

# Kamerabasierte Fahrbahnerkennung zur automatisierten Fahrbahnführung eines Modellauto

Bahri Enis Demirtel

Master Thesis – 02. November 2017

Betreuer: Dr. -Ing. Eric Lenz



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

REGELUNGSTECHNIK *rtm*  
UND MECHATRONIK



---

## Aufgabenstellung

---

Für schriftliche Arbeiten (Pro-/Projektseminar, Studien-, Bachelor-, Master-, Diplomarbeiten, etc.) soll Studierenden ein L<sup>A</sup>T<sub>E</sub>X-Dokument zur Verfügung gestellt werden, das die Vorgaben aus den *Richtlinien zur Anfertigung von Studien- und Diplomarbeiten* [?] umsetzt. Die Dokumentation soll die Funktionen des Dokumentes beschreiben und Hinweise zu ihrer Anwendung geben.

Grundlage ist die tudreport-Klasse. Die damit erstellten Arbeiten müssen sowohl zum Ausdrucken geeignet sein als auch für die Bildschirmdarstellung und die elektronische Archivierung als PDF-Datei.

Beginn: 02. May 2017

Ende: 02. November 2017

Seminar: 15. November 2017

---

Prof. Dr.-Ing. Ulrich Konigorski

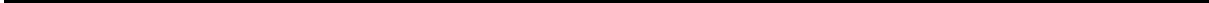
---

Dr. -Ing. Eric Lenz

Technische Universität Darmstadt  
Institut für Automatisierungstechnik und Mechatronik  
Fachgebiet Regelungstechnik und Mechatronik  
Prof. Dr.-Ing. Ulrich Konigorski

Landgraf-Georg-Straße 4  
64283 Darmstadt  
Telefon 06151/16-25200  
[www.rtm.tu-darmstadt.de](http://www.rtm.tu-darmstadt.de)





---

## **Erklärung**

---

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 02. November 2017

---

Bahri Enis Demirtel

---

## Kurzfassung

---

Das  $\text{\LaTeX}$ -Dokument `sada_tudreport` ist eine Vorlage für schriftliche Arbeiten (Proseminar-, Projektseminar-, Studien-, Bachelor-, Master- und Diplomarbeiten, etc.) am Institut für Automatisierungstechnik der TU Darmstadt. Das Layout ist an die *Richtlinien zur Anfertigung von Studien- und Diplomarbeiten* [?] angepasst und durch Modifikation der Klasse `tudreport` realisiert, so dass in der Arbeit die gewohnten  $\text{\LaTeX}$ -Befehle benutzt werden können. Die vorliegende Anleitung beschreibt die Klasse und gibt grundlegende Hinweise zum Verfassen wissenschaftlicher Arbeiten. Sie ist außerdem ein Beispiel für den Aufbau einer Studien-, Bachelor-, Master- bzw. Diplomarbeit.

**Schlüsselwörter:** Studienarbeit, Bachelorarbeit, Masterarbeit, Diplomarbeit, Vorlage,  $\text{\LaTeX}$ -Klasse

---

## Abstract

---

The  $\text{\LaTeX}$  document `sada_tudreport` provides a template for student's research reports and diploma theses ("Proseminar-, Projektseminar-, Studien-, Bachelor-, Master- und Diplomarbeiten") at the Institute of Automatic Control, Technische Universität Darmstadt. The layout is adapted to the "*Richtlinien zur Anfertigung von Studien- und Diplomarbeiten*" [?] and is implemented by modification of the standard `tudreport` class, so that common  $\text{\LaTeX}$  commands can be used in the text. This manual describes the class and dwells on general considerations on how to write scientific reports. Additionally, it is an example for the structure of a thesis.

**Keywords:** Research reports, diploma theses, template,  $\text{\LaTeX}$  class

---

# Inhaltsverzeichnis

<b>Symbole und Abkürzungen</b>	<b>vii</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Introduction . . . . .	1
1.2. Problem Statement and Objective Target . . . . .	1
1.3. Structure of Paper . . . . .	2
<b>2. Fundamentals</b>	<b>3</b>
1. Phase: Einarbeitung und Literaturrecherche . . . . .	3
Generelles zu Treffen mit dem Betreuer . . . . .	3
Literaturrecherche und -management . . . . .	3
2. Phase: Bearbeitung des Problems . . . . .	4
3. Phase: Ergebnisse zusammen stellen . . . . .	4
Ergebnissicherung . . . . .	4
4. Phase: Dokumentation und Präsentation . . . . .	4
Wissenschaftliches Schreiben . . . . .	5
Verteidigung und Präsentation . . . . .	5
Organisatorisches . . . . .	6
Allgemeines zu Hilfsmitteln bei der Erstellung und Bearbeitung . . . . .	6
Empfohlene Programme . . . . .	7
<b>3. Implementation</b>	<b>9</b>
<b>4. Evaluation and Discussion</b>	<b>11</b>
4.1. Editor . . . . .	11
4.2. Rechtschreibung . . . . .	11
4.3. Zitate und korrekte Zitierweise . . . . .	11
4.4. Gliederung des Dokuments . . . . .	13
4.5. Bilder . . . . .	13
4.6. Tabellen . . . . .	15
4.7. Mathematische Formeln . . . . .	15
4.8. Auszeichnungen und Hervorhebungen . . . . .	17
4.9. Einbinden von Quellcode . . . . .	17
4.10. Abstände und Sonderzeichen . . . . .	18
4.11. Definition eigener Befehle . . . . .	19

---

4.12. Sonstiges . . . . .	19
<b>5. Related Works</b>	<b>21</b>
5.1. Verzeichnisse . . . . .	21
5.2. Verzeichnis common . . . . .	21
5.3. Angaben über die Arbeit . . . . .	22
5.4. „Entwurfsmodus“ . . . . .	23
<b>6. Conclusion</b>	<b>25</b>
<b>A. Checkliste</b>	<b>27</b>
<b>B. Programme zur Erstellung von Grafiken</b>	<b>29</b>
B.1. Vektorgrafiken . . . . .	29
B.2. Pixelgrafiken . . . . .	30
<b>C. Das T<sub>E</sub>X-System</b>	<b>31</b>
<b>D. Befehle in commonmacros.tex</b>	<b>35</b>



---

# Symbole und Abkürzungen

## Lateinische Symbole und Formelzeichen

Symbol	Beschreibung	Einheit
$I$	Strom	A
$R$	Widerstand	$\Omega$
$U$	Spannung	V

## Griechische Symbole und Formelzeichen

Symbol	Beschreibung	Einheit
$\Psi$	Datenmatrix	
$\sigma$	Standardabweichung	
$\omega$	Kreisfrequenz	$s^{-1}$

## Abkürzungen

Kürzel	vollständige Bezeichnung
Dgl.	Differentialgleichung
LS	Kleinste Quadrate ( <i>Least Squares</i> )
PRBS	Pseudo-Rausch-Binär-Signal ( <i>Pseudo Random Binary Signal</i> )
ZVF	Zustandsvariablenfilter



---

# 1 Introduction

---

## 1.1. Introduction

---

Technology is also developing in the automotive industry like in other industries over time. The number of sensors and accordingly features in automobiles increases exponentially. One of these sensors is color camera. At the beginning, the cameras were used in the automobile industry just for helping to park and to drive backwards. Nowadays, one of the main functions of color cameras is lane detection in autonomous cars and in the cars which have lane departure warning system. In this master thesis must be the lanes detected and formulated mathematically.

The results of this master thesis will be expanded by the students, which will be participated to the Echtzeitsysteme Projektseminar at the Technical University of Darmstadt. One of the aim of this seminar is to attend the Carolo-Cup which is organized annual by Technical University of Braunschweig. Because of that, in this master thesis, width of track, its curvature and changes of curvature of which, are used from Carolo-Cup. In real life, they are more problems for detecting lanes, they are shadows of which can be caused by appearances of trees and buildings, the dirt left on the road surface etc.

Thereby, the lanes of track must be detected in enough short time and there shouldn't be any dead time between lane detection and formulating mathematically. Other main thing, the lane detection must be enough robust sein. The lanes must be detected also certainly with expected light conditions.

---

## 1.2. Problem Statement and Objective Target

---

Autonomous driving is one of the active researching themes. Autonomous driving can be studied in two fundamental tasks. These tasks are : lane detection and lane guidance. For lane detection, they are different scientific techniques in the literature. All of these techniques have advantages and disadvantages for different situations. For example, some of these techniques are suitable just "lines" not for curves. Some of them are suit for also curves but they are not good in different light conditions and some of them are robust and suit for curves but they are so much computationally expensive. In these master thesis, our aim is, to research and implement the best method for Carola-Cup.

---

### 1.3. Structure of Thesis

---

In Chapter 2 is explained the fundamentals of lane detection. All methods, which are used in this thesis, are also explained in this chapter with their reasons. Some methods are also compared with their advantages and disadvantages.

In Chapter 3 is explained steps of implementation. These steps are properties of track, hardware of model car and used software libraries and programs. In this chapter will also explained detailed program flow.

In Chapter 4

In Chapter 5

In Chapter 6

---

## 2 Fundamentals

---

### 1. Phase: Einarbeitung und Literaturrecherche

---

- dient zur Verdeutlichung der gegebenen Problemstellung
- führt den Bearbeiter auf den „State-of-the-Art“ hin → Literatur
- zeigt Möglichkeiten zur Problemlösung auf, die dann in Phase 2 genutzt werden können
- dient der Ideenfindung
- regelmäßige Treffen mit dem Betreuer

---

### Generelles zu Treffen mit dem Betreuer

---

- persönliche Treffen sollten gut vorbereitet werden (z.B. Fragenkatalog)
- Ergebnisse des Treffens und Anmerkungen des Betreuers festhalten (Notizen)

---

### Literaturrecherche und -management

---

Zur Sammlung und Verwaltung von Literatur dienen sog. Literaturdatenbanken. Diese Programme ermöglichen oft die direkte Verwendung als Literaturverzeichnis im späteren Text.

