Modelling and System Identification

WS 2024 - Course Organization

Team Introduction

Lecture Organization

General

- Lecture
  - Monday, 8:15 - 9:55, HS 026
  - Wednesday, 8:15 - 9:55, HS 036
- Exercises
  - Pen & Paper
  - Coding
  - Teams of up to 3 people
- Exercise Sessions with Tutors
  - Group 1: Wednesday, 14:15 - 15:45, SR 00-031, Building 051 <- currently: too many people
  - Group 2: Thursday, 10:15 - 11:45, SR 01-009/13, Building 101
  - Group 3: Thursday, 14:15 - 15:45, SR 03-026, Building 051

Exercises - Points

- 9 graded exercises @ 10 points
- 3 microexams @ 10 points
- 3 blocks
- Studienleistung: at least **20 points in all 3 blocks**
- Optional exercise 0, bonus exercise 10

Exercise 1 Exercise 2 Exercise 3 Microexam 1
Exercise 4 Exercise 5 Exercise 6 Microexam 2
Exercise 7 Exercise 8 Exercise 9 (Exercise 10) Microexam 3

Exercises - Pen & Paper

- Handwritten solution to exercise sheet, or LateX if you want
- Hand-in: **Monday, before the lecture (<8:15)**
- Or: Box in Front of Prof. Diehl's Office (<8:15)
- Put names of team members and matriculation number on sheets!
- Returned to you in the exercise sessions
- Solution in exercise session

Exercises for Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2022-2023

Exercise 1: Resistance Estimation Example
(to be returned before October 31st, 8:00)

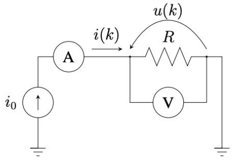
Prof. Dr. Moritz Diehl, Katrin Baumgärtner, Jakob Harzer, Yizhen Wang, Rashmi Dabir

In this exercise you investigate some important facts from statistics in numerical experiments.

Pen-and-paper exercises can be uploaded on the Ilias course page as a pdf or handed in during the lecture.

Exercise Tasks

1. We consider the following experimental setup:



Imagine you are sitting in a class of 200 electrical engineering students and you want to estimate the value of R using Ohm's law. Since the value of the current i_0 flowing through the resistor is not known exactly, an ammeter is used to measure the current $i(k)$ and a voltmeter to measure $u(k)$. Every student is taking 1000 measurements. The measurement number is represented by k. We assume that the measurements are noisy:

$$i(k) = i_0 + n_i(k) \qquad \text{and} \qquad u(k) = u_0 + n_u(k)$$

where $u_0 = 10\,\text{V}$ is the true values of the voltage across the resistor, $i_0 = 5\,\text{A}$ is the true value of the current flowing through the resistor and $n_i(k)$ and $n_u(k)$ are the values of the noise.
Please consider the data-set with all measurements of all students provided on the course website.

Let us now investigate the behaviour of the three different estimators which were introduced in the lecture:

$$\hat{R}_{\text{SA}}(N) = \frac{1}{N}\sum_{k=1}^{N}\frac{u(k)}{i(k)} \qquad \hat{R}_{\text{LS}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N}u(k)i(k)}{\frac{1}{N}\sum_{k=1}^{n}i(k)^2} \qquad \hat{R}_{\text{EV}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N}u(k)}{\frac{1}{N}\sum_{k=1}^{N}i(k)}$$

We will write code to simulate the behavior of these estimators. For each of the three estimators, carry out the following tasks.

(a) CODE: Compute the result of the function $\hat{R}_*(N)$, for $N = 1, \ldots, N_{\text{max}}$ using your personal measurements (student 1 or experiment 1). Do this for each estimator ($*$ can be either SA, LS or EV). Plot the three curves in one plot.
PAPER: Do the estimators converge for $N \to \infty$? (5 points)

Exercises - Code

- Python Scripts
- Use Numpy, Scipy
- Tutorial
- Github Classrooms to distribute, collect and grade exercises

Exercises for Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2022-2023

Exercise 1: Resistance Estimation Example
(to be returned before October 31st, 8:00)

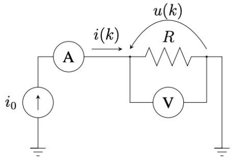
Prof. Dr. Moritz Diehl, Katrin Baumgärtner, Jakob Harzer, Yizhen Wang, Rashmi Dabir

In this exercise you investigate some important facts from statistics in numerical experiments.

