 below. Note that the given division lines are not the only ones in existence. There is a lot of characteristics, which allow for improved treatment of dynamic systems. Here, we only focus on the main research lines.

Simple type	**Complex type**
Linear The system is linear in the input and output variable u, y. Example: $$\ddot{y}(t) = -\theta_1 y(t) + \theta_2 u(t)$$	**Nonlinear** The system may not be linear in either the input or the output variable u, y. Example: $$\ddot{y}(t) = -\theta_1 \sin(y(t)) + \theta_2 u(t)$$
Time invariant All parameters are constants, i.e. „Today is tomorrow.“ Example: both linear and nonlinear systems displayed above	**Time varying** At least one parameter is time dependent. Example: $$\ddot{y}(t) = -\theta_1(t) y(t) + \theta_2 u(t)$$
Discrete time Time is given by sampling instants in $\mathbb{Z}$. Example: $$y(n+1) = \theta_1 y(n) + \theta_2 u(n)$$	**Continuous time** Time is given by a real valued variable. Example: all examples displayed above
Input output model Input is directly mapped to output. Example: all exampled displayed above	**State space model** Input triggers changes of an internal state, output is a linear combination of these internal states. Example: $$x(n+1) = Ax(n) + Bu(n)$$ $$y(n) = Cx(n)$$

Table 5.1: Division lines for dynamic systems

Within our analysis, we will mainly focus on the most simple case of a discrete time LTI input-output model. In fact, these properties are easily seen as shown in the following example:

Example 5.7
Consider a pendulum with a spring at the rope shown in Figure 5.2. Here, we have

$$p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \dot{p}_1 \\ \dot{p}_2 \end{pmatrix}$$

where p denotes the position and v the velocity of the pendulum and (c_1, c_2) represents the anker position. From Newtons Law we know

$$F = m\ddot{p}$$
$$F = m\dot{v}$$

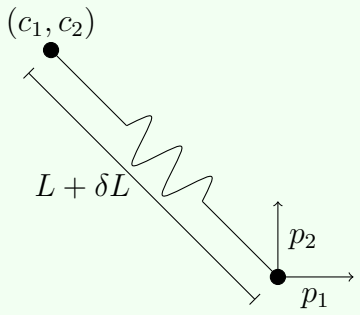


Figure 5.2: Pendulum with a spring at the rope

where

$$F = \frac{c-p}{\|c-p\|_2}(\|c-p\|_2 - L)K + \begin{pmatrix} 0 \\ -mg \end{pmatrix}$$
$$\dot{v} = \frac{F}{m} = (c-p)\left(1 - \frac{L}{\|c-p\|_2}\right)\frac{K}{m} + \begin{pmatrix} 0 \\ -g \end{pmatrix}$$

and K denotes the hook constant. Hence, we obtain $\dot{x} = f(x)$ with $x = [p_1, p_2, v_1, v_2]^\top$ and

$$f(x) = \begin{pmatrix} v_1 \\ v_2 \\ (c_1 - p_1)\left(1 - \frac{L}{\|c_1 - p_1\|_2}\right)\frac{K}{m} \\ (c_2 - p_2)\left(1 - \frac{L}{\|c_2 - p_2\|_2}\right)\frac{K}{m} - g \end{pmatrix}$$

Now, we can conclude that the dynamic system is a

- *nonlinear*
- *time invariant*
- *continuous time*
- *state space model.*

5.4 Estimators in the dynamic setting

Here, we focus on the discrete time LTI case where the dynamic model is given by

$$y(n) = \theta_u(1)u(n-1) + \ldots + \theta_u(N)u(n-N) + \theta_y(1)y(n-1) + \ldots + \theta_y(N)y(n-N) \quad (5.2)$$

as defined in Definition 5.2. Then, we obtain our standard model equation

$$y_0(k) = g\left(u_0(k), \theta\right) \quad (3.1)$$

by considering

$$u_0(k) = (u(n-1), \ldots, u(n-N), y(n-1), \ldots, y(n-N))$$

and $\theta = (\theta_u, \theta_y)$ as in Definition 5.2. Then, we similarly introduce the additive error, which is called *equation error* in the dynamic model case, and again obtain

$$y(k, X_x) = y_0(k) + X_x(k) \tag{3.2}$$

where $y(k, X_x)$ and $y_0(k)$ represent the modeled and nominal output and $X_x(k)$ denotes the *random equation error variable.*

Different from our approach in Chapter 4, the inputs and the outputs are typically not known exactly, i.e.

$$\begin{aligned} y^m(k) &= y(k) + X_y(N) \\ u^m(k) &= u(k) + X_u(N), \end{aligned}$$

where $X_y(k)$ and $X_u(k)$ denote the *random output and input error variables.*

5.4.1 Maximum likelihood estimator

In the previous Section 4.1, we introduced the general approach for the Maximum likelihood estimator in Algorithm 4.1 and provided a simple static example. The approach, however, is not restricted to the static setting but may also be applied for dynamic system models.

Defining the input variable as

$$u_0 := (u^m(n-1), \ldots, u^m(n-N), y^m(n-1), \ldots, y^m(n-N))$$

and the parameter vector via

$$\theta := (\theta_u, \theta_y, u(n-1), \ldots, u(n-N), y(n-1), \ldots, y(n-N))$$

we can apply Algorithm 4.1 and evaluate formula

$$\begin{aligned} \hat{\theta}_{\mathrm{ML}}(N) &= \operatorname*{argmax}_{\theta} f(z \mid u_0, \theta) && (4.2) \\ &= \operatorname*{argmax}_{\theta} f_{X_u}\left(u^m(k) - u(k)\right) \cdot f_{X_y}\left(y^m(k) - y(k)\right) \cdot f_{X_x}\left(z_k - g\left(u_0(k), \theta\right)\right). && (5.5) \end{aligned}$$

Even for LTI systems, this typically leads to a nonlinear optimization problem, which can be solved using standard techniques, see, e.g., [**?**]. However, if the probability density functions f_{X_u}, f_{X_y} and f_{X_g} are normal with zero mean and respective variance, that is $f_{X_u} \in \mathcal{N}\left(0, \sigma^2\left(X_u\right)\right)$, $f_{X_y} \in \mathcal{N}\left(0, \sigma^2\left(X_y\right)\right)$ and $f_{X_g} \in \mathcal{N}\left(0, \sigma^2\left(X_f\right)\right)$, then we can utilize the approach from Example 4.3 and obtain the loglikelihood function

$$\begin{aligned} \ln f(z \mid u_0, \theta) = -\frac{1}{2} \sum_{k=1}^{N} \Bigg[& \left(\frac{u^m(k) - u(k)}{\sigma^2\left(X_u\right)} \right)^2 + \left(\frac{y^m(k) - y(k)}{\sigma^2\left(X_y\right)} \right)^2 \\ & + \left(\frac{z_k - g\left(u(k-1), \ldots, u(k-N), y(k-1), \ldots, y(k-N), \theta\right)}{\sigma^2\left(X_g\right)} \right)^2 \Bigg]. \end{aligned} \tag{5.6}$$

If the model is LTI, then the last part of the loglikelihood function becomes are linear square function, and the entire loglikelihood function is a linear least square function by itself.