*Empfehlung: JabRef (Freeware, auf Java-Basis, für alle Plattformen)*

Das Suchen und Lesen der aktuellsten Literatur ist wichtig für eine gute Übersicht über die Problemstellung. Über die ULB ist es möglich viele Online-Quellen zu nutzen, um Texte auch als pdf Dokumente erhalten zu können. Die für unseren Bereich wichtigsten Seiten sind:

- Elektronische Zeitschriftenbibliothek (DB aller Zeitschriften und Berechtigungen)
- IEEEExplore (Zugang zu allen Medien der IEEE)
- Sciencedirect (Zugang zu Zeitschriften aus dem ELSEVIER Verlag)
- CiteSeer (freie DB mit Zugang zu vielen Volltexten)
- IFAC-PapersOnline (Archiv mit Artikeln aller IFAC Konferenzen)
- Springer E-Books (Archiv des Springer Verlags mit vielen eBooks und Zeitschriften)

auch mal auf den Webseiten der Autoren schauen und natürlich „googlen“

---

## 2. Phase: Bearbeitung des Problems

---

- dient zur Lösung der Problemstellung
- kreativste und anstrengendste Phase der Arbeit
- Verwendung professioneller Hilfsmittel (Programme wie Matlab oder Mathematica etc.)
- Treffen wenn Bedarf besteht, keine Regelmäßigkeit mehr

---

## 3. Phase: Ergebnisse zusammen stellen

---

- dient der Reorganisation der Arbeit
- sämtliche Ergebnisse werden festgehalten
- Strukturierung und Gliederung der Ergebnisse, so dass die nächste Phase (Schreiben) gut durchgeführt werden kann
- wenige, längere Treffen zur Ergebnisbesprechung mit dem Betreuer

---

### Ergebnissicherung

---

- alles zusammentragen was erreicht wurde → guter Überblick notwendig (S.O.)
- auch Programmcode ist ein Ergebnis → verständlich kommentieren (Englisch)
- auf Wiederverwendbarkeit von Grafiken achten (Linienstärke, Farbe, Beschriftung, ...)
- nur noch kleine Änderungen durchführen (z.B. Parametereinstellungen)

*Ergebnisse sollten für sich sprechen und für jeden verständlich sein (ohne Erklärung)*

---

## 4. Phase: Dokumentation und Präsentation

---

- Verfassen der Arbeit und erstellen der Präsentation
- letzte Verfeinerungen an den Ergebnissen (falls notwendig)
- schwierigste Phase der Arbeit
- regelmäßige Treffen zur Korrektur des Textes / der Präsentation

---

## Wissenschaftliches Schreiben

---

- nicht am Layout der Arbeit aufhalten → Vorlage verwenden
- sehr aufwendiger Prozess von Schreiben - Verbessern - neu Schreiben - ...
- Aufwand darf nicht unterschätzt werden (Richtwert: 1-2 Seiten / Tag)
- einfach erst einmal aufschreiben - korrigiert wird dann später
- Struktur einer wissenschaftlichen Arbeit ist vorgegeben
  - **Titelseite** mit Art der Arbeit, Titel, Namen des Autors sowie Abgabedatum.
  - **Aufgabenstellung** wird vom Betreuer der Arbeit zur Verfügung gestellt.
  - **Erklärung** zur Selbständigkeit. Der Text ist vorgegeben und wird bei Verwendung der Vorlage automatisch erzeugt.
  - **Kurzfassung** der Arbeit. Der Umfang soll so bemessen sein, dass die englische Version (**Abstract**) auf die gleiche Seite passt.
  - **Inhaltsverzeichnis** wird in  $\text{\LaTeX}$  durch `\tableofcontents` automatisch erzeugt.
  - **Symbole und Abkürzungen**. Dieses Verzeichnis erstellt man am Besten von Hand. Die Einteilung in lateinische und griechische Symbole und Formelzeichen kann nach Bedarf geändert werden (zum Beispiel nach Kapiteln oder Konzepten) oder ganz weggelassen werden
  - **Hauptteil der Arbeit**, in einzelne Kapitel und Abschnitte unterteilen.
  - **Anhang**. Hier können Abschnitte stehen, die beim Lesen der Arbeit stören würden, z. B. Programmcode, technische Daten oder lange mathematische Beweise.
  - **Literaturverzeichnis** wird entweder von Hand erstellt oder automatisch generiert (in  $\text{\LaTeX}$  z. B. mit. `BIBTeX`)
  - **Zusätzliches Material** wie z. B. der vollständige Programmcode eines Software-Projekts gehört nicht in die Arbeit, sondern kann in einem separaten Ordner abgelegt werden.

$P_{\text{rägnanz}}$   $O_{\text{rdnung}}$   $E_{\text{infachheit}}$   $M_{\text{otivation}}$  (einfache, kurze, strukturierte und anregende Sätze)

---

## Verteidigung und Präsentation

---

- Inhalt der Arbeit auf ca. 10 wesentliche Punkte reduzieren
- je Punkt eine Folie maximal zwei
- je Folie 1-2 min Gesprächszeit
- Struktur der Präsentation:

- 
- Motivation / Einleitung (Aufgabenstellung)
  - Grundlagen
  - Lösungsweg
  - Ergebnisse
  - Zusammenfassung (und Ausblick)
- komplexe Sachverhalte durch Abbildungen verdeutlichen
  - nur Stichpunkte schreiben, keine ganzen Sätze
  - klare, einfach nachvollziehbare Notation (z.B. bei Variablen) verwenden
  - mehrmaliges Üben des Vortrags (z.B. vor dem Spiegel, vor der Familie, ...)

*nichts weglassen was auf einer Folie steht, gerne zusätzliche Dinge erwähnen*

---

## Organisatorisches

---

- Bewertung
  - Arbeitsstil: 40% - Selbständigkeit, Verständnis, Kreativität, Fleiß, Zusammenarbeit mit Betreuer, Systematik
  - Ergebnisse: 20% - Qualität, Nutzbarkeit, Innovations-, Erfüllungsgrad
  - Ausarbeitung: 30% - Aufbau, äußere Form, Sprache, Grundlagen, Vollständigkeit
  - Vortrag: 10% - Inhalt, Stil, Folien, Vorführung, Diskussion
- jeder Studierende sollte an mindestens einem Regelungstechnischen Seminar teilnehmen
- Abgabetermin ist ein fixer Termin → Prüfungsleistung (Durchfallen möglich)
- jeder Studierende erhält auf Wunsch einen eigenen Account für unsere Rechner (Passwort legt Admin fest)
- zur Erstellung der Ausarbeitung und der Präsentation wird die Verwendung des Textsatzprogramms LaTeX empfohlen, das Template für die Ausarbeitung erhältet Ihr vom Betreuer

---

## Allgemeines zu Hilfsmitteln bei der Erstellung und Bearbeitung

---

- es gibt viele Bücher die bei der Erstellung von wissenschaftlichen Arbeiten helfen
- viele verwendete Programme bieten ausführliche Hilfen an (z.B. Matlab, Mathematica)
- bei Problemen immer zuerst in der Hilfe schauen, dann "googlen", dann Betreuer fragen



---

## Empfohlene Programme

---

- Literaturdatenbank → JabRef (<http://jabref.sourceforge.net/>)
- Ausarbeitung und Präsentation → L<sup>A</sup>T<sub>E</sub>X/ PowerPoint
- Simulation und Regelungstechnik → Matlab/Simulink
- symbolische und numerische Berechnungen → Mathematica
- Plots → pgfplots
- Blockschaltbilder → TikZ



---

## 3 Implementation

Das vorliegende Dokument beschreibt die Formalitäten, die bei der Erstellung von Studien-, Bachelor-, Master- und Diplomarbeiten zu beachten sind. Außerdem gibt es Tipps, worauf bei der Durchführung zu achten ist. Das Dokument entstand durch die Zusammenführung der `iat-sada`-Klasse mit dem TU-Design. Die Tipps und die Beschreibung des  $\text{T}_{\text{E}}\text{X}$ -Systems sind aus der Beschreibung der `iat-sada`-Klasse von Michael Vogt übernommen.



---

## 4 Evaluation and Discussion

Dieses Kapitel gibt allgemeine Hinweise zur Erstellung einer wissenschaftlichen Arbeit mit dem Textsatzprogramm  $\text{\LaTeX}$ . Das Kapitel sollte auch von Studierenden gelesen werden, die sich gegen eine Erstellung der Arbeit mit  $\text{\LaTeX}$  entschieden haben. Da das Layout der Arbeit in diesem Fall gemäß den TUD Designvorgaben zusätzlich selbst erstellt werden muss, dient die vorliegende Anleitung auch als Vorlage.

---

### 4.1 Editor

---

Für  $\text{\LaTeX}$  gibt es eine Vielzahl von freien und kommerziellen Texteditoren. Das  $\text{\TeX}$ NICCENTER ist ein freier open-source Editor, der sich bei uns am Institut bewährt hat. Eine Anleitung zur Einrichtung des  $\text{\TeX}$ NICCENTER befindet sich in dieser zip-Datei.

---

### 4.2 Rechtschreibung

---

Studentische Abschlussarbeiten am IAT sind nach *neuer* deutscher Rechtschreibung zu verfassen. Die wesentlichen Änderungen betreffen die Groß- und Kleinschreibung sowie die Getrennt- und Zusammenschreibung, siehe [?]. Das offensichtlichste Merkmal ist die neue „ß“-Regel: Nach kurzem Vokal steht jetzt *immer* „ss“ (wie in Abschluss), nach langem Vokal „ß“ (wie in Fuß). Die neue deutsche Rechtschreibung wird in  $\text{\LaTeX}$  mit dem Paket `babel` über die Option `ngerman` aktiviert.

Im Internet findet man unter <http://www.duden.de/> einen Crashkurs zur neuen deutschen Rechtschreibung.

---

### 4.3 Zitate und korrekte Zitierweise

---

Zitate sind wörtliche oder sinngemäße Wiedergaben von Gedanken, Ideen oder Meinungen anderer Autoren. Werden Ideen oder Inhalte aus Quellen wörtlich oder sinngemäß in die eigene Arbeit übernommen, besteht die **Pflicht**, diese zu kennzeichnen. Wird dies unterlassen, liegt ein **Täuschungsversuch** (Plagiat) vor und die Arbeit kann als **nicht bestanden** bewertet werden.

Laut § 38 Abs. 2 der *Allgemeinen Prüfungsbestimmungen* liegt „[...] ein Täuschungsversuch [...] vor, wenn eine falsche Erklärung nach §§ 22 Abs. 7, 23 Abs. 7 abgegeben worden ist oder ein anderes Werk, eine Bearbeitung eines anderen Werkes, eine Umgestaltung eines anderen Werkes ganz oder teilweise in der Prüfungsarbeit wiedergegeben werden, ohne dieses zu zitieren (Plagiat).“ [?]