Pen-and-paper exercises can be uploaded on the Ilias course page as a pdf or handed in during the lecture.

Exercise Tasks

1. We consider the following experimental setup:



Imagine you are sitting in a class of 200 electrical engineering students and you want to estimate the value of R using Ohm's law. Since the value of the current i_0 flowing through the resistor is not known exactly, an ammeter is used to measure the current $i(k)$ and a voltmeter to measure $u(k)$. Every student is taking 1000 measurements. The measurement number is represented by k. We assume that the measurements are noisy:

$$i(k) = i_0 + n_i(k) \quad \text{and} \quad u(k) = u_0 + n_u(k)$$

where $u_0 = 10\,\text{V}$ is the true values of the voltage across the resistor, $i_0 = 5\,\text{A}$ is the true value of the current flowing through the resistor and $n_i(k)$ and $n_u(k)$ are the values of the noise.
Please consider the data-set with all measurements of all students provided on the course website.

Let us now investigate the behaviour of the three different estimators which were introduced in the lecture:

$$\hat{R}_{\text{SA}}(N) = \frac{1}{N}\sum_{k=1}^{N}\frac{u(k)}{i(k)} \qquad \hat{R}_{\text{LS}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N}u(k)i(k)}{\frac{1}{N}\sum_{k=1}^{n}i(k)^2} \qquad \hat{R}_{\text{EV}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N}u(k)}{\frac{1}{N}\sum_{k=1}^{N}i(k)}$$

We will write code to simulate the behavior of these estimators. For each of the three estimators, carry out the following tasks.

(a) CODE: Compute the result of the function $\hat{R}_*(N)$, for $N = 1, \dots, N_{\max}$ using your personal measurements (student 1 or experiment 1). Do this for each estimator ($*$ can be either SA, LS or EV). Plot the three curves in one plot.
PAPER: Do the estimators converge for $N \to \infty$? (5 points)

Exercises - Code

- Each exercise contains:
  - `dataset.npz`
  - **`taskXX.py`**
  - `test_taskXX.py`
  - `refSolution.npz`
- Run tests with pytest
- 1 point per successful test

Exercises for Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2022-2023

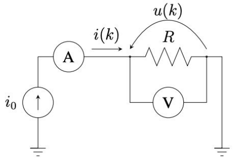
Exercise 1: Resistance Estimation Example
(to be returned before October 31st, 8:00)

Prof. Dr. Moritz Diehl, Katrin Baumgärtner, Jakob Harzer, Yizhen Wang, Rashmi Dabir

In this exercise you investigate some important facts from statistics in numerical experiments.
Pen-and-paper exercises can be uploaded on the Ilias course page as a pdf or handed in during the lecture.

Exercise Tasks

1. We consider the following experimental setup:



Imagine you are sitting in a class of 200 electrical engineering students and you want to estimate the value of R using Ohm's law. Since the value of the current i_0 flowing through the resistor is not known exactly, an ammeter is used to measure the current $i(k)$ and a voltmeter to measure $u(k)$. Every student is taking 1000 measurements. The measurement number is represented by k. We assume that the measurements are noisy:

$$i(k) = i_0 + n_i(k) \qquad \text{and} \qquad u(k) = u_0 + n_u(k)$$

where $u_0 = 10\,\text{V}$ is the true values of the voltage across the resistor, $i_0 = 5\,\text{A}$ is the true value of the current flowing through the resistor and $n_i(k)$ and $n_u(k)$ are the values of the noise.
Please consider the data-set with all measurements of all students provided on the course website.

Let us now investigate the behaviour of the three different estimators which were introduced in the lecture:

$$\hat{R}_{\text{SA}}(N) = \frac{1}{N}\sum_{k=1}^{N}\frac{u(k)}{i(k)} \qquad \hat{R}_{\text{LS}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N} u(k)i(k)}{\frac{1}{N}\sum_{k=1}^{n} i(k)^2} \qquad \hat{R}_{\text{EV}}(N) = \frac{\frac{1}{N}\sum_{k=1}^{N} u(k)}{\frac{1}{N}\sum_{k=1}^{N} i(k)}$$

We will write code to simulate the behavior of these estimators. For each of the three estimators, carry out the following tasks.