Corollary 5.8
If the dynamic g represents an LTI model, $f_{X_u} \in \mathcal{N}(0, \sigma^2(X_u))$, $f_{X_y} \in \mathcal{N}(0, \sigma^2(X_y))$ and $f_{X_g} \in \mathcal{N}(0, \sigma^2(X_f))$, then the $\hat{\theta}_{ML}(N) = \hat{\theta}_{LS}(N)$.

Hence, the LTI problem can be solved using techniques we discussed in Chapter 3.

5.4.2 Bayes estimator

Similar to the Maximum likelihood estimator, we also introduced the Bayes estimator

$$\hat{\theta}_{\mathrm{BA}}(N) = \operatorname*{argmax}_{\theta} f_\theta(\theta \mid u, z) \tag{4.6}$$

in a very general way. In the dynamic case, we require identical knowledge, i.e. the probability density function of the parameters f_θ must be at hand. Additionally, as displayed in the Maximum likelihood estimator case, we require the probability density functions of the „measurements“, which in the dynamic case corresponds to the input pdf f_{X_u}, the output pdf f_{X_y} and the equation error pdf f_{X_g}.

If this data is given, then the approach of Section 4.2 can be directly transfered. To this end, we apply the Bayes rule

$$f_\theta(\theta \mid u, z) = \frac{f(z \mid \theta, u) f_\theta(\theta)}{f(z)},$$

where again $f(z \mid \theta, u)$ is given the likelihood function

$$f(z \mid u_0, \theta) = f_{X_u}(u^m(k) - u(k)) \cdot f_{X_y}(y^m(k) - y(k)) \cdot f_{X_x}(z_k - g(u_0(k), \theta)). \tag{5.5}$$

Hence, we again obtain the following result:

Corollary 5.9
If f_{X_u}, f_{X_y}, f_{X_x} and f_θ are normal and measurements z_k are iid, then the Bayes estimator problem is a linear least square problem.

Chapter 6

Kalman filtering

The famous Kalman filter belongs to the class of so called recursive identification methods. The idea of such methods is to iteratively update the estimate utilizing new measurements at hand. Following this approach, an online processing of the results is possible. Additionally, one could generalize the approach by introducing a „forgetting factor“ to the cost function, which renders the method to be adaptive by design.

In contrast to our previous analysis of input–output models, the Kalman filter is designed for a state space systems of the form

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n), \end{aligned} \tag{6.1}$$

which is subject to equation and measurement noise. Hence, we have a systematic distinction between the internal state x and the externally viewable measurements y. Here, we discuss basic properties of the Kalman filter and construct a respective algorithm. Since the Kalman filter idea is quite involved, we start of by a simple introduction into recursive identification based on the mean value calculation.

6.1 Recursive identification

There exist two systematically different ways to compute an estimator like those we discussed in Chapters 3 and 4: In the first case, the optimization is postponed till all measurements are available. The second case, on the other hand, evaluates the estimate each time a new sample is available. So far, we have seen the postponement approach, but now we focus on the second recursive case.

A straightforward solution to generate such a procedure is to redo all the calculations after each sample. Such an approach is numerically robust and requires no further insight, yet it may be computationally expensive depending on the number of samples and the complexity of the computation process. For example, it is simple to recompute the mean value, but it is a complex task to solve a nonlinear optimization problem for a dynamical ARX model. Hence, reformulating the problem such that only the newly required calculations are made, recuperating all the previous results, may allow us to generate a more efficient solution method.

Before coming to a more elaborate variant of this approach, we consider the simple example of the mean value computation

$$\hat{\theta}(N) = \frac{1}{N} \sum_{k=1}^{N} z_k.$$

Using this formula, we can recompute the mean value once a new measurement is available via

$$\hat{\theta}(N+1) = \frac{1}{N+1}\sum_{k=1}^{N+1} z_k.$$

To recuperate the previous sum, we can equivalently evaluate

$$\begin{aligned}\hat{\theta}(N+1) &= \frac{1}{N+1}\sum_{k=1}^{N} z_k + \frac{1}{N+1} z_{N+1} \\ &= \frac{N}{N+1}\hat{\theta}(N) + \frac{1}{N+1} z_{N+1}.\end{aligned}$$

Although this form already meets our requirements of reusing previous computations, it is possible to rearrange it to a more suitable expression:

$$\hat{\theta}(N+1) = \hat{\theta}(N) + \frac{1}{N+1}\Big(z_{N+1} - \hat{\theta}(N)\Big)$$

Although this expression is very simple, it is very informative because almost every recursive algorithm ca be reduced to a similar form. The following observations can be made:

- The new estimate $\hat{\theta}(N+1)$ equals the old estimate $\hat{\theta}(N)$ plus a correction term, that is $\frac{1}{N+1}\Big(z_{N+1} - \hat{\theta}(N)\Big)$.
- The correction term consists of two terms by itself: a gain factor $\frac{1}{N+1}$ and an error term.
- The gain factor decreases towards zero as more measurements are already accumulated in the previous estimate. This means that in the beginning of the experiment, less importance is given to the old estimate $\hat{\theta}(N)$, and more attention is paid to the new incoming measurements. When N starts to grow, the error term becomes small compared to the old estimate. The algorithm relies more and more on the accumulated information in the old estimate $\hat{\theta}(N)$ and it does not vary it that much for accidental variations of the new measurements. The additional bit of information in the new measurement becomes small compared with the information that is accumulated in the old estimate.
- The second term $z_{N+1} - \hat{\theta}(N)$ is an error term. it makes the difference between the predicted value of the next measurement on the basis of the model and the actual measurement z_{k+1}.
- When properly initiated, i.e. $\hat{\theta}(1) = z_1$, this recursive result is exactly equal to the non recursive implementation. However, from a numerical point of view, it is a very robust procedure as calculation errors etc. are compensated in each step.

6.2 Construction of the Kalman filter

As stated earlier, the Kalman filter we discuss here deals with state space models of the form (6.1), which are excited by the known input signal u and disturbed by the equation noise source X_x. Additionally, the output quantities y are disturbed by a measurement noise source X_y. The aim of the Kalman filter is to estimate the state x of the system from the measurements z. As we will see in this chapter, the Kalman filter operates by propagating the mean and covariance of the state through time. Our approach to deriving the Kalman filter will involve the following steps:

1. First, we discuss a mathematical description of the model dynamics whose states we want to estimate. Here, we focus on LTI state space models of the form (6.1).
2. Next, we implement equations that describe the propagation of the mean and the covariance of the state with time respectively. These equations form a dynamic system by themselves.
3. These equations are used to implement a recursive algorithm and form the basis for the derivation of the Kalman filter because:
   (a) The mean of the state is the Kalman filter estimate of the state.
   (b) The covariance of the state is the covariance of the Kalman filter state estimate.
4. Every time we receive a new measurement, we update the mean and covariance of the state similar to the simple example displayed in Section 6.1.

In order to classify the Kalman filter problem, we first require a formal distinction of problems regarding information and time dependency:

- $x(n-k)$: an interpolation problem,
- $x(n)$: a filtering problem,
- $x(n+k)$: an prediction (or extrapolation) problem.