---

## Zitierfähigkeit

---

Zitierfähig sind nur veröffentlichte Werke aus allgemein zugänglichen Quellen (Bücher, Artikel, etc.). Quellen, bei denen die Verfügbarkeit nicht garantiert werden kann (Internetquellen) oder der Urheber nicht klar nachvollziehbar ist (Wikipedia, o. ä.), sind problematisch und sollten möglichst vermieden werden. Wird doch solch eine Quelle zitiert, ist die aktuelle Version zum Zeitpunkt des Zitates beizulegen.

---

## Wörtliche Zitate

---

Wörtliche Zitate in ingenieurwissenschaftlichen Arbeiten sind unüblich. Sollte doch ein wörtliches Zitat in die Arbeit übernommen werden, muss dieses buchstaben- und zeichengetreu, inklusive eventueller Rechtschreibfehler übernommen werden. Das wörtliche Zitat wird in Anführungszeichen eingefasst.

---

## Sinngemäße Zitate

---

Weit häufiger werden in wissenschaftlichen Arbeiten Ideen oder Meinungen anderer Autoren sinngemäß übernommen. Diese müssen durch einen Verweis auf die Quelle gekennzeichnet werden. Durch die Position des Verweises muss der Umfang der sinngemäßen Übernahme klar hervorgehen.

---

## Quellenangaben im Literaturverzeichnis

---

Für die Darstellung der Verweise, als auch für die Darstellung der Quellen im Literaturverzeichnis gibt es verschiedene Zitierweisen. Üblich sind die Harvard-Variante mit Autor und Veröffentlichungsjahr in runden Klammern, wie zum Beispiel (Isermann, 2001) oder eine fortlaufende Nummerierung in eckigen Klammern, wie in dieser Vorlage. Die Sortierung des Literaturverzeichnisses kann alphabetisch oder nach dem Erscheinen der Verweise erfolgen. Das `bibgerm` Paket ermöglicht verschiedene (im Deutschen übliche) Zitierweisen.

Ein Verweis wird mit `\cite{...}` eingefügt und mit einem festen Leerzeichen `~` mit dem vorherigen Wort getrennt. Schließt der Verweis einen Satz ab, folgt der Punkt *hinter* dem Verweis.

Für die Erstellung des Literaturverzeichnisses bietet sich die Erweiterung `BibTeX` an. Das Literaturverzeichnis kann per Hand oder automatisch erstellt werden. Viele Literaturverwaltungsprogramme, wie zum Beispiel `JABREF` ermöglichen den direkten Export der Datenbank nach `BibTeX`.

---

## 4.4 Gliederung des Dokuments

---

Im Inhaltsverzeichnis wird die Gliederung der Arbeit dargestellt. Die Überschriften und Seitenangaben der Kapitel, Unterkapitel und Abschnitte müssen mit den Elementen im Inhaltsverzeichnis übereinstimmen. Überschriften sind kurz und prägnant zu formulieren und dürfen keine vollständigen Sätze sein. Gibt es Unterpunkte in der Gliederung, so müssen immer mindestens zwei davon existieren und inhaltlich auf der gleichen Ebene sein. Die einzelnen Punkte des Inhaltsverzeichnis müssen nummeriert werden. Die Übersichtlichkeit des Inhaltsverzeichnisses kann durch Einrücken der Unterpunkte erhöht werden.

Das Inhaltsverzeichnis wird in  $\text{\LaTeX}$  automatisch erstellt. Innerhalb der einzelnen Kapitel `\chapter{...}` werden weitere Unterteilungen mit den Befehlen `\section{...}`, `\subsection{...}` usw. vorgenommen. Werden diese mit einem `*` versehen, dann erhält der jeweilige Abschnitt keine Nummer und erscheint nicht im Inhaltsverzeichnis. Dies kann in manchen Fällen nützlich sein.

Um eine korrekte Darstellung des Inhaltsverzeichnisses zu erhalten, muss ggf. mehrmals hintereinander kompiliert werden, da sich aufgrund von Gleitobjekten Seitenzahlen ändern können. Dreimaliges Kompilieren reicht in der Regel.

Bei Bildern und Tabellen, die in eine `figure`- bzw. `table`-Umgebung eingeschlossen sind, handelt es sich um sog. *Gleitobjekte*, d. h. sie erscheinen nicht an der Stelle, an der sie eingebunden werden, sondern oben oder unten auf einer Seite, siehe z. B. Tabelle 4.2 auf Seite 19. Optional können in `[]` noch Positionierungswünsche angegeben werden. Mit dem Befehl `\caption{...}` erhalten Bilder eine *Unterschrift* und Tabellen eine *Überschrift*.

Kapitel, Abschnitte, Bilder, Tabellen und Gleichungen können mittels `\label{...}` benannt werden. Dadurch ist es möglich, sie später mit `\ref{...}` oder `\pageref{...}` zu referenzieren. Es empfiehlt sich, den Namen (Labels) eine Markierung voranzustellen, aus der hervorgeht, um welches Objekt es sich handelt. Üblich sind `cha`: für „Chapter“, `sec`: für „Section“, `fig`: für „Figure“, `tab`: für „Table“ und `eq`: für „Equation“. Ein Bild benennt man also z. B. mit `\label{fig:Ausgangssignal}`.

---

## 4.5 Bilder

---

Wenn möglich, sollten Bilder als Vektorgrafik eingebunden werden, damit sichergestellt werden, dass alle Details beim Ausdrucken erhalten bleiben. Es ist dabei auf eine ausreichende Strichstärke zu achten. Ebenfalls sollten die im Bild verwendete Schrift die gleiche sein, wie im übrigen Dokument.

Werden doch Pixelgrafiken verwendet, so ist die richtige Wahl der Auflösung von besonderer Bedeutung. Einerseits sollten die Bilder auf dem ausgedruckten Dokument gut aussehen, andererseits aber auch eine zügige Bildschirmdarstellung und kleine Dateigröße ermöglichen.

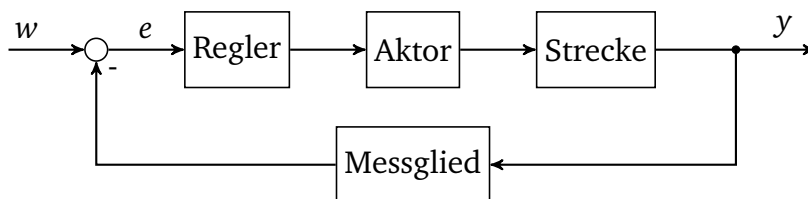
Für die Erstellung von Plots eignet sich das  $\text{\LaTeX}$ -Paket PGFPLOTS sehr gut. Eine Anleitung zu diesem Paket befindet sich in dieser zip-Datei. In Anhang B sind weitere geeignete Programme zur Erstellung von Bildern und Plots mit ihren Eigenschaften aufgelistet.

Eine Grafik lässt sich am einfachsten mit dem Befehl `\includegraphics{}` aus dem `graphicx`-Paket einbinden. Eine vollständige Beschreibung des Befehls und weiterer nützlicher Grafikbefehle findet man in der Dokumentation des `graphicx`-Pakets. Diese liegt – wie die Beschreibung aller anderen  $\text{\LaTeX}$ -Pakete – im doc-Verzeichnis des  $\text{\TeX}$ -Systems. In welchem Format `graphicx` die Grafiken benötigt, hängt davon ab, ob man  $\text{\TeX}$  in Verbindung mit Dvips verwendet oder stattdessen pdf $\text{\TeX}$ , siehe Anhang C.

Für  **$\text{\TeX}$ /Dvips** (empfohlen) müssen alle Grafiken im *Encapsulated-PostScript-Format* (\*.eps) vorliegen. EPS-Dateien lassen sich aus praktisch jeder Software erzeugen und können sowohl Vektor- als auch Pixelgrafiken enthalten – zusätzlich sind auch Preview-Grafiken möglich, was aber in der  $\text{\TeX}$ -Welt im Allgemeinen nicht erforderlich ist.

**pdf $\text{\TeX}$**  verarbeitet dagegen Grafiken im *Portable-Document-Format* (\*.pdf) sowie die Pixelformate JPEG und PNG. Den direkten Export von PDF-Grafiken bieten derzeit zwar nur wenige Programme an, sie lassen sich aber einfach aus dem EPS-Format mit Hilfe des Acrobat Distiller oder Ghostscript erzeugen.

Bilder müssen zentriert sein (`\centering`) und eine *Bildunterschrift* (`\caption{...}`) besitzen. Um auf eine Abbildung zu referenzieren, kann auch sie mit einem `\label{fig:...}` versehen werden. Nur in besonderen Ausnahmefällen sollten Bilder mit Text umflossen werden. Wurden Abbildungen einer Quelle entnommen, muss dies entsprechend mit einem Verweis auf die Quelle im Literaturverzeichnis gekennzeichnet werden. Werden Abbildungen eine Quelle nachempfunden, angepasst oder abgeändert, ist dies mit dem Zusatz „in Anlehnung an...“ oder ähnlich anzugeben.



**Abbildung 4.1.:** Standard-Regelkreis; Bild erstellt mit TikZ

Generell ist die Arbeit (und insbesondere die Grafiken) so zu gestalten, dass sie auch schwarz-weiß gedruckt werden kann. Farbige Fotos und Screenshots verursachen dabei i. Allg. keine zusätzlichen Probleme. Werden jedoch z. B. farbige Kurven in einem Diagramm dargestellt, hat dies folgende Konsequenzen:

- Keine zu hellen Farben für die Linien verwenden, da diese sonst beim Drucken nicht zu erkennen sind. RGB-Grün (0,1,0) ist tabu!



- Bei mehreren Kurvenverläufen darf deren Farbe nicht das einzige Unterscheidungsmerkmal sein: Entweder sind unterschiedliche Linienformen zu verwenden oder die Kurven im Diagramm beschriften.

Bei der Erstellung von Plots ist unbedingt die DIN 461 Grafische Darstellung in Koordinatensystemen zu beachten. Hilfreich in diesem Zusammenhang und überhaupt für die korrekte Schreibweise von Zahlen und Einheiten im Fließtext und in Formeln sind die Hinweise der TU-Chemnitz: [www.tu-chemnitz.de/physik/FPRAK/Grundsatz/Literatur/si\\_v1.pdf](http://www.tu-chemnitz.de/physik/FPRAK/Grundsatz/Literatur/si_v1.pdf).