(a) CODE: Compute the result of the function $\hat{R}_*(N)$, for $N = 1, \ldots, N_{\max}$ using your personal measurements (student 1 or experiment 1). Do this for each estimator ($*$ can be either SA, LS or EV). Plot the three curves in one plot.
PAPER: Do the estimators converge for $N \to \infty$? (5 points)

Exercises - Github Classrooms

- register your (mock) github account and your matriculation number with us
- To accept an exercise with your team, click on the exercise link on the website

What is your immatriculation number? (It should look like `1234567`)

Enter your answer

What is your GitHub name? (It should look like `@someName`)

If you don't want to connect your real GitHub account, you can create a mock GitHub account for the purpose of this course.

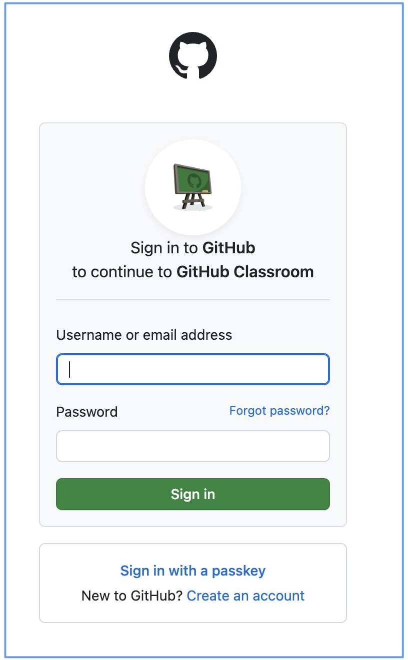
Enter your answer

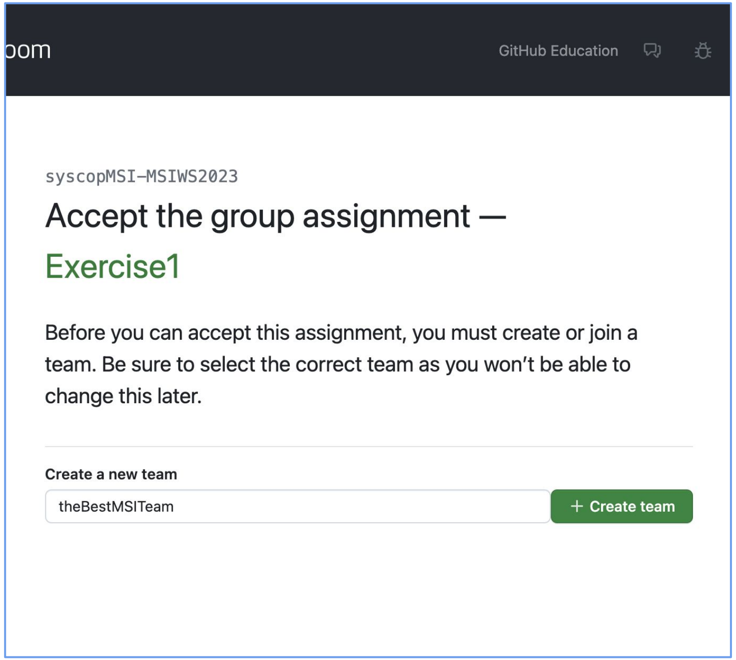
Submit

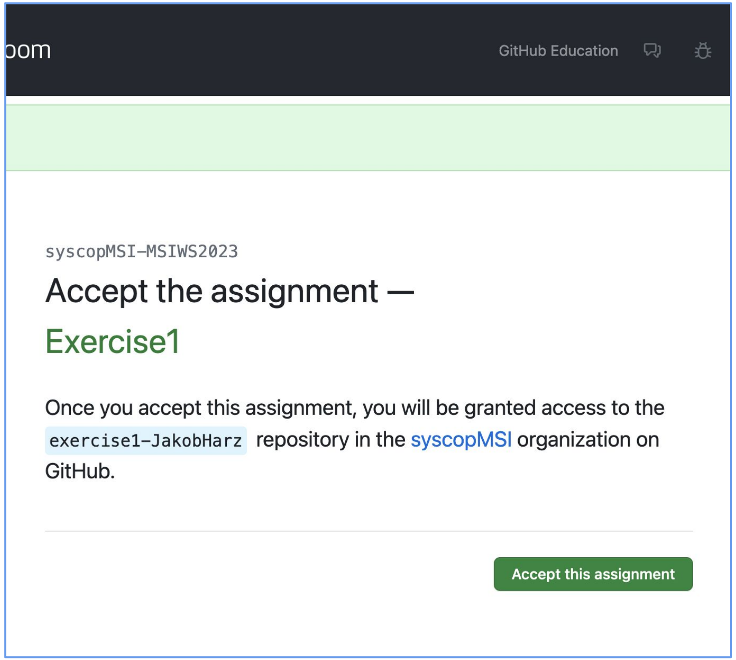
Exercises Sheets

Sheet	GH Classroom Link	Deadline
Sheet 0: Intro (optional)	Click to Accept Exercise 0	October 23
Sheet 1: Resistance Estimation Example	Click to Accept Exercise 1	October 30
Sheet 2: Statistics + Parameter Estimation		November 6
Sheet 3: Optimality Conditions and Linear Least Squares		November 13
Sheet 4: Weighted Linear Least-Squares		November 20
Sheet 5: Ill-Posed Linear Least-Squares & Regularization		December 4
Sheet 6: Maximum Likelihood and MAP Estimation		December 11
Sheet 7: Recursive Least Squares		January 8
Sheet 8: Nonlinear Least Squares		January 15
Sheet 9: Kalman Filter		January 29

