# Lecture Notes on Modelling and System Identification (Preliminary Draft)

Moritz Diehl

November 26, 2014

# Contents

# Preface to the Preliminary Manuscript

This lecture manuscript is written to accompany a lecture course on "Modelling and System Identification" given in the winter term 2014/15 at the University of Freiburg. Some parts and figures are based on a previous manuscript from the same course a year earlier, which was compiled by Benjamin Völker, and on the lecture notes from a course on numerical optimization the author has previously taught at the University of Leuven. Aim of the present manuscript is that it shall serve to the students as a reference for study during the semester. It follows the general structure of the lecture course and is written during the semester. The current version is a preliminary version only. A complete version shall be finalized in the weeks after the end of the semester, such that the final script can be used for exam preparation.

Freiburg, November 2014 -February 2015 *Moritz Diehl*

# Chapter 1

# Introduction

The lecture course on Modelling and System Identification (MSI) has as its aim to enable the students to create models that help to predict the behaviour of systems. Here, we are particularly interested in dynamic system models, i.e. systems evolving in time. With good system models, one can not only predict the future (like in weather forecasting), but also control or optimize the behaviour of a technical system, via feedback control or smart input design. Having a good model gives us access to powerful engineering tools. This course focuses on the process to obtain such models. It builds on knowledge from three fields: Systems Theory, Statistics, and Optimization. We will recall necessary concepts from these three fields on demand during the course. For a motivation for the importance of statistics in system identification, we first look at a very simple example taken from the excellent lecture notes of J. Schoukens [Sch13]. The course will then first focus on identification methods for static models and their statistical properties, and review the necessary concepts from statistics and optimization where needed. Later, we will look at different ways to model dynamic systems and how to identify them. For a much more detailed and complete treatment of system identification, we refer to the textbook by L. Ljung [Lju99].

## 1.1 Mathematical Notation

Within this lecture we use $\mathbb{R}$ for the set of real numbers, $\mathbb{R}_+$ for the non-negative ones and $\mathbb{R}_{++}$ for the positive ones, $\mathbb{Z}$ for the set of integers, and $\mathbb{N}$ for the set of natural numbers including zero, i.e. we identify $\mathbb{N} = \mathbb{Z}_+$. The set of real-valued vectors of dimension $n$ is denoted by $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ denotes the set of matrices with $n$ rows and $m$ columns. By default, all vectors are assumed to be column vectors, i.e. we identify $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. We usually use square brackets when presenting vectors and matrices elementwise. Because will often deal with concatenations of several vectors, say $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, yielding a vector in $\mathbb{R}^{n+m}$, we abbreviate this concatenation sometimes as $(x, y)$ in the text, instead of the correct but more clumsy equivalent notations $[x^\top, y^\top]^\top$ or

$$\begin{bmatrix} x \\ y \end{bmatrix}.$$

Square and round brackets are also used in a very different context, namely for intervals in $\mathbb{R}$, where for two real numbers $a < b$ the expression $[a, b] \subset \mathbb{R}$ denotes the closed interval containing both boundaries $a$ and $b$, while an open boundary is denoted by a round bracket, e.g. $(a, b)$ denotes the open interval and $[a, b)$ the half open interval containing $a$ but not $b$.

When dealing with norms of vectors $x \in \mathbb{R}^n$, we denote by $\|x\|$ an arbitrary norm, and by $\|x\|_2$ the Euclidean norm, i.e. we have $\|x\|_2^2 = x^\top x$. We denote a weighted Euclidean norm with a positive definite weighting matrix $Q \in \mathbb{R}^{n\times n}$ by $\|x\|_Q$, i.e. we have $\|x\|_Q^2 = x^\top Q x$. The $L_1$ and $L_\infty$ norms are defined by $\|x\|_1 = \sum_{i=1}^n |x_i|$ and $\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$. Matrix norms are the induced operator norms, if not stated otherwise, and the Frobenius norm $\|A\|_{\mathrm{F}}$ of a matrix $A \in \mathbb{R}^{n\times m}$ is defined by $\|A\|_{\mathrm{F}}^2 = \mathrm{trace}(AA^\top) = \sum_{i=1}^n \sum_{j=1}^m A_{ij} A_{ij}$.

When we deal with derivatives of functions $f$ with several real inputs and several real outputs, i.e. functions $f : \mathbb{R}^n \to \mathbb{R}^m, x \mapsto f(x)$, we define the Jacobian matrix $\frac{\partial f}{\partial x}(x)$ as a matrix in $\mathbb{R}^{m\times n}$, following standard conventions. For scalar functions with $m = 1$, we denote the gradient vector as $\nabla f(x) \in \mathbb{R}^n$, a column vector, also following standard conventions. Slightly less standard, we generalize the gradient symbol to all functions $f : \mathbb{R}^n \to \mathbb{R}^m$ even with $m > 1$, i.e. we generally define in this lecture

$$\nabla f(x) = \frac{\partial f}{\partial x}(x)^\top \in \mathbb{R}^{n\times m}.$$

Using this notation, the first order Taylor series is e.g. written as

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^\top (x - \bar{x})) + o(\|x - \bar{x}\|)$$

The second derivative, or Hessian matrix will only be defined for scalar functions $f : \mathbb{R}^n \to \mathbb{R}$ and be denoted by $\nabla^2 f(x)$.

For square symmetric matrices of dimension $n$ we sometimes use the symbol $\mathbb{S}_n$, i.e. $\mathbb{S}_n = \{A \in \mathbb{R}^{n \times n} | A = A^\top\}$. For any symmetric matrix $A \in \mathbb{S}_n$ we write $A \succcurlyeq 0$ if it is a positive semi-definite matrix, i.e. all its eigenvalues are larger or equal to zero, and $A \succ 0$ if it is positive definite, i.e. all its eigenvalues are positive. This notation is also used for *matrix inequalities* that allow us to compare two symmetric matrices $A, B \in \mathbb{S}_n$, where we define for example $A \succcurlyeq B$ by $A - B \succcurlyeq 0$.

When using logical symbols, $A \Rightarrow B$ is used when a proposition $A$ implies a proposition $B$. In words the same is expressed by "If $A$ then $B$". We write $A \Leftrightarrow B$ for "$A$ if and only if $B$", and we sometimes shorten this to "$A$ iff $B$", with a double "f", following standard practice.

## 1.2 A Simple Example: Resistance Estimation

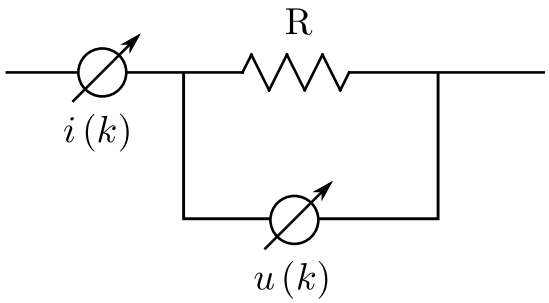


Figure 1.1: Resistance estimation example with resistor $R$ , current measurements $i(k)$, and voltage measurements $u(k)$ for $k = 1, 2, ...., N$. [Sch13]

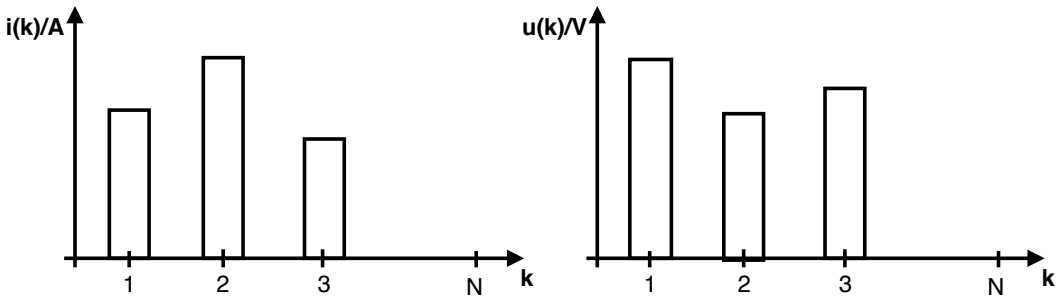


Figure 1.2: The measurements are a time series of discrete noisy values.

We know by Ohm's law that $u = Ri$. Given the measurements of $u$ and $i$, how should we compute an estimate $\hat{R}$ for the unknown resistance ? Let us look at three different approaches.

- Simple Approach

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

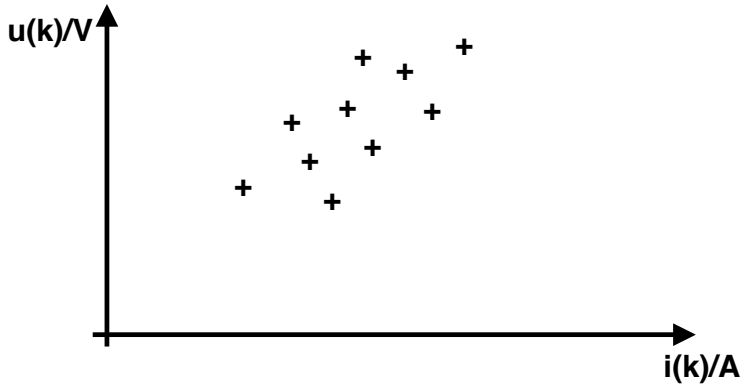


Figure 1.3: The same measurements in a voltage-current diagram.

- Error-in-Variables

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

- Least Squares

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

$$= \frac{\frac{1}{N}\sum_{k=1}^{N} u(k) \cdot i(k)}{\frac{1}{N}\sum_{k=1}^{N} i(k)^2} \tag{1.4}$$

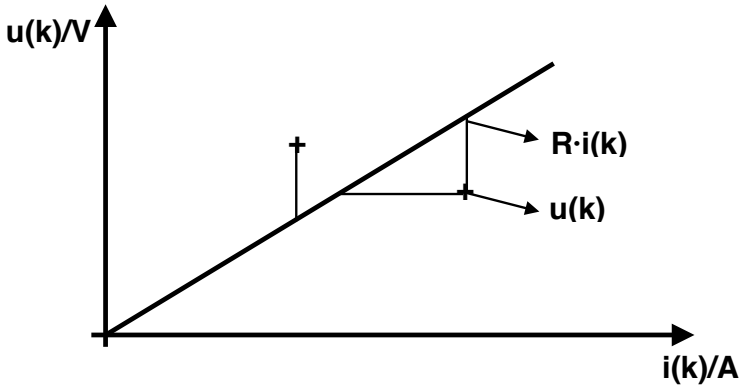


Figure 1.4: Principle behind the Least Squares approach: minimize the squared differences between $u(k)$ and $Ri(k)$.

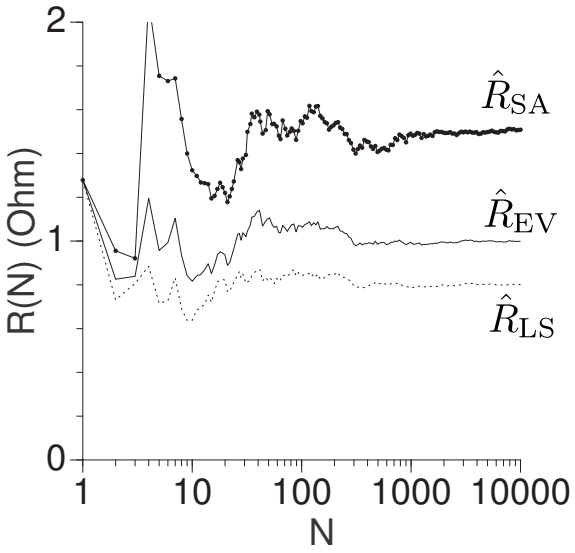


Figure 1.5: Different estimations of $R$ for increasing sample size $N$. At least two of the estimators are wrong. But which estimator gives us the true value of $R$ when $N$ goes to infinity ?

# Chapter 2

# Probability and Statistics in a Nutshell

In this chapter we review concepts from the fields of mathematical statistics and probability that we will need often in this course.

## 2.1 Random Variables and Probabilities

*Random variables* are used to describe the possible outcomes of experiments. Random variables are slightly different than the usual mathematical variables, because a random variable does not yet have a value. A random variable $X$ can take values from a given set, typically the real numbers. A specific value $x \in \mathbb{R}$ of the random variable $X$ will typically be denoted by a lower case letter, while the random variable itself will usually be denoted by an upper case letter; we will mostly, but not always stick to this convention. Note that the random variable $X$ itself is not a real number, but just takes values in $\mathbb{R}$. Nevertheless, we sometimes write sloppily $X \in \mathbb{R}$, or $Y \in \mathbb{R}^n$ to quickly indicate that a random variable takes scalars or vectors as values.