## 4.6 Tabellen

Tabellen müssen ebenfalls zentriert sein und besitzen eine zentrierte Tabellenüberschrift (`\caption{...}`). Hier gelten die gleichen Regeln zur Quellenangabe wie bei den Bildern. Ein Referenzieren wird auch hier mit `\label{tab:...}` ermöglicht.

Damit Tabellen „schön“ aussehen, empfiehlt es sich, einige Grundregeln zu beachten. Es gilt das Prinzip: *weniger ist mehr*. So sollte auf die Verwendung von senkrechten Linien verzichtet werden und nur wichtige Zeilen, wie zum Beispiel Überschriften, Sinnabschnitte, Unterpunkte, etc. mit horizontalen Linien getrennt werden. Das Paket `booktabs` stellt Linientypen für den Kopf und Fuß einer Tabelle zur Verfügung.

**Tabelle 4.1.: Parameter<sup>1</sup>**

	Ref.	Mod.	Einheit		Ref.	Mod.	Einheit
$m_1$	4,0	4,63	kg	$d_1$	0	0	$\frac{\text{Nms}}{\text{rad}}$
$m_2$	10,1	11,15	kg	$d_2$	0	0	$\frac{\text{Nms}}{\text{rad}}$
$m_3$	45,7	42,5	kg	$l_1$	0,5	0,45	m
$J_1$	0,967	0,993	$\text{kg m}^2$	$l_2$	1,5	1,59	m
$J_2$	0,571	0,599	$\text{kg m}^2$	$\varphi_{1,0}$	100	98,5	°
$g$	9,81	9,81	$\frac{\text{m}}{\text{s}^2}$	$\varphi_{2,0}$	5	4.66	°

## 4.7 Mathematische Formeln

Mathematische Formeln, wie

$$\int_0^\infty g(x) dx \approx \sum_{i=1}^n w_i e^{x_i} g(x_i) \quad (4.1)$$

<sup>1</sup> Eine schönere Formatierung der Tabelle (Ausrichtung am Trennzeichen, etc.) erlaubt das `siunitx`-Paket, welches sich aber mit irgendwas in der Vorlage beißt (nur Warnmeldungen, aber trotzdem...)

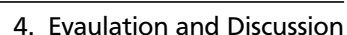
Alle mathematischen Ausdrücke (auch wenn es nur einzelne Zeichen sind) werden im mathematischen Modus `$...$` geschrieben, damit sie in der richtigen Schrift erscheinen. Hier gilt die Faustregel, dass gewöhnliche mathematische Größen *kursiv* geschrieben werden, Ausdrücke mit konventioneller (feststehender) Bedeutung dagegen in normaler (steiler, aufrechter) Schrift, siehe [?]. Es sind insbesondere Einheiten, Standardfunktionen und -operatoren sowie mathematische Konstanten steil zu schreiben,

Da der mathematische Modus zunächst alle Größen *kursiv* setzt, müssen Ausdrücke, die in aufrechter Schrift erscheinen sollen, mit dem Befehl `\mathrm{...}` gekennzeichnet werden; bei Textteilen innerhalb einer Formel verwendet man besser `\mbox{...}` oder `\text{...}` (aus dem `amsmath`-Paket). Für Standardfunktionen werden von L<sup>A</sup>T<sub>E</sub>X die entsprechenden Befehle `\sin`, `\log`, `\max` usw. zur Verfügung gestellt. Konsequenter Weise muss dieses Prinzip auch auf Indizes angewendet werden, also z. B.

Als Ausnahme von der o. g. Faustregel werden große griechische Buchstaben meist *nicht* kursiv geschrieben, so auch im mathematischen Modus von L<sup>A</sup>T<sub>E</sub>X. Matrizen und Vektoren werden in **fetten** Buchstaben gesetzt. Damit sie sich besser von den übrigen Symbolen abheben, werden auch sie nicht **fett-kursiv** sondern **fett-steil** geschrieben. Dazu definiert man zweckmäßigerweise im Vorspann des Dokuments den Befehl

den man dann sowohl für Matrizen (Großbuchstaben, z.B.  $\mat{A}$ ) als auch für Vektoren (Kleinbuchstaben, z.B.  $\mat{x}$ ) verwenden kann.

Chemische Formelzeichen schreibt man grundsätzlich in aufrechter Schrift. Variable Größen sind aber auch hier kursiv:



---

Beim Referenzieren von Gleichungen muss diese nummeriert werden. Ist die Gleichung

$$\int_0^{\infty} g(x) dx \approx \sum_{i=1}^n w_i e^{x_i} g(x_i) \quad (4.2)$$

mit `\label{eq:Approx}` bezeichnet, erzeugt die Referenz `\eqref{eq:Approx}` den Ausdruck „(4.2)“. Kapitel, Abschnitte, Bilder und Tabellen bekommen keine Klammern, also z.B. `Kapitel~\ref{cha:Intro}` für „Kapitel 1“. Es ist darauf zu achten, ob es sich um *Kapitel* oder *Abschnitte* handelt. Vor `\ref{...}` steht ein festes Leerzeichen `~`, damit dort kein Umbruch erfolgen kann. Literaturangaben werden mit `\cite{...}` anstelle von `\ref{...}` referenziert.

Werden Gleichungen oder Listen in den laufenden Text eingefügt, darf dazwischen kein Absatz (d.h. eine Leerzeile im Quelltext) sein. Um den Quelltext besser zu gliedern, kann an dieser Stelle eine Zeile mit einem `%`-Zeichen eingefügt werden. Eine Leerzeile darf nur dann im Quelltext stehen, wenn auch wirklich ein Absatz erwünscht ist. Der Zeilentrenner `\` erzeugt übrigens keinen Absatz und darf im laufenden Text überhaupt nicht vorkommen.

---

## 4.8 Auszeichnungen und Hervorhebungen

---

Wichtige Begriffe werden durch eine andere Schrift hervorgehoben (ausgezeichnet). Man unterscheidet dabei integrierte und aktive Auszeichnungen. Integrierte Auszeichnungen sollen erst beim Lesen wahrgenommen werden, sich aber ansonsten in den Text eingliedern. Die typische Form einer integrierten Auszeichnung ist die *kursive* Schrift, die mit `\textit{...}` oder `\emph{...}` erzeugt wird. Aktive Auszeichnungen sollen dagegen sofort beim Betrachten der Seite auffallen. Der wichtigste Vertreter ist hier die **fette** Schrift, die man durch Verwendung von `\textbf{...}` erhält. In wissenschaftlichen Arbeiten werden vorwiegend integrierte Auszeichnungen benutzt.

Grundsätzlich sollte bei Auszeichnungen immer nur *ein* Attribut geändert werden, also *nicht* gleichzeitig **fett**, *kursiv* und unterstrichen. Programmcode und Befehle setzt man üblicherweise mit `\texttt{...}` oder `\verb|...|` in Schreibmaschinenschrift, Namen gelegentlich mit `\textsc{...}` in KAPITÄLCHEN. Für manche Bezeichnungen kommt eine **fette serifenlose** Schrift `\textsf{\textbf{...}}` in Frage. Hier müssen ausnahmsweise *zwei* Attribute geändert werden, da sich die serifenlose Schrift zu wenig vom übrigen Text abhebt.

„Anführungszeichen“ (siehe Abschnitt 4.10) sind sparsam zu verwenden, z.B. bei umgangssprachlichen Begriffen oder wörtlichen Zitaten. Unterstreichen und **S p e r r e n** sollen überhaupt nicht benutzt werden. Es ist wichtig, sich zu Beginn der Arbeit zu überlegen, welche Begriffe in welcher Schrift gesetzt werden, und dies konsequent einzuhalten.

---

## 4.9 Einbinden von Quellcode

---

Wird Quellcode (MATLAB, C, ...) in der Arbeit angegeben, ist grundsätzlich eine Monospace-Schriftart zu verwenden, da nur so die Lesbarkeit des Codes gewährleistet werden kann.

---

Quellcode (MATLAB, C, ...) kann auf verschiedene Arten eingebunden werden. Allgemein sollte mittels `\linespread{1}` der ursprüngliche L<sup>A</sup>T<sub>E</sub>X-Zeilenaabstand benutzt werden. Manchmal kann es auch erforderlich sein, die Schrift zu verkleinern oder notfalls sogar die Seiten im Querformat zu beschreiben. Im laufenden Text sollten nur kleinere Code-Fragmente abgedruckt sein, längere Programme gehören grundsätzlich in den Anhang oder in einen separaten Ordner.

Die einfachste Möglichkeit zum Einbinden von Quellcode ist die `verbatim`- bzw. die `verbatim*`-Umgebung. Der Code wird in Schreibmaschinenschrift *exakt* (inkl. aller Leer- und Sonderzeichen) so wiedergegeben, wie er im L<sup>A</sup>T<sub>E</sub>X-Quelltext steht.

Komfortablere Möglichkeiten bietet das `listings`-Paket, z. B. Syntax-Highlighting mit verschiedenen Schriften oder das Einbinden externer Dateien. Die Umschaltung auf den einfachen Zeilenaabstand muss aber von Hand erfolgen, z. B. mittels

```
\lstset{\basicstyle=\linespread{1}\selectfont}
```

---

## 4.10 Abstände und Sonderzeichen

---

L<sup>A</sup>T<sub>E</sub>X interpretiert ein Leerzeichen `\` im Quelltext als normalen Wortzwischenraum. Nach Befehlen wird es jedoch ignoriert, da es dort nur das Ende des Befehls kennzeichnet. Soll z. B. in dem Satz „T<sub>E</sub>X ist toll!“ nach „T<sub>E</sub>X“ ein Leerzeichen erscheinen, dann muss im Quelltext entweder `\TeX\` oder `\TeX{}` geschrieben werden. Im ersten Fall wird durch `\` ein Leerzeichen erzwungen, im zweiten Fall wird die leere Umgebung `{}` benutzt, um den Befehl `\TeX` zu beenden.