The Kalman filter is no only a classical filtering problem, i.e. the data is not computed based on current information. Instead, an internal dynamic for the mean value is constructed and propagated, such that new information can be integrated recursively. To this end, we have to distinguish between an a priori and an a posteriori estimate of the expected value.

6.2.1 Model dynamics and assumptions

The Kalman filter system can be stated in both continuous and discrete time. Within this work, we focus on the discrete time version given by

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) + X_x(n) \\ y(n) &= Cx(n) + X_y(n), \end{aligned} \tag{6.2}$$

where x, u, X_x, y and X_y are vectors and A, B and C are matrices, see also Figure 6.1 for a corresponding block diagram. Here, we suppose the following to hold:

Assumption 6.1
Regarding system (6.2) we have that

- the matrices A, B and C are known,
- the matrix B satisfies $B = 0$,
- the random variables X_x and X_y are independent variables,
- the probability density functions f_{X_x} and f_{X_y} are normal distributions,
- the expected values satisfy $\mathrm{E}\left(X_x(k)\right) = 0$ and $\mathrm{E}\left(X_y(k)\right) = 0$ and
- the covariance matrices are given by

$$\mathrm{Cov}\left(X_x(k), X_x(j)\right) = R_x\delta_{kj} \quad \text{and} \quad \mathrm{Cov}\left(X_y(k), X_y(j)\right) = R_y\delta_{kj}.$$

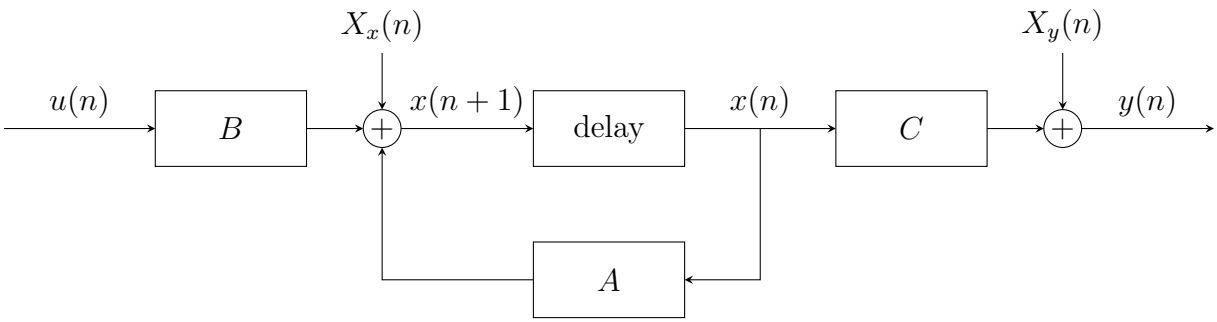


Figure 6.1: Block diagram of the state space system (6.2)

To shorten the notation, we introduce the vector

$$Y(n) := \{y(1), \dots, y(n)\}$$

and denote

$$P(n) := \mathrm{Cov}(x(n) \mid Y(n)) = \mathrm{E}\left([x(n) - \mathrm{E}(x(n) \mid Y(n))] [x(n) - \mathrm{E}(x(n) \mid Y(n))]^\top \right)$$
$$Q(n) := AP(n)A^\top + R_x(n).$$

Given this problem setting, we can now start to derive internal dynamic of the Kalman filter, i.e. the dynamic of the mean value and the covariance.

6.2.2 Propagation of mean and covariance

To write down the Kalman filter dynamics, we first need to construct the propagation of the mean value and the covariance regarding past information. Casually speaking, we need to know how these properties evolve regarding past information without new measurements.

Lemma 6.2
Given a system (6.2) *such that Assumption 6.1 holds. Then we have*

$$\mathrm{E}(x(n+1) \mid Y(n)) = A\mathrm{E}(x(n) \mid Y(n)). \tag{6.3}$$

Proof. Since we have

$$\begin{aligned} \mathrm{E}(x(n+1) \mid Y(n)) &= \mathrm{E}(Ax(n) + X_x(n) \mid Y(n)) \\ &= A\mathrm{E}(x(n) \mid Y(n)) + \mathrm{E}(X_x(n) \mid Y(n)) \\ &= A\mathrm{E}(x(n) \mid Y(n)). \end{aligned}$$

the assertion follows directly. □

Lemma 6.3
Given a system (6.2) *such that Assumption 6.1 holds. Then we have*

$$\mathrm{Cov}(x(n+1) \mid Y(n)) = AP(n)A^\top + R_x = Q(n). \tag{6.4}$$

Proof. Incorporating the definition of the notation, we see

$$\begin{aligned}
&\mathrm{Cov}(x(n+1) \mid Y(n)) = \mathrm{Cov}(Ax(n) + X_x(n) \mid Y(n)) \\
&= \mathrm{E}\left(\left(Ax(n) + X_x(n) - \mathrm{E}\left(Ax(n) + X_x(n) \mid Y(n)\right)\right)\right. \\
&\qquad \left.\left(Ax(n) + X_x(n) - \mathrm{E}\left(Ax(n) + X_x(n) \mid Y(n)\right)\right)^\top\right) \\
&= \mathrm{E}\left(\left(Ax(n) + X_x(n) - A\mathrm{E}\left(x(n) \mid Y(n)\right)\right)\left(Ax(n) + X_x(n) - A\mathrm{E}\left(x(n) \mid Y(n)\right)\right)^\top\right) \\
&= \mathrm{E}\left(A\left(x(n) - \mathrm{E}\left(x(n) \mid Y(n)\right)\right)\left(x(n) - \mathrm{E}\left(x(n) \mid Y(n)\right)\right)^\top A^\top\right) \\
&\quad + \mathrm{E}\left(X_x(n)x(n)^\top \mid Y(n)\right)A^\top + A\mathrm{E}\left(x(n)X_x(n)^\top \mid Y(n)\right) \\
&\quad - \mathrm{E}\left(X_x(n)\mathrm{E}\left(x(n) \mid Y(n)\right)^\top \mid Y(n)\right)A^\top + A\mathrm{E}\left(\mathrm{E}\left(x(n) \mid Y(n)\right)X_x(n)^\top \mid Y(n)\right) \\
&\quad + \mathrm{E}\left(\mathrm{E}\left(x(n) \mid Y(n)\right)\mathrm{E}\left(x(n) \mid Y(n)\right)^\top \mid Y(n)\right) \\
&= A\mathrm{E}\left(\left(x(n) - \mathrm{E}\left(x(n) \mid Y(n)\right)\right)\left(x(n) - \mathrm{E}\left(x(n) \mid Y(n)\right)\right)^\top\right)A^\top + R_x \\
&= AP(n)A^\top + R_x = Q(n).
\end{aligned}$$

which concludes the proof. □

Now that we know the estimate of the mean value and the covariance under the system dynamics, we can move forward to integrate a new measurement.