The probability that a certain event $A$ occurs is denoted by $P(A)$, and $P(A)$ is a real number in the interval $[0, 1]$. The event $A$ is typically defined by a condition that a random variable can satisfy or not. For example, the probability that the value of a random variable $X$ is larger than a fixed number $a$ is denoted by $P(X > a)$. If the event contains all possible outcomes of the underlying random variable $X$, its probability is one. If two events $A$ and $B$ are *mutually exclusive*, i.e. are never true at the same time, the probability that one or the other occurs is given by the sum of the two probabilities: $P(A \vee B) = P(A) + P(B)$. Two events can also be *independent* from each other. In that case, the probability that they both occur is given by the product: $P(A \wedge B) = P(A)P(B)$. If this is not the case, the two events are called *dependent*. We will often also write $P(A, B)$ for the joint probabily $P(A \wedge B)$. One can define the *conditional probability* $P(A|B)$ that an event $A$ occurs given that event $B$ has already occured. It is easy to verify the identity

$$P(A|B) = \frac{P(A, B)}{P(B)}.$$

An immediate consequence of this identity is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

which is known as *Bayes Theorem* after Thomas Bayes (17011761), who investigated how new evidence (event $B$ has occured) can update prior beliefs (a-priori probability $P(A)$).

## 2.2 Scalar Random Variables and Probability Density Functions

For a real valued random variable $X$, one can define the *Probability Density Function (PDF)* $p_X(x)$ which describes the behaviour of the random variable, and which is a function from the real numbers to the real numbers, i.e. $p_X : \mathbb{R} \to \mathbb{R}$, $x \mapsto p_X(x)$. Note that we use the name of the underlying random variable (here $X$) as index, when needed, and that the input argument $x$ of the PDF is just a real number. We will sometimes drop the index when the underlying random variable is clear from the context.

The PDF $p_X(x)$ is related to the probability that $X$ takes values in any interval $[a, b]$ in the following way:

$$P(X \in [a, b]) = \int_a^b p_X(x)\ \mathrm{d}x$$

Conversely, one can define the the PDF as

$$p_X(x) = \lim_{\Delta x \to 0} \frac{P(X \in [x, x + \Delta x])}{\Delta x}$$

Two random variables $X, Y$ are independent if the joint PDF $p_{X,Y}(x, y)$ is the product of the individual PDFs, i.e. $p_{X,Y}(x, y) = p_X(x)p_Y(y)$, otherwise they are dependent. The conditional PDF $p_{X|Y}$ of $X$ for given $Y$ is defined by

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x, y)}{p_Y(y)}.$$

As the above notation can become very cumbersome, we will occasionally also omit the index of the PDF and for example just express the above identity as

$$p(x|y) = \frac{p(x, y)}{p(y)}.$$

### 2.2.1 Mean and Variance

The *expectation value* or *mean* of a random variable is often denoted by $\mu_X$ and computed as $\int_{-\infty}^{\infty} x\ p_X(x)\mathrm{d}x$. More generally, one can compute the expectation of any function $f(X)$ of a random variable, which is by itself a random variable. It is convenient to introduce the expectation operator $\mathbb{E}\{\cdot\}$, which is defined by

$$\mathbb{E}\{f(X)\} := \int_{-\infty}^{\infty} f(x)\ p_X(x)\mathrm{d}x.$$

Due to the linearity of the integral, the expectation operator is also linear. This means that for any affine function $f(X) = a + bX$ with fixed numbers $a, b \in \mathbb{R}$, we have that

$$\mathbb{E}\{a + bX\} = a + b\ \mathbb{E}\{X\}.$$

Note that this is not possible for nonlinear functions $f(X)$, i.e. in general $\mathbb{E}\{f(X)\} \neq f(\mathbb{E}\{X\})$.

The *variance* of a random variable $X$ is a measure of how much the variable varies around the mean and is denoted by $\sigma_X^2$. It is defined as

$$\sigma_X^2 := \mathbb{E}\{(X - \mu_X)^2\}.$$

The square root of the variance, $\sigma_X = \sqrt{\sigma_X^2}$, is called the *standard deviation*.

### 2.2.2 Examples

...

## 2.3 Multidimensional Random Variables

...

### 2.3.1 Mean and Covariance Matrix

The expectation operator can also be applied to vector valued random variables, where the expectation is just computed for each component separately. We denote the mean of a random vector $X$ by $\mu_X = \mathbb{E}\{X\}$. Note that $\mu_X$ is a vector of the same dimension as $X$. We generalize the variance to the so-called *covariance matrix* $\Sigma_X \in \mathbb{R}^{n \times n}$, which contains all variances and covariances in a single matrix. It is given by $\Sigma_X = \mathrm{Cov}(X)$ where the covariance operator is defined by

$$\mathrm{Cov}(X) = \mathbb{E}\{(X - \mu_X)(X - \mu_X)^\top\}.$$

It is easy to verify the identity $\mathrm{Cov}(X) = \mathbb{E}\{XX^\top\} - \mu_X\mu_X^\top$.

### 2.3.2 Multidimensional Normal Distribution

We say that a vector valued random variable $X$ is normally distributed with mean $\mu$ and covariance $\Sigma$ if its PDF $p(x)$ is given by a multidimensional Gaussian as follows

$$p(x) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

As a shorthand, one also writes $X \sim \mathcal{N}(\mu, \Sigma)$ to express that $X$ follows a normal distribution with mean $\mu$ and covariance $\Sigma$ . One can verify by integral computations that indeed $\mathbb{E}\{X\} = \mu$ and $\mathrm{Cov}(X) = \Sigma$.

## 2.4 Statistical Estimators

An *estimator* uses possibly many measurements in order to estimate the value of some parameter vector that we typically denote by $\theta$ in this script. The parameter is not random, but its true value, $\theta_0$, is not known to the estimator. If we group all the measurements in a vector valued random variable $Y_N \in \mathbb{R}^N$, the estimator is a function of $Y_N$. We can denote this function by $\hat{\theta}_N(Y_N)$. Due to its dependence on $Y_N$, the estimate $\hat{\theta}_N(Y_N)$ is itself a random variable, for which we can define mean and covariance. Ideally, the expectation value of the estimator is equal to the true parameter value $\theta_0$. We then say that the estimator is unbiased.

**Definition 1 (Biased- and Unbiasedness)** *An estimator $\hat{\theta}_N$ is called* unbiased *iff $\mathbb{E}\{\hat{\theta}_N(Y_N)\} = \theta_0$, where $\theta_0$ is the true value of a parameter. Otherwise, it is called* biased*.*

**Example for unbiasedness: estimating the mean by an average** One of the simplest estimators tries to estimate the mean $\theta \equiv \mu_Y$ of a scalar random variable $Y$ by averaging $N$ measurements of $Y$. Each of these measurements $Y(k)$ is random, and overall the random vector $Y_N$ is given by $Y_N = [Y(1), \ldots, Y(N)]^\top$. The estimator $\hat{\theta}_N(Y_N)$ is given by

$$\hat{\theta}_N(Y_N) = \frac{1}{N}\Sigma_{k=1}^N Y(k).$$

It is easy to verify that this estimator is unbiased, because

$$\mathbb{E}\{\hat{\theta}_N(Y_N)\} = \frac{1}{N}\Sigma_{k=1}^N \mathbb{E}\{Y(k)\} = \frac{1}{N}\Sigma_{k=1}^N \mu_Y = \mu_Y$$

Because this estimator is often used it has a special name. It is called the *sample mean*.

In order to assess the performance of an unbiased estimator, one can regard the covariance matrix of the estimates, i.e.

$$\mathrm{Cov}(\hat{\theta}_N(Y_N))$$

The smaller this symmetric positive semi-definite matrix, the better the estimator. If two estimators $\hat{\theta}^A$ and $\hat{\theta}^B$ are both unbiased, and if the matrix inequality $\mathrm{Cov}(\hat{\theta}^A) \succcurlyeq \mathrm{Cov}(\hat{\theta}^B)$ holds, we can conclude that the estimator $\hat{\theta}^B$ has a better performance than estimator $\hat{\theta}^A$. Typically, the covariance of an estimator becomes smaller when an increasing number $N$ of measurements is used. Often the covariance even tends to zero as $N \to \infty$.

Some estimators are not unbiased, but if $N$ tends to infinity, their bias – i.e. the difference between true value and the mean of the estimate – tends to zero.

**Definition 2 (Asymptotic Unbiasedness)** *An estimator $\hat{\theta}_N$ is called* asymptotically unbiased *iff*

$$\lim_{N\to\infty} \mathbb{E}\{\hat{\theta}_N(Y_N)\} = \theta_0.$$

**Example for asymptotically unbiasedness: estimating the variance by the mean squared deviations** One of the simplest biased, but asymptotically unbiased estimators is tries to estimate the variance $\theta \equiv \sigma_Y^2$ of a scalar random variable $Y$ by taking $N$ measurements of $Y$, computing the experimental mean $M(Y_N) = \frac{1}{N}\Sigma_{k=1}^N Y(k)$, and then averaging the squared deviations from the mean

$$\hat{\theta}_N(Y_N) = \frac{1}{N}\Sigma_{k=1}^N (Y(k) - M(Y_N))^2$$

To show that it is biased, one has to consider that the sample mean $M(Y_N)$ is a random variable that is not independent from $Y_N$. One can compute its expectation value, which after some algebra is evaluated to be

$$\mathbb{E}\{\hat{\theta}_N\} = \frac{N-1}{N}\sigma_Y^2.$$

Only for $N \to \infty$, this estimator tends to the true value, so it is indeed *asymptotically unbiased*.

Because the bias is very easy to correct, in practice one rarely uses the above formula. Instead, to estimate the variance of a random variable $Y$, one uses the so called *sample variance* $S^2$ that is defined by

$$S^2 = \frac{1}{N-1}\Sigma_{n=1}^{N}(Y(n) - M(Y_N))^2.$$

Note the division by $N-1$ instead of $N$.

A stronger and even more desirable property than asymptotic unbiasedness is called *consistency*.

**Definition 3 (Consistency)** *An estimator $\hat{\theta}_N(Y_N)$ is called* consistent *if, for any $\epsilon > 0$, the probability $P(\hat{\theta}_N(Y_N) \in [\theta_0 - \epsilon, \theta_0 + \epsilon])$ tends to one as $N \to \infty$.*

It can be shown that an estimator is consistent if (a) it is asymptotically unbiased and (b) its covariance tends to zero as $N \to 0$.

## 2.5 Analysis of the Resistance Estimation Example

...

# Chapter 3

# Linear Least Squares Estimation

Linear least squares (LLS or just LS) is a technique that helps us to find a model that is linear in some unknown parameters $\theta \in \mathbb{R}^d$ . For this aim, we regard a sequence of measurements $y(1), \ldots, y(N) \in \mathbb{R}$ that shall be explained – they are also called the *dependent variables* – and another sequence of *regression vectors* $\phi(1), \ldots, \phi(N) \in \mathbb{R}^d$, which are regarded as the inputs of the model and are also called the *independent or explanatory variables*. Prediction errors are modelled by additive measurement noise $\epsilon(1), \ldots, \epsilon(N)$ with zero mean such that the overall model is given by

$$y(k) = \phi(k)^\top \theta + \epsilon(k), \quad \text{for} \quad k = 1, \ldots, N.$$

Let us in this section regard only scalar measurements $y(k)$, though LLS can be generalized easily to the case of several dependent variables. The task of LLS is to find an estimate $\hat{\theta}_{\mathrm{LS}}$ for the true but unknown parameter vector $\theta_0$. Often the ultimate aim is to be able to predict a $y$ for any given new values of the regression vector $\phi$ by the model $y = \phi^\top \hat{\theta}_{\mathrm{LS}}$.

## 3.1 Least Squares Problem Formulation

Idea of linear least squares is to find the $\theta$ that minimizes the sum of the squares of the prediction errors $y(k) - \phi(k)^\top \theta$, i.e. the *least squares cost function*

$$\sum_{k=1}^{N} \left( y(k) - \phi(k)^\top \theta \right)^2 .$$

Stacking all values $y(k)$ into one long vector $y_N \in \mathbb{R}^N$ and all regression vectors as rows into one matrix $\Phi_N \in \mathbb{R}^{N \times d}$, i.e.,

$$y_N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \quad \text{and} \quad \Phi_N = \begin{bmatrix} \phi(1)^\top \\ \vdots \\ \phi(N)^\top \end{bmatrix}$$

we can write the least squares cost function[1] as

$$f(\theta) = \|y_N - \Phi_N \theta\|_2^2.$$

The least squares estimate $\hat{\theta}_{\mathrm{LS}}$ is the value of $\theta$ that minimizes this function. Thus, we are faced with an *unconstrained optimization problem* that can be written as

$$\min_{\theta \in \mathbb{R}^d} f(\theta).$$

In estimation, we are mainly interested in the input arguments of $f$ that achieve the minimal value, which we call the minimizers. The set of minimizers $S^*$ is denoted by

$$S^* = \arg\min_{\theta \in \mathbb{R}^d} f(\theta).$$

[1] We recall that for any vector $x \in \mathbb{R}^n$, we define the Euclidean norm as $\|x\|_2 = \left( \sum_{i=1}^n x_i^2 \right)^{1/2} = (x^\top x)^{1/2}$.