Manchmal führt ein normales Leerzeichen zu unerwünschten Ergebnissen. Bei fest verbundenen Begriffen benutzt man ein *festes* Leerzeichen, z. B. bei Dr.~Müller oder 3~Uhr, das weder umgebrochen noch gedehnt werden kann, s. a. Abschnitt 4.4. Bei zusammengesetzten Abkürzungen, beispielsweise „d. h.“, „u. a.“ oder „z. B.“, wird ein *kleiner* Zwischenraum `\`, verwendet. Hinter dem zweiten Punkt sollte wieder ein `\` stehen, damit dieser nicht als Satzende interpretiert wird. Der kleine Zwischenraum `\`, steht auch zwischen Zahl und Einheit bei physikalischen Größen, siehe Tabelle 4.2.

Im *mathematischen* Modus wird das Komma als Aufzählungszeichen interpretiert und dahinter ein kleiner Abstand eingefügt. Dies ist jedoch problematisch, da das Komma im Deutschen auch als *Dezimalkomma* verwendet wird. Um den zusätzlichen Abstand zu unterdrücken schreibt man z. B. `$2{,}5x$` für „2,5x“.

Unterschiede sind auch bei den „Strichen“ zu beachten. Der *Bindestrich* – steht bei zusammengesetzten Wörtern oder Trennungen und wird ohne zusätzlichen Zwischenraum benutzt. Der *Gedankenstrich* -- steht bei eingeschobenen Satzteilen und als „Bis-Strich“. Als Gedankenstrich wird er immer mit einem Leerzeichen davor und dahinter benutzt, als „Bis-Strich“ ohne Leerzeichen. Das *Minuszeichen* \$-\$ gibt es nur im mathematischen Modus. Tabelle 4.2 zeigt Beispiele für die drei Fälle.

---

**Tabelle 4.2.:** Die wichtigsten Abstände und Sonderzeichen.

Bezeichnung	Beispiel	Eingabe
Leerzeichen	$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ist toll!	<code>\TeX\_{ist\_toll!}</code> <code>\TeX\_{\_ist\_toll!}</code>
festes Leerzeichen	Dr. Müller	<code>Dr.\~Müller</code>
kleines Leerzeichen	d. h. 3,5 km	<code>d.\,h.\_3,5\,km</code>
Bindestrich	$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -Datei	<code>\TeX-Datei</code>
Gedankenstrich	S. 153–165	<code>S.\~153--165</code>
Minuszeichen	$y = 5x - 2$	<code>\$y=5x-2\$</code>
Anführungszeichen	„Beispiel“	<code>\glqq\_Beispiel\grqq\_</code> <code>\glqq\_Beispiel\grqq{}</code>

Die deutschen Anführungszeichen werden mit `\glqq` und `\grqq{}` bzw. `\grqq\_` gesetzt. Keinesfalls dürfen stattdessen englische Anführungszeichen ‘...’ oder gar das Zoll-Zeichen "..." benutzt werden. Wie oben erläutert, muss der Befehl `\grqq` mit `\_` oder mit `{}` abgeschlossen werden, falls danach ein Leerzeichen folgen soll.

---

## 4.11 Definition eigener Befehle

---

Die Möglichkeit eigene Befehle in  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  zu definieren und zu verwenden erleichtert das Erstellen einer wissenschaftlichen Arbeit deutlich. So dient ein eigener Befehl oft dazu, häufig verwendete Befehlsfolgen kürzer und schneller schreiben zu können. Von zentraler Bedeutung ist außerdem, dass man diesen Befehl einfach ändern kann. Hat man z. B. alle Matrizen mit einem eigenen Befehl versehen, der diese fett formatiert, so lässt sich dies auch schnell für alle Matrizen wieder ändern. Entscheidet man sich Matrizen mit einem Unterstrich zu kennzeichnen, so ist lediglich die Anpassung des entsprechenden Befehls notwendig.

Dies lässt sich auch auf Variablenamen übertragen. Definiert man z. B. für  $\tilde{x}_{b2}$  einen neuen Befehl `\xb2`, so verkürzt sich zum einen der Schreibaufwand. Zum anderen lässt sich selbst in der Endphase der Arbeit die Variable umbenennen, z. B. in  $z_2$ , indem lediglich der Befehl verändert wird. Die sinnvolle Verwendung eines Befehls setzt damit voraus, dass er auch immer verwendet wird.

Beispielhafte selbst definierte Befehle, die teilweise hier am Institut verwendet werden, sind in Kapitel D zu finden.

---

## 4.12 Sonstiges

---

Das hyperref-Paket wird benutzt, um die Ausgabe für das *Portable Document Format* (PDF) zu optimieren. Dies funktioniert sowohl mit  $\text{T}_{\text{E}}\text{X}/\text{Dvips}$  als auch mit  $\text{pdfT}_{\text{E}}\text{X}$ . Besondere Einstellungen sind dafür nicht erforderlich.

Im PDF-Dokument können dann alle Verweise auf Kapitel, Gleichungen, Bilder, Literatur usw. angeklickt werden. Um eine gute Druckqualität zu gewährleisten, sind diese Links allerdings *nicht* farblich hervorgehoben.

Außerdem werden in das PDF-Dokument *Bookmarks* (Lesezeichen) eingebettet, die später als Baumstruktur erscheinen und die Navigation erleichtern. In den Bookmarks erscheint die Titelseite sowie alle Einträge des Inhaltsverzeichnisses. Schließlich werden auch noch *Pagelabels* (also „wahre“ Seitenzahlen) erzeugt, die ebenfalls die Navigation erleichtern und von Vorteil sind, wenn nur Teile des Dokuments gedruckt werden.

---

## Verwenden von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Paketen

---

Durch das Einbinden von Zusatzpaketen kann  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  angepasst und erweitert werden. Die Pakete werden mit dem Befehl `\usepackage{...}` eingebunden, ggf. können auch noch Optionen in [...] angegeben werden. Folgende Pakete sollten auf jeden Fall benutzt werden:

<code>inputenc</code>	Ermöglicht mit der Option <code>latin1</code> Umlaute im Quelltext.
<code>babel</code>	Mit Option <code>ngerman</code> für neue deutsche Rechtschreibung.
<code>graphicx</code>	Standard-Grafikpaket, z. B. zum Einbinden von EPS-Bildern.

Zu sämtlichen Paketen findet man im Verzeichnis `TEXMF/doc/latex` eine ausführliche Dokumentation. Welche Pakete sonst noch zum Einsatz kommen, hängt vom Einzelfall ab – allerdings sollten nicht mehr als nötig verwendet werden. Insbesondere sollte man auf Pakete verzichten, die das Layout verändern oder zu stark in das Font-System eingreifen. Im Folgenden sind in alphabetischer Reihenfolge einige Pakete aufgelistet, die für Abschlussarbeiten interessant sein könnten:

<code>amsmath</code>	<code>dcolum</code>	<code>lscap</code>	<code>longtable</code>	<code>subfig</code>
<code>amssymb</code>	<code>europs</code>	<code>latexsym</code>	<code>natbib</code>	<code>textcomp</code>
<code>array</code>	<code>flafter</code>	<code>listings</code>	<code>siunits</code>	<code>verbatim</code>

---

## 5 Related Works

Es handelt sich bei diesem  $\text{\LaTeX}$ -Dokument um ein für studentische Arbeiten am Institut für Automatisierungstechnik vorbereitetes Dokument auf Basis der Klasse `tudreport`. Es ist keine neue, abgeleitete Klasse definiert!

Die Klasse `tudreport` ist aus der Standard-Klasse `scrreprt` abgeleitet und stellt nur wenige neue Befehle zur Verfügung; weitere Funktionen können bei Bedarf durch Zusatzpakete eingebunden oder selbst definiert werden. Die Klasse ist daher auch so aufgebaut, dass sie mit möglichst vielen Paketen zusammen arbeitet. Im Wesentlichen wird das Layout angepasst, wie es in [?] festgelegt ist und sich für solche Arbeiten bewährt hat, z. B.:

- Es wird doppelseitig auf DIN-A4-Papier geschrieben. In die zu erstellende PDF-Version werden Bookmarks und Hyperlinks (nicht farbig!) integriert.
- Der Abstand der Zeilen beträgt das 1,25-fache des Standard-Abstands von  $\text{\LaTeX}$ . Da technische Arbeiten viele Formeln und Bilder enthalten, werden Absätze durch einen zusätzlichen vertikalen Zwischenraum statt durch einen Einzug getrennt.
- Kapitel beginnen immer auf einer neuen Seite.
- Die Titelseite hat ein festes Layout mit dem Logo der TU Darmstadt.

---

### 5.1 Verzeichnisse

---

- `bib`  
Hier wird standardmäßig die Datei `literature.bib` mit den Bibtex-Einträgen erwartet.
- `common`  
Allgemeinere Dateien, in die Teile der Definitionen ausgelagert sind, damit die Hauptdatei nicht überfrachtet wird.
- `images`  
Vorgesehen für Bilder
- `inc`  
Vorgesehen für `tex`-Dateien mit eigentlichem Inhalt

---

### 5.2 Verzeichnis `common`

---

Damit das Hauptdokument nicht überfrachtet wird, sind die folgenden längeren „Abschnitte“ in die angegebenen Dateien im Unterverzeichnis `common` ausgelagert:

- `commonmacros.tex`  
Enthält eine Reihe nützlicher selbst definierter Befehle, Mathe- und Beispielumgebungen sowie die Formate `Matlab_colored_smallfont` und `Matlab_colored` zur formatierten Darstellung von Matlab-Code (siehe Kapitel D für eine Übersicht).
- `includes.tex`  
Beinhaltet alle `\usepackage`-Befehle
- `preface.tex`  
Generiert die ersten Seiten der Arbeit (Aufgabenstellung, Erklärung, Inhaltsverzeichnis, etc. und nimmt weitere Einstellungen vor)
- `SADA_Abstract.tex`  
Kurzfassung der Arbeit in deutscher und englischer Sprache.
- `SADA_Aufgabenstellung.tex`  
Aufgabenstellung bei einer studentischen Arbeit. Achtung: für den FB16 muss für das offizielle Exemplar die im Original unterschriebene Aufgabenstellung an dieser Stelle mit gebunden werden.
- `SADA_Erklaerung.tex`  
Erklärung zu studentischen Arbeiten. (Je nach Fachbereich und Art der Arbeit muss ggf. etwas angepasst werden.)
- `setup.tex`  
Nimmt generelle Einstellungen vor. Diese sollten nur von kundigen Benutzern geändert werden.
- Logos  
Außerdem sind im Verzeichnis `common` noch die Logo-Grafiken im eps- und pdf-Format abgelegt.