6.2.3 Derivation of the Kalman dynamics

To derive an update formula of the estimate of the mean value and the covariance computed in the previous section, we need to construct the probability density function of $x(n+1)$. The idea here is to compute an estimate of $x(n+1)$ such that the probability of a respective realization after the measurement of $y(n+1)$ is maximal. This probability density function, in turn, requires an extension of Bayes' rule, which can be derived from the conditional probability density functions

$$\begin{aligned}
f(a,b,c) &= f(a \mid b,c)f(b,c) = f(a \mid b,c)f(b \mid c)f(c) \\
f(a,b,c) &= f(a,b \mid c)f(c).
\end{aligned}$$

Combining these two equations, we obtain

$$f(a \mid b,c) = \frac{f(a,b,c)}{f(b \mid c)f(c)} = \frac{f(a,b \mid c)f(c)}{f(b \mid c)f(c)} = \frac{f(a,b \mid c)}{f(b \mid c)}.$$

Substituting $a = x(n+1)$, $b = y(n+1)$ and $c = Y(n)$ reveals

$$\begin{aligned}
f(x(n+1) \mid y(n+1), Y(n)) &= \frac{f(x(n+1), y(n+1) \mid Y(n))}{f(y(n+1) \mid Y(n))} \\
&= \frac{f(y(n+1) \mid x(n+1), Y(n))f(x(n+1) \mid Y(n))}{f(y(n+1) \mid Y(n))} \\
&= \frac{f_{X_y}(y(n+1) - CAx(n))f(x(n+1) \mid Y(n))}{f(y(n+1) \mid Y(n))}
\end{aligned} \tag{6.5}$$

where we have used that given $Y(n)$, we obtain $x(n+1) = Ax(n)$. Expression (6.5) is very informative. On the left hand side, we find the socalled „a posteriori“ probability density

function of $x(n+1)$, which includes the knowledge obtained from the measurement $y(n+1)$. The a posteriori pdf is calculated form the „a priori“ pdf by taking the latest measurement $y(n+1)$ into account.

In the following part, we are going to determine $x(n+1)$ such that the probability of realizing $x(n+1)$ after the measurement $y(n+1)$ is maximal. Note that we imposed the limitation that the probability density function of the noise X_x and X_y are normal distributions, cf. Assumption 6.1. Since the covariance matrix $\mathrm{Cov}(x(n+1) \mid Y(n))$ is given by Lemma 6.3 and R_x, R_y are given by Assumption 6.1, the probability density functions f_{X_x} and f_{X_y} are determined completely. The denominator of (6.5) is independent of $x(n+1)$ and can therefore be considered as constant when finding the maximum. Hence, we have

$$\begin{aligned}
&\max_{x(n+1)} f(x(n+1) \mid y(n+1), Y(n)) = \\
&= \max_{x(n+1)} e^{-\frac{1}{2}(y(n+1)-CA\mathrm{E}(x(n)|Y(n)))^\top R_y^{-1}(y(n+1)-CA\mathrm{E}(x(n)|Y(n)))} \\
&\qquad \cdot e^{-\frac{1}{2}(x(n+1)-A\mathrm{E}(x(n)|Y(n)))^\top Q^{-1}(n+1)(x(n+1)-A\mathrm{E}(x(n)|Y(n)))} \\
&= \max_{x(n+1)} e^{-\frac{1}{2}(x(n+1)-A\mathrm{E}(x(n)|Y(n)))^\top \left(Q^{-1}(n+1)+C^\top R_y^{-1}C\right)(x(n+1)-A\mathrm{E}(x(n)|Y(n)))}
\end{aligned}$$

From this equation, we directly obtain

$$\mathrm{Cov}\left(x(n+1) \mid Y(n+1)\right) = P(n+1) = Q(n+1)^{-1} + C^\top R_y^{-1} C. \tag{6.6}$$

In order to compute the maximizer of $f(x(n+1) \mid y(n+1), Y(n))$, it is sufficient to minimize the exponent of the above expression. Considering the necessary first order condition, we obtain

$$\left(Q^{-1}(n+1) + C^\top R_y^{-1} C\right)\left(x(n+1) - A\,\mathrm{E}\left(x(n) \mid Y(n)\right)\right) = 0$$

In order to obtain stationarity of the evolution, we require $x(n+1) = \mathrm{E}\left(x(n+1) \mid Y(n+1)\right)$. Inserting this into the necessary condition reveals

$$\begin{aligned}
&\left(Q^{-1}(n+1) + C^\top R_y^{-1} C\right) \mathrm{E}\left(x(n+1) \mid Y(n+1)\right) \\
&\quad = Q^{-1}(n+1) A\,\mathrm{E}\left(x(n) \mid Y(n)\right) + C^\top R_y^{-1} C A\,\mathrm{E}\left(x(n) \mid Y(n)\right)
\end{aligned}$$

Now, we can use the matrix inverse lemma

$$P = \left(Q^{-1} + C^\top R_y C\right)^{-1} = Q - QC^\top \left(CQC^\top + R_y\right)^{-1} CQ$$

and the relation

$$\left(Q + C^\top R_y^{-1} C\right)^{-1} C^\top R_y^{-1} = QC^\top \left(CQC^\top + R_y\right)^{-1}$$

to obtain

$$\begin{aligned}
&\mathrm{E}\left(x(n+1) \mid Y(n+1)\right) = \\
&\quad = A\,\mathrm{E}\left(x(n) \mid Y(n)\right) + Q(n+1)C^\top \left(CQ(n+1)C^\top + R_y\right)^{-1} \left(y(n+1) - CA\,\mathrm{E}\left(x(n) \mid Y(n)\right)\right)
\end{aligned} \tag{6.7}$$

6.2.4 Integration of mean and covariance into a recursive algorithm

To shorten notation, we introduce the abbreviation

$$X(n) := \mathrm{E}\left(x(n) \mid Y(n)\right),$$

we can formulate the following recursive algorithm:

Algorithm 6.4 (Kalman filter for LTI systems without external input)
Given a given LTI model with initial conditions R_x, R_y and $X(1)$, set $P(1) = R_x$.
For $n = 1, \dots$ do

$$Q(n+1) = AP(n)A^\top + R_x \tag{6.8}$$
$$K(n+1) = Q(n+1)C^\top \left(CQ(n+1)C^\top + R_y\right)^{-1} \tag{6.9}$$
$$P(n+1) = (\mathrm{Id} - K(n+1)C)\, Q(n+1) \tag{6.10}$$
$$X(n+1) = AX(n) + K(n+1)\left(y(n+1) - CAX(n+1)\right) \tag{6.11}$$

The algorithm contains several factors, which exhibit a good interpretation regarding the computations made earlier in this chapter. Here, the time component plays an important role.

- The matrix $Q(n+1) = P(n+1 \mid n)$ represents the a priori covariance matrix of $X(n+1) = \mathrm{E}\left(x(n+1) \mid Y(n+1)\right)$ using n measurements only.
- Similarly, the matrix $P(n+1)$ corresponds to the a posteriori covariance matrix of $X(n+1) = \mathrm{E}\left(x(n+1) \mid Y(n+1)\right)$ using $n+1$ measurements.
- Considering the dynamic of the system, the vector $AX(n)$ reveals the extrapolated state variable based on the model dynamics A and n measurements.
- Projecting on the output, the vector $CA\mathrm{E}\,(n)$ represents the expected output given the extrapolated state of the system.

We like to note that, within the algorithm, the matrices Q, P and K are independent of the measurements. For this reason, they can be computed beforehand which lowers the computational complexity of the filter. Additionally, the method remains usable when the noise is not normally distributed. In that case, however, the solution found by the filter is no longer an optimal one.