Note that there can be several minimizers. If the minimizer is unique, we have only one value in the set, that we denote $\theta^*$, and we can slightly sloppily identify $\theta^*$ with $\{\theta^*\}$. The least squares estimator $\hat{\theta}_{\rm LS}$ is given by this unique minimizer, such that we will often write write

$$\hat{\theta}_{\rm LS} = \arg\min_{\theta\in\mathbb{R}^d} f(\theta).$$

But in order to compute the minimizer (or the set of minimizers), we need to solve an optimization problem. Let us therefore recall a few concepts from optimization, and then give an explicit solution formula for $\hat{\theta}_{\rm LS}$.

## 3.2 A Micro-Review of Unconstrained Optimization

Let us in this section use, as customary in optimization textbooks, the variable $x \in \mathbb{R}^n$ instead of $\theta \in \mathbb{R}^d$ as the unknown decision variable in the optimization problem. Throughout the course, we often want to solve unconstrained optimization problems of the form

$$\min_{x\in D} f(x), \tag{3.1}$$

where we regard objective functions $f : D \to \mathbb{R}$ that are defined on some open domain $D \subset \mathbb{R}^n$. We are only interested in minimizers that lie inside of $D$. We might have $D = \mathbb{R}^n$, but often this is not the case, e.g. as in the following example:

$$\min_{x\in(0,\infty)} \frac{1}{x} + x. \tag{3.2}$$

Let us state a few simple and well-known results from unconstrained optimization that are often used in this course.

**Theorem 1 (First Order Necessary Conditions)** *If $x^* \in D$ is local minimizer of $f : D \to \mathbb{R}$ and $f \in C^1$ then*

$$\nabla f(x^*) = 0. \tag{3.3}$$

**Definition 4 (Stationary Point)** *A point $\bar{x}$ with $\nabla f(\bar{x}) = 0$ is called a stationary point of $f$.*

Given the above theorem, stationarity is a necessary, but of course not a suffcient condition for optimality. There is one surprisingly large class of functions $f(x)$, however, for which stationarity is both necessary and sufficent for global optimality: the class of convex functions.

**Theorem 2 (Convex First Order Sufficient Conditions)** *Assume that $f : D \to \mathbb{R}$ is $C^1$ and convex. If $x^* \in D$ is a stationary point of $f$, then $x^*$ is a global minimizer of $f$.*

We will extensively make use of this theorem, because many of the optimization problems formulated in system identification are convex. An important convex objective function is the least squares cost function $f(x) = \|y - \Phi x\|_2^2$ that is the subject of this chapter. For general nonlinear cost functions $f(x)$, however, we need to look at second order derivatives in order to decide if a stationary point is a minimizer or not. There exist necessary and sufficient conditions that are straightforward generalizations of well-known results one dimensional analysis to $\mathbb{R}^n$.

**Theorem 3 (Second Order Necessary Conditions)** *If $x^* \in D$ is local minimizer of $f : D \to \mathbb{R}$ and $f \in C^2$ then*

$$\nabla^2 f(x^*) \succcurlyeq 0. \tag{3.4}$$

Note that the matrix inequality is identical with the statement that all eigenvalues of the Hessian $\nabla^2 f(x^*)$ must be non-negative. It is possible that the Hessian has one or more zero eigenvalues – whose eigenvectors corresponds to directions of zero-curvature in the cost function. Due to this fact, the second order necessary condition (3.4) is not sufficient for a stationary point $x^*$ to be a minimizer. This is illustrated by the simple one-dimensional functions $f(x) = x^3$ or $f(x) = -x^4$ for which $x^* = 0$ is a saddle point and a maximizer, respectively, though for both the necessary conditions $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succcurlyeq 0$ are satisfied. How can we obtain a sufficient optimality condition for general nonlinear, but smooth functions $f$?

**Theorem 4 (Second Order Sufficient Conditions and Stability under Perturbations)** *Assume that $f : D \to \mathbb{R}$ is $C^2$. If $x^* \in D$ is a stationary point and*

$$\nabla^2 f(x^*) \succ 0. \tag{3.5}$$

*then $x^*$ is a strict local minimizer of $f$. In addition, this minimizer is locally unique and is stable against small perturbations of $f$, i.e.there exists a constant $C$ such that for sufficiently small $p \in \mathbb{R}^n$ holds*

$$\|x^* - \arg\min_x (f(x) + p^\top x)\| \leq C\|p\|.$$

## 3.3 Solution of the Linear Least Squares Problem

The function $f(\theta) = \frac{1}{2} \parallel y_N - \Phi_N \theta \parallel_2^2$ is convex. Therefore local minimizers are found by just setting the gradient to zero. For notational convenience, we will in this section omit the subindex $N$ and write $f(\theta) = \frac{1}{2} \parallel y - \Phi\theta \parallel_2^2$, and we will refer to the components of $y$ with a simple subindex, i.e. write $y_k$ instead of $y(k)$. Also, we have introduced a factor $\frac{1}{2}$ in the objective, which does not change the minimizer. We introduced it because it will cancel a factor two that would otherwise be present in the first and second derivatives of $f$. To find the minimizer, let us compute the gradient of $f$.

$$\begin{aligned} \nabla f(\theta^*) = 0 \quad &\Leftrightarrow \quad \Phi^\top \Phi \theta^* - \Phi^\top y = 0 \\ &\Leftrightarrow \quad \theta^* = \underbrace{(\Phi^\top \Phi)^{-1} \Phi^\top}_{=\Phi^+} y \end{aligned} \tag{3.6}$$

**Definition 5 (Pseudo inverse)** *$\Phi^+$ is called the pseudo inverse of the matrix $\Phi$ and is a generalization of the inverse matrix. If $\Phi^\top \Phi \succ 0$, the pseudo inverse $\Phi^+$ is given by*

$$\Phi^+ = (\Phi^\top \Phi)^{-1} \Phi^\top \tag{3.7}$$

So far, $(\Phi^\top \Phi)^{-1}$ is only defined when $\Phi^\top \Phi \succ 0$. This holds if and only if $\mathrm{rank}(\Phi) = n$, i.e., if the columns of $\Phi$ are linearly independent. In this context, it is interesting to note that $\nabla^2 f(\theta) = \Phi^\top \Phi$, i.e. the pseudo inverse is well-defined if and only if the second order sufficient conditions for optimality are satisfied.

Later, we will generalize the pseudo inverse to the case that $\Phi$ has linearly dependent column vectors, i.e. that the matrix $\Phi^\top \Phi$ has one or more zero eigenvalues. Due to convexity of $f$, points with $\nabla f(\theta) = 0$ will still be minimizers in that case, but they will not be unique anymore. But let us first illustrate the regular case with $\Phi^\top \Phi \succ 0$ in two examples.

**Example 1 (Fitting a constant equals taking the average)** *Let us regard the simple optimization problem:*

$$\min_{\theta \in \mathbb{R}} \ \frac{1}{2} \sum_{i=1}^{N} (y_i - \theta)^2.$$

*This is a linear least squares problem, where the vector $y$ and the matrix $\Phi \in \mathbb{R}^{N \times 1}$ are given by*

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \ \Phi = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{3.8}$$

*Because $\Phi^\top \Phi = N$, it can be easily seen that*

$$\Phi^+ = (\Phi^\top \Phi)^{-1} \Phi^\top = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \tag{3.9}$$

*so we conclude that the local minimizer equals the average of the given points $y_i$:*

$$\theta^* = \Phi^+ y = \frac{1}{N} \sum_{i=1}^{N} y_i. \tag{3.10}$$

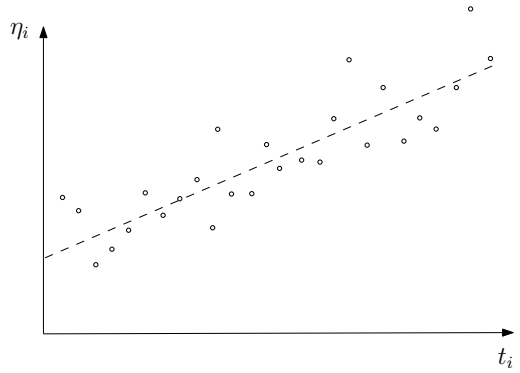


Figure 3.1: Linear regression for a set of data points $(t_i, y_i)$

**Example 2 (Fitting a line)** *Given data points $\{t_i\}_{i=1}^N$ with corresponding values $\{y_i\}_{i=1}^N$, find the 2-dimensional parameter vector $\theta = (\theta_1, \theta_2)$, so that the polynomial of degree one $p(t;\theta) = \theta_1 + \theta_2 t$ provides a prediction of $y$ at time $t$. The corresponding optimization problem looks like:*

$$\min_{\theta\in\mathbb{R}^2} \frac{1}{2}\sum_{i=1}^{N}(y_i - p(t_i;\theta))^2 = \min_{\theta\in\mathbb{R}^2} \frac{1}{2}\left\| y - \Phi \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right\|_2^2 \tag{3.11}$$

*where $y$ is the same vector as in* (3.8) *and $\Phi$ is given by*

$$\Phi = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{bmatrix}. \tag{3.12}$$

*The local minimizer is found by equation* (3.6)*, where the calculation of $(\Phi^\top\Phi)$ is straightforward:*

$$\Phi^\top\Phi = \left[\begin{array}{c|c} N & \sum t_i \\ \hline \sum t_i & \sum t_i^2 \end{array}\right] \tag{3.13}$$

## 3.4 Weighted Least Squares

One might want to give different weights to different residuals in the sum of the linear least squares cost function. This is important if the measurement errors $\epsilon(k)$ have zero mean and are independent, but are not identically distributed, such that they have different variances $\sigma_\epsilon^2(k)$. We would intuitively like to give less weight to those measurements which are corrupted by stronger noise. Weighting is mandatory if different measurements represent different physical units, if we want to avoid that we add squared apples to squared pears. Fortunately, the variance of each measurement has the same unit as the measurement squared, such that a division of each residual by the variance would make all terms free of units. For this reason one nearly always uses the following weighted least squares cost function:

$$f_{\mathrm{WLS}}(\theta) = \sum_{k=1}^{N} \frac{(y(k) - \phi^\top\theta)^2}{\sigma_\epsilon^2(k)},$$

and we will see that this cost function ensures the best possible performance of the least squares estimator. To bring this into a more compact notation, we can introduce a diagonal weighting matrix

$$W = \begin{bmatrix} \sigma_\epsilon^2(1) & & \\ & \ddots & \\ & & \sigma_\epsilon^2(N) \end{bmatrix}$$

and then write[2]

$$f_{\mathrm{WLS}}(\theta) = \|y - \Phi\theta\|_W^2.$$

Even more general, one might use any symmetric positive definite matrix $W \in \mathbb{R}^{N\times N}$ as weighting matrix. The optimal solution is given by

$$\hat{\theta}_{\mathrm{WLS}} = \arg\min f_{\mathrm{WLS}}(\theta) = (\Phi^\top W\Phi)^{-1}\Phi^\top W y.$$

There is an alternative way to represent the solution, using the matrix $\tilde{\Phi} = W^{\frac{1}{2}}\Phi$ and its pseudo inverse. To derive this alternative way, let us first state the fact that there exists a unique symmetric square root $W^{\frac{1}{2}}$ for any symmetric positive definite matrix $W$. For example, for a diagonal weighting matrix as above, the square root is given by

$$W^{\frac{1}{2}} = \begin{bmatrix} \sigma_\epsilon(1) & & \\ & \ddots & \\ & & \sigma_\epsilon(N) \end{bmatrix}.$$

With this square root matrix, we have the trivial identity $\|x\|_W^2 = \|W^{\frac{1}{2}}\|_2^2$ for any vector $x \in \mathbb{R}^N$ and can therefore write

$$f_{\mathrm{WLS}}(\theta) = \|\underbrace{W^{\frac{1}{2}}y}_{=:\tilde{y}} - \underbrace{W^{\frac{1}{2}}\Phi}_{=:\tilde{\Phi}}\theta\|_W^2.$$

Thus, the weighted least squares problem is nothing else than an unweighted least squares problem with rescaled measurements $\tilde{y} = W^{\frac{1}{2}}y$ and rescaled regressor matrix $\tilde{\Phi} = W^{\frac{1}{2}}\Phi$, and the solution can be computed using the pseudo inverse of $\tilde{\Phi}$ and is simply given by

$$\hat{\theta}_{\mathrm{WLS}} = \tilde{\Phi}^+\tilde{y}.$$

This way of computing the estimate is numerically more stable so it is in general preferable. An important observation is that the resulting solution vector $\hat{\theta}_{\mathrm{WLS}}$ does not depend on the total scaling of the entries of the weighting matrix, i.e. for any positive real number $\alpha$, the weighting matrices $W$ and $\alpha W$ deliver identical results $\hat{\theta}_{\mathrm{WLS}}$. Only for this reason it is meaningful to use unweighted least squares – they deliver the optimal result in the case that the measurement errors are assumed to be independent and identically distributed. But generally speaking, all least squares problems are in fact weighted least squares problems, because one always has to make a choice of how to scale the measurement errors. If one uses unweighted least squares, one implicitly chooses the unit matrix as weighting matrix, which makes sense for i.i.d. measurement errors, but otherwise not. For ease of notation, we will in the following nevertheless continue discussing the unweighted LS formulation, keeping in mind that any weighted least squares problem can be brought into this form by the above rescaling procedure.