---

### 5.3 Angaben über die Arbeit

---

Es sind im Hauptdokument die Befehle

```
\newcommand{\SADATyp}{Diplomarbeit}
\newcommand{\SADATitel}{Titel der Arbeit}
\newcommand{\SADASTadt}{Darmstadt}
\newcommand{\SADAAutor}{Martin Mustermann}
\newcommand{\SADABetreuer}{Dipl.-Ing. Rudi Ratlos,}
\newcommand{\SADABetreuerII}{Dipl.-Ing. Hans Hilflos}
\newcommand{\SADABetreuerIII}{}
\newcommand{\SADABegin}{01. Januar 2000}
\newcommand{\SADAAbgabe}{01. Juli 2000}
\newcommand{\SADASeminar}{01. August 2000}
```



---

sinnvoll zu setzen.

Der Fachbereich 16 (Maschinenbau) schreibt je nachdem, ob es sich um eine Studien- oder Diplomarbeit handelt, unterschiedliche Angaben auf der „Erklärungsseite“ vor. Es ist im Hauptdokument eine der drei Zeilen

```
\def\SADAVarianteErklaerung{ETIT}  
%\def\SADAVarianteErklaerung{MBDA}  
%\def\SADAVarianteErklaerung{MBSA}
```

einzukommentieren. (ETIT: „Normal“, MBDA: Fachbereich 16, Diplomarbeit, MBSA: Fachbereich 16, Studienarbeit.)

Je nach Fachgebiet ist der Schalter

```
\rtmtrue    % rtm  
%\rtmfalse  % rtp
```

einzukommentieren.

---

## 5.4 „Entwurfsmodus“

---

Damit man nicht gezwungen ist, das tudesign-Paket und die TU-Schriftarten auf jedem Rechner zu installieren, um mit dem Dokument zu arbeiten, ist es vorgesehen, auch die Klasse scrreprt behelfsweise als Basis zu verwenden. Dazu ist Folgendes zu beachten:

Je nachdem, welche Klasse als Basis ausgewählt ist, muss der Schalter

```
\TUDdesigntrue    % TUD-Design  
%\TUDdesignfalse  % Für Rechner ohne installierte TUDdesign-Pakete
```

passend gesetzt werden. Dadurch wird erreicht, dass das Dokument auch mit scrreprt fehlerfrei umgewandelt wird und die Seitenränder grob stimmen, so dass man einen Eindruck vom späteren Layout bekommt.



---

## 6 Conclusion

Das vorliegende Dokument beschreibt die Formalitäten, die bei der Erstellung von Studien-, Bachelor-, Master- und Diplomarbeiten zu beachten sind. Außerdem gibt es Tipps, worauf bei der Durchführung zu achten ist. Das Dokument entstand durch die Zusammenführung der `iat-sada`-Klasse mit dem TU-Design. Die Tipps und die Beschreibung des  $\text{T}_{\text{E}}\text{X}$ -Systems sind aus der Beschreibung der `iat-sada`-Klasse von Michael Vogt übernommen.



---

# A Checkliste

Von der Arbeit sind zwei gedruckte Exemplare (doppelseitig; gebunden mit schwarzem Karton; vorne Karton oder Klarsichtfolie) für die Bibliothek und den Betreuer abzugeben, sowie eine Datei im *Portable Document Format* (PDF). Dazu sollten folgende Punkte überprüft werden:

## Im Quelltext

- ☐ Wurden alle der Literatur entnommenen Stellen mit Literaturverweisen belegt?
- ☐ keine wörtlichen Zitate, falls doch müssen diese in Anführungszeichen!
- ☐ Alle Literaturangaben im Literaturverzeichnis vollständig? (Autor(en), Titel, Verlag/Journal/Konferenzband/etc., Jahr, usw.)
- ☐ Alle Seitenangaben im Inhaltsverzeichnis korrekt?
- ☐ Keine einzelnen Abschnitte/Unterabschnitte im Literaturverzeichnis?
- ☐ Prägnante Kapitelnamen? (keine ganzen Sätze!)
- ☐ Alle Bilder haben *Bildunterschriften*, alle Tabellen haben *Überschriften*?
- ☐ Wurden die Hinweise aus Kapitel ?? berücksichtigt (allgemeiner Aufbau, Abstände, Sonderzeichen, ...)?
- ☐ Sind die Bilder gut erkennbar und alle Elemente beschriftet? Passt die Schriftgröße in den Bildern zum Text? Sitzen die Gleitobjekte an der richtigen Stelle?
- ☐ Treten beim Aufruf von  $\text{\TeX}$  bzw. pdf $\text{\TeX}$  Fehler oder Warnungen auf? Sind *alle* Bilder, Tabellen und Literaturstellen im Text zitiert? Sind falsche oder doppelte Referenzen vorhanden? Dies lässt sich anhand der Log-Datei feststellen.

## In der PDF-Datei

- ☐ Die Datei ist im **doppelseitigen** Layout mit Hypertext-Elementen zu erstellen; die Optionen `draft`, `oneside` und `nohyperref` sind **nicht** aktiviert. Die Seitengröße des PDF-Dokuments überprüfen: 210 mm × 297 mm (DIN A4).
- ☐ Sind die PDF-Bookmarks und Seitenzahlen vorhanden? Zumindest im Vorspann sollte überprüft werden, ob die Bookmarks auf die richtige Seite verweisen.
- ☐ Sind die PDF-Infofelder (in Acrobat Reader: Datei → Dokumenteigenschaften) richtig eingetragen? Notfalls mit `\hypersetup{...}` korrigieren.
- ☐ In den Bookmarks und Infefeldern können nicht alle Zeichen dargestellt werden. In einem solchen Fall z. B. `\texorpdfstring` (aus `hyperref`) verwenden.

---

## Im Ausdruck

- ☐ Die Arbeit ist **doppelseitig**, vorzugsweise schwarzweiß auf einem Laserdrucker auszu-  
drucken. Sind Alle Grafiken gut zu erkennen (Farbe, Linienstärke, etc.)? Wurden alle  
Sonderzeichen korrekt gedruckt?
- ☐ Beim Ausdrucken aus dem Acrobat Reader darf die Seitenanpassung **nicht** aktiviert sein,  
da der Textblock sonst verkleinert wird. Linker Rand muss 30 mm, rechter Rand muss  
20 mm betragen (nachmessen).
- ☐ Selbstständigkeitserklärung unterschreiben!

---

## B Programme zur Erstellung von Grafiken

---

### B.1 Vektorgrafiken

---

Vektorgrafiken bestehen aus geometrischen Formen, deren Beschreibung unabhängig von der Auflösung ist. Sie sind z. B. für Diagramme und mathematische Plots sinnvoll und lassen sich aus vielen Programmen direkt als EPS oder PDF speichern. Es ist besonders darauf zu achten, dass die Schriften und Strichstärken zum Rest des Dokuments passen.

- $\text{\LaTeX}$  selbst bietet mit den Paketen `TikZ` und `pgfplots` zwei sehr mächtige Werkzeuge um direkt in  $\text{\LaTeX}$  Vektorgrafiken, wie zum Beispiel Blockschaltbilder oder Plots zu erzeugen. Schriftart und -größe sind automatisch identisch mit dem übrigen laufenden Text, so dass hier keine Anpassungen mehr vorgenommen werden müssen.
  - `TikZ` eignet sich hervorragend für Blockschaltbilder
  - Mit `pgfplots` können aus einer `.txt`-Datei mit den Variablenwerten Plots direkt in  $\text{\LaTeX}$  erzeugt werden, so dass sich ein einheitliches Gesamtbild ergibt. Die benötigte `.txt`-Datei lässt sich mit `MATLAB` einfach erzeugen. Achsenbeschriftungen, Legenden, etc. können mit `MATLAB`-ähnlichen Befehlen einfach hinzugefügt werden.

Die Dokumentationen zu `TikZ` [?] und `pgfplots` [?] sind sehr ausführlich und mit vielen Beispielen leicht verständlich erklärt.

- `MATLAB` kann über den Befehl `print -deps name.eps` Grafiken direkt als EPS abspeichern. Allerdings wird die Grafik so skaliert, als würde sie auf einem Drucker ausgegeben. Deshalb müssen unbedingt die Schrift/Linien oder die Größe angepasst werden. Hierzu gibt es verschiedene Techniken. Der direkte PDF-Export aus `MATLAB` ist derzeit noch nicht zu empfehlen.
- Zeichenprogramme können in der Regel ebenfalls EPS direkt exportieren. Der am IAT verwendete Adobe Illustrator erzeugt auch sehr gutes PDF, da sein eigenes Format fast identisch mit PDF ist.
- Bei allen Programmen, die eine Druckfunktion besitzen, kann man die Ausgabe eines beliebigen PostScript-Druckers in eine Datei umlenken. Dazu stellt man den Treiber auf das EPS-Format um, so dass nur noch in den wenigsten Fällen eine Nachbearbeitung der PostScript-Ausgabe z. B. mit `GSview` erforderlich ist.
- Liegt eine Grafik als *Windows Metafile* (`*.wmf`) oder *Enhanced Metafile* (`*.emf`) vor, lässt sie sich mit dem Tool `WMF2EPS` konvertieren. Zahlreiche Windows-Programme bieten eine

---

Export-Möglichkeit in diese beiden Formate, allerdings kann es aufgrund des einfachen Grafikmodells zu Verlusten kommen.

- Arbeitet ein Programm mit der Windows-Zwischenablage zusammen, kann über WMF2EPS eine EPS-Datei direkt aus der Zwischenablage erstellt werden. Für Vektorgrafiken sollte WMF2EPS aber generell nur im Notfall für verwendet werden.

Zur Verwendung in pdf $\TeX$  lässt sich eine Grafik, die im EPS-Format vorliegt, einfach in PDF konvertieren. Hierfür eignen sich der Acrobat Distiller oder Ghostscript in Verbindung mit EPSTOPDF (das in allen gängigen TeX-Distributionen enthalten ist).