Similar to the case defined by Assumption 6.1, we can consider the more general LTI case with external inputs, i.e. $B \neq 0$. Recall, that the remaining assumptions are still in place, that is

Assumption 6.5
Regarding system (6.2) we have that

- the matrices A, B and C are known,
- the random variables X_x and X_y are independent variables,
- the probability density functions f_{X_x} and f_{X_y} are normal distributions,
- the expected values satisfy $\mathrm{E}\left(X_x(k)\right) = 0$ and $\mathrm{E}\left(X_y(k)\right) = 0$ and
- the covariance matrices are given by

$$\mathrm{Cov}\left(X_x(k), X_x(j)\right) = R_x\delta_{kj} \quad \text{and} \quad \mathrm{Cov}\left(X_y(k), X_y(j)\right) = R_y\delta_{kj}.$$

Given these assumptions, the computations displayed before in this chapter can be modified and the following algorithm can be derived:

Algorithm 6.6 (Kalman filter for LTI systems with external input)
Given a given LTI model with initial conditions R_x, R_y and $X(1)$, set $P(1) = R_x$.
For $n = 1, \dots$ do

$$Q(n+1) = AP(n)A^\top + R_x \tag{6.12}$$
$$K(n+1) = Q(n+1)C^\top \left(CQ(n+1)C^\top + R_y\right)^{-1} \tag{6.13}$$
$$P(n+1) = (\mathrm{Id} - K(n+1)C)\, Q(n+1) \tag{6.14}$$
$$X(n+1) = AX(n) + Bu(n) + K(n+1)\left(y(n+1) - CAX(n+1) - CBu(n)\right) \tag{6.15}$$

6.3 Example

Consider an inertial measurement unit (IMU) to be given, which is capable of measuring all three angular velocities around the body fixed coordinate (BFC) axis of the unit as well as the three acceleration forces in the directions of the BFC axis. The measurements are obtained from gyros and accelerometers respectively. IMUs are typically used to maneuver aircraft, including unmanned aerial vehicles (UAVs), among many others, and spacecraft, including satellites and landers. This data can then be used by a computer to continually calculate the vehicle's current position. One way to do this is to integrate over time the sensed acceleration, together with an estimate of gravity, to calculate the current velocity for each of the six degrees of freedom. In a second step, one can integrate the velocity to calculate the current position, which leads to a typical double integrator system.

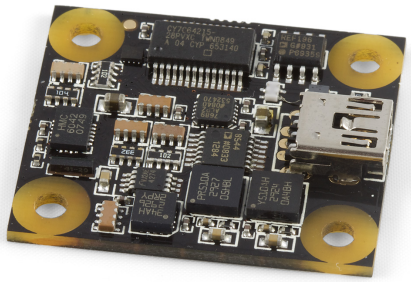
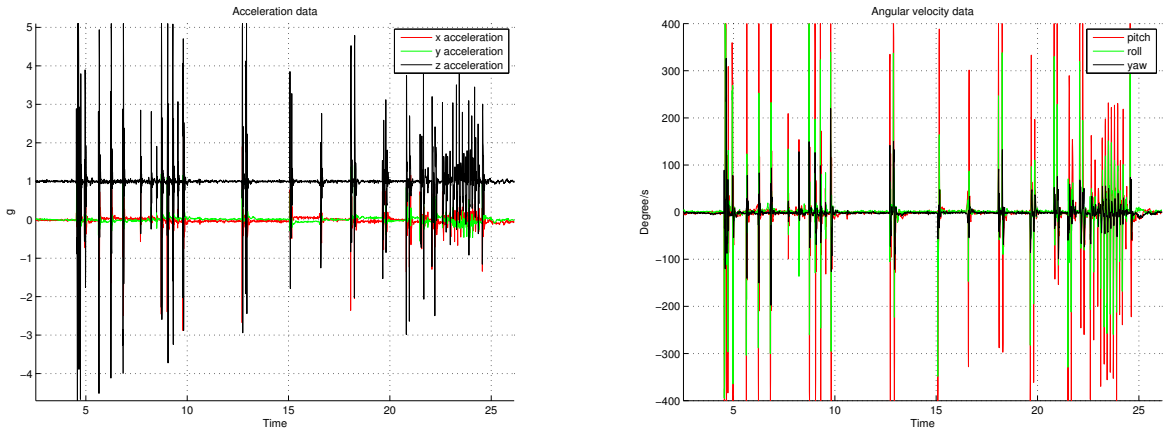


Figure 6.2: Inertial measurement unit (IMU)

Unfortunately, such a method suffers from accumulated error. The reason for this error lies in the construction of the method: The sensors detect accelerations and velocity only once within a sampling interval. Hence, these states may change within the interval and the method cannot recognize that change. Instead, the integration accumulates the error, which may grow exponentially over time. Note that reducing the length of the sampling intervals will not solve the problem as the error is systemic.

A sample of such measurements are displayed in Figures 6.3 and 6.4. Figure 6.3 shows details for common results in car experiments for bumpy roads, where we can see high vertical accelerations. The data set was recorded for a 1:8 model car running an off road track. From the data, we can see that the vertical forces are extremely large, up to $5g$ in upwards and $-4g$

downwards. The angular velocity rates, on the other hand, are rather small. Note that the high values for pitch and roll are due to a singularity in the sensors, which provide data from a $[0, 360]$ degree interval.

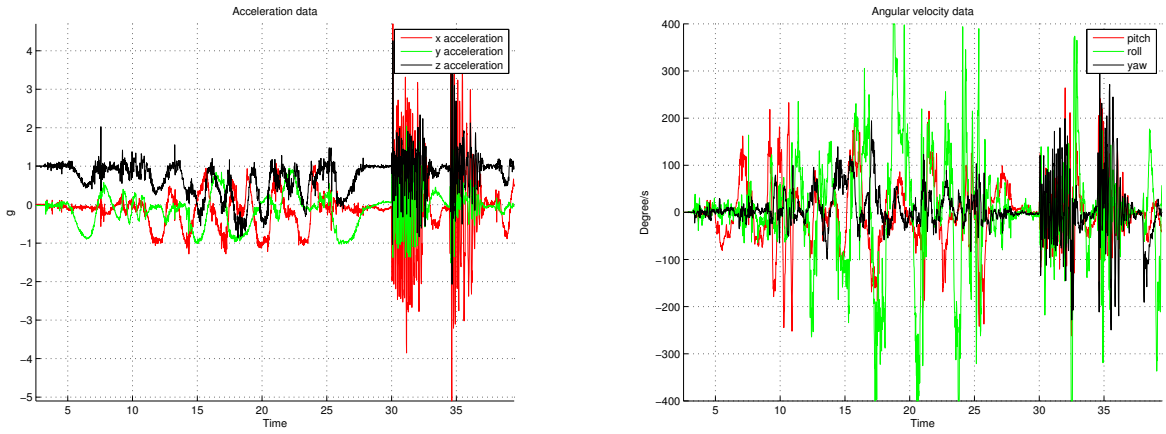


(a) IMU acceleration data in BFC

(b) IMU velocity data in BFC

Figure 6.3: IMU measurement data from gyros and accelerometers for sudden strikes

The second measurement data was obtained during quadcopter flights, where curvy maneuvers in all three axis occurred. From the data, one can see that the maneuvers were rather extreme and the copter crashed twice at the end. Here, the data changes are much smaller compared to the first case, at least until the crash occurred.

