
Metrology Lecture Book

Katharina-S. Isleif

Sep 23, 2022

CONTENTS

I Part	5
1 Über dieses Skript	7
1.1 Vorlesungsinhalte	7
1.2 Praktikum	7
1.3 Tutorials	8
1.4 Programmieren mit Python	8
1.5 Quick introduction to Jupyter Notebook	8
2 Infos zum Praktikum	9
2.1 Grundidee: Messdaten sammeln	9
2.2 Diagramme zeichnen	10
II Grundlagen	13
3 Messtechnik	15
3.1 Messen und Einheiten	15
4 Messdaten analysieren	29
4.1 Zusammenfassung	29
4.2 Quellen und Ursachen von Messunsicherheiten	30
4.3 Statistische Größen	35
4.4 Fortpflanzung von Messunsicherheiten	46
4.5 Kurvenanpassung	52
III Messsignale und -systeme	61
5 Stationäre Messsysteme	63
5.1 Grundstruktur eines Messsystems	63
5.2 Ideale Kennlinie	64
5.3 Reale Kennlinie	69
6 Messsignale	79
6.1 Grundlagen	79
6.2 Kenngrößen	85
6.3 Modulation von Messsignalen	88
6.4 Digitalisierung	95
6.5 Fourier-Analyse	98
6.6 Rauschen und spektrale Dichte	107

7 Dynamische Messsysteme	109
7.1 Linear zeitinvariante (LZI) Systeme	109
7.2 Differentialgleichung	111
7.3 Impuls- und Sprungantwort	113
7.4 Übertragungsfunktion	117
7.5 Faltung und Korrelation	125
IV Sensoren	127
8 Messen elektrischer Größen	129
8.1 Elektronische Minimum	129
8.2 Spannungsmesseingang	129
8.3 Strommesseingang	129
8.4 Operationsverstärker	129
8.5 Leistungsmessung	129
8.6 Widerstandsmessung	129
8.7 Frequenzmessung	129
9 Sensoren	131
9.1 Einführung	131
9.2 Dehnungsmessstreifen	131
9.3 Piezoelektrische Sensoren	131
9.4 Kapazitive Sensoren	131
9.5 Drucksensoren	131
9.6 Temperaturmessungen	131
9.7 Messung von Wärmestrahlung	131
9.8 Optische Sensoren	131
V Tutorials	133
10 Jupyter Notebooks	135
10.1 Klimadatei (.csv) einlesen und analysieren	135
10.2 Fehlerfortpflanzung	141
10.3 Vergleich verschiedener Fit-Funktionen	146
10.4 Lineare Regression mit Berücksichtigung von Fehlerbalken	153
10.5 Fourier-Transformation: Entfernung von Hintergrundrauschen in einem Audiosignal	162
Bibliography	173

Willkommen zur Vorlesung Messtechnik an der HSU. Auf den folgenden Seiten findet ihr begleitende Informationen zur Vorlesung und zum Praktikum.

Warning: Dieses *Metrology Lecture Book* befindet sich aktuell im Aufbau und ihr könnt euch noch nicht auf Vollständigkeit verlassen.

See also:

Weitere Infos zum [Lecture Book](#).

Tips für's Praktikum

Messdaten richtig sammeln, dokumentieren und analysieren.

Über dieses Skript

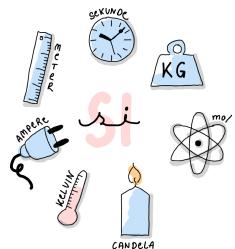
Einführung in Python und Jupyter-Notebooks.

Tutorials

Übungen zur Messtechnik mit Jupyter-Notebooks.

Vorlesungsinhalt

Messen, Einheiten



SI-Einheiten, Nicht-SI, Kalibrieren, Eichen, Prüfen

Messunsicherheiten



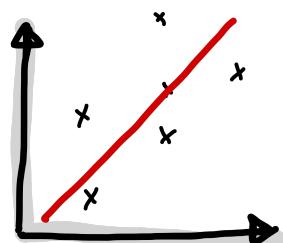
Zufällige
Abweichung



Systematische
Abweichung

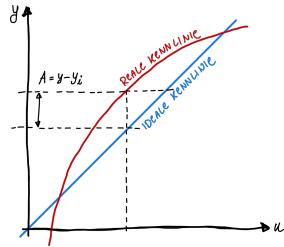
Systematisch, zufällig, Vertrauensintervall, Normalverteilung, Fehlerfortpflanzung

Kurvenanpassung



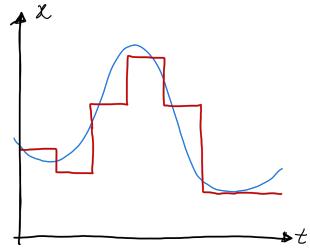
Regression, Fit, Korrelation, Kovarianz, Least-Squares

Stationäre Messsysteme



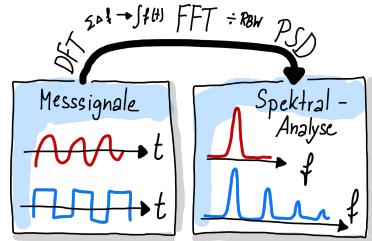
Ideale und reale Kennlinie

Messsignale



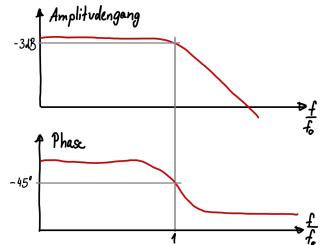
Signale als Informationsträger, Digitalisierung, Kenngrößen

Fourier-Analyse



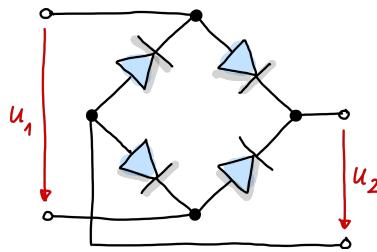
Fourier-Reihen, Fourier-Transformation

Dynamische Messsysteme



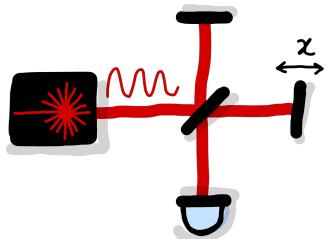
LZI-Systeme, DGLs, Impuls- und Sprungantwort, Faltung, Übertragungsfunktion

Messen elektrischer Größen



Messeingänge, Widerstandsbrücken, Wechselspannung, Messverstärker

Sensoren



optisch, kapazitiv, piezo-elektrisch, resistiv, magnetisch, gravitativ

Literatur

Parthier 2020 [Par20]



Messtechnik: Vom SI-Einheitensystem über Bewertung von Messergebnissen zu Anwendungen der elektrischen Messtechnik

Explore this book

Léon 2019 [PL19]



Messtechnik: Grundlagen, Methoden und Anwendungen

Explore this book

Part I

Part

CHAPTER
ONE

ÜBER DIESES SKRIPT

Das *Metrology Lecture Book* ist eine Mischung aus Vorlesungsskript, Praktikumshinweisen und Tutorials als Jupyter-Notebooks. Es handelt sich hierbei um ein Jupyter-Book was es ermöglicht Python-Code direkt einzubinden.

1.1 Vorlesungsinhalte

Um dem **Vorlesungsskript** zu folgen sind **keine Python-Kenntnisse erforderlich**.