## 3.5 Ill-Posed Least Squares and the Moore Penrose Pseudo Inverse

In some cases, the matrix $\Phi^\top\Phi$ is not invertible, i.e. it contains at least one zero eigenvalue. In this case the estimation problem is called ill-posed, because the solution is not unique. But there is still the possibility to obtain a solution of the least squares problem that might give a reasonable result. For this we have to use a special type of pseudo inverse. Let us recall that definition (3.7) of the pseudo inverse does only hold if $\Phi^\top\Phi$ is invertible. This implies that the set of optimal solutions $S^*$ has only one optimal point $\theta^*$, given by $S^* = \{\theta^*\} = (\Phi^\top\Phi)^{-1}\Phi y$. If $\Phi^\top\Phi$ is not invertible, the set of solutions $S^*$ is given by

$$S^* = \{\theta \mid \nabla f(\theta) = 0\} = \{\theta | \Phi^\top\Phi\theta - \Phi^\top y = 0\} \tag{3.14}$$

In order to pick a unique point out of this set, we might choose to search for the "minimum norm solution", i.e. the vector $\theta^*$ with minimum norm satisfying $\theta^* \in S^*$.

$$\min_{\theta\in\mathbb{R}^n} \quad \frac{1}{2}\|\theta\|_2^2 \quad \text{subject to} \quad \theta \in S^* \tag{3.15}$$

We will show below that this minimal norm solution is given by the so called "Moore Penrose Pseudo Inverse".

[2]Recall that for any positive definite matrix $W$ the weighted Euclidean norm $\|x\|_W$ is defined as $\|x\|_W = \sqrt{x^\top Wx}$.

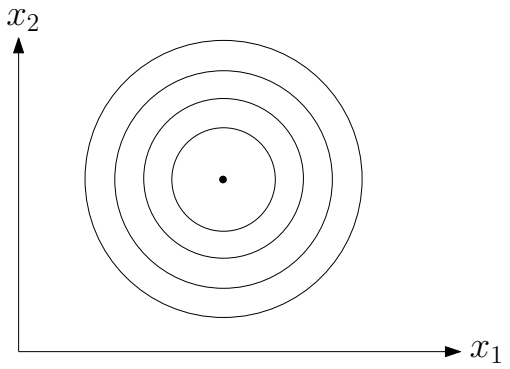


Figure 3.2: $\Phi^\top \Phi$ is invertible, resulting in a unique minimum.

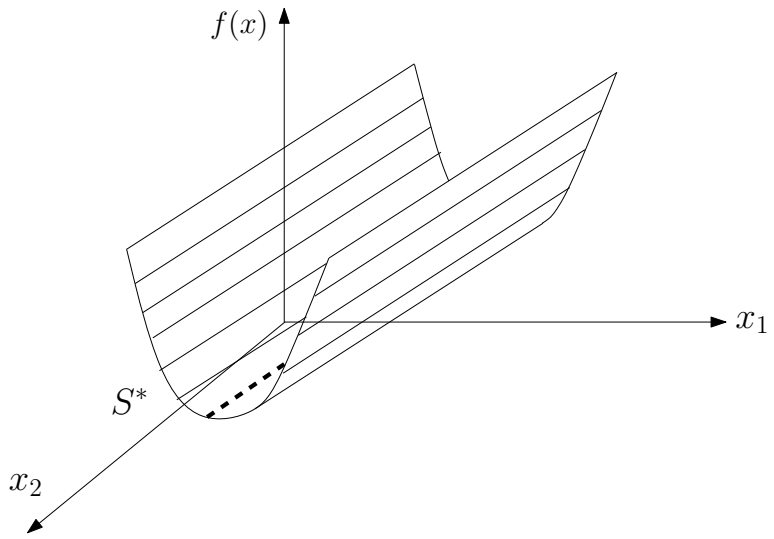


Figure 3.3: An example of an ill-posed problem, $\Phi^\top \Phi$ is not invertible

**Definition 6 (Moore Penrose Pseudo Inverse)** *Assume $\Phi \in \mathbb{R}^{m\times n}$ and that the singular value decomposition (SVD) of $\Phi$ is given by $\Phi = USV^\top$. Then, the Moore Penrose Pseudo Inverse $\Phi^+$ is given by:*

$$\Phi^+ = VS^+U^\top, \tag{3.16}$$

*where for*

$$S = \left[\begin{array}{cccccc} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_r & & \\ & & & & \ddots & \\ & & & & & 0 \\ \hline 0 & \dots & 0 & \dots & & 0 \end{array}\right] \quad \textit{holds} \quad S^+ = \left[\begin{array}{cccccc|c} \sigma_1^{-1} & & & & & & 0 \\ & \sigma_2^{-1} & & & & & \vdots \\ & & \ddots & & & & 0 \\ & & & \sigma_r^{-1} & & & \vdots \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & 0 & 0 \end{array}\right] \tag{3.17}$$

*If $\Phi^\top\Phi$ is invertible, then $\Phi^+ = (\Phi^\top\Phi)^{-1}\Phi^\top$ what easily can be shown:*

$$\begin{aligned} (\Phi^\top\Phi)^{-1}\Phi^\top &= (VS^\top U^\top USV^\top)^{-1}VS^\top U^\top \\ &= V(S^\top S)^{-1}V^\top VS^\top U^\top \\ &= V(S^\top S)^{-1}S^\top U^\top \\ &= V\left[\begin{array}{cccc} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_r^2 \end{array}\right]^{-1}\left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & \sigma_r & \end{array}\right. \left.\begin{array}{c} \\ \\ \\ 0 \end{array}\right] U^\top \\ &= VS^+U^\top \end{aligned}$$

### 3.5.1 Regularization for Least Squares

The minimum norm solution can be approximated by a "regularized problem"

$$\min_\theta \ \frac{1}{2}\|y - \Phi\theta\|_2^2 + \frac{\alpha}{2}\|\theta\|_2^2, \tag{3.18}$$

with small $\alpha > 0$, to get a unique solution

$$\begin{aligned} \nabla f(\theta) &= \Phi^\top\Phi\theta - \Phi^\top y + \alpha\theta && (3.19)\\ &= (\Phi^\top\Phi + \alpha\mathbb{I})\theta - \Phi^\top y, && (3.20)\\ \text{thus} \quad \theta^* &= (\Phi^\top\Phi + \alpha\mathbb{I})^{-1}\Phi^\top y && (3.21)\\ & && (3.22) \end{aligned}$$

**Lemma 1** $\lim_{\alpha\to 0}(\Phi^\top\Phi + \alpha\mathbb{I})^{-1}\Phi^\top = \Phi^+$, *with $\Phi^+$ the Moore Penrose Pseudo Inverse.*

**Proof:** Taking the SVD of $\Phi = USV^\top$, $(\Phi^\top\Phi + \alpha\mathbb{I})^{-1}\Phi^\top$ can be written in the form:

$$\begin{aligned} (\Phi^\top\Phi + \alpha\mathbb{I})^{-1}\Phi^\top &= (VS^\top U^\top USV^\top + \alpha\underbrace{\mathbb{I}}_{VV^\top})^{-1}\underbrace{\Phi^\top}_{US^\top V^\top} \\ &= V(S^\top S + \alpha\mathbb{I})^{-1}V^\top VS^\top U^\top \\ &= V(S^\top S + \alpha\mathbb{I})^{-1}S^\top U^\top \end{aligned}$$

Rewriting the right hand side of the equation explicitly:

$$= V \begin{bmatrix} \sigma_1^2+\alpha & & & & & \\ & \ddots & & & & \\ & & \sigma_r^2+\alpha & & & \\ & & & \alpha & & \\ & & & & \ddots & \\ & & & & & \alpha \end{bmatrix}^{-1} \left[\begin{array}{cccccc|c} \sigma_1 & & & & & & 0 \\ & \ddots & & & & & \vdots \\ & & \sigma_r & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & 0 & 0 \end{array}\right] U^\top$$

Calculating the matrix product simplifies the equation:

$$= V \left[\begin{array}{cccccc|c} \frac{\sigma_1}{\sigma_1^2+\alpha} & & & & & & 0 \\ & \ddots & & & & & \vdots \\ & & \frac{\sigma_r}{\sigma_r^2+\alpha} & & & & \\ & & & \frac{0}{\alpha} & & & \\ & & & & \ddots & & \\ & & & & & \frac{0}{\alpha} & 0 \end{array}\right] U^T$$

It can be easily seen that for $\alpha \to 0$ each diagonal element has the solution:

$$\lim_{\alpha \to 0} \frac{\sigma_i}{\sigma_i^2+\alpha} = \begin{cases} \frac{1}{\sigma_i} & \text{if} \quad \sigma_i \neq 0 \\ 0 & \text{if} \quad \sigma_i = 0 \end{cases} \tag{3.23}$$

With the above lemma, we have shown that the Moore Penrose Pseudo Inverse $\Phi^+$ solves the problem (3.18) for infinitely small $\alpha > 0$. Thus it selects $\theta^* \in S^*$ with minimal norm.

## 3.6 Statistical Analysis of the Weighted Least Squares Estimator

In the following, we analyse the weighted least squares estimator, and we make the following assumptions:

- the $N$ measurements $y(k)$ are generated by a model $y(k) = \phi(k)^\top \theta_0 + \epsilon(k)$ with $\theta_0$ the true but unknown parameter value
- the $N$ regression vectors $\phi(k)$ are deterministic and not corrupted by noise (attention: this assumption is often violated in practice)
- the $N$ noise terms $\epsilon(k)$ have zero mean. Often, we can also assume that they are independent from each other, or even that they are i.i.d., and this will have consequences on the optimal choice of weighting matrix, as we will see. We abbreviate $\epsilon_N = [\epsilon(1), \ldots, \epsilon(N)]^\top$.
- the weighted least squares estimator is computed as $\hat{\theta}_{\text{WLS}} = (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W y_N$.

We are most interested in the following questions: is the estimator biased or unbiased? What is its performance, i.e. what covariance matrix does the estimate have?

### 3.6.1 The expectation of the least squares estimator

Let us compute the expectation of $\hat{\theta}_{\text{WLS}}$.

$$\begin{aligned} \mathbb{E}\{\hat{\theta}_{\text{WLS}}\} &= \mathbb{E}\left\{(\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W y_N\right\} && (3.24) \\ &= (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \;\; \mathbb{E}\{y_N\} && (3.25) \\ &= (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \quad \mathbb{E}\{\Phi_N \theta_0 \qquad + \epsilon_N \} && (3.26) \\ &= (\Phi_N^\top W \Phi_N)^{-1} (\Phi_N^\top W \Phi_N) \quad \theta_0 \qquad + \; (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \; \mathbb{E}\{\epsilon_N\} && (3.27) \\ &= \qquad \theta_0 \qquad + \; 0 && (3.28) \end{aligned}$$

Thus, the weighted least squares estimator has the true parameter value $\theta_0$ as expectation value, i.e. it is an unbiased estimator. This fact is true independent of the choice of weighting matrix $W$.