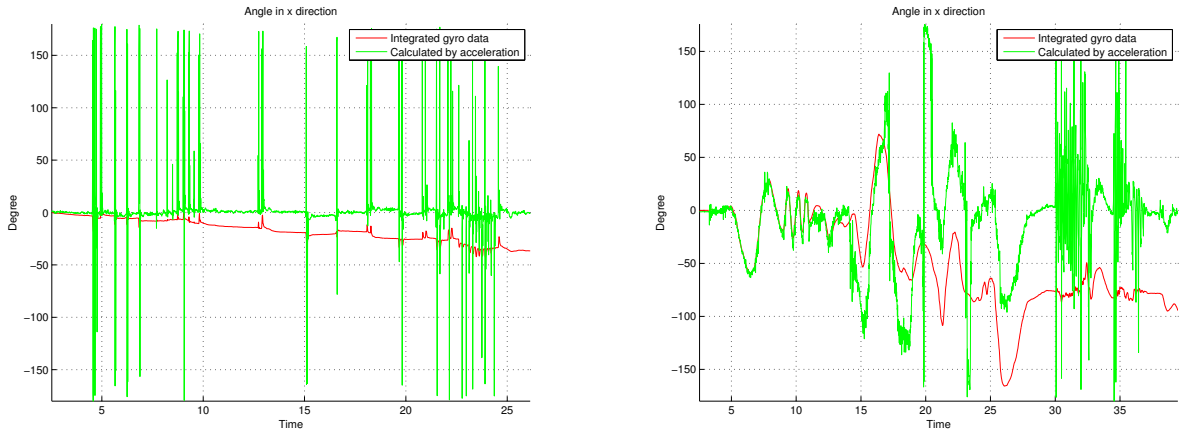


(a) IMU acceleration data in BFC

(b) IMU velocity data in BFC

Figure 6.4: IMU measurement data from gyros and accelerometers for continuous changes

In both cases, we were interested in the current state of the system. For the sake of simplicity, we focus on the pitch angle. Here, we like to note that the accelerometer data typically jitters and is not that accurate. The gyros on the other hand give us quite good data, but as we have discussed before, a simple integration may result in exponential errors illustrated in Figure 6.5.



(a) Results for angles using one sensor family only

(b) Results for angles using one sensor family only

Figure 6.5: IMU angular results for using sensor families separately

Within Figure 6.5, we displayed the results of two different computations: For one, we used the accelerometer data to evaluate

$$\hat{\theta}_1 = \frac{180^\circ}{\pi} \cdot \arctan 2(\ddot{x}_{3,\mathrm{BFC}}, \ddot{x}_{2,\mathrm{BFC}}). \tag{6.16}$$

Since this estimate is based on accelerometer data only, there is no drift in the result. Secondly, we used a simple integration of the timestamped angular velocity data

$$\hat{\theta}_1(n+1) = \hat{\theta}_{\mathrm{pitch}}(n) + (t_{n+1} - t_n)\, \dot{x}_{1,\mathrm{BFC}}. \tag{6.17}$$

We observe that in both cases the angle computed by (6.17) diverges from the result of (6.16).

Now, we apply the Kalman filter to this problem to fuse the advantages of the gyro (no jitter) and the accelerometer (no drift). To this end, we define the model dynamics (6.1) by

$$x(n) = \begin{pmatrix} x_{1,\mathrm{BFC}}(n) \\ \dot{x}_{1,\mathrm{BFC}}(n) \end{pmatrix}, \qquad u(n) = \frac{\pi}{180^\circ} \cdot \dot{x}_{1,\mathrm{BFC}}(n)$$
$$A(n) = \begin{pmatrix} 1 & -(t_{n+1} - t_n) \\ 0 & 1 \end{pmatrix}, \qquad B(n) = \begin{pmatrix} (t_{n+1} - t_n) \\ 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

which gives us the system

$$x(n+1) = \begin{pmatrix} 1 & -(t_{n+1} - t_n) \\ 0 & 1 \end{pmatrix} x(n) + \begin{pmatrix} (t_{n+1} - t_n) \\ 0 \end{pmatrix} u(n) \tag{6.18}$$
$$y(n) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(n) \tag{6.19}$$

The Kalman filter is initialized using the accelerometer data to generate an initial value of the estimator

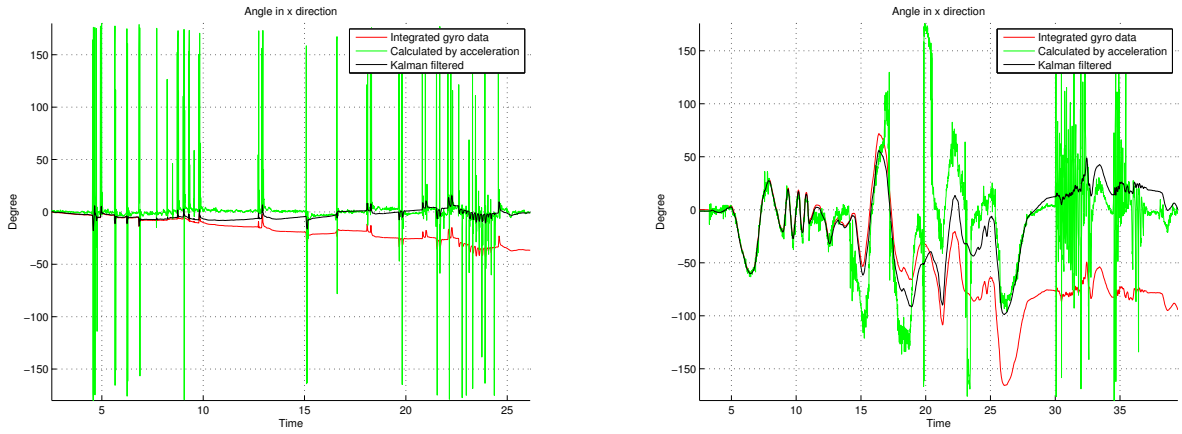
$$x(0) = \begin{pmatrix} \frac{180^\circ}{\pi} \cdot \arctan 2(\ddot{x}_{3,\mathrm{BFC}}, \ddot{x}_{2,\mathrm{BFC}}) \\ 0 \end{pmatrix},$$

and the approximated covariance matrices of the disturbances

$$R_x = \begin{pmatrix} \mathrm{E}\left(\frac{\pi}{180^\circ} \cdot 0.0257 \cdot (t_{n+1} - t_n)^2\right) & 0 \\ 0 & 10^{-8} \end{pmatrix}, \qquad R_y = \frac{\pi}{180^\circ} \cdot 15,$$

which are based on physical properties of the sensors and a freely chosen bias correction value for $R_{x2,2}$.

The resulting estimates of the Kalman filter based on the above are shown in Figure 6.6.