- Den **Code** zu den Berechnungen und Plots sind meist versteckt und können über einen Klick auf dein *Pfeil* angezeigt werden.
- **Anmerkungen, Textmarker und Vorlesungsnotizen** könnt ihr direkt auf der Seite machen (oben ganz rechter Seitenrand). Hierfür müsst ihr euch unter <https://hypothes.is/> registrieren. Möchtet ihr die Notizen mit mir oder der Jahrgangs-Gruppe teilen, dann müsst ihr der entsprechenden Hypothes-Gruppe beitreten. Dadurch könnt ihr direkt Fragen zum Skript stellen (und Antworten bekommen) oder auf Fehler hinweisen. Den Link für eure Jahrgangsgruppe findet ihr im **ILIAS**.

1.2 Praktikum

Für das **Praktikum** müsst ihr *Daten aufnehmen, auswerten und grafisch darstellen*. Hierfür könnt ihr prinzipiell ein Programm eurer Wahl nehmen, wie z.B.

- Matlab (Lizenzen sind über die HSU erhältlich)
- Python, frei-erhältlich und die *Tutorials* im *Lecture Book* nutzen Python (Jupyter-Notebooks).
- QtiPlot, freil erhältlich
- Von einer Datenanalyse in Excel rate ich ab.

Im *Metrology Lecture Book* findet ihr jedoch viele Beispiele zur Analyse und Darstellung von Messdaten in python.

1.3 Tutorials

Die Tutorials sind freiwillig. Hiermit lernt ihr aber nicht nur die Auswertung und Darstellung von Messdaten, sondern gleichzeitig auch die Programmiersprache Python.

1.4 Programmieren mit Python

Es gibt zahlreiche Tutorials und Lernseiten über Python im Internet, womit ihr Python-Programmierung schnell und einfach lernen könnt. Für die Tutorials benötigt ihr folgende Grundkenntnisse, bzw. lernt sie innerhalb der Tutorials:

- **Basics:** Schleifen, Funktionen, Listen, Tuples, Sets, Dictionaries
- **Algebra:** `numpy`, `scipy`
 - **Fit/Modellanpassungen:** `lmfit`, `scipy` (z.B. `curve_fit`), `numpy` (z.B. `polyfit`)
 - **Spektralanalyse:** `scipy` (z.B. `rfft`)
- **Datenverarbeitung:** `pandas` (z.B. `DataFrames`)
- **Datenvisualisierung:** `matplotlib`

Die einfachste Möglichkeit mit der Python Programmierung zu starten ist das ausführen von Jupyter-Notebooks.

1.5 Quick introduction to Jupyter Notebook

Es gibt verschiedene Möglichkeiten Jupyter-Notebooks auszuführen.

- **Ohne Installation: Code direkt im Browser ausprobieren:**
 - **Jupyter-Lab** sofort ausprobieren: <https://jupyter.org/try-jupyter/lab/>. Um einzelne *Zellen* auszuführen macht einen *Doppelklick* in den Abschnitt und führt diesen mit Shift+Enter aus.
 - **Jupyter-Notebooks** sofort ausprobieren: <https://jupyter.org/try-jupyter/retro/notebooks/?path=notebooks/Intro.ipynb>. Um einzelne *Zellen* auszuführen macht einen *Doppelklick* in den Abschnitt und führt diesen mit Shift+Enter aus.
 - **Tutorials** im *Lecture Book* sofort ausprobieren: klickt oben auf die *Rakete* und dann auf *Colab*. Diese ist eine weitere interaktive Umgebung, in der das Jupyter-Notebook gestartet wird. Die *Rakete* erscheint nur auf den Seiten, die als Jupyter-Notebook hinterlegt sind.
- **Installation auf dem Rechner:**
 - ich empfehle die Installation von Python/Jupyter-Notebooks/Jupyter-Lab über `Anaconda`
 - außerdem gibt es neuerdings eine Desktop-Version von `Jupyter-Lab`, die es erlaubt `.ipynb` Dateien ganz einfach per Doppelklick aus eurem lokalen Datei-System zu öffnen.

CHAPTER
TWO

INFOS ZUM PRAKTIKUM

In den Vorlesungsunterlagen findest du Infos zu den Grundlagen des Messtechnik, die auch für das Praktikum gelten. Folgende Fragen solltest du die vor deinem ersten Praktikumsversuch einmal beantworten:

- **Einheiten:** Welche Maßeinheiten gibt es? Was sind SI-Einheiten? Welche Vorsätze gibt es?
- **Messunsicherheiten:** Welche Arten von Unsicherheiten gibt es? Welche statistischen Methoden gibt es um Messreihen auszuwerten? Wie gebe ich Messunsicherheiten an (Messwert und Fehlerbalken)?
- **Fehlerfortpflanzung:** Wie betreibe ich Fehlerfortpflanzung? Was muss ich bei systematischen, zufälligen und korrelierten Unsicherheiten beachten?
- **Kurvenanpassung:** Wie plotte ich Messdaten sinnvoll? Wie kann ich Zusammenhänge in den Daten darstellen?

2.1 Grundidee: Messdaten sammeln

Aufgabe der Messtechnik ist es physikalische Messgrößen quantitativ zu beobachten und ist somit wichtiger Bestandteil in der Physik. Eine *quantitative* Beschreibung bedeutet immer, dass eine Messgröße mit einem Zahlenwert, x , und einer Maßeinheit, E , ausgedrückt wird. Um einen möglichst *guten* Zahlenwert experimentell zu ermitteln, wird neben der konkreten Durchführung ein Großteil der Experimentierzeit damit verbracht das Experiment vorzubereiten und zu planen, aber auch die Daten auszuwerten und darzustellen, Ergebnisse zu überprüfen und ggf. Messungen zu wiederholen. Was sich beim Messen nicht umgehen lässt, und was der ein oder andere sicherlich schon im Praktikum beobachten konnte, ist, dass Beobachtungen immer statistischen (zufälligen) Schwankungen unterliegen. Dies führt dazu, dass sich das Messergebnis immer verändert. eine Messung wird prinzipiell niemals den *wahren* Wert liefern können, weshalb wir sorgfältig messen und auswerten müssen. Es gibt *best practice* Methoden und Techniken, die in der Wissenschaft benutzt werden, um Unsicherheiten und Schwankungen der Messgröße quantitativ zu beschreiben um so ein Qualitätsmaß der Messung bzw. unseres Experiments zu erhalten. Durch die Einhaltung der *best practice* Methoden kannst du jederzeit Rechenschaft ablegen und das Ergebnis begründen. Die grundlegende Norm für die Messtechnik ist in der *DIN-Norm DIN 1319* und dem *GUM* (Guide to the Expression of Uncertainty in Measurement) offiziell festgehalten.

- **Planung:** Was soll gemessen werden? Was wird hierfür benötigt? Welche Fehlerquellen/Störeinflüsse könnten auftreten, bzw. welche sind bekannt? Welche systematischen Unsicherheiten sind bekannt?
- **Durchführung:** Führe Protokoll! Wurde **alles**, was wichtig sein könnte, protokolliert und in Tabellen zusammengefasst, aufgeschrieben, fotografiert?
- **Auswertung:** Prüfe die Ergebnisse auf Vollständigkeit und Übersichtlichkeit! Hierzu gehört auch eine vollständige Abschätzung von Messunsicherheiten.
- **Prüfung:** Ergeben die Ergebnisse Sinn und sind diese konsistent mit anderen Ergebnissen aus der Literatur? Haben wir die Ergebnisse erwartet?
- **Darstellung:** Vollständige Angabe des Messergebnisses, bestehend aus Zahlenwert, Maßeinheit und Messunsicherheit. Verwende die wissenschaftliche Notation für Zehnerpotenzen.