---

## B.2 Pixelgrafiken

---

Pixelgrafiken besitzen eine feste Anzahl von Bildpunkten, d. h. ihre Auflösung hängt von der Größe der Darstellung ab. Man benutzt Pixelformate z. B. für Fotos, Screenshots oder eingescannte Grafiken. Hierbei ist die Wahl der Auflösung besonders wichtig. Die Bilder sollen einerseits eine gute Druckqualität ergeben, andererseits aber auch eine zügige Bildschirmdarstellung und kleine Dateigröße ermöglichen. Beim Scannen ist außerdem zu beachten, dass sich die Auflösung ändert, wenn die Grafiken nicht in Originalgröße eingebunden werden.

- Fotos liegen in der Regel im JPEG-Format vor; eine Auflösung von 100–150 dpi ist häufig bereits ausreichend. Während pdf $\TeX$  eine JPEG-Grafik direkt einlesen kann, muss sie für  $\TeX$ /Dvips nach EPS konvertiert werden. Das Kommandozeilen-Tool JPEG2PS erledigt dies, ohne die JPEG-Kompression zu verlieren.
- Sonstige Farb- oder Graustufen-Grafiken (insb. wenn sie „harte“ Farbübergänge besitzen) werden am besten zunächst im PNG-Format gespeichert. Die richtige Wahl der Farbtiefe hat großen Einfluss auf die spätere Dateigröße. Bzgl. der Auflösung gibt es hier keine allgemeine Regel; oft liegt sie bereits fest (z. B. bei Screenshots). pdf $\TeX$  kann PNG-Grafiken direkt verarbeiten, für  $\TeX$ /Dvips müssen sie wieder in EPS konvertiert werden. Dazu kopiert man die Grafik mit einer beliebigen Pixelgrafik-Software in die Zwischenablage und benutzt WMF2EPS.
- Schwarzweiße Strichzeichnungen müssen in relativ hoher Auflösung ( $\geq 300$  dpi) vorliegen, um beim Drucken eine ausreichende Qualität zu gewährleisten. In Fall von  $\TeX$ /Dvips speichert man sie am besten im TIFF-Format mit der sog. *CCITT Group 4* Kompression und konvertiert sie über die Zwischenablage und WMF2EPS in eine EPS-Datei. Da pdf $\TeX$  keine TIFF-Grafiken einlesen kann, ersetzt man sie hier durch PNG oder bettet die TIFF-Grafik in eine PDF-Datei ein (z. B. mit Adobe Acrobat), um die bessere Kompression zu erhalten.



---

## C Das T<sub>E</sub>X-System

Die Klasse `tudreport` unterstützt die Workflows  $\text{L}^{\text{T}}_{\text{E}}\text{X} \rightarrow \text{DVI} \rightarrow \text{PostScript} \rightarrow \text{PDF}$  und  $\text{L}^{\text{T}}_{\text{E}}\text{X} \rightarrow \text{PDF}$ . Dazu sind mehrere Programme nötig, die zusammenfassend als *T<sub>E</sub>X-System* bezeichnet werden. Hinzu kommen noch verschiedene Hilfsprogramme.

---

### Bestandteile des T<sub>E</sub>X-Systems

---

Als Basis wird am IAT derzeit die T<sub>E</sub>X-Distribution MikT<sub>E</sub>X 2.7 unter Windows XP eingesetzt. Darin sind neben dem (pdf)T<sub>E</sub>X-Interpreter und L<sup>T</sup><sub>E</sub>X (samt einer Vielzahl von Zusatzpaketen) auch Programme wie Yap (DVI-Previewer) und Dvips (zum Konvertieren von DVI in PostScript) enthalten.

Zusätzlich benötigt man noch Ghostscript und GSview zum Ansehen von PostScript-Dateien und zum Konvertieren von PostScript und EPS in PDF. Im Hinblick auf die Konvertierung ist vor allem das aktuelle Ghostscript 8.x zu empfehlen. Um die fertige PDF-Datei ansehen zu können, muss schließlich noch ein PDF-Viewer (z. B. Acrobat Reader oder SumatraPDF) installiert sein. Alternativ kann auch das kommerzielle Acrobat-Paket verwendet werden, das mit dem Acrobat Distiller eine sehr gute Möglichkeit zum Konvertieren von PostScript/EPS in PDF enthält.

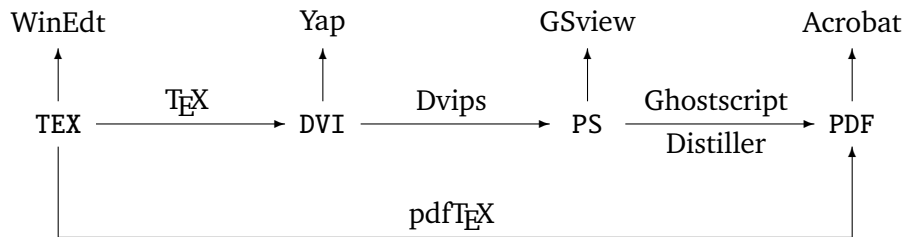
Um die Bedienung der einzelnen Teilprogramme zu erleichtern, kommen spezielle T<sub>E</sub>X-Shells zum Einsatz. Diese besitzen neben einem komfortablen Editor (Rechtschreibprüfung, Syntax-Highlighting, ...) Bedienelemente zum Aufruf von L<sup>T</sup><sub>E</sub>X, Dvips usw. Am IAT ist auf den meisten Rechnern das Shareware-Programm WinEdt installiert, aber auch das kostenlose T<sub>E</sub>XnicCenter ist inzwischen sehr gut. Eine weitere Möglichkeit ist der Editor Emacs mit der Erweiterung AUCT<sub>E</sub>X. Die letzte Komponente sind die beiden Grafik-Tools JPEG2PS und WMF2EPS aus Abschnitt 4.5.

---

### Umgang mit dem T<sub>E</sub>X-System

---

Zu Beginn der Arbeit ist die prinzipielle Entscheidung zu treffen, ob man T<sub>E</sub>X/Dvips oder pdfT<sub>E</sub>X verwendet. Bild C.1 zeigt die Zusammenhänge. Während der T<sub>E</sub>X-Interpreter sein eigenes Format DVI liefert, das dann weiter in PostScript und PDF konvertiert wird, ist pdfT<sub>E</sub>X eine neue T<sub>E</sub>X-Variante, die direkt PDF produziert. Beide Wege funktionieren problemlos; aus Sicht des Anwenders unterscheiden sie sich in erster Line beim Einbinden von Grafiken, siehe Tabelle C.1 und Abschnitt 4.5. T<sub>E</sub>X/Dvips ist von Vorteil, wenn viele Grafiken als EPS vorliegen. Für pdfT<sub>E</sub>X entscheidet man sich, wenn vorwiegend Pixel-Bilder und PDF-Grafiken (z. B. aus Adobe Illustrator) vorhanden sind.



**Abbildung C.1.:** TeX-Workflows und beteiligte Programme

**Tabelle C.1.:** Einbinden von Grafiken in TeX-Dokumente.

Format	TeX/Dvips	pdfTeX
EPS	(direkt)	Acrobat, EPSTOPDF
PDF	Acrobat, Ghostscript	(direkt)
JPEG	JPEG2PS	(direkt)
PNG	WMF2EPS	(direkt)
TIFF	WMF2EPS	über PDF

Im Fall von TeX/Dvips reicht es während der Erstellung des Dokuments in der Regel aus, eine DVI-Datei zu erzeugen. Gelegentlich sollte man diese aber auch in PostScript und weiter in PDF konvertieren und das Ergebnis überprüfen. Bei pdfTeX arbeitet man ohnehin nur mit der PDF-Datei.

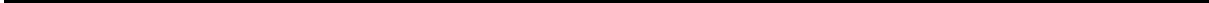
Beim Schreiben ist es manchmal hilfreich, wenn man die Option `draft` verwendet. Statt der Bilder werden dann nur Rahmen der gleichen Größe angezeigt, und es sind alle Stellen markiert, an denen der Text über den Rand hinausragt. Solange man an der Arbeit schreibt, empfiehlt es sich, bei den Umgebungen `figure` und `table` keine Positionsangaben zu verwenden (insb. nicht `[h]` oder gar `[h!]`). Erst *ganz am Ende* überprüft man die Platzierung und ändert sie bei Bedarf durch Positionsangaben, durch Verschieben im Quelltext oder mit Hilfe des `flafter`-Pakets. In jedem guten L<sup>A</sup>TeX-Buch ist außerdem beschrieben, wie man den Mechanismus zur Positionierung für das gesamte Dokument anpassen kann.

Häufig tritt auch die Frage auf, wie oft das Dokument mit L<sup>A</sup>TeX bearbeitet werden muss, damit alle Bezüge und Referenzen stimmen. L<sup>A</sup>TeX legt während der Bearbeitung Dateien (`*.aux`, `*.toc`, `*.lof`, usw.) an, in denen Informationen über die Gliederung, Bilder, Tabellen usw. abgelegt sind. Bei einem erneuten Aufruf von L<sup>A</sup>TeX werden diese ausgelesen und die Informationen an den entsprechenden Stellen eingesetzt. Daraus ergeben sich folgende Regeln:

- 1× bearbeiten bei kleineren Änderungen.
- 2× bearbeiten, falls sich die Referenzen verändern, d. h. wenn neue Gleichungen oder Bilder eingefügt werden oder wenn sich Seitenumbrüche verschieben.
- 3× bearbeiten, wenn sich ein Verzeichnis (Inhalt, Bilder, ...) um eine Seite verlängert, weil dann die Seitenzahlen im Verzeichnis nicht mehr stimmen.

---

Da während des Schreibens häufig nur kleine Änderungen zwischen zwei Durchläufen vorgenommen werden, genügt meist der einmalige Aufruf von  $\text{\LaTeX}$ . Spätestens bei der nächsten Iteration stimmen die Bezüge wieder. Bei der Benutzung von  $\text{\BibTeX}$  sollte  $\text{\LaTeX}$  *vorher* 1× und *nachher* 2× aufgerufen werden, damit alle Zitate stimmen.



---

## D Befehle in commonmacros.tex

Hier sind im Folgenden kurz die in `commonmacros.tex` definierten Befehle aufgelistet.