(a) Results for Kalman filter using data from Figure 6.3 (b) Results for Kalman filter using data from Figure 6.4

Figure 6.6: IMU Kalman filter fusion results in comparison to single sensor family results

Within this figure, we can clearly see the improvement of the Kalman filter sensor fusion. Note only is the jitter of the accelerometer reduced drastically, but also the drift of the gyros is eliminated after a certain startup phase. Within the startup, the Kalman filter estimate resembles the gyro results. Then, the filter reaches a point where it recognizes the growing error and compensates for it. Within Figure 6.6, we can identify this point by the starting divergence of the filter result from the integration one. From that point forward, the Kalman filter estimate converges towards the accelerometer result, yet without it strong pulses.

Appendices

Appendix A

Programs

Here, we display the programs used throughout the script. The programs may be used for a better personal understanding of the estimation process and the involved notions.

A.1 Motivating example of the electric circuit

```
clear all;
close all;

N = 100;

% Generating the measurements
% Group A
rng(1);
voltage = 1+0.2*randn(1,N);
rng(1);
current = 1+0.3*randn(1,N);

% % Group B
% rng(1);
% voltage = 1+0.2*randn(1,N);
% rng(1);
% current = 1+0.3*(rand(1,N)-0.5);

% Plotting measured voltages
figure(1);
hold on;
plot(1:N,voltage);
axis on;
axis ([1 N 0 2]);
xlabel('Measurement number', 'Fontsize', 14);
ylabel('Measured voltage (V)', 'Fontsize', 14);
grid on;
hold off;

% Plotting measured currents
figure(2);
hold on;
plot(1:N,current);
axis on;
axis ([1 N 0 2]);
xlabel('Measurement number', 'Fontsize', 14);
```

```
ylabel('Measured current (I)', 'Fontsize', 14);
grid on;
hold off;

% Computing and plotting the resistance
for i=1:size(voltage,2)
    resistance(i) = voltage(1,i)/current(1,i);
end
figure(3);
hold on;
plot(1:N,resistance);
axis on;
axis ([1 100 0 5]);
xlabel('Measurement number', 'Fontsize', 14);
ylabel('Measured value (R)', 'Fontsize', 14);
grid on;
hold off;
```

Program A.1: Matlab program for generating measurements for the electric circuit shown in Figure 1.1.

```
clear all;
close all;

Nmax = 100000;

% Group A
rng(1);
voltage = 1+0.2*randn(1,Nmax);
rng(1);
current = 1+0.3*randn(1,Nmax);

% % Group B
% rng(1);
% voltage = 1+0.2*randn(1,Nmax);
% rng(1);
% current = 1+0.3*(rand(1,Nmax)-0.5);

% Compute different estimators
for i=1:Nmax
    R_SA(i) = 1/i * sum(voltage(1,1:i)./current(1,1:i));
    R_EV(i) = sum(voltage(1,1:i))/sum(current(1,1:i));
    R_LS(i) = (1/i * sum(voltage(1,1:i).*current(1,1:i)))/(1/i * sum(current
        (1,1:i).*current(1,1:i)));
end

% Plot estimated resistances
figure(1);
semilogx(1:Nmax,R_SA,'-b', ...
     1:Nmax,R_EV,'-r', ...
     1:Nmax,R_LS,'-g');
xlabel('Number of measurements', 'Fontsize', 14);
ylabel('Estimated resistance', 'Fontsize', 14);
grid on;
legend('R_{SA}','R_{EV}','R_{LS}');

% Compute frequencies of estimation results
```

```
R_SA_min = min(R_SA);
R_SA_max = max(R_SA);
R_SA_pdf_x = linspace(R_SA_min,R_SA_max,100);
%histc(R_SA,R_SA_pdf_x);
R_SA_pdf_y=histc(R_SA,R_SA_pdf_x);

R_EV_min = min(R_EV);
R_EV_max = max(R_EV);
R_EV_pdf_x = linspace(R_EV_min,R_EV_max,100);
%hist(R_EV,R_EV_pdf_x);
R_EV_pdf_y=histc(R_EV,R_EV_pdf_x);

R_LS_min = min(R_LS);
R_LS_max = max(R_LS);
R_LS_pdf_x = linspace(R_LS_min,R_LS_max,100);
%hist(R_LS,R_LS_pdf_x);
R_LS_pdf_y=histc(R_LS,R_LS_pdf_x);

% Plot frequencies of estimation results
figure(2);
plot(R_SA_pdf_x,R_SA_pdf_y,'-b', ...
     R_EV_pdf_x,R_EV_pdf_y,'-r',...
     R_LS_pdf_x,R_LS_pdf_y,'-g');
xlabel('Estimated resistance', 'Fontsize', 14);
ylabel('frequency', 'Fontsize', 14);
grid on;
axis on;
axis ([0.9 1.1 0 90000]);
legend('R_{SA}','R_{EV}','R_{LS}');

% Comparing frequency development of estimation results
for i=3:4
    R_SA_min = min(R_SA(1:10^i));
    R_SA_max = max(R_SA(1:10^i));
    R_SA_pdf_x = linspace(R_SA_min,R_SA_max,100);
    R_SA_pdf_y = histc(R_SA(1:10^i),R_SA_pdf_x);

    R_EV_min = min(R_EV(1:10^i));
    R_EV_max = max(R_EV(1:10^i));
    R_EV_pdf_x = linspace(R_EV_min,R_EV_max,100);
    R_EV_pdf_y = histc(R_EV(1:10^i),R_EV_pdf_x);

    R_LS_min = min(R_LS(1:10^i));
    R_LS_max = max(R_LS(1:10^i));
    R_LS_pdf_x = linspace(R_LS_min,R_LS_max,100);
    R_LS_pdf_y = histc(R_LS(1:10^i),R_LS_pdf_x);

    figure(i);
    plot(R_SA_pdf_x,R_SA_pdf_y,'-b', ...
         R_EV_pdf_x,R_EV_pdf_y,'-r',...
         R_LS_pdf_x,R_LS_pdf_y,'-g');
    xlabel('Estimated resistance', 'Fontsize', 14);
    ylabel('frequency', 'Fontsize', 14);
    grid on;
    axis on;
    axis ([0.9 1.1 0 90000]);
    legend('R_{SA}','R_{EV}','R_{LS}');
end

```

```
%%%%%%%%%%%

% Computing standard deviation
for i=1:Nmax
    R_SA_var(i)=var(R_SA(1:i));
    R_EV_var(i)=var(R_EV(1:i));
    R_LS_var(i)=var(R_LS(1:i));
end

% Plotting standard deviation
figure(5);
loglog(1:Nmax,R_SA_var,'-b', ...
     1:Nmax,R_EV_var,'-r', ...
     1:Nmax,R_LS_var,'-g');
xlabel('Number of measurements', 'Fontsize', 14);
ylabel('Standard deviation', 'Fontsize', 14);
grid on;
legend('R_{SA}','R_{EV}','R_{LS}');

% Plotting analysis of realizations
figure(6);
current_min = min(current);
current_max = max(current);
current_x = linspace(current_min,current_max,100);
hist(current,current_x);
xlabel('Current', 'Fontsize', 14);
ylabel('# Realization', 'Fontsize', 14);
grid on;
```

Program A.2: Matlab program for analyzing the outcome of the electric circuit estimation problem