See also:

In den Vorlesungsunterlagen findest du Infos zu *Messen, Einheiten, Kalibrieren, Quellen und Ursachen von Messunsicherheiten, statistischen Messunsicherheiten* und *Fehlerfortpflanzung*.

2.2 Diagramme zeichnen

Mittels Diagrammen (engl. *Plots*) werden Messwerte dargestellt. Die folgenden Regeln helfen dabei, dass die Diagramme anschaulich sind und der Betrachter direkt erkennt, worum es geht:

- **Achsenbeschriftung:** Beschrifte die Achsen richtig, eindeutig und mit vollständiger Angabe: physikalische Größe und Maßeinheit!
- **Skalierung:** Wähle eine passende Skalierung in 1er-/2er-/5er oder 10er- (Dakaden) Schritten
- **Markierungen:** Wähle eine gute erkennbare Markierung für Messpunkte und ggf. eine angebrachte Linienbreite für Kurven. Hierbei können unterschiedliche Farben, Strichdicken, Stricharten und Markierungspunkte verwendet werden, oder eine Kombination.
- **Titel:** Nutze passende Über- / oder Unterschriften für das Diagramm, insbesondere wenn diese in der Auswertung im Text erwähnt werden.
- **Anderes:** Weitere Punkte und Linien, die nicht gemessen wurden, sondern nur als *Hilfe* dienen (z.B. Fit-Funktionen, Modelle, Referenzlinien) oder Kommentare sind, sollten besonders gekennzeichnet werden.
- **Messunsicherheiten:** Für Messwerte (in Form von Fehlerbalken), aber auch für Funktionsterme und Ausgleichsgeraden, müssen Messunsicherheiten in den Graphen angegeben werden.

Verwende eine sinnvolle Software für die Datenanalyse und die grafische Darstellung, welche auch Fit-Analysen unterstützen. Wir werden hier im folgenden Beispiele in `python` aufführen und Jupyter Notebooks verwenden. Weitere Software ist Qti-Plot (kostenlos) oder Matlab (Lizenzen über HSU verfügbar).

See also:

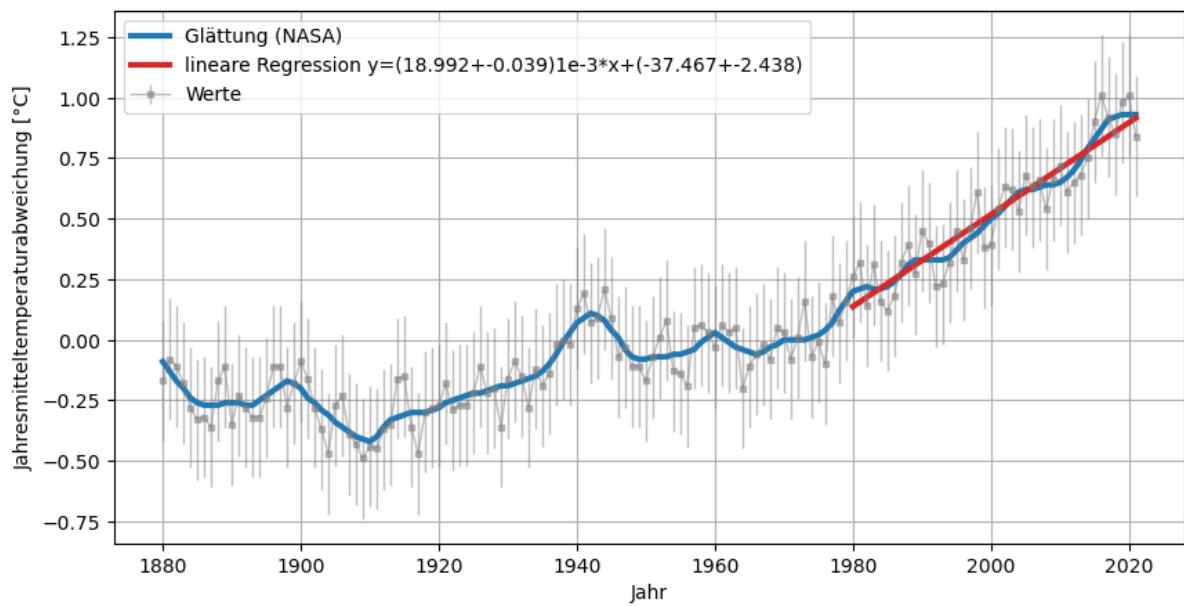
Das Einlesen und darstellen von Messdaten in `python` lernst du z.B. in diesem [Tutorial](#) anhand von echten Klimadaten der NASA. Die Codes auf dieser Seite können direkt benutzt und sogar ausgeführt werden.

Im Folgenden sind zwei Diagramme dargestellt, die jeweils die gleichen (!) Klima-Messdaten der NASA zeigen. Das zweite Diagram weist einige Defizite: Aufgrund der Darstellung wurde eine lineare Regression über den kompletten Messwertebereich durchgeführt, wodurch der Temperaturanstieg der letzten Jahre um einen Faktor 2 zu gering abgeschätzt wurde!

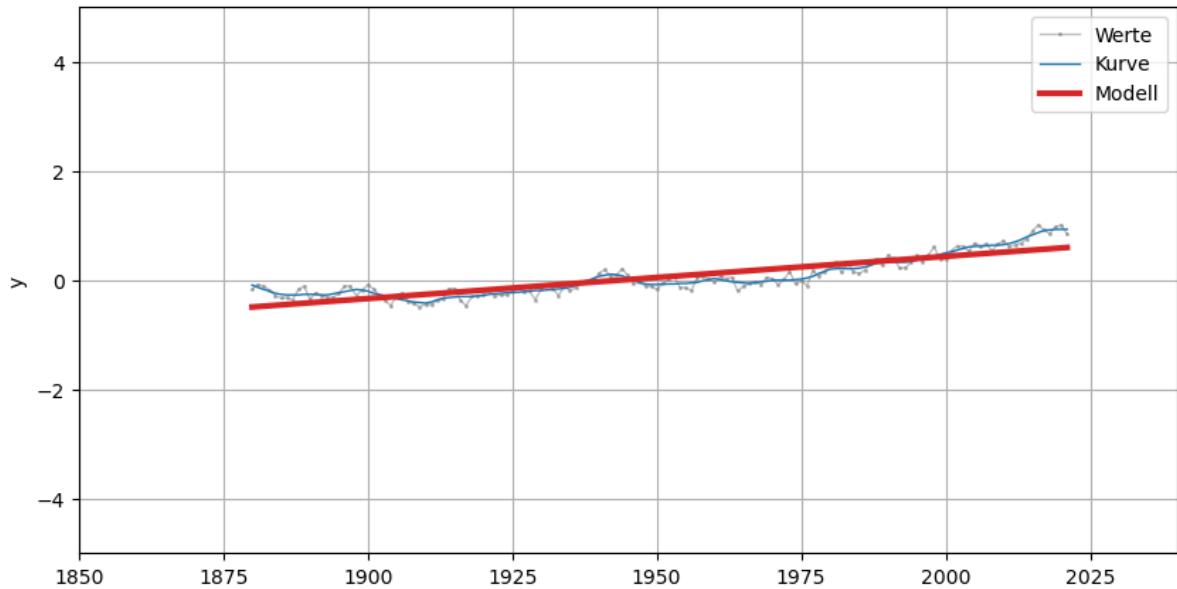
Aufgabe

Welche anderen Defizite fallen dir im Vergleich zum ersten Bild auf?