---

### Einheiten

---

Die folgenden Befehle funktionieren im Mathe-und Textmodus (d.h. es wird im Textmodus automatisch für den Befehl in den Mathemodus umgeschaltet):

- Einheit (Aufrechte Schrift im Mathemodus)  
 $\text{\unit{\frac{N}{m}}} \rightarrow \frac{N}{m}$
- Zahl mit Einheit  
(Setzt „kleines“ Leerzeichen zwischen Zahl und Einheit, Zahl und Einheit automatisch im Mathemodus, Einheit in aufrechter Schrift)  
 $\text{\valunit{34,3}{cm}} \rightarrow 34,3\text{ cm}$
- (Das aufrechte  $\mu$  gibt es mit dem Befehl `\upmu` aus dem Paket `upgreek`)  
 $\text{\valunit{4}{\upmu m}} \rightarrow 4\mu\text{m}$

Besondere Einheiten

- Gradzeichen (Funktioniert im Text- und Mathemodus)  
 $\text{\degree} \rightarrow ^\circ$
- Grad Celsius (Funktioniert im Text- und Mathemodus)  
 $\text{\degC} \rightarrow ^\circ\text{C}$

---

### Vektoren und Matrizen

---

- Vektor  
 $\text{\ve{x}} \rightarrow \mathbf{x}$
- Matrix  
 $\text{\ma{A}} \rightarrow \mathbf{A}$
- Vektor Sonderzeichen  
 $\text{\ves{\lambda}} \rightarrow \boldsymbol{\lambda}$
- Matrix Sonderzeichen  
 $\text{\mas{\Lambda}} \rightarrow \mathbf{\Lambda}$

Wichtig: Mathematische Akkzente müssen dabei geklammert werden!

- $\dot{\ve{x}} \rightarrow \dot{x}$
- $\dot{\tilde{\ve{x}}} \rightarrow \dot{\tilde{x}}$

$\ve{}$  und  $\ma{}$  bzw.  $\ves{}$  und  $\mas{}$  machen jeweils genau das gleiche. Die Unterscheidung dient nur zur besseren Lesbarkeit.

- Transponiert-Zeichen (aufrechtes T)  
 $\ma{A}^{\transp} \rightarrow A^T$

---

## Funktionen und Abkürzungen

---

- Unterstreichen  
 $\ul{x} \rightarrow \underline{x}$
- Innenprodukt  
 $\inprod{f}{g} \rightarrow \langle f, g \rangle$
- Exponentialschreibweise  
 $45\E{-2} \rightarrow 45 \cdot 10^{-2}$
- e-Funktion  
 $\eexp{t} \rightarrow e^t$
- Rang  
 $\rang{\ma{A}} \rightarrow \text{rg}(A)$
- Imaginäre Einheit (aufrechtes j)  
 $5+\iu\,2 \rightarrow 5 + j2$
- „Von-Bis-Punkte“ mit Kommas und schönen Abständen  
 $1 \,\,\text{todots} \,\, n \rightarrow 1, \dots, n$
- i abgeleitet  
 $\dot{i} \rightarrow (i)'$
- Aufrechte Schrift (Abkürzung für  $\mathrm{}$ )  
 $\mrm{abc} \rightarrow \text{abc}$
- Normaler Text in Formel (Abkürzung für  $\textnormal{}$ )  
 $\tn{ab \,\,\text{für}} \rightarrow \text{ab für}$
- Geklammerte Gruppe mit Subscript  
 $\grpsb{\frac{1}{2}}{x} \rightarrow \left(\frac{1}{2}\right)_x$
- Geklammerte Gruppe mit aufrechtem Subscript  
 $\grprsb{\frac{1}{2}}{x} \rightarrow \left(\frac{1}{2}\right)_x$

---

## Ableitungen und Integrale

---

- Normale Ableitung  
 $\text{\normd{f}{x}} \rightarrow \frac{df}{dx}$
- Materielle Ableitung  
 $\text{\matd{f}{x}} \rightarrow \frac{Df}{Dx}$
- Partielle Ableitung  
 $\text{\partialald{f}{x}} \rightarrow \frac{\partial f}{\partial x}$
- Beispiel höhere Ableitung  
 $\text{\normd{^2 f}{x^2}} \quad \text{\quad} \text{\partialald{^2 f}{x \partial y}} \rightarrow \frac{d^2 f}{dx^2} \quad \frac{\partial^2 f}{\partial x \partial y}$
- Normale Ableitung an  
 $\text{\normdat{f}{x}{x=0}} \rightarrow \left. \frac{df}{dx} \right|_{x=0}$
- Materielle Ableitung an  
 $\text{\matdat{f}{x}{x=0}} \rightarrow \left. \frac{Df}{Dx} \right|_{x=0}$
- Partielle Ableitung an  
 $\text{\partialalddat{f}{x}{x=0}} \rightarrow \left. \frac{\partial f}{\partial x} \right|_{x=0}$
- Aufrechtes „d“ für Integral  
 $\text{\ud} \rightarrow d$
- Beispiel für Integral  
 $\text{\int f(x) \ud x} \rightarrow \int f(x) dx$

---

## Transformationen

---

- $\text{\Laplace{x}} \rightarrow \mathcal{L}(x)$
- $\text{\InvLaplace{X}} \rightarrow \mathcal{L}^{-1}(X)$
- $\text{x \trans X} \rightarrow x \quad \circ \text{---} \bullet \quad X$
- $\text{X \invtrans x} \rightarrow X \quad \bullet \text{---} \circ \quad x$
- $\text{\FT{x}} \rightarrow \mathfrak{F}\{x\}$
- $\text{\FTabs{x}} \rightarrow |\mathfrak{F}\{x\}|$
- $\text{\IFT{x}} \rightarrow \mathfrak{F}^{-1}\{x\}$
- $\text{\DFT{x}} \rightarrow \text{DFT}\{x\}$
- $\text{\DFTabs{x}} \rightarrow |\text{DFT}\{x\}|$

---

## Matlab/Simulink

---

- `\mlfct{abc}` → abc
- `\mlvar{abc}` → abc

---

## Verweise

---

Verweise auf verschiedene Objekte mit passendem Text („Abbildung X“, „Tabelle X“). Dabei ist dann immer der komplette Text ein Hyperlink, und nicht nur die Zahl.

- Abbildung  
`\figref{label}`
- Tabelle  
`\tabref{label}`
- Gleichung  
`\equref{label}`
- Definition  
`\defref{label}`
- Kapitel  
`\charef{label}`
- Abschnitt  
`\secref{label}`
- Listing  
`\lstref{label}`
- Seite  
`\pageref{label}`

Z. T. auch auf Varioref basierend („Abbildung 23 auf dieser Seite“, „Abbildung 23 auf Seite 45“)

- Abbildung  
`\figvref{label}`
- Tabelle  
`\tabvref{label}`
- Gleichung  
`\equivref{label}`



---

## Abkürzungen

---

Abkürzungen mit Punkt „dazwischen“ (wird mit kleinen Abständen gesetzt)

`\dah` → d. h., `\Dah` → D. h., `\iA` → i. A., `\IA` → i. A., `\ua` → u. a., `\Ua` → U. a., `\uU` → u. U., `\UU` → u. U., `\zB` → z. B., `\ZB` → Zum Beispiel, `\zT` → z. T., `\ZT` → Z. T.

Abkürzungen mit Punkt, bei denen der Punkt nicht als Satzende interpretiert wird:

`\bspw` → bspw., `\Bspw` → Bspw., `\bzw` → bzw., `\Bzw` → Bzw., `\bzgl` → bzgl., `\ca` → ca., `\evtl` → evtl., `\ggf` → ggf., `\Ggf` → Ggf., `\usw` → usw., `\vgl` → vgl., `\Vgl` → Vgl.

---

## Mathe-Umgebungen

---

- **theorem**  
„Satz“, selber Zähler wie lemma (Lemma)
- **lemma**  
„Lemma“, selber Zähler wie theorem (Satz)
- **definition**  
„Definition“, eigener Zähler
- **example**  
„Beispiel“, eigener Zähler, größerer linker Rand

```
\begin{theorem}
  Beispiel für Theorem
\end{theorem}
```

**Satz 1** *Beispiel für Theorem*

```
\begin{lemma}
  Beispiel für Lemma
\end{lemma}
```

**Lemma 2** *Beispiel für Lemma*

```
\begin{definition}
  Beispiel für Definition
\end{definition}
```

**Definition 1** *Beispiel für Definition*

```
\begin{example}
```

---

```
    Beispiel für Beispiel
\end{example}
```

### Beispiel D.1:

Beispiel für Beispiel

```
\begin{example}[Test]
    Beispiel für Beispiel mit "'Namen'"
\end{example}
```

### Beispiel D.2: Test

Beispiel für Beispiel mit „Namen“

---

## Listingdefintionen

---

- Matlab\_colored
- Matlab\_colored\_smallfont

Verwendung:

```
\begin{lstlisting}[style=Matlab_colored, %
    caption = {Beispiellisting, style=Matlab\_colored}, %
    label={lst:Listing1}]
    [...]
\end{lstlisting}
```

---

### Listing D.1: Beispiellisting, style=Matlab\_colored

---

```
1 function [] = animierePunkt(inY, inX)
```

```
    temp = length(inY);
```

```
    %% [...]
```

```
6
```

```
    %% -----
```

```
    for i=1:temp
```

```
        if i>1
```

---

```

        delete(p(i-1));
11    end
    p(i) = plot(inX(i),inY(i),'Marker','o','MarkerSize',10);
        pause(0.025);
end
hold off;

```

---

```

\begin{lstlisting}[style=Matlab_colored_smallfont, %
    caption = {Beispiellisting, style=Matlab\_colored\_smallfont}, %
    label={lst:Listing2}]
    [...]
\end{lstlisting}

```

---

#### Listing D.2: Beispiellisting, style=Matlab\_colored\_smallfont

---

```

function [] = animierePunkt(inY, inX)

temp = length(inY);

5  %% [...]

%% -----
for i=1:temp
    if i>1
10     delete(p(i-1));
        end
        p(i) = plot(inX(i),inY(i),'Marker','o','MarkerSize',10);
        pause(0.025);
    end
15 hold off;

```

---



---

## Sonstiges

---

Latex gibt beim Umwandeln z. T. Fehler aus, wenn Zeichen aus dem textcomp-Paket verwendet werden, da diese nicht in den TU-Schriften vorhanden sind. Mit `\textcompstdfont{}` wird die Schriftart für den Text im Argument explizit umgeschaltet, und so der Fehler vermieden:

- `\textcompstdfont{\textuparrow}` → ↑