### 3.6.2 The covariance of the least squares estimator

In order to assess the performance of an unbiased estimator, one can look at the covariance matrix of $\hat{\theta}_{\mathrm{WLS}}$. The smaller this covariance matrix in the matrix sense, the better is the estimator. Let us therefore compute the covariance matrix of $\hat{\theta}_{\mathrm{WLS}}$. Using the identity $\hat{\theta}_{\mathrm{WLS}} - \theta_0 = (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \epsilon_N$, it is given by

$$\begin{aligned}
\mathrm{Cov}(\hat{\theta}_{\mathrm{WLS}}) &= \mathbb{E}\{(\hat{\theta}_{\mathrm{WLS}} - \theta_0)(\hat{\theta}_{\mathrm{WLS}} - \theta_0)^\top\} && (3.29)\\
&= (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \, \mathbb{E}\left\{\epsilon_N \epsilon_N^\top\right\} W \Phi_N (\Phi_N^\top W \Phi_N)^{-1} && (3.30)\\
&= (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \quad \Sigma_{\epsilon_N} \quad W \Phi_N (\Phi_N^\top W \Phi_N)^{-1}. && (3.31)
\end{aligned}$$

Here, we have used the shorthand $\Sigma_{\epsilon_N} = \mathrm{Cov}(\epsilon)N)$. For different choices of $W$, the covariance $\mathrm{Cov}(\hat{\theta}_{\mathrm{WLS}})$ will be different. However, there is one specific choice that makes the above formula very easy: if we happen to know $\Sigma_{\epsilon_N}$ and would choose $W := \Sigma_{\epsilon_N}^{-1}$, we would obtain

$$\begin{aligned}
\mathrm{Cov}(\hat{\theta}_{\mathrm{WLS}}) &= (\Phi_N^\top W \Phi_N)^{-1} \Phi_N^\top W \quad W^{-1} \quad W \Phi_N (\Phi_N^\top W \Phi_N)^{-1} && (3.32)\\
&= (\Phi_N^\top W \Phi_N)^{-1} (\Phi_N^\top W \Phi_N)(\Phi_N^\top W \Phi_N)^{-1} && (3.33)\\
&= (\Phi_N^\top W \Phi_N)^{-1} && (3.34)\\
&= (\Phi_N^\top \Sigma_{\epsilon_N}^{-1} \Phi_N)^{-1}. && (3.35)
\end{aligned}$$

Interestingly, it turns out that this choice of weighting matrix is the optimal choice, i.e. for all other weighting matrices $W$ one has

$$\mathrm{Cov}(\hat{\theta}_{\mathrm{WLS}}) \succcurlyeq (\Phi_N^\top \Sigma_{\epsilon_N}^{-1} \Phi_N)^{-1}.$$

Even more, one can show that in case of Gaussian noise with zero mean and covariance $\Sigma_{\epsilon_N}$, the weighted linear least squares estimator with optimal weights $W = \Sigma_{\epsilon_N}^{-1}$ achieves the lower bound on the covariance matrix that any unbiased estimator can achieve (the so called Cramer-Rao lower bound).

## 3.7 Measuring the Goodness of Fit using R-Squared

One ... Coefficient of Determination - $R^2$ ...

## 3.8 Estimating the Covariance with a Single Experiment

So far, we have analysed the theoretical properties of the LS estimator, and we know that for independent identically distributed measurement errors, the unweighted least squares estimater gives us the optimal estimator. If the variance of the noise is $\sigma_\epsilon^2$, the least squares estimator $\hat{\theta}_{\mathrm{LS}} = \Phi_N^+ y_N$ is a random variable with the true parameter value $\theta_0$ as mean and the following covariance matrix:

$$\Sigma_{\hat{\theta}} := \mathrm{Cov}(\hat{\theta}_{\mathrm{LS}}) = \sigma_\epsilon^2 (\Phi_N^\top \Phi_N)^{-1}.$$

In addition, if the number of measurement $N$ in one experiment is large, by a law of large numbers, the distribution of $\hat{\theta}_{\mathrm{LS}}$ follows approximately a normal distribution, even if the measurement errors were not normally distributed. Thus, if one repeats the same experiment with the same $N$ regression vectors many times, the estimates $\hat{\theta}_{\mathrm{LS}}$ would follow a normal distribution characterized by these two parameters, i.e. $\hat{\theta}_{\mathrm{LS}} \sim \mathcal{N}(\theta_0, \Sigma_{\hat{\theta}})$. In a realistic application, however, the situation is quite different than in this analysis:

- First, we do of course *not* know the true value $\theta_0$
- Second, we do not repeat our experiment many times, but just do *one single experiment*.
- Third, we typically do *not* know the variance of the measurement noise $\sigma_\epsilon^2$.

Nevertheless, and surprisingly, if one makes the assumption that the noise is independent identically distributed, one is able to make a very good guess of the covariance matrix of the estimator, which we will describe here. The main reason is that we know the deterministic matrix $\Phi_N$ exactly. The covariance is basically given by the matrix $(\Phi_N^\top \Phi)^{-1}$, which only needs to be scaled by a factor, the unknown $\sigma_\epsilon^2$. Thus, we only need to find an

estimate for the noise variance. Fortunately, we have $N$ measurements $y(k)$ as well as the corresponding model predictions $\phi(k)^\top \hat{\theta}_{\mathrm{LS}}$ for $k = 1, \ldots, N$, so their average difference can be used to estimate the measurement noise. Because the predictions are based on fitting the $d$-dimensional vector $\hat{\theta}_{\mathrm{LS}}$ to the same measurements $y(k)$ that we want to use to estimate the measurement errors, we should not just take the average of the squared deviations $(y(k) - \phi(k)^\top \hat{\theta})^2$ – this would be a biased (though asymptotically unbiased) estimator. It can be shown that an unbiased estimate for $\sigma_\epsilon^2$ is obtained by

$$\hat{\sigma}_\epsilon^2 := \frac{1}{(N-d)} \sum_{k=1}^{N} (y(k) - \phi(k)^\top \hat{\theta}_{\mathrm{LS}})^2 = \frac{\|y_N - \Phi_N \hat{\theta}_{\mathrm{LS}}\|_2^2}{(N-d)}$$

Thus, our final formula for a good estimate $\hat{\Sigma}_{\hat{\theta}}$ of the true but unknown covariance $\mathrm{Cov}(\theta_{\mathrm{LS}})$ is

$$\hat{\Sigma}_{\hat{\theta}} := \hat{\sigma}_\epsilon^2 (\Phi_N^\top \Phi_N)^{-1} = \frac{\|y_N - \Phi_N \hat{\theta}_{\mathrm{LS}}\|_2^2}{(N-d)} (\Phi_N^\top \Phi_N)^{-1}.$$

What we have now are two quantities, an estimate $\hat{\theta}_{\mathrm{LS}}$ of the true parameter value $\theta_0$, as well as an estimate $\hat{\Sigma}_{\hat{\theta}}$ for the covariance matrix of this estimate. This knowledge helps us to make a strong statement about how probable it is that our our estimate is close to the true parameter value. Under the assumption that our linear model structure is correct and thus our estimator is unbiased, and the (slightly optimistic) assumption that our covariance estimate $\hat{\Sigma}_{\hat{\theta}}$ is equal to the true covariance $\Sigma_{\hat{\theta}}$ of the estimator $\hat{\theta}_{\mathrm{LS}}$, we can compute the probability that an uncertainty ellipsoid around the estimated $\hat{\theta}_{\mathrm{LS}}$ contains the (unknown) true parameter value $\theta_0$.

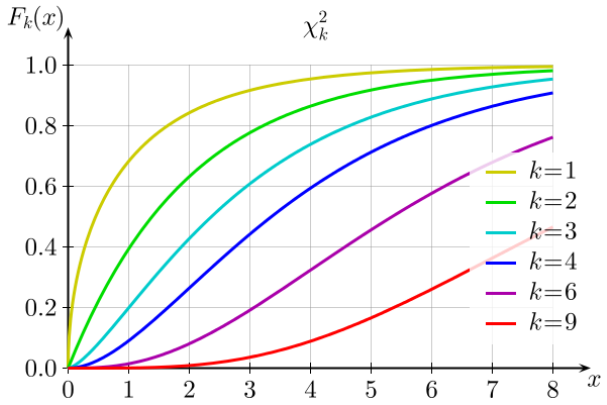


Figure 3.4: Cumulative density function $F(x, k)$ for the $\chi^2$-distribution with $k$ degrees of freedom (image: wikipedia).

In order to compute this probability, we need to use the cumulative density function (CDF) – i.e. the integral of the PDF – of the so called $\chi^2$-distribution (Chi-squared) with $k := d$ degrees of freedom. We denote this CDF, which is illustrated for up to $k = 9$ degrees of freedom in Figure 3.4, by $F(x, k)$. This function tells us how probable it is that the square of a $k$-dimensional, normally distributed variable $X \sim \mathcal{N}(0, \mathbb{I})$ with zero mean and unit covariance has a value smaller than $x$, i.e.

$$P(\|X\|_2^2 \leq x) = F(x, k).$$

Using the fact that under the above assumptions, the random variable $X := \Sigma_{\hat{\theta}}^{-\frac{1}{2}} (\hat{\theta}_{\mathrm{LS}} - \theta_0)$ is normally distributed with zero mean and unit covariance, we thus know that for any positive $x$ we have

$$P(\|\theta_0 - \hat{\theta}_{\mathrm{LS}}\|_{\Sigma_{\hat{\theta}}^{-1}}^2 \leq x) = F(x, d).$$

We can give another interpretation to the same fact: the probability that the true value $\theta_0$ is contained in the *confidence ellipsoid* $\mathcal{E}_x(\hat{\theta}_{\mathrm{LS}})$ defined by

$$\mathcal{E}_x(\hat{\theta}_{\mathrm{LS}}) := \{\theta \in \mathbb{R}^d \mid \|\theta - \hat{\theta}_{\mathrm{LS}}\|_{\Sigma_{\hat{\theta}}^{-1}}^2 \leq x\}$$

is given by

$$P\left(\ \theta_0 \in \mathcal{E}_x(\hat{\theta}_{\mathrm{LS}})\ \right) = F(x,d).$$

Note that in this expression, it is the ellipsoid which is random, not the true, but unknown, value $\theta_0$. We call the confidence ellipsoid for $x = 1$, i.e. the set

$$\mathcal{E}_1(\hat{\theta}_{\mathrm{LS}}) := \{\theta \in \mathbb{R}^d \mid \|\theta - \hat{\theta}_{\mathrm{LS}}\|^2_{\Sigma^{-1}_{\hat{\theta}}} \leq 1\}$$

the *one-sigma confidence ellipsoid*. The probability that the true value is contained in it decreases with increasing dimension $d = k$ of the parameter space and can be found in Figure 3.4 at $x = 1$.

Note that the variance for one single component of the parameter vector can be found as a diagonal entry in the covariance matrix, and that the probability that the true value of this single component is inside the one sigma interval around the estimated value is always 68.3%, independent of the parameter dimension $d$. This is due to the fact that each single component of $\hat{\theta}$ follows a one dimensional normal distribution.

For mathematical correctness, we have to note that we had to assume that the covariance matrix $\Sigma_{\hat{\theta}}$ is exactly known in order to make use of the $\chi^2$-distribution. On the other hand, in practice, we can only use its estimate $\hat{\Sigma}_{\hat{\theta}}$ in the definition of the confidence ellipsoid. A refined analysis, which is beyond our ambitions, would need to take into account that also $\hat{\Sigma}_{\hat{\theta}}$ is a random variable, which implies that $X := \hat{\Sigma}^{-\frac{1}{2}}_{\hat{\theta}}(\hat{\theta}_{\mathrm{LS}} - \theta_0)$ follows a distribution which is similar to, but not equal to a standard normal distribution. For the practice of least squares estimation, however, the above characterization of confidence ellipsoids with the $\chi^2$-distribution is accurate enough and can help us to assess the quality of an estimation result after a single experiment.

# Chapter 4

# Maximum Likelihood and Bayesian Estimation

...

## 4.1 Maximum Likelihood Estimation

**Definition 7 (Likelihood)** *The likelihood function $L(\theta)$ is a function of $\theta$ for given measurements $y$ that describes how likely the measurements would have been if the parameter would have the value $\theta$. It is defined as $L(\theta) = p(y|\theta)$, using the PDF of $y$ for given $\theta$.*

**Definition 8 (Maximum-Likelihood Estimate)** *The maximum-likelihood estimate of the unknown parameter $\theta$ is the parameter value that maximizes the likelihood function $L(\theta) = p(y|\theta)$.*

Assume $y_i = M_i(\overline{\theta}) + \epsilon_i$ with $\overline{\theta}$ the "true" parameter, and $\epsilon_i$ *Gaussian noise* with expectation value $\mathbb{E}(\epsilon_i) = 0$, $\mathbb{E}(\epsilon_i\,\epsilon_i) = \sigma_i^2$ and $\epsilon_i$, $\epsilon_j$ independent for $i \neq j$ . Then holds

$$p(y|\theta) = \prod_{i=1}^{m} p(y_i \mid \theta) \tag{4.1}$$

$$= C \prod_{i=1}^{m} \exp\left(\frac{-(y_i - M_i(\theta))^2}{2\sigma_i^2}\right) \tag{4.2}$$

with $C = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_i^2}}$. Taking the logarithm of both sides gives

$$\log p(y|\theta) = \log(C) + \sum_{i=1}^{m} -\frac{(y_i - M_i(\theta))^2}{2\sigma_i^2} \tag{4.3}$$

with a constant $C$. Due to monotonicity of the logarithm holds that the argument maximizing $p(y|\theta)$ is given by

$$\arg\max_{\theta\in\mathbb{R}^n} p(y|\theta) = \arg\min_{\theta\in\mathbb{R}^n} -\log(p(y|\theta)) \tag{4.4}$$

$$= \arg\min_{\theta\in\mathbb{R}^n} \sum_{i=1}^{m} \frac{(y_i - M_i(\theta))^2}{2\sigma_i^2} \tag{4.5}$$

$$= \arg\min_{\theta\in\mathbb{R}^n} \frac{1}{2} \left\|S^{-1}(y - M(\theta))\right\|_2^2 \tag{4.6}$$

Thus, the least squares problem has a statistical interpretation. Note that due to the fact that we might have different standard deviations $\sigma_i$ for different measurements $y_i$ we need to scale both measurements and model functions in

order to obtain an objective in the usual least squares form $\|\hat{y} - \hat{M}(\theta)\|_2^2$, as

$$\min_\theta \ \frac{1}{2} \sum_{i=1}^{n} \left( \frac{y_i - M_i(\theta)}{\sigma_i} \right)^2 = \min_\theta \ \frac{1}{2} \|S^{-1}(y - M(\theta))\|_2^2 \tag{4.7}$$

$$= \min_\theta \ \frac{1}{2} \|S^{-1}y - S^{-1}M(\theta)\|_2^2 \tag{4.8}$$

$$\text{with } S = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix}.$$

**Statistical Interpretation of Regularization terms:** Note that a regularization term like $\alpha\|\theta - \bar{\theta}\|_2^2$ that is added to the objective can be interpreted as a "pseudo measurement" $\bar{\theta}$ of the parameter value $\theta$, which includes a statistical assumption: the smaller $\alpha$, the larger we implicitly assume the standard deviation of this pseudo-measurement. As the data of a regularization term are usually given before the actual measurements, regularization is also often interpreted as "a priori knowledge". Note that not only the Euclidean norm with one scalar weighting $\alpha$ can be chosen, but many other forms of regularization are possible, e.g. terms of the form $\|A(\theta - \bar{\theta})\|_2^2$ with some matrix $A$.