Temperaturanstieg pro Jahr (von 1980 bis 2020): $0.019^{\circ}\text{C/Jahr}$



Temperaturanstieg pro Jahr (von 1980 bis 2020): $0.008^{\circ}\text{C}/\text{Jahr}$



See also:

Wie du eine Kurve an deine Messdaten anpasst findet du unter [Kurvenanpassung](#) und [Übungen in Python](#).

Warning: Alle Beispiele und Tutorials im *Metrology Lecture Book* benutzen python.

Part II

Grundlagen

MESSTECHNIK

Was ist die Aufgabe der Messtechnik? Wozu braucht man sie? Wo taucht sie auf? Wer nutzt sie? Im Alltag findet man die Messtechnik überall, zum Beispiel im Handy, bei Temperatur- oder Stickstoffmessungen in der Umwelt oder in der Prozess- und Fertigungstechnik. Es bedarf einer bestimmten Struktur und Vorgehensweise, um physikalische oder chemische Größen zu beschreiben und auswerten zu können. Wir werden die Fragen beantworten, was die Eigenheiten und Einheiten bestimmter physikalischer Messgrößen sind, welches Messgeräte für diese existieren und was bei der Anwendung beachtet werden muss.

Heutzutage bestehen messtechnische Lösungen fast ausschließlich aus elektronischen Systemen. Häufig werden eigenständige Messgeräte oder elektronische Messmodule für den PC benutzt, welche stets elektronische Bauelemente und Schaltungen nutzen. Deshalb werden wir uns insbesondere auch mit der Messung von elektrischen Kenngrößen (Spannungen, Ströme, Leistungen, Widerstände, Kapazitäten, Induktivitäten) beschäftigen werden.

Für das Messen von nicht-elektrischer Größen - was in der Mehrheit der Anwendungen von Interesse ist - werden in der Messtechnik zusätzliche Sensoren eingesetzt. Ein Sensor wandelt nach einem bestimmten physikalischen (oder chemischen) Funktionsprinzip die nicht-elektrische Größe in ein elektrisches Signal um, das mit elektronischen Schaltungen weiterverarbeitet werden kann. Beispiele für über Sensoren erfassbare nicht-elektrische Größen sind beispielsweise Temperatur, Druck, Feuchte, Durchfluss, Weg, Winkel, Kraft, Druck, Beschleunigung, CO₂-Konzentration, Schalldruck etc.

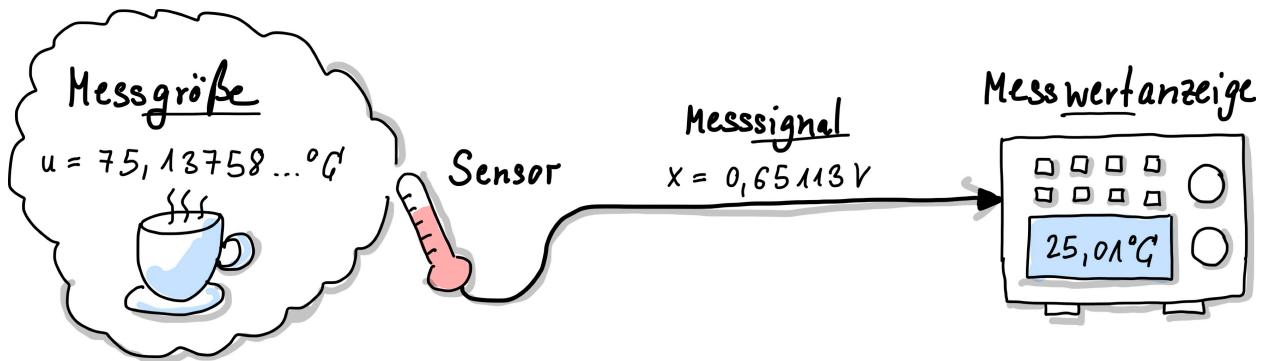
- *Messen und Einheiten*

3.1 Messen und Einheiten

Ein Messobjekt hat eine bestimmte Messgröße (physikalische Größe, Temperatur, Windstärke, ...) von welcher der Messwert bestimmt werden soll. Hierfür wird ein Verfahren benötigt, um die Größe zu extrahieren, was durch ein passendes Messgerät geschieht.

Im Bild wird das Beispiel einer Temperaturmessung gezeigt. An einem bestimmten Ort herrschende **Messgröße**, hier 24,1286941... °C, wird mittels eines geeigneten Aufbaus in einen **Messwert** von 25,01°C überführt. Der Messwert kann uns direkt angezeigt werden oder er kann in nachfolgenden elektronischen Systemen zur Weiterverarbeitung in geeigneter Form zugeführt werden.

Von einem **Messsignal**, x_1 (im Gegensatz zur Begrifflichkeit *Messgröße*) spricht man, wenn direkt mit der Messgröße zusammenhängende elektrische Signale zwischen den beiden Stellen, an denen Messgröße und Messwert anfallen, gemessen werden (zum Beispiel das Kabel in der Skizze). Messsignale tragen die Information über die Messgröße, welche auf unterschiedlichste Weise realisiert werden z.B. als analoger Spanungs- oder Stromwert, als frequenzmoduliertes Signal, als Digitalwort oder ähnliches.

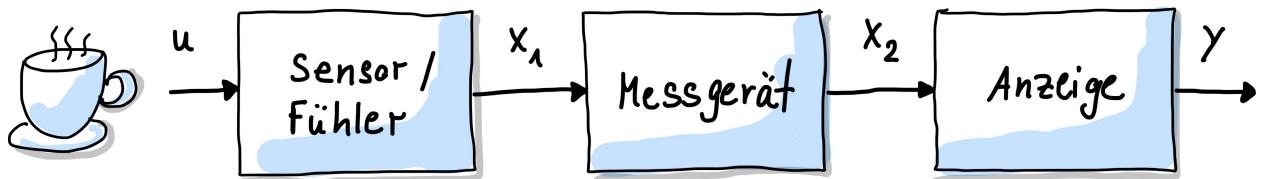


Wird ein Messwert, y , bestimmt, so entspricht dieser im Rahmen der Messtechnik einem Vielfachen, n , von einer Einheit, E :

$$y = n \cdot E$$

Ein Messgerät bestimmt ein Vielfaches einer Einheit. Damit dies überall auf der Welt gleich gut funktioniert muss ein Messgerät entsprechend *geeicht* oder *kalibriert* werden. Außerdem gibt es auch noch den Begriff des *Justierens*. Dieser beschreibt die Anpassung eines Messgeräts an verschiedene Umgebungen. Evtl. müssen Messgeräte bei unterschiedlichen Temperaturen anders behandelt werden und entsprechend *einjustiert* werden.

Häufig werden Prozesse in der Messtechnik mittels **Blockdiagrammen** dargestellt:



Wie man an der oben dargestellten Temperaturmessung sieht, sind die Werte, die gemessen werden, nicht unbedingt *exakt*. Später befassen wir uns noch mal genauer mit den so genannten *Messabweichungen*.