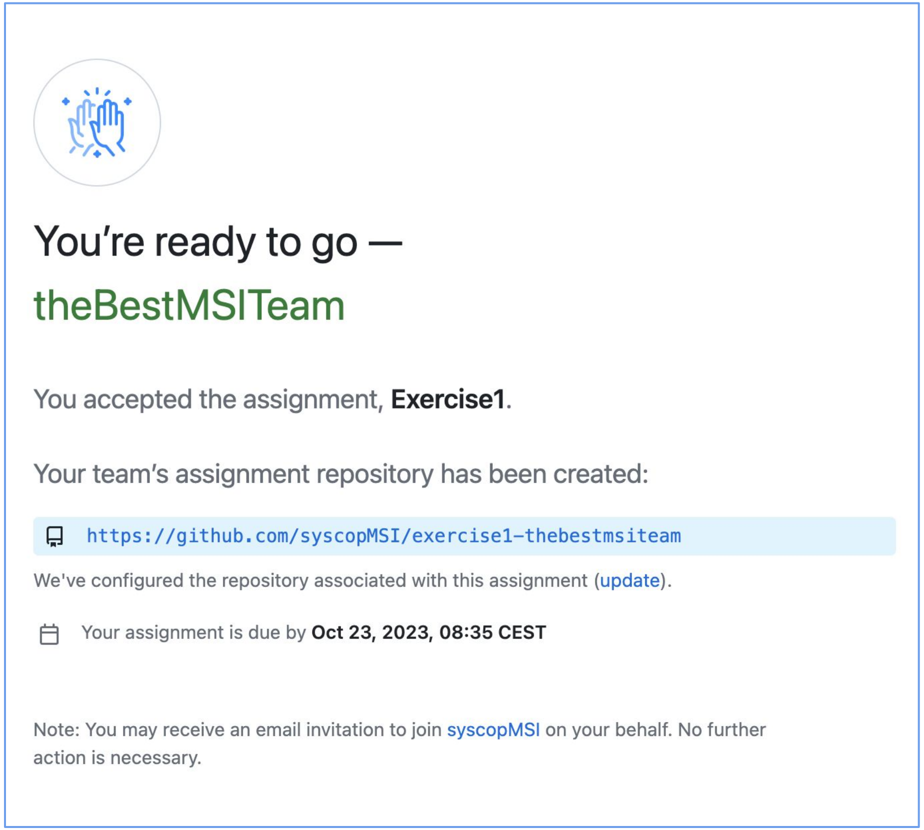
Exercises - Github Classrooms

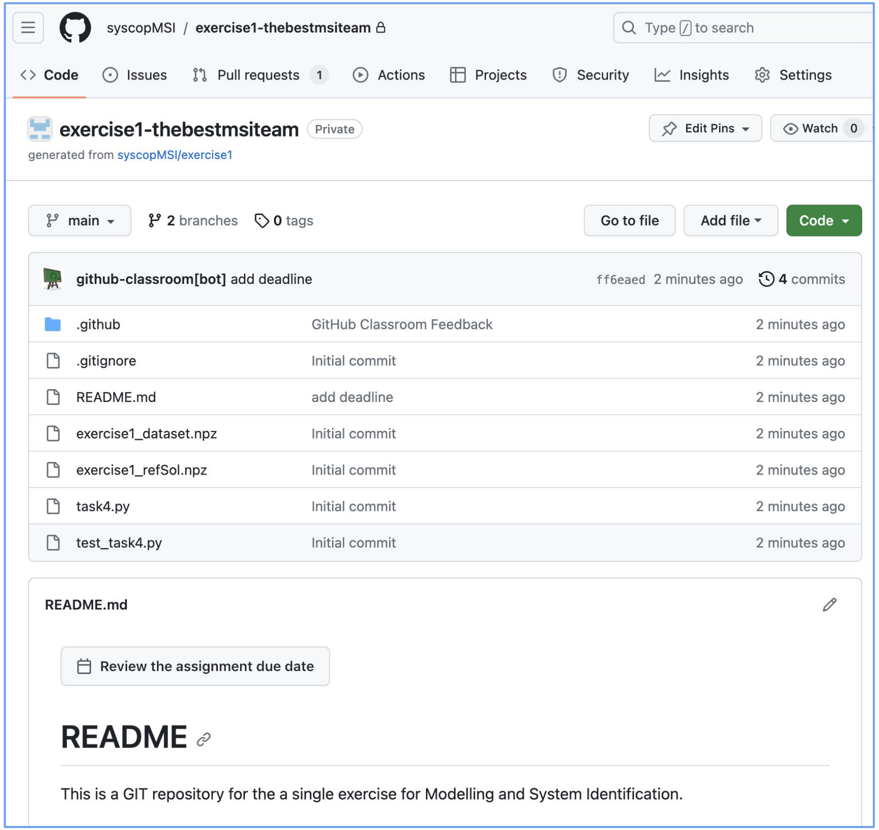






Exercises - Github Classrooms





Exercises - Github Classrooms

- clone the repository
- make your changes
- run the tests locally
- commit & push

```
[~/Google Drive/My Drive/Uni/Teaching/MSI/WS2023/Presentation]$ git clone https://github.com/syscopMSI/exercise1-thebestmsiteam.git
Cloning into 'exercise1-thebestmsiteam'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 16 (delta 3), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (16/16), 9.98 MiB | 4.27 MiB/s, done.
Resolving deltas: 100% (3/3), done.
[~/Google Drive/My Drive/Uni/Teaching/MSI/WS2023/Presentation]$
```