### 4.1.1 $L_1$-Estimation

Instead of using $\|.\|_2^2$, i.e. the $L_2$-norm in the fitting problem, we might alternatively use $\|.\|_1$, i.e., the $L_1$-norm. This gives rise to the so called $L_1$-estimation problem:

$$\min_\theta \|y - M(\theta)\|_1 = \min_\theta \ \sum_{i=1}^{m} |y_i - M_i(\theta)| \tag{4.9}$$

Like the $L_2$-estimation problem, also the $L_1$-estimation problem can be interpreted statistically as a maximum-likelihood estimate. However, in the $L_1$-case, the measurement errors are assumed to follow a Laplace distribution instead of a Gaussian.

An interesting observation is that the optimal $L_1$-fit of a constant $\theta$ to a sample of different scalar values $y_1, \ldots, y_m$ just gives the median of this sample, i.e.

$$\arg\min_{\theta \in \mathbb{R}} \ \sum_{i=1}^{m} |y_i - \theta| \quad = \quad \text{median of } \{y_1, \ldots, y_m\}. \tag{4.10}$$

Remember that the same problem with the $L_2$-norm gave the average of $y_i$. Generally speaking, the median is less sensitive to outliers than the average, and a detailed analysis shows that the solution to general $L_1$-estimation problems is also less sensitive to a few outliers. Therefore, $L_1$-estimation is sometimes also called "robust" parameter estimation.

## 4.2 Bayesian Estimation and the Maximum A Posteriori Estimate

...

## 4.3 Recursive Linear Least Squares

...

# Chapter 5

# Dynamic Systems in a Nutshell

In this lecture, our major aim is to model and identify *dynamic systems*, i.e. processes that are evolving with time. These systems can be characterized by *states* $x$ and parameters $p$ that allow us to predict the future behavior of the system. If the state and the parameters are not known, we first need to *estimate* them based on the available measurement information. The estimation process is very often optimization-based, and thus, derivatives play a crucial role in this chapter. Often, a dynamic system can be controlled by a suitable choice of inputs that we denote as *controls* $u$ in this script, and the ultimate purpose of modelling and system identification is to be able to design and test control strategies.

As an example of a dynamic system, we might think of an electric train where the state $x$ consists of the current position and velocity, and where the control $u$ is the engine power that the train driver can choose at each moment. We might regard the motion of the train on a time interval $[t_{\text{init}}, t_{\text{fin}}]$, and the ultimate aim of controller design could be to minimize the consumption of electrical energy while arriving in time. Before we can decide on the control strategy, we need to know the current state of the train. Even more important, we should know important model parameters such as the mass of the train or how the motor efficiency changes with speed.

To determine the unknown system parameters, we typically perform experiments and record measurement data. In optimization-based state and parameter estimation, the objective function is typically the misfit between the actual measurements and the model predictions.

A typical property of a dynamic system is that knowledge of an *initial state* $x_{\text{init}}$ and a *control input trajectory* $u(t)$ for all $t \in [t_{\text{init}}, t_{\text{fin}}]$ allows one to determine the whole *state trajectory* $x(t)$ for $t \in [t_{\text{init}}, t_{\text{fin}}]$. As the motion of a train can very well be modelled by Newton's laws of motion, the usual description of this dynamic system is deterministic and in continuous time and with continuous states.

But dynamic systems and their mathematical models can come in many variants, and it is useful to properly define the names given commonly to different dynamic system classes, which we do in the next section. Afterwards, we will discuss two important classes, continuous time and discrete time systems, in more mathematical detail.

## 5.1 Dynamic System Classes

In this section, let us go, one by one, through the many dividing lines in the field of dynamic systems.

#### Continuous vs Discrete Time Systems

Any dynamic system evolves over time, but time can come in two variants: while the physical time is continuous and forms the natural setting for most technical and biological systems, other dynamic systems can best be modelled in discrete time, such as digitally controlled sampled-data systems, or games.

We call a system a *discrete time system* whenever the time in which the system evolves only takes values on a predefined time grid, usually assumed to be integers. If we have an interval of real numbers, like for the physical time, we call it a *continuous time system*. In this lecture, we usually denote the continuous time by the variable $t \in \mathbb{R}$ and write for example $x(t)$. In case of discrete time systems, we typically use the index variable $k \in \mathbb{N}$, and write $x_k$ or $x(k)$ for the state at time point $k$.

### Continuous vs Discrete State Spaces

Another crucial element of a dynamic system is its state $x$, which often lives in a continuous state space, like the position of the train, but can also be discrete, like the position of the figures on a chess game. We define the *state space* $\mathbb{X}$ to be the set of all values that the state vector $x$ may take. If $\mathbb{X}$ is a subset of a real vector space such as $\mathbb{R}^{n_x}$ or another differentiable manifold, we speak of a *continuous state space*. If $\mathbb{X}$ is a finite or a countable set, we speak of a *discrete state space*. If the state of a system is described by a combination of discrete and continuous variables we speak of a *hybrid state space*.

### Finite vs Infinite Dimensional State Spaces

The class of continuous state spaces can be further subdivided into the finite dimensional ones, whose state can be characterized by a finite set of real numbers, and the infinite dimensional ones, which have a state that lives in function spaces. The evolution of finite dimensional systems in continuous time is usually described by *ordinary differential equations (ODE)* or their generalizations, such as *differential algebraic equations (DAE)*.

Infinite dimensional systems are sometimes also called *distributed parameter systems*, and in the continuous time case, their behaviour is typically described by *partial differential equations (PDE)*. An example for a controlled infinite dimensional system is the evolution of the airflow and temperature distribution in a building that is controlled by an air-conditioning system. Systems with delay are another class of systems with infinite dimensional state space.

### Continuous vs Discrete Control Sets

We denote by $\mathbb{U}$ the set in which the controls $u$ live, and exactly as for the states, we can divide the possible control sets into *continuous control sets* and *discrete control sets*. A mixture of both is a *hybrid control set*. An example for a discrete control set is the set of gear choices for a car, or any switch that we can can choose to be either on or off, but nothing in between.

### Time-Variant vs Time-Invariant Systems

A system whose dynamics depend on time is called a *time-variant system*, while a dynamic system is called *time-invariant* if its evolution does not depend on the time and date when it is happening. As the laws of physics are time-invariant, most technical systems belong to the latter class, but for example the temperature evolution of a house with hot days and cold nights might best be described by a time-variant system model. While the class of time-variant systems trivially comprises all time-invariant systems, it is an important observation that also the other direction holds: each time-variant system can be modelled by a nonlinear time-invariant system if the state space is augmented by an extra state that takes account of the advancement of time, and which we might call the "clock state".

### Linear vs Nonlinear Systems

If the state trajectory of a system depends linearly on the initial value and the control inputs, it is called a *linear system*. If the dependence is affine, one should ideally speak of an *affine system*, but often the term linear is used here as well. In all other cases, we speak of a *nonlinear system*.

A particularly important class of linear systems are *linear time invariant (LTI)* systems. An LTI system can be completely characterized in at least three equivalent ways: first, by two matrices that are typically called $A$ and $B$; second, by its *step response function*; and third, by its *frequency response function*. A large part of the research in the control community is devoted to the study of LTI systems.

### Controlled vs Uncontrolled Dynamic Systems

While we are in this lecture mostly interested in *controlled dynamic systems*, i.e. systems that have a control input that we can choose, it is good to remember that there exist many systems that cannot be influenced at all, but that only evolve according to their intrinsic laws of motion. These *uncontrolled systems* have an empty control set, $\mathbb{U} = \emptyset$. If a dynamic system is both uncontrolled and time-invariant it is also called an *autonomous system*.

#### Stable vs Unstable Dynamic Systems

A dynamic system whose state trajectory remains bounded for bounded initial values and controls is called a *stable system*, and an *unstable system* otherwise. For autonomous systems, *stability* of the system around a fixed point can be defined rigorously: for any arbitrarily small neighborhood $\mathcal{N}$ around the fixed point there exists a region so that all trajectories that start in this region remain in $\mathcal{N}$. *Asymptotic stability* is stronger and additionally requires that all considered trajectories eventually converge to the fixed point. For autonomous LTI systems, stability can be computationally characterized by the eigenvalues of the system matrix.

#### Deterministic vs Stochastic Systems

If the evolution of a system can be predicted when its initial state and the control inputs are known, it is called a *deterministic system*. When its evolution involves some random behaviour, we call it a *stochastic system*.

The movements of assets on the stockmarket are an example for a stochastic system, whereas the motion of planets in the solar system can usually be assumed to be deterministic. An interesting special case of deterministic systems with continuous state space are *chaotic systems*. These systems are so sensitive to their initial values that even knowing these to arbitrarily high, but finite, precisions does not allow one to predict the complete future of the system: only the near future can be predicted. The partial differential equations used in weather forecast models have this property, and one well-known chaotic system of ODE, the *Lorenz attractor*, was inspired by these.

#### Open-Loop vs Closed-Loop Controlled Systems

When choosing the inputs of a controlled dynamic system, one first way is decide in advance, before the process starts, which control action we want to apply at which time instant. This is called *open-loop control* in the systems and control community, and has the important property that the control $u$ is a function of time only and does not depend on the current system state.

A second way to choose the controls incorporates our most recent knowledge about the system state which we might observe with the help of measurements. This knowledge allows us to apply feedback to the system by adapting the control action according to the measurements. In the systems and control community, this is called *closed-loop control*, but also the more intuitive term *feedback control* is used. It has the important property that the control action does depend on the current state or the latest measurements.

## 5.2 Continuous Time Systems

Most systems of interest in science and engineering are described in form of deterministic differential equations which live in continuous time. On the other hand, all numerical simulation methods have to discretize the time interval of interest in some form or the other and thus effectively generate discrete time systems. We will thus briefly sketch some relevant properties of continuous time systems in this section, and show how they can be transformed into discrete time systems. Later, we will mainly be concerned with discrete time systems, while we occasionally come back to the continuous time case.

#### Ordinary Differential Equations

A controlled dynamic system in continuous time can in the simplest case be described by an ordinary differential equation (ODE) on a time interval $[t_{\text{init}}, t_{\text{fin}}]$ by

$$\dot{x}(t) = f(x(t), u(t), t), \quad t \in [t_{\text{init}}, t_{\text{fin}}] \tag{5.1}$$

where $t \in \mathbb{R}$ is the time, $u(t) \in \mathbb{R}^{n_u}$ are the controls, and $x(t) \in \mathbb{R}^{n_x}$ is the state. The function $f$ is a map from states, controls, and time to the rate of change of the state, i.e. $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times [t_{\text{init}}, t_{\text{fin}}] \to \mathbb{R}^{n_x}$. Due to the explicit time dependence of the function $f$, this is a time-variant system.