3.1.1 Wissenschaft des Messens

Wunder an Präzision (2600 v. Chr.): Cheops-Pyramide

Die Cheops-Pyramide ist die älteste und größte der 3 Pyramiden von Gizeh. Sie wurde als Grabmal für den ägyptischen König (Pharao) Cheops errichtet, der während der 4. Dynastie im Alten Reich regierte (2620-2580 v. Chr.) und sie gehört zu den 7 Weltwundern der Antike. Sie ist eines der einzigen Weltwunder, welches bis heute erhalten geblieben ist. Die Seitenlänge beträgt $230,33 \text{ m} \pm 4 \text{ cm}$ und die Höhe $146,59 \text{ m}$ und war damit 4000 Jahre lang das höchste Bauwerk der Welt. Ihre Einmessung wurde in sehr hoher Genauigkeit vorgenommen, welches in nachfolgenden Bauten nicht mehr erreicht wurde. Sie ist genau nach den vier Himmelsrichtungen ausgerichtet und der Unterschied in den Längen ihrer vier Seiten beträgt weniger als ein Promille!

Im Allgemeinen sind „nur“ drei Parameter maßgeblich, um die Präzision einer Pyramide zu bestimmen:

- die waagrechte Ausrichtung des Fundaments und aller folgenden Bauschichten,
- die Orientierung nach den Himmelsrichtungen,
- die Seitenneigung der Flächen.

Alle drei Parameter müssen nicht nur bei Baubeginn exakt festgelegt, sondern vor allem auch während des Baus kontinuierlich kontrolliert und nachgemessen werden, sonst wird das ganze Bauwerk sichtbar unregelmäßig. Es genügt nicht,

beispielsweise die horizontale Ausrichtung lediglich bei Baubeginn festzulegen und dann drauflos zu bauen. Bei der enormen Höhe der großen Pyramiden von Gizeh würde ein sich wiederholender Messfehler von wenigen Millimetern nach oben hin multipliziert. Erwägt man also mögliche Messmethoden im Hinblick auf ihre Tauglichkeit beim Pyramidenbau, müssen drei wesentliche Bedingungen erfüllt werden:

- Die Messtechnik ist mit steinzeitlichem Werkzeug und Wissen möglich.
- Sie ist realistisch geeignet, bei den beträchtlichen Dimensionen der Pyramiden zumindest die erreichte Genauigkeit zu liefern.
- Sie ist auch in großer Höhe und auf kleiner Fläche anwendbar, so dass sie bei jeder neuen Schicht wiederholt werden kann.

Wie hat man die Präzision im Pyramidenbau im Jahre 2600 v. Chr. erreicht? Bei den Ägyptern verwandte man sogenannten Körpermaße. Üblich waren z.B. Elle und Fuß. Verbindliche und reproduzierbare Maßeinheiten bei den Pyramidenbauern waren sogenannte Längennormale aus Holz. Diese *Normale* mussten jeden Monat mit der *königlichen Elle* oder *meh* – einem *Primärnormal* aus Granit, das der Unterarmlänge des Pharaos entsprach – neu kalibriert werden. Dieses Verfahren funktionierte erstaunlich gut, denn eine Vernachlässigung des Kalibriergebots wurde schließlich mit dem Tod bezahlt. Die erreichte Präzision beim Pyramidenbau wird daran deutlich, dass die Abweichungen zwischen den Kantenlängen der Basis einer Pyramide teilweise lediglich 0,06 % betragen.

```
# diesen Code durch Eingabe von <shift><ret> ausführen
laenge_pyramide = 230.33      # in m
abweichung = 0.14             # in m
relative_abweichung = abweichung / laenge_pyramide
print('relative Messabweichung der Cheops-Pyramide: ', relative_abweichung*100, '%')
```

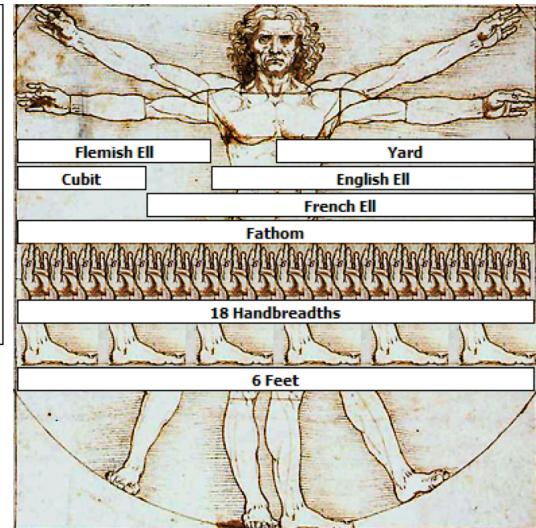
relative Messabweichung der Cheops-Pyramide: 0.06078235575044501 %



Für die Maße yard und inch wurde der König als Referenz genommen.



Längeneinheiten im alten Ägypten.



Quelle: Wikipedia

Bis zum 18. Jhd. orientierten sich die meisten Maßeinheiten weiterhin am Menschen, wobei natürlich regionale Abweichungen beachtet werden mussten! Die Regensburger Elle war etwa 81,1 cm lang, während die Bremer Elle dagegen nur 54,7 cm aufwies.

Wunder an Präzision (2600 v. Chr.): Cheops-Pyramide

- Die Griechen übernahmen beispielsweise die ägyptischen Längenmaße und führten das Stadion ein (die Länge, die

ein geübter Läufer schnell zurücklegen kann, etwa 180 m).

- Die Römer führten zur Messung der großen Entfernungen in ihrem Straßennetz die Meile als neues Längenmaß hinzu.
 - 1101 führt Heinrich I. von England die Längeneinheit Yard (Abstand von seiner Nasenspitze bis zum Daumen seines ausgestreckten Armes) und Inch (Breite seines Daumens) ein.
 - Eduard II. von England erklärt die Länge von einem Zoll zum Längenmaß. Es hat die Länge dreier hintereinandergelegter Gerstenkörner.
 - Der Mathematiker J. Kölbel schlägt an Stelle eines Körpermaßes ein sogenanntes Naturmaß vor: "16 Männer groß und klein", die nach einer Messe der Reihe nach aus der Kirche kommen, stellen ihre Füße hintereinander. Der sechzehnte Teil der Gesamtlänge soll dann ein Fuß sein.
-

Es wird deutlich, wie zufällig jedes Herzogtum seine eigenen Einheiten eingeführt hatte was teilweise zu großem Chaos geführt hatte. Die Körpermaße einzelner Herrscher wurden als Längeneinheit benutzt, die nur lokal Gültigkeit besaßen und auch lokal individuell festgelegten Einheiten erschwerten internationalen Handel und Probleme in Forschung, Technik und Kommunikation.