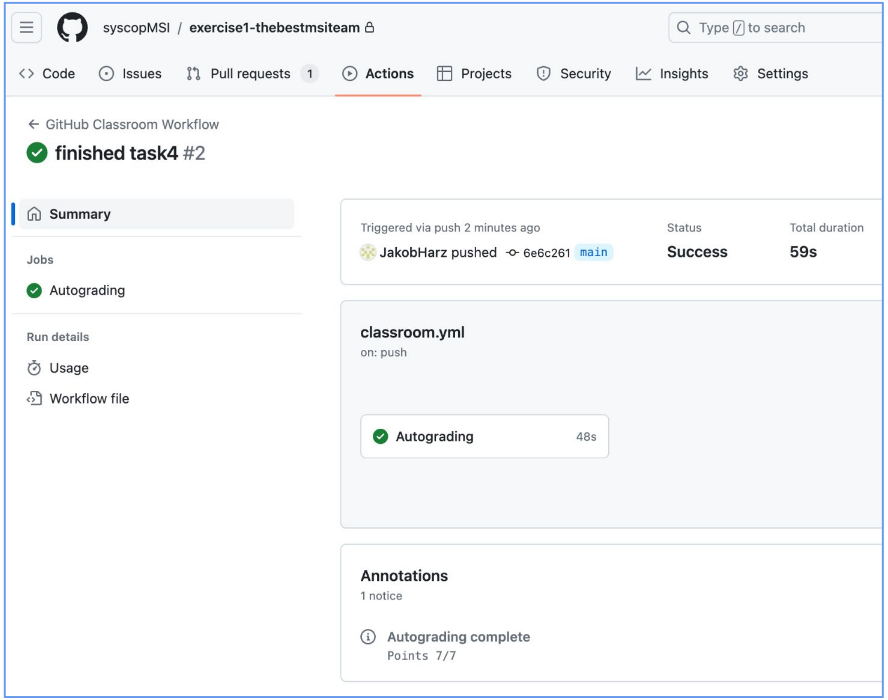
you work on the exercise

```
[main][~/Google Drive/My Drive/Uni/Teaching/MSI/WS2023/Presentation/exercise1-thebestmsiteam]$ pytest
============================================================ test session starts ============================================================
platform darwin -- Python 3.8.16, pytest-7.2.1, pluggy-1.0.0
rootdir: /Users/jakobharzer/MyDrive/Uni/Teaching/MSI/WS2023/Presentation/exercise1-thebestmsiteam
plugins: anyio-3.6.2
collected 7 items

test_task4.py .......                                                                                                                [100%]

============================================================= 7 passed in 1.38s =============================================================
```

```
*[main][~/Google Drive/My Drive/Uni/Teaching/MSI/WS2023/Presentation/exercise1-thebestmsiteam]$ git add task4.py
*[main][~/Google Drive/My Drive/Uni/Teaching/MSI/WS2023/Presentation/exercise1-thebestmsiteam]$ git commit -m "finished task4"
[main a7cd2ed] finished task4
 1 file changed, 65 insertions(+), 87 deletions(-)