We are first interested in the question if this differential equation has a solution if the initial value $x(t_{\text{init}})$ is fixed and also the controls $u(t)$ are fixed for all $t \in [t_{\text{init}}, t_{\text{fin}}]$. In this context, the dependence of $f$ on the fixed controls $u(t)$ is equivalent to a a further time-dependence of $f$, and we can redefine the ODE as $\dot{x} = \tilde{f}(x, t)$ with $\tilde{f}(x, t) := f(x, u(t), t)$. Thus, let us first leave away the dependence of $f$ on the controls, and just regard the time-dependent uncontrolled ODE:

$$\dot{x}(t) = f(x(t), t), \quad t \in [t_{\text{init}}, t_{\text{fin}}]. \tag{5.2}$$

### Initial Value Problems

An initial value problem (IVP) is given by (5.2) and the initial value constraint $x(t_{\text{init}}) = x_{\text{init}}$ with some fixed parameter $x_{\text{init}}$. Existence of a solution to an IVP is guaranteed under continuity of $f$ with respect to to $x$ and $t$ according to a theorem from 1886 that is due to Giuseppe Peano. But existence alone is of limited interest as the solutions might be non-unique.

**Example 3 (Non-Unique ODE Solution)** *The scalar ODE with $f(x) = \sqrt{|x(t)|}$ can stay for an undetermined duration in the point $x = 0$ before leaving it at an arbitrary time $t_0$. It then follows a trajectory $x(t) = (t - t_0)^2/4$ that can be easily shown to satisfy the ODE* (5.2)*. We note that the ODE function $f$ is continuous, and thus existence of the solution is guaranteed mathematically. However, at the origin, the derivative of $f$ approaches infinity. It turns out that this is the reason which causes the non-uniqueness of the solution.*

As we are only interested in systems with well-defined and deterministic solutions, we would like to formulate only ODE with unique solutions. Here helps the following theorem by Charles Émile Picard (1890) and Ernst Leonard Lindelöf (1894).

**Theorem 5 (Existence and Uniqueness of IVP)** *Regard the initial value problem* (5.2) *with $x(t_{\text{init}}) = x_{\text{init}}$, and assume that $f : \mathbb{R}^{n_x} \times [t_{\text{init}}, t_{\text{fin}}] \to \mathbb{R}^{n_x}$ is continuous with respect to $x$ and $t$. Furthermore, assume that $f$ is Lipschitz continuous with respect to $x$, i.e., that there exists a constant $L$ such that for all $x, y \in \mathbb{R}^{n_x}$ and all $t \in [t_{\text{init}}, t_{\text{fin}}]$*

$$\|f(x,t) - f(y,t)\| \leq L\|x - y\|. \tag{5.3}$$

*Then there exists a unique solution $x : [t_{\text{init}}, t_{\text{fin}}] \to \mathbb{R}^{n_x}$ of the IVP.*

Lipschitz continuity of $f$ with respect to $x$ is not easy to check. It is much easier to verify if a function is differentiable. It is therefore a helpful fact that every function $f$ that is differentiable with respect to $x$ is also locally Lipschitz continuous, and one can prove the following corollary to the Theorem of Picard-Lindelöf.

**Corollary 1 (Local Existence and Uniqueness)** *Regard the same initial value problem as in Theorem 5, but instead of global Lipschitz continuity, assume that $f$ is continuously differentiable with respect to $x$ for all $t \in [t_{\text{init}}, t_{\text{fin}}]$. Then there exists a possibly shortened, but non-empty interval $[t_{\text{init}}, t'_{\text{fin}}]$ with $t'_{\text{fin}} \in (t_{\text{init}}, t_{\text{fin}}]$ on which the IVP has a unique solution.*

Note that for nonlinear continuous time systems – in contrast to discrete time systems – it is very easily possible to obtain an "explosion", i.e., a solution that tends to infinity for finite times, even with innocently looking and smooth functions $f$.

**Example 4 (Explosion of an ODE)** *Regard the scalar example $f(x) = x^2$ with $t_{\text{init}} = 0$ and $x_{\text{init}} = 1$, and let us regard the interval $[t_{\text{init}}, t_{\text{fin}}]$ with $t_{\text{fin}} = 10$. The IVP has the explicit solution $x(t) = 1/(1 - t)$, which is only defined on the half open interval $[0, 1)$, because it tends to infinity for $t \to 1$. Thus, we need to choose some $t'_{\text{fin}} < 1$ in order to have a unique and finite solution to the IVP on the shortened interval $[t_{\text{init}}, t'_{\text{fin}}]$. The existence of this local solution is guaranteed by the above corollary. Note that the explosion in finite time is due to the fact that the function $f$ is not globally Lipschitz continuous, so Theorem 5 is not applicable.*

### Discontinuities with Respect to Time

It is important to note that the above theorem and corollary can be extended to the case that there are finitely many discontinuities of $f$ with respect to $t$. In this case the ODE solution can only be defined on each of the continuous time intervals separately, while the derivative of $x$ is not defined at the time points at which the discontinuities of $f$ occur, at least not in the strong sense. But the transition from one interval to the next can be determined by continuity of the state trajectory, i.e. we require that the end state of one continuous initial value problem is the starting value of the next one.

The fact that unique solutions still exist in the case of discontinuities is important because many state and parameter estimation problems are based on discontinuous control trajectories $u(t)$. Fortunately, this does not cause difficulties for existence and uniqueness of the IVPs.

### Linear Time Invariant (LTI) Systems

A special class of tremendous importance are the linear time invariant (LTI) systems. These are described by an ODE of the form

$$\dot{x} \quad = \quad Ax + Bu \tag{5.4}$$

with fixed matrices $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$. LTI systems are one of the principal interests in the field of automatic control and a vast literature exists on LTI systems. Note that the function $f(x, u) = Ax + Bu$ is Lipschitz continuous with respect to $x$ with Lipschitz constant $L = \|A\|$, so that the global solution to any initial value problem with a piecewise continuous control input can be guaranteed.

For system identification, we usually need to add output equations $y = Cx + Du$ to our model, where the outputs $y$ may be the only physically measurable quantities. In that context, it is important to remark that the states are not even unique, because different state space realizations of the same input-output behavior exist.

Many important notions such as *controllability* or *stabilizability*, and *observability* or *detectability*, and concepts such as the *impulse response* or *frequency response function* can be defined in terms of the matrices $A, B, C$ and $D$ alone. In particular, the *transfer function* $G(s)$ of an LTI system is the Laplace transform of the impulse response can be shown to be given by

$$G(s) = C(sI - A)^{-1}B + D.$$

The frequency response is given by the transfer function evaluated at values $s = j\omega$ where $j$ is the imaginary unit.

### Zero Order Hold and Solution Map

In the age of digital control, the inputs $u$ are often generated by a computer and implemented at the physical system as piecewise constant between two sampling instants. This is called *zero order hold*. The grid size is typically constant, say of fixed length $\Delta t > 0$, so that the sampling instants are given by $t_k = k \cdot \Delta t$. If our original model is a differentiable ODE model, but we have piecewise constant control inputs with fixed values $u(t) = u_k$ wtih $u_k \in \mathbb{R}^{n_u}$ on each interval $t \in [t_k, t_{k+1}]$, we might want to regard the transition from the state $x(t_k)$ to the state $x(t_{k+1})$ as a discrete time system. This is indeed possible, as the ODE solution exists and is unique on the interval $[t_k, t_{k+1}]$ for each initial value $x(t_k) = x_{\text{init}}$.

If the original ODE system is time-invariant, it is enough to regard one initial value problem with constant control $u(t) = u_{\text{const}}$

$$\dot{x}(t) = f(x(t), u_{\text{const}}), \quad t \in [0, \Delta t], \quad \text{with} \quad x(0) = x_{\text{init}}. \tag{5.5}$$

The unique solution $x : [0, \Delta t] \to \mathbb{R}^{n_x}$ to this problem is a function of both, the initial value $x_{\text{init}}$ and the control $u_{\text{const}}$, so we might denote the solution by

$$x(t; x_{\text{init}}, u_{\text{const}}), \quad \text{for} \quad t \in [0, \Delta t]. \tag{5.6}$$

This map from $(x_{\text{init}}, u_{\text{const}})$ to the state trajectory is called the *solution map*. The final value of this short trajectory piece, $x(\Delta t; x_{\text{init}}, u_{\text{const}})$, is of major interest, as it is the point where the next sampling interval starts. We might define the transition function $f_{\text{dis}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ by $f_{\text{dis}}(x_{\text{init}}, u_{\text{const}}) = x(\Delta t; x_{\text{init}}, u_{\text{const}})$. This function allows us to define a discrete time system that uniquely describes the evolution of the system state at the sampling instants $t_k$:

$$x(t_{k+1}) = f_{\text{dis}}(x(t_k), u_k). \tag{5.7}$$

### Solution Map of Linear Time Invariant Systems

Let us regard a simple and important example: for linear continuous time systems

$$\dot{x} = Ax + Bu$$

with initial value $x_{\text{init}}$ at $t_{\text{init}} = 0$, and constant control input $u_{\text{const}}$, the solution map $x(t; x_{\text{init}}, u_{\text{const}})$ is explicitly given as

$$x(t; x_{\text{init}}, u_{\text{const}}) = \exp(At)x_{\text{init}} + \int_0^t \exp(A(t - \tau))Bu_{\text{const}} \mathrm{d}\tau,$$

where $\exp(A)$ is the matrix exponential. It is interesting to note that this map is well defined for all times $t \in \mathbb{R}$, as linear systems cannot explode. The corresponding discrete time system with sampling time $\Delta t$ is again a linear time invariant system, and is given by

$$f_{\text{dis}}(x_k, u_k) = A_{\text{dis}} x_k + B_{\text{dis}} u_k \tag{5.8}$$

with

$$A_{\text{dis}} = \exp(A\Delta t) \quad \text{and} \quad B_{\text{dis}} = \int_0^{\Delta t} \exp(A(\Delta t - \tau)) B \mathrm{d}\tau. \tag{5.9}$$

One interesting observation is that the discrete time system matrix $A_{\text{dis}}$ resulting from the solution of an LTI system in continuous time is by construction an invertible matrix, with inverse $A_{\text{dis}}^{-1} = \exp(-A\Delta t)$. For systems with strongly decaying dynamics, however, the matrix $A_{\text{dis}}$ might have some very small eigenvalues and will thus be nearly singular.

### Sensitivities

In the context of estimation, derivatives of the dynamic system simulation are often needed. Following Theorem 5 and Corollary 1 we know that the solution map to the IVP (5.5) exists on an interval $[0, \Delta t]$ and is unique under mild conditions even for general nonlinear systems. But is it also differentiable with respect to the initial value and control input?

In order to discuss the issue of derivatives, which in the dynamic system context are often called *sensitivities*, let us first ask what happens if we call the solution map with different inputs. For small perturbations of the values $(x_{\text{init}}, u_{\text{const}})$, we still have a unique solution $x(t; x_{\text{init}}, u_{\text{const}})$ on the whole interval $t \in [0, \Delta t]$. Let us restrict ourselves to a neighborhood $\mathcal{N}$ of fixed values $(x_{\text{init}}, u_{\text{const}})$. For each fixed $t \in [0, \Delta t]$, we can now regard the well defined and unique solution map $x(t; \cdot) : \mathcal{N} \to \mathbb{R}^{n_x}$, $(x_{\text{init}}, u_{\text{const}}) \mapsto x(t; x_{\text{init}}, u_{\text{const}})$. A natural question to ask is if this map is differentiable. Fortunately, it is possible to show that if $f$ is $m$-times continuously differentiable with respect to both $x$ and $u$, then the solution map $x(t; \cdot)$, for each $t \in [0, \Delta t]$, is also $m$-times continuously differentiable with respect to $(x_{\text{init}}, u_{\text{const}})$.

In the general nonlinear case, the solution map $x(t; x_{\text{init}}, u_{\text{const}})$ can only be generated by a numerical simulation routine. The computation of derivatives of this numerically generated map is a delicate issue. The reason is that most numerical integration routines are adaptive, i.e., might choose to do different numbers of integration steps for different IVPs. This renders the numerical approximation of the map $x(t; x_{\text{init}}, u_{\text{const}})$ typically non-differentiable in the inputs $x_{\text{init}}, u_{\text{const}}$. Thus, multiple calls of a black-box integrator and application of finite differences might result in very wrong derivative approximations.