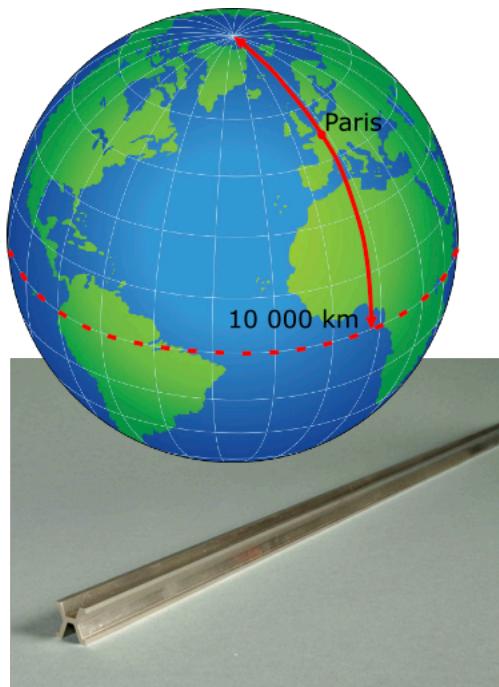
3.1.2 Maßeinheiten

Aufgrund der im vorherigen Kapitel erläuterten regionalen Unterschiede, beschloss 1790 die französische Nationalversammlung im Geiste der französischen Revolution und unter dem Motto „À tous les temps, à tous les peuples, les“ (Für alle Zeiten, für alle Völker) die Schaffung eines neuen Einheitensystems. Der erste Antrag wurde am 07. Oktober für die Maßeinheit Meter gestellt. Ein Meter sollte als der zehnmillionste Teil des Erdmeridianquadranten definiert sein. Die Gelehrtenkommission (Borda, Condorcet, Lagrange, Laplace und Monge) hat zwei Maßeinheiten wifolgt definiert:

- Der Meter als universelle Maßeinheit der Länge sollte den zehnmillionsten Teil der Entfernung vom Nordpol zum Äquator über Paris betragen;
- das Kilogramm als universelle Maßeinheit der Masse sollte der Masse eines Kubikdezimeters Wasser entsprechen.

1799 wurde das Naturmaß Meter wieder durch ein Kunstmaß ersetzt, da die Meterfestlegung mit der Erde als Referenz, messtechnisch nur sehr aufwendig zu wiederholen ist:

- Man fertigte einen Maßstab aus Platin an, der als Urmeterstab in Paris aufbewahrt wurde
- 1889 wurde der Platinstab durch einen Platin-Iridium-Körper mit X-förmigem Querschnitt ersetzt (90% Platin und 10% Iridium). Auf diesem wurden 2 Mittelstriche markiert, welche den Meter angeben. Bei Temperaturveränderungen von 0°C auf 20°C verlängert sich das "Meter" um 0,3 mm, wobei die Ablesegenauigkeit hierbei 0,01 mm betrug.



Der erste Meter wurde mit der Erde als Referenz definiert.
1889 wurde ein Maßstab aus Platin angefertigt.

SI-Einheiten

Im Rahmen der Meterkonvention im Jahr 1960 wurde das **Internationale Einheitensystem**, kurz SI, benannt nach „le Système Internationale d'unités“, eingeführt. Die Definitionen der Basiseinheiten basierten nach wie vor teilweise auf materiellen Prototypen (bis 2019 war dies tatsächlich beim Kilogramm der Fall). Das SI basiert auf der Idee, dass sich im Prinzip alle relevanten Messgrößen über physikalische Gesetze auf genau 7 Basisgrößen zurückführen lassen. Diese 7 Basisgrößen sind die Basiseinheiten, aus denen alle weiteren Einheiten abgeleitet werden können:

- **Meter (m)** als Einheit für die Länge
- **Kilogramm (kg)** als Einheit für die Masse
- **Sekunde (s)** als Einheit für die Zeit
- **Ampere (A)** als Einheit für die elektrische Stromstärke
- **Kelvin (K)** als Einheit für die thermodynamische Temperatur
- **Candela (Cd)** als Einheit für die Lichtstärke und
- **Mol (mol)** als Einheit für die Stoffmenge



Das Ur-Kilo in Paris

Mit Ausnahme des Kilogramms wurden bis vor Kurzem alle Basiseinheiten über reproduzierbare Experimente eindeutig festgelegt. Die Sekunde ist zum Beispiel darüber definiert, als das sie das 9192631770-fache der Periodendauer einer bestimmten Strahlung ist, nämlich der des Übergangs zwischen den beiden Hyperfeinstrukturniveaus des Grundzustands von Atomen des Nuklids 133-Cäsium. Im Prinzip könnte sich also jeder das Element Cäsium besorgen, eine Atomuhr betreiben, und somit seine Sekunde zuhause definieren. Oder man spart sich die Arbeit und sucht eine der Kalibrierbehörden auf. 1983 wird die Länge eines Meters als „jene Wegstrecke, die das Licht im Vakuum während der Dauer von $1/299792458$ -tel einer Sekunde zurücklegt“, festgelegt. Somit war das Meter die erste Einheit, welche durch eine Naturkonstanten, nämlich die Lichtgeschwindigkeit $c = 299792458$ m/s definiert, bzw. festgelegt wurde. Andere Einheiten waren in der Praxis schwieriger umzusetzen, wie z.B. das Ampere. Für die Definition eines Amperes wurde die Kraft zwischen 2 stromdurchflossenen Leitern gemessen, was extrem unpraktisch ist. Eine große Ausnahme ist und blieb das Ur-Kilo, welches als Prototyp von 1889-2018 in Paris als internationale Referenz sicher (bzw. vermeintlich sicher) aufbewahrt wurde. Das Ur-Kilo war:

- ein 3,9 cm hoher und 3,9 cm dicker Metallzyylinder, der zu 90% aus Platin und zu 10% aus Iridium besteht
- Seit 1889 ist dieser „Block“ das Referenznormal für Kilogramm
- es wird unter drei Glasglocken in einem Tresor des „Internationalen Büros für Maß und Gewicht“ (BIPM) in Paris aufbewahrt
- keine Lösung für die Ewigkeit, denn - keiner weiß warum - aber das Ur-Kilo wird immer leichter
- 50 Mikrogramm hat es in den letzten 129 Jahren im Vergleich zu seinen 70 offiziellen Kopien weltweit verloren



[spiegel.de]

ABO SHOP AKADEMIE JOBS MEHR ▾ E-PAPER AUDIO APPS ARCHIV MERKLISTE ANMELDEN

ZEIT ONLINE

Suche

Politik Gesellschaft Wirtschaft Kultur ▾ Wissen Digital Campus ▾ Arbeit Entdecken Sport ZEITmagazin Podcasts mehr ▾ Z+

Physikalische Größen

Das Ur-Kilo ist abgelöst

130 Jahre diente ein Kiloklotz als Maß aller Dinge – bis jetzt. Ab sofort gelten für Kilogramm, Sekunde, Meter, Stromstärke, Temperatur und Lichtstärke neue Messmethoden.

20. Mai 2019, 19:05 Uhr / Quelle: ZEIT ONLINE, dpa, AFP, tgr / 102 Kommentare /

Silizium-Atome statt Klotz: Das Ur-Kilogramm als Maß aller Dinge hat ausgedient. Das Internationale Einheitensystem für physikalische Größen arbeitet ab sofort mit anderen Definitionen und Messmethoden. Künftig wird ein Kilogramm mithilfe einer bestimmten Anzahl von Silizium-Atomen und weiterer Faktoren festgelegt. Die neue Definition richtet sich nach den vom Begründer der Quantenphysik, Max Planck, eingeführten Planck-Konstanten (\hbar).

Das klingt erstmal nicht viel, wird aber in unserer Hightech-Welt, in der schon in Nanometern (Millionstel Millimeter) oder Femtosekunden (Millionstel einer Milliardstel Sekunde) gemessen wird, mehr und mehr zum Problem.