[main][~/Google Drive/My Drive/Uni/Teaching/MSI/WS2023/Presentation/exercise1-thebestmsiteam]$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.17 KiB | 1.17 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/syscopMSI/exercise1-thebestmsiteam.git
   ff6eaed..a7cd2ed  main -> main
```

Exercises - Github Classrooms

- for grading, the test are rerun automatically on GH (with slightly different data)



Polls

Tutorials

Python

2D Numpy in Python

Welcome! This notebook will teach you about using Numpy in the Python Programming Language. By the end of this lab, you'll know what Numpy is and the Numpy operations.

Table of Contents

- Create a 2D Numpy Array
- Accessing different elements of a Numpy Array
- Basic Operations

Estimated time needed: **20 min**

Create a 2D Numpy Array

```python
# Import the libraries

import numpy as np
import matplotlib.pyplot as plt
```

Python

Consider the list a, the list contains three nested lists **each of equal size**.

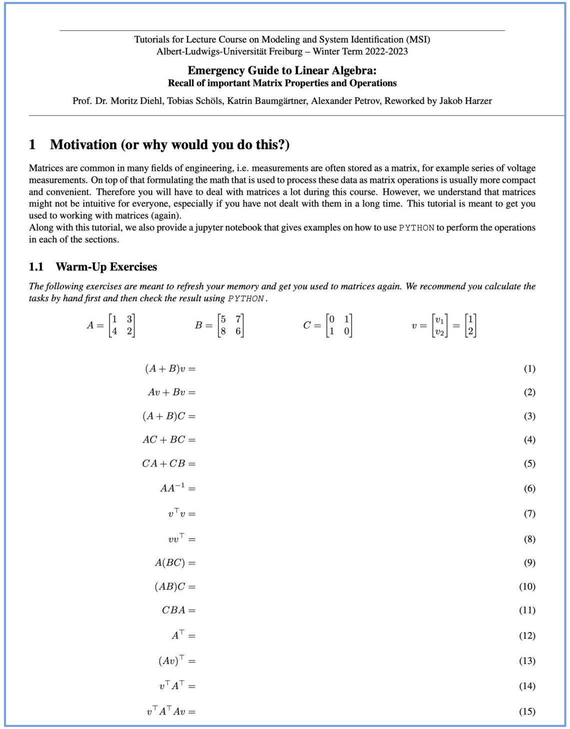
```python
# Create a list

a = [[11, 12, 13], [21, 22, 23], [31, 32, 33]]
a
```

Python

We can cast the list to a Numpy Array as follow

```python
# Convert list to Numpy Array
# Every element is the same type

A = np.array(a)
A
```