### Numerical Integration Methods

A numerical simulation routine that approximates the solution map is often called an *integrator*. A simple but very crude way to generate an approximation for $x(t; x_{\text{init}}, u_{\text{const}})$ for $t \in [0, \Delta t]$ is to perform a linear extrapolation based on the time derivative $\dot{x} = f(x, u)$ at the initial time point:

$$\tilde{x}(t; x_{\text{init}}, u_{\text{const}}) = x_{\text{init}} + t f(x_{\text{init}}, u_{\text{const}}), \quad t \in [0, \Delta t]. \tag{5.10}$$

This is called one *Euler integration step*. For very small $\Delta t$, this approximation becomes very good. In fact, the error $\tilde{x}(\Delta t; x_{\text{init}}, u_{\text{const}}) - x(\Delta t; x_{\text{init}}, u_{\text{const}})$ is of second order in $\Delta t$. This motivated Leonhard Euler to perform several steps of smaller size, and propose what is now called the *Euler integration method*. We subdivide the interval $[0, \Delta t]$ into $M$ subintervals each of length $h = \Delta t / M$, and perform $M$ such linear extrapolation steps consecutively, starting at $\tilde{x}_0 = x_{\text{init}}$:

$$\tilde{x}_{j+1} = \tilde{x}_j + h f(\tilde{x}_j, u_{\text{const}}), \quad j = 0, \ldots, M-1. \tag{5.11}$$

It can be proven that the Euler integration method is *stable*, i.e. that the propagation of local errors is bounded with a constant that is independent of the step size $h$. Therefore, the approximation becomes better and better when we decrease the step size $h$: since the *consistency* error in each step is of order $h^2$, and the total number of steps is of order $\Delta t / h$, the accumulated error in the final step is of order $h\Delta t$. As this is linear in the step size $h$, we say that the Euler method has the *order one*. Taking more steps is more accurate, but also needs more computation time. One measure for the computational effort of an integration method is the number of evaluations of $f$, which for the Euler method grows linearly with the desired accuracy.

In practice, the Euler integrator is rarely competitive, because other methods exist that deliver the desired accuracy levels at much lower computational cost. We discuss several numerical simulation methods later, but present here already one of the most widespread integrators, the *Runge-Kutta Method of Order Four*, which we will often abbreviate as *RK4*. One step of the RK4 method needs four evaluations of $f$ and stores the results in four intermediate quantities $k_i \in \mathbb{R}^{n_x}$, $i = 1, \ldots, 4$. Like the Euler integration method, the RK4 also generates a sequence of values $\tilde{x}_j$, $j = 0, \ldots, M$, with $\tilde{x}_0 = x_{\text{init}}$. At $\tilde{x}_j$, and using the constant control input $u_{\text{const}}$, one step of the RK4 method proceeds as follows:

$$\begin{aligned} k_1 &= f(\tilde{x}_j, u_{\text{const}}) && \text{(5.12a)} \\ k_2 &= f(\tilde{x}_j + \frac{h}{2} k_1, u_{\text{const}}) && \text{(5.12b)} \\ k_3 &= f(\tilde{x}_j + \frac{h}{2} k_2, u_{\text{const}}) && \text{(5.12c)} \\ k_4 &= f(\tilde{x}_j + h\, k_3, u_{\text{const}}) && \text{(5.12d)} \\ \tilde{x}_{j+1} &= \tilde{x}_j + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) && \text{(5.12e)} \end{aligned}$$

One step of RK4 is thus as expensive as four steps of the Euler method. But it can be shown that the accuracy of the final approximation $\tilde{x}_M$ is of order $h^4 \Delta t$. In practice, this means that the RK4 method usually needs tremendously fewer function evaluations than the Euler method to obtain the same accuracy level.

From here on, and throughout the major part of the lecture, we will leave the field of continuous time systems, and directly assume that we control a discrete time system $x_{k+1} = f_{\text{dis}}(x_k, u_k)$. Let us keep in mind, however, that the transition map $f_{\text{dis}}(x_k, u_k)$ is usually not given as an explicit expression but can instead be a relatively involved computer code with several intermediate quantities. In the exercises of this lecture, we will usually discretize the occuring ODE systems by using only one Euler or RK4 step per control interval, i.e. use $M = 1$ and $h = \Delta t$. The RK4 step often gives already a sufficient approximation at relatively low cost.

## 5.3 Discrete Time Systems

Let us now discuss in more detail the discrete time systems that are at the basis of the control problems in the first part of this lecture. In the general time-variant case, these systems are characterized by the dynamics

$$x_{k+1} = f_k(x_k, u_k), \quad k = 0, 1, \ldots, N-1 \tag{5.13}$$

on a time horizon of length $N$, with $N$ control input vectors $u_0, \ldots, u_{N-1} \in \mathbb{R}^{n_u}$ and $(N+1)$ state vectors $x_0, \ldots, x_N \in \mathbb{R}^{n_x}$.

If we know the initial state $x_0$ and the controls $u_0, \ldots, u_{N-1}$ we could recursively call the functions $f_k$ in order to obtain all other states, $x_1, \ldots, x_N$. We call this a *forward simulation* of the system dynamics.

**Definition 9 (Forward simulation)** *The* forward simulation *is the map*

$$\begin{aligned} f_{\text{sim}} : \quad \mathbb{R}^{n_x + N n_u} &\to \mathbb{R}^{(N+1) n_x} \\ (x_0; u_0, u_1, \ldots, u_{N-1}) &\mapsto (x_0, x_1, x_2, \ldots, x_N) \end{aligned} \tag{5.14}$$

*that is defined by solving Equation* (5.13) *recursively for all* $k = 0, 1, \ldots, N-1$.

The inputs of the forward simulation routine are the initial value $x_0$ and the controls $u_k$ for $k = 0, \ldots, N-1$. In many practical problems we can only choose the controls while the initial value is fixed. Though this is a very natural assumption, it is not the only possible one. In optimization, we might have very different requirements: We might, for example, have a free initial value that we want to choose in an optimal way. Or we might have both a fixed initial state and a fixed terminal state that we want to reach. We might also look for periodic sequences with $x_0 = x_N$, but do not know $x_0$ beforehand. All these desires on the initial and the terminal state can be expressed by suitable constraints. For the purpose of this manuscript it is important to note that the fundamental equation that is characterizing a dynamic optimization problem are the system dynamics stated in Equation (5.13), but no initial value constraint, which is optional.

### Linear Time Invariant (LTI) Systems

As discussed already for the continuous time case, linear time invariant (LTI) systems are not only one of the simplest possible dynamic system classes, but also have a rich and beautiful history. In the discrete time case, they are determined by the system equation

$$x_{k+1} \quad = \quad Ax_k + Bu_k, \quad k = 0, 1, \ldots, N-1. \tag{5.15}$$

with fixed matrices $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$. An LTI system is asymptotically stable if all eigenvalues of the matrix $A$ are strictly inside the unit disc of the complex plane, i.e. have a modulus smaller than one. It is easy to show that the forward simulation map for an LTI system on a horizon with length $N$ is given by

$$f_{\text{sim}}(x_0; u_0, \ldots, u_{N-1}) = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_0 \\ Ax_0 + Bu_0 \\ A^2x_0 + ABu_0 + Bu_1 \\ \vdots \\ A^Nx_0 + \sum_{k=0}^{N-1} A^{N-1-k}Bu_k \end{bmatrix}$$

In order to check controllability, due to linearity, one might ask the question if after $N$ steps any terminal state $x_N$ can be reached from $x_0 = 0$ by a suitable choice of control inputs. Because of

$$x_N = \underbrace{\begin{bmatrix} A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{=\mathcal{C}_N} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

this is possible if and only if the matrix $\mathcal{C}_N \in \mathbb{R}^{n_x \times Nn_u}$ has the rank $n_x$. Increasing $N$ can only increase the rank, but one can show that the maximum possible rank is already reached for $N = n_x$, so it is enough to check if the so called *controllability matrix* $\mathcal{C}_{n_x}$ has the rank $n_x$.

### Eigenvalues and Eigenvectors of LTI Systems

Every square matrix $A \in \mathbb{R}^{n_x \times n_x}$ can be brought into the Jordan canonical form $A = QJQ^{-1}$ with non-singular $Q \in \mathbb{C}^{n_x \times n_x}$ and $J$ block diagonal, consisting of $m$-Jordan blocks $J_i$. Thus, it holds that

$$J = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_m \end{bmatrix} \quad \text{with} \quad J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & 1 & \\ & & \ddots & \ddots \\ & & & \lambda_i \end{bmatrix}.$$

Many of the Jordan blocks might just have size one, i.e. $J_i = [\lambda_i]$. To better understand the uncontrolled system evolution with dynamics $x_{k+1} = Ax_k$ and initial condition $x_0 = x_{\text{init}}$, one can regard the solution map $x_N = A^N x_0$ in the eigenbasis, which yields the expression

$$x_N = Q\, J^N\, (Q^{-1}x_0)$$

First, it is seen that all Jordan blocks evolve independently, after the initial condition is represented in the eigenbasis. Second, a simple Jordan block $J_i$ will just result in the corresponding component being multiplied by a factor $\lambda_i^N$. Third, for nontrivial Jordan blocks, one obtains more complex expressions with $N$ upper diagonals of the form

$$J_i^N = \begin{bmatrix} \lambda_i^N & N\lambda_i^{N-1} & \cdots & 1 & \\ & \lambda_i^N & N\lambda_i^{N-1} & & \ddots \\ & & & \ddots & \\ & & & & \lambda_i^N \end{bmatrix}.$$

If one eigenvalue has a larger modulus $|\lambda_i|$ than all others, the Jordan block $J_i^N$ will grow faster (or shrink slower) than the others for increasing $N$. The result is that the corresponding eigenvector(s) will dominate the final state $x_N$ for large $N$, while all others "die out". Here, the second largest eigenvalues will result in the most slowly decaying components, and their corresponding eigenvectors will keep a visible contribution in $x_N$ the longest.

Interestingly, complex eigenvalues as well as eigenvectors appear in complex conjugate pairs. If an eigenvalue $\lambda_i$ is complex, the (real part of) the corresponding eigenvector will perform oscillatory motion. To understand the behaviour of complex eigenvectors, let us regard a complex conjugate pair of simple eigenvalues $\lambda_i$ and $\lambda_j = \bar{\lambda}_i$, and their eigenvectors $v_i, v_j \in \mathbb{C}^{n_x}$, i.e. $Av_i = \lambda_i v_i$ and $Av_j = \bar{\lambda}_i v_j$. It is easy to see that, because $A$ is real, $v_j = \bar{v}_i$ is a possible choice for the eigenvector corresponding to $\bar{\lambda}_i$. Then holds that $\mathrm{Re}\{v_i\} = \frac{1}{2}(v_i + v_j)$. Therefore,

$$A^N \,\mathrm{Re}\{v_i\} = \frac{1}{2}(\lambda_i^N v_i + \lambda_j^N v_j) = \frac{1}{2}(\lambda_i^N v_i + \bar{\lambda}_i^N \bar{v}_i) = \mathrm{Re}\{\lambda_i^N v_i\}.$$

If we represent $\lambda_i$ as $\lambda_i = re^{\phi i}$ (where the $i$ in the exponent is the imaginary unit while the other $i$ remains just an integer index), then $\lambda_i^N = r^N e^{N\phi i}$. If $\phi$ is a fraction of $2\pi$, there is an $N$ such that $N\phi = 2\pi$, and after $N$ iterations we will obtain the same real part as in the original eigenvector, but multiplied with $r^N$. We can conclude that the real part of the eigenvector to a complex eigenvalue $re^{\phi i}$ performs a form of damped or growing oscillatory motion with period duration $N = 2\pi/\phi$ and growth constant $r$.

#### Affine Systems and Linearizations along Trajectories

An important generalization of linear systems are affine time-varying systems of the form

$$x_{k+1} \quad = \quad A_k x_k + B_k u_k + c_k, \quad k = 0, 1, \ldots, N-1. \tag{5.16}$$

These often appear as linearizations of nonlinear dynamic systems along a given reference trajectory. To see this, let us regard a nonlinear dynamic system and some given reference trajectory values $\bar{x}_0, \ldots, \bar{x}_{N-1}$ as well as $\bar{u}_0, \ldots, \bar{u}_{N-1}$. Then the Taylor expansion of each function $f_k$ at the reference value $(\bar{x}_k, \bar{u}_k)$ is given by

$$(x_{k+1} - \bar{x}_{k+1}) \approx \frac{\partial f_k}{\partial x}(\bar{x}_k, \bar{u}_k)(x_k - \bar{x}_k) + \frac{\partial f_k}{\partial u}(\bar{x}_k, \bar{u}_k)(u_k - \bar{u}_k) + (f_k(\bar{x}_k, \bar{u}_k) - \bar{x}_{k+1})$$

thus resulting in affine time-varying dynamics of the form (5.16). Note that even for a time-invariant nonlinear system the linearized dynamics becomes time-variant due to the different linearization points on the reference trajectory.

It is an important fact that the forward simulation map of an affine system (5.16) is again an affine function of the initial value and the controls. More specifically, this affine map is for any $N \in \mathbb{N}$ given by:

$$x_N = (A_{N-1} \cdots A_0)\, x_0 + \sum_{k=0}^{N-1} \left(\Pi_{j=k+1}^{N-1} A_j\right)(B_k u_k + c_k)\,.$$

## 5.4 Input Output Models

...

# Bibliography

[Lju99] L. Ljung. *System identification: Theory for the User*. Prentice Hall, Upper Saddle River, N.J., 1999.

[Sch13] J. Schoukens. System identification, April 2013. Lecture Manuscript.