Seit 2018 werden *alle* SI-Einheiten von Naturkonstanten abgeleitet. Bei den drei Basiseinheiten Meter, Sekunde und Candela hat sich nichts substantielles geändert, lediglich die Formulierung der Definition wurde angepasst, z.B.:

- Die **Sekunde** ist ab sofort dadurch definiert, dass die Frequenz der Cäsium-Strahlung, $\Delta\nu_{133\text{Cs}}$, exakt den Wert 9192631770 annimmt, wenn man diese in 1/s ausdrückt:

$$1 \text{ s} = \frac{9192631770}{\Delta\nu_{133\text{Cs}}}$$

(Cäsiumuhren haben übrigens eine Störanfälligkeit von 1:1e13, das entspricht einer Abweichung von 1s in 300000 Jahren.)

- Das **Meter** wird ähnlich wie zuvor über die Lichtgeschwindigkeit c ausgedrückt:

$$1 \text{ m} = \frac{c}{299792458} \text{ s} = 30,663318\dots \frac{c}{\Delta\nu_{133\text{Cs}}}$$

- Die **Candela** wird von der photometrische Strahlungsäquivalent K_{cd} (ebenfalls eine Naturkonstante) abgeleitet. Sie wird über die SI-Einheiten kg, m, s und Steradian (sr = m^2/m^2) definiert.

Anders sieht es bei den weiteren vier Basiseinheiten aus, für die Naturkonstanten gefunden und festgelegt wurden:

- Das **Kilogramm** ist nun durch Ableitung aus dem Planckschen Wirkungsquantum $h = 6,62607015 \cdot 10^{-34} \text{ Js}$ definiert, wobei die Einheit J (Joule), wie unten noch aufgeführt wird, nichts anderes als $\text{kg}\text{m}^2/\text{s}^2$ ist.

$$1 \text{ kg} = \frac{h}{6,626070040 \cdot 10^{-34}} \text{ m}^{-2}\text{s} = 1,475521\dots \cdot 10^{40} h \cdot \frac{\Delta\nu_{133\text{Cs}}}{c}$$

h wird dabei in Kooperation der metrologischen Institutionen in Form aufwendiger Experimente in entsprechender Genauigkeit bestimmt.

- Das **Ampere** wird dadurch definiert, dass die Elementarladung $e = 1,6021766208 \cdot 10^{-19} \text{ As}$ beträgt:

$$1 \text{ A} = \frac{e}{1,6021766208 \cdot 10^{-19}} \text{ s}^{-1} = 6,789687\dots \cdot 10^8 \Delta\nu_{133\text{Cs}} \cdot e$$

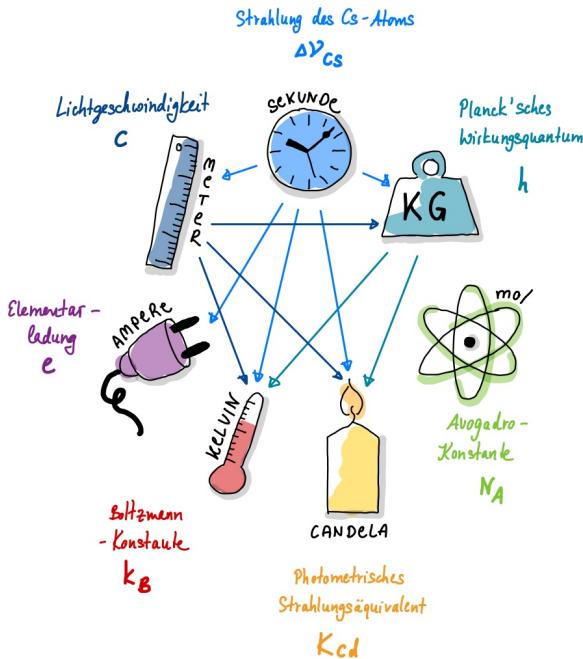
- Das **Kelvin** ist die Einheit der thermodynamischen Temperatur, über die Boltzmann-Konstante $k_B = 1,38064852 \cdot 10^{-23} \text{ kg}\text{m}^2\text{s}^{-2}\text{K}^{-1}$.

$$1 \text{ K} = \frac{1,38064852 \cdot 10^{-23}}{k_B} \text{ kg}\text{m}^2\text{s}^{-1} = 2,266665 \Delta\nu_{133\text{Cs}} \cdot \frac{h}{k_B}$$

- Das **Mol** ist dadurch definiert, dass die Avogadro-Konstante $N_A = 6,022140857 \cdot 10^{23} \text{ mol}^{-1}$ beträgt.

$$1 \text{ mol} = \frac{6,022140857 \cdot 10^{23}}{N_A}$$

N_A ist die Zahl, der in einem Mol enthaltenen Atome. Sie ist so definiert, dass 12 g Kohlenstoff (12C) genau einem Mol entspricht.



Physikalische Konstanten		
Konstante	Exakter Wert	Seit
$\Delta\nu_{Cs}$	Strahlung des Cs Atoms	9 192 631 770 Hz
c	Lichtgeschwindigkeit	299 792 458 m/s
h	Plank'sches Wirkungsquantum	$6,626\ 070\ 15 \times 10^{-34} \text{ Js}$
e	Elementarladung	$1,602\ 176\ 634 \times 10^{-19} \text{ C}$
k_B	Boltzmann-Konstante	$1,380\ 649 \times 10^{-23} \text{ J/K}$
N_A	Avogadro-Konstante	$6,022\ 140\ 76 \times 10^{23} \text{ mol}^{-1}$
K_{cd}	Photometrisches Strahlungsäquivalent	683 lm/W

Abgeleitete / Ergänzende SI-Einheiten

SI umfasst auch eine Aufzählung weiterer Einheiten, welche von den 7 Basiseinheiten, oder über physikalische Gesetzmäßigkeiten, abgeleitet werden können. Hier nur einige Beispiele:

- 1 Hz (Hertz für Frequenz) = 1/s
- 1 N (Newton für Kraft) = kgm/s²
- 1 Pa (Pascal für Druck) = 1 N/m² = 1 kg/ms²
- 1 J (Joule für Energie) = 1 Nm = 1 kg²/s²
- 1 W (Watt für Leistung) = 1 J/s = 1 kgm²/s³
- 1 V (Volt für Spannung) = 1 W/A = 1 kgm²/s³A
- 1 H (Henry für Induktivität) = 1 Vs/A = 1 kgm²/s²A²
- 1 F (Farad für Kapazität) = 1 As/V = 1 s⁴A²/kgm²

Zwischen verschiedenen physikalischen Teildisziplinen kann nun auch mit den Einheiten hin und her jongliert werden. So kommt die Leistung (W) sowohl in mechanischen, als auch auch elektrische Gesetzmäßigkeiten vor. Man kann durch die elektrische Spannung (V) durch eine Kombination des Amperes (elektrische Basiseinheit) mit mechanischen verknüpfen.

Ergänzende Einheiten im SI-System sind beispielsweise:

- 1 rad (Radian) = 1 m/m , welches der ebene Winkel zwischen zwei Radien eines Kreises ist, falls der dadurch beschriebene Kreisbogen genauso groß ist wie der Radius. Der Umfang eines Kreises ist bekannterweise $2\pi \cdot r$, wobei r der Kreisradius ist. Dadurch entspricht eine komplette Drehung einem Winkel von $2\pi \text{ rad}$
- 1 sr (Steradian) = $1 \text{ m}^2/\text{m}^2$ ist der räumliche Winkel (analog zum Radian). Dieser schließt mit der Kugelmitte als Scheitelpunkt eine Fläche auf der Kugeloberfläche ein. Diese Fläche ist quadratisch mit einer Seitenlänge die dem Kugelradius entspricht. Die Einheit kann also ebenfalls auf Basiseinheiten zurückgeführt werden, hier $1 \text{ sr} = \text{m}^2/\text{m}^2$.