Linear Algebra



Tutorials for Lecture Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2022-2023

Emergency Guide to Linear Algebra:
Recall of important Matrix Properties and Operations

Prof. Dr. Moritz Diehl, Tobias Schöls, Katrin Baumgärtner, Alexander Petrov, Reworked by Jakob Harzer

1 Motivation (or why would you do this?)

Matrices are common in many fields of engineering, i.e. measurements are often stored as a matrix, for example series of voltage measurements. On top of that formulating the math that is used to process these data as matrix operations is usually more compact and convenient. Therefore you will have to deal with matrices a lot during this course. However, we understand that matrices might not be intuitive for everyone, especially if you have not dealt with them in a long time. This tutorial is meant to get you used to working with matrices (again).
Along with this tutorial, we also provide a jupyter notebook that gives examples on how to use PYTHON to perform the operations in each of the sections.

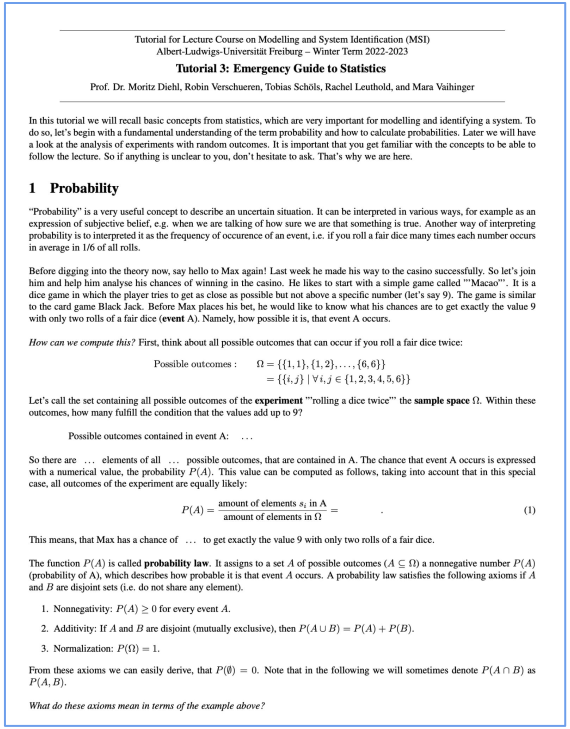
1.1 Warm-Up Exercises

The following exercises are meant to refresh your memory and get you used to matrices again. We recommend you calculate the tasks by hand first and then check the result using PYTHON.

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 5 & 7 \\ 8 & 6 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$(A+B)v = \tag{1}$$
$$Av + Bv = \tag{2}$$
$$(A+B)C = \tag{3}$$
$$AC + BC = \tag{4}$$
$$CA + CB = \tag{5}$$
$$AA^{-1} = \tag{6}$$
$$v^\top v = \tag{7}$$
$$vv^\top = \tag{8}$$
$$A(BC) = \tag{9}$$
$$(AB)C = \tag{10}$$
$$CBA = \tag{11}$$
$$A^\top = \tag{12}$$
$$(Av)^\top = \tag{13}$$
$$v^\top A^\top = \tag{14}$$
$$v^\top A^\top A v = \tag{15}$$

Statistics



Tutorial for Lecture Course on Modelling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2022-2023

Tutorial 3: Emergency Guide to Statistics

Prof. Dr. Moritz Diehl, Robin Verschueren, Tobias Schöls, Rachel Leuthold, and Mara Vaihinger

In this tutorial we will recall basic concepts from statistics, which are very important for modelling and identifying a system. To do so, let's begin with a fundamental understanding of the term probability and how to calculate probabilities. Later we will have a look at the analysis of experiments with random outcomes. It is important that you get familiar with the concepts to be able to follow the lecture. So if anything is unclear to you, don't hesitate to ask. That's why we are here.

1 Probability

"Probability" is a very useful concept to describe an uncertain situation. It can be interpreted in various ways, for example as an expression of subjective belief, e.g. when we are talking of how sure we are that something is true. Another way of interpreting probability is to interpreted it as the frequency of occurence of an event, i.e. if you roll a fair dice many times each number occurs in average in 1/6 of all rolls.