A.2 Linear least square

```
clear all;
close all;

% Set sample size
N = 100;

% Generate inputs (here: no inputs)

% Get measurements (in case of real measurements delete random parameters
% and replace measurement data)
% Generate random parameters
rng(1);
parameter(:,1) = 1 + 0.2 * randn(N,1);
% Generate measurements
rng(1);
measurement = parameter(:,1);

% Define model
K = ones(N,1);
% Evaluate linear least square formula
estimate = inv(transpose(K)*K)*transpose(K)*measurement;
```

```
22
23 % Plot results
24 figure(1);
25 hold on;
26 plot(1:N,measurement,'xb');
27 plot(1:N,K*estimate,'-r','LineWidth',2);
28 axis on;
29 axis tight;
30 xlabel('Measurement number', 'Fontsize', 14);
31 ylabel('Measured output', 'Fontsize', 14);
32 grid on;
33 hold off;
```

Program A.3: Matlab program for computing the linear least square estimator for Example 3.5

```
1  clear all;
2  close all;
3
4  % Set sample size
5  N = 100;
6
7  % Generate inputs
8  rng(1);
9  input(:,1) = linspace(1,2,N)';
10 input(:,2) = linspace(1,10,N)';
11
12 % Get measurements (in case of real measurements delete random parameters
13 % and replace measurement data)
14 % Generate random parameters
15 rng(1);
16 parameter(:,1) = 1 + 2.0 * randn(N,1);
17 parameter(:,2) = 2 + 1.0 * randn(N,1);
18 % Generate measurements
19 rng(1);
20 measurement = input(:,1) .* parameter(:,1) + input(:,2).^2 .* parameter(:,2);
21
22 % Define model
23 K = [input(:,1), input(:,2).^2];
24 % Evaluate linear least square formula
25 estimate = inv(transpose(K)*K)*transpose(K)*measurement;
26
27 % Plot results
28 figure(1);
29 hold on;
30 plot(1:N,measurement,'xb');
31 plot(1:N,K * estimate,'-r','LineWidth',2);
32 axis on;
33 axis tight;
34 xlabel('Measurement number', 'Fontsize', 14);
35 ylabel('Measured output', 'Fontsize', 14);
36 grid on;
37 hold off;
```

Program A.4: Matlab program for computing the linear least square estimator for Example 3.6

```
1 clear all;
```

```
close all;

% Set sample size
N = 100;

% Generate inputs
rng(1);
input(:,1) = linspace(1,2,N)';
input(:,2) = linspace(1,10,N)';

% Get measurements (in case of real measurements delete random parameters
% and replace measurement data)
% Generate random parameters
rng(1);
parameter(:,1) = 1 + 2.0 * randn(N,1);
parameter(:,2) = 2 + 1.0 * randn(N,1);
% Generate measurements
rng(1);
measurement = input(:,1) .* parameter(:,1) + input(:,2).^2 .* parameter(:,2);

% Define model
K = [input(:,1), input(:,2).^2];
% Define weighting matrix
W = diag(linspace(0,1,N));
% Evaluate linear least square formula
estimate = inv(transpose(K)*W*K)*transpose(K)*W*measurement;

% Plot results
figure(1);
hold on;
plot(1:N,measurement,'xb');
plot(1:N,K * estimate,'-r','LineWidth',2);
axis on;
axis tight;
xlabel('Measurement number', 'Fontsize', 14);
ylabel('Measured output', 'Fontsize', 14);
grid on;
hold off;
```

Program A.5: Matlab program for computing the linear least square estimator for Example 3.13

A.3 Kalman filtering

```
clear all;
close all;

% Read data
%data = load('kalman_data/data-zero');
%data = load('kalman_data/data-360x');
%data = load('kalman_data/data-x');
%data = load('kalman_data/data-x2');
data = load('kalman_data/data-smallhits');

% Reassign data to arrays
```

```
discreteTime = data(:,1); % in s
acceleration_x = data(:,2); % in G = 9.81 m/s^2
acceleration_y = data(:,3); % in G = 9.81 m/s^2
acceleration_z = data(:,4); % in G = 9.81 m/s^2
angular_velocity_x = data(:,5); % in degree/s
angular_velocity_y = data(:,6); % in degree/s
angular_velocity_z = data(:,7); % in degree/s
numberOfSamples = length(discreteTime);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute angle by integration of velocity data from gyros
angle_x_version1 = zeros(numberOfSamples,1);
for n=1:numberOfSamples-1
    timestep = discreteTime(n+1) - discreteTime(n);
    angle_x_version1(n+1) = angle_x_version1(n) + timestep * angular_velocity_x(
       n);
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute angle via accelerometer data
angle_x_version2 = 180/pi * atan2( acceleration_y, acceleration_z);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kalman Filter

% Set measurement noise of output
Ry_z = 15*pi/180; % standard deviation of measurement noise


% Initial state vector ( phi_x, bias_x )
x = [ atan2( acceleration_y(1), acceleration_z(1) ); 0 ];
% Set filter output for initial state
angle_x_version3 = zeros(numberOfSamples,1);
angle_x_version3(1) = 180/pi * x(1);

% Set equation error (process noise)
hApprox = 0.008;
varW1 = hApprox^2 * 0.0257*pi/180;
varW2 = 1e-8; % critical parameter for bias correction
Rx_z = [ varW1 0; 0 varW2];

% Initial covariance
P = Rx_z;

for n=1:numberOfSamples-1
    % System matrices
    timestep = discreteTime(n+1) - discreteTime(n);
    A = [ 1 -timestep; 0 1 ];
    B = [ timestep; 0];
    C = [ 1   0];

    % measurement in rad
    u = angular_velocity_x(n) / 180 * pi;
    y = atan2( acceleration_y(n+1), acceleration_z(n+1) );

    Q = A * P * A' + Rx_z;
    K = Q * C' * inv( C * Q * C' + Ry_z);
```

```
    P = (eye(2) - K * C) * Q;
    x = A * x + B * u + K * (y - C * A * x - C * B * u);

    angle_x_version3(n+1) = 180/pi * x(1);
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
% Plots
% Data plot
figure(1);
hold on;
plot(discreteTime,acceleration_x,'r-');
plot(discreteTime,acceleration_y,'g-');
plot(discreteTime,acceleration_z,'k-');
title('Acceleration data');
xlabel('Time');
ylabel('g');
legend('x acceleration', 'y acceleration', 'z acceleration');
axis tight;
grid on;

figure(2);
hold on;
plot(discreteTime,angular_velocity_x,'r-');
plot(discreteTime,angular_velocity_y,'g-');
plot(discreteTime,angular_velocity_z,'k-');
title('Angular velocity data');
xlabel('Time');
ylabel('Degree/s');
legend('pitch ', 'roll', 'yaw');
axis tight;
grid on;

% Results plot
figure(3);
hold on;
plot(discreteTime,angle_x_version1,'r-');
plot(discreteTime,angle_x_version2,'g-');
plot(discreteTime,angle_x_version3,'k-');
title('Angle in x direction');
xlabel('Time');
ylabel('Degree');
legend('Integrated gyro data', 'Calculated by acceleration', 'Kalman filtered');
axis tight;
grid on;
```

Program A.6: Matlab program for evaluating the Kalman filter for Example **??**