Nicht-SI-Einheiten

Es gibt diverse zusätzliche Einheiten, welche keine offiziellen SI-Einheiten sind, aber aufgrund ihrer großen Beliebtheit und Handbarkeit gerne benutzt werden. Im Allgemeinen gibt es aber immer Zusammenhänge zu den SI-Einheiten, sodass sie sich in solche umformen lassen. Beispiele sind z.B.:

- Grad Celsius: $1^\circ\text{C} = \text{K} + 273,15$
- Grad Fahrenheit: $9/5 \text{ K} - 459,67$
- Minute: $1 \text{ min} = 60 \text{ s}$
- (Winkel-)Grad: $1^\circ = \pi/180 \text{ rad}$
- (Winkel-)Minute: $1' = 1/60^\circ$
- Liter: $1 \text{ l} = 1 \text{ dm}^3$
- Tonne: $1 \text{ t} = 10^3 \text{ kg}$
- Bar: $1 \text{ bar} = 10^5 \text{ Pa}$

Dann gibt es noch historisch gewachsene Einheiten, wie z.B. die Meile, yard, foot, inch, once, pound, gallon, welche sich analog in SI-Einheiten umrechnen lassen. Diese Umrechnung ist global nicht immer die gleiche und es existieren für dieselbe Einheit unterschiedliche Umrechnungen (USA und UK sind hier die wohl bekanntesten Beispiele). Doch auch je nach Anwendungsbereich gibt es Unterschiede:

- 1 mile = 1 Landmeile = 1.609,344 m (US)
- 1 nautical mile = 1 Seemeile (oder Luftfahrt) = 1.853,2 m (UK)
- 1 mile = exakt 1.852 m (international)

Einheiten, die zwar in Gebrauch sind, aber nicht auf SI-Einheiten zurückzuführen sind, wurden für spezifische Einsatzgebiete konkret festgelegt:

- die Wasserhärte
- das Mostgewicht
- den Feingehalt von Gold- und Silberlegierungen
- die Windstärke
- den Seegang
- die Stärke von Erdbeben

Vorsätze und Präfix im SI

Zum SI, bzw. prinzipiell angewendet in allen anderen Einheiten, sind sogenannte Präfixe / Vorsätze definiert. Teile oder Vielfach von SI-Einheiten können in Kurzform geschrieben werden, was das Lesen erleichtert. So können besonders große oder besonders kleine Zahlen übersichtlicher dargestellt werden. Dafür muss der oder die Forschende oder Ingeneur:in lediglich ein paar Vokabeln können:

Zeichen	Name	Multiplikator
Y	Yotta	10^{24}
Z	Zetta	10^{21}
E	Exa	10^{18}
P	Peta	10^{15}
T	Tera	10^{12}
G	Giga	10^9
M	Mega	10^6
k	Kilo	10^3
h	Hekto	10^2
da	Deka	10^1

Zeichen	Name	Multiplikator
d	Dezi	10^{-1}
c	Zenti	10^{-2}
m	Milli	10^{-3}
µ	Mikro	10^{-6}
n	Nano	10^{-9}
p	Piko	10^{-12}
f	Femto	10^{-15}
a	Atto	10^{-18}
z	Zepto	10^{-21}
y	Yokto	10^{-24}

Logarithmische Einheiten

In der Messtechnik können unter Umständen Messwerte in ganz unterschiedlichen Größenordnungen anfallen. Daher haben wir uns ja im Kapitel vorher die Präfixe bzw. Vorsätze angesehen. Für eine Darstellung im Diagramm, bei dem die Achsen typischerweise eine feste Einheit besitzen, wird es dennoch schwierig, die Gesamtheit der Messreihe übersichtlich darzustellen. Daher bedient man sich häufig der logarithmischen Darstellung, welche im ersten Moment relativ umständlich und kompliziert erscheint, aber einen hohen Nutzen hat. Diese Darstellung ist auch im SI-System vorgesehen.

Der eigentlich Messwert auf einen wohl definierten Referenzwert bezogen wird. Man bildet also den Quotienten aus Messwert und Referenzwert, P/P_{ref} (bei Leistungen) oder U/U_{ref} bei Spannungen. Danach werden diese Quotienten logarithmiert, *fast ausschließlich* mit der 10er-Logarithmus (log). Der neue Wert ist per Definition einheitenlos, wird aber die Einheit **Dezibel** (dB) zugeordnet, also das Zehntel eines **Bels**. Ganz selten wird der natürliche Logarithmus benutzt, dann wird die Einheit Neper (Np) angewendet.

In der Messtechnik hat es sich etabliert (ungeschriebenes Gesetz), dass in erster Linie Leistungen gemäß der eben beschriebenen Gesetzmäßigkeit in der Einheit dB umgewandelt werden, man spricht hierbei von der **Leistungsgroße**:

$$1 \text{ dB} = 10 \cdot \log \frac{P}{P_{\text{ref}}}$$

Man spricht von der **Feldgröße**, wenn Spannungen in die Einheit dB umgewandelt werden:

$$1 \text{ dB} = 20 \cdot \log \frac{U}{U_{\text{ref}}}$$

Aufgabe

Beweise die Umformung von Leistungsgröße in Feldgröße! Hinweise: Es gilt $P \propto U^2$

Logarithmische Darstellungen finden meistens nur bei elektrischen Leistungen und Spannungen statt. Häufig wird das Dezibel z.B. in der Hochfrequenztechnik verwendet oder bei der Charakterisierung von Frequenzgängen (dazu kommen wir später noch). Im Allgemeinen spricht man von **Pegeln**, sobald die Messwerte logarithmisch angegeben sind. Bei der Angabe von Messwerten in der Einheit Dezibel **muss** stets darauf geachtet werden den Refrenzwert mitanzugeben. Typische Schreibweisen hierzu sind z.B.

- dB(mW): es handelt sich um einen Leistungspegel und der Referenzwert ist 1 mW
- dB(mV): es handelt sich um einen Spannungspegel und der Referenzwert ist 1 mV
- dB(μ V): es handelt sich um einen Spannungspegel und der Referenzwert ist 1 μ V Ohne die Angabe des Referenzwertes ist weder die physikalische Größe, noch der Skalierungsfaktor bekannt, und daher sehr wichtig, falls der Dezibel-Wert später in absolute Einheiten zurück konvertiert werden sollte.

Für Leistungspegel gilt also im Allgemeinen (für einen typischen Referenzwert von 1 mW)

- $1 \mu\text{W} = 0,000001 \text{ W} = -30 \text{ dB (mW)}$

Aufgabe

In folgendem Code-Block können Umrechnungen für verschiedene Messwerte ausprobiert werden:

```
import numpy as np
P = 1e-6      # Messwerts, hier 1 uW
P_ref = 1e-3 # Referenzwert = 1 mW
P_dB = 10 * np.log10(P/P_ref) # Achtung: in numpy ist "log" der natürlich Logarithmus
                             # → ln
print('Leistung = %.e W = %.10f W' %(P, P))
print('Pegel in dB: ', P_dB, '(% .3f)' %(P_ref)) # In diesem Code Block können
                                                 # → Umrechnungen für verschiedene Messwerte ausprobiert werden:
```