Before digging into the theory now, say hello to Max again! Last week he made his way to the casino successfully. So let's join him and help him analyse his chances of winning in the casino. He likes to start with a simple game called "'Macao"'. It is a dice game in which the player tries to get as close as possible but not above a specific number (let's say 9). The game is similar to the card game Black Jack. Before Max places his bet, he would like to know what his chances are to get exactly the value 9 with only two rolls of a fair dice (**event** A). Namely, how possible it is, that event A occurs.

How can we compute this? First, think about all possible outcomes that can occur if you roll a fair dice twice:

$$\text{Possible outcomes}: \quad \Omega = \{\{1,1\},\{1,2\},\ldots,\{6,6\}\} = \{\{i,j\} \mid \forall i,j \in \{1,2,3,4,5,6\}\}$$

Let's call the set containing all possible outcomes of the **experiment** "'rolling a dice twice"' the **sample space** Ω. Within these outcomes, how many fulfill the condition that the values add up to 9?

Possible outcomes contained in event A: …

So there are … elements of all … possible outcomes, that are contained in A. The chance that event A occurs is expressed with a numerical value, the probability $P(A)$. This value can be computed as follows, taking into account that in this special case, all outcomes of the experiment are equally likely:

$$P(A) = \frac{\text{amount of elements } s_i \text{ in A}}{\text{amount of elements in } \Omega} = \quad . \tag{1}$$

This means, that Max has a chance of … to get exactly the value 9 with only two rolls of a fair dice.

The function $P(A)$ is called **probability law**. It assigns to a set A of possible outcomes ($A \subseteq \Omega$) a nonnegative number $P(A)$ (probability of A), which describes how probable it is that event A occurs. A probability law satisfies the following axioms if A and B are disjoint sets (i.e. do not share any element).

1. Nonnegativity: $P(A) \geq 0$ for every event A.
2. Additivity: If A and B are disjoint (mutually exclusive), then $P(A \cup B) = P(A) + P(B)$.
3. Normalization: $P(\Omega) = 1$.

From these axioms we can easily derive, that $P(\emptyset) = 0$. Note that in the following we will sometimes denote $P(A \cap B)$ as $P(A, B)$.

What do these axioms mean in terms of the example above?

The next few weeks …

Week	Lecture	Exercise	Tutorial Session
1	Mon: Lecture & Orga Wed: Lecture	Exercise 0 (optional)	**Python & GIT Tutorial,** Exercise 0 hints
2	**Mon: LA Tutorial** Wed: Lecture	Exercise 1	Exercise 0 solution
3	**Mon: Statistics Tutorial** Wed: Lecture	Exercise 2	Exercise 1 Solution
4	Mon: Lecture Wed: Lecture	Exercise 3	Exercise 2 Solution
5	Mon: Lecture Wed: Lecture	Exercise 4	Exercise 3 Solution

Course Website

https://www.syscop.de/teaching/ws2024/modelling-and-system-identification

Systems Control and Optimization Laboratory

IMTEK, Faculty of Engineering, University of Freiburg

Home People Software Projects Publications Teaching Events Job Offers Contact

Modelling and System Identification

Prof. Moritz Diehl, Jakob Harzer, Katrin Baumgärtner

Modelling and System Identification (MSI) is concerned with the search for mathematical models for real-life systems. The course is based on statistics, optimization and simulation methods for differential equations. The exercises will be based on pen-and-paper exercises and computer exercises with Python.

Course language is English and all course communication is via this course homepage.

If you have any questions regarding the exercises/lectures, please send an **email to the tutors**, syscop.msi@gmail.com

---

Lectures. The lectures will take place on Mondays, *8:30 - 10:00 a.m in Building 101, HS 026* and Wednesdays, *8:30-10:00 a.m. in Building 101, HS 036*. If you cannot attend, you may watch the lecture recordings, see below.

Exercises. The exercise sheets include both pen-and-paper exercises as well as programming exercises using Python. Exercise sheets can be hand-ed in before the lecture on Monday, 8:30 or put into the Mailbox in front of room 00-075 in the ground floor of building 101 before. Programming exercis-es are handed in via *Github Classrooms* (more on that below). You have one week to work on the sheet and you might work in groups of at most three students.

Script

Questions?

syscop.msi@gmail.com

syscop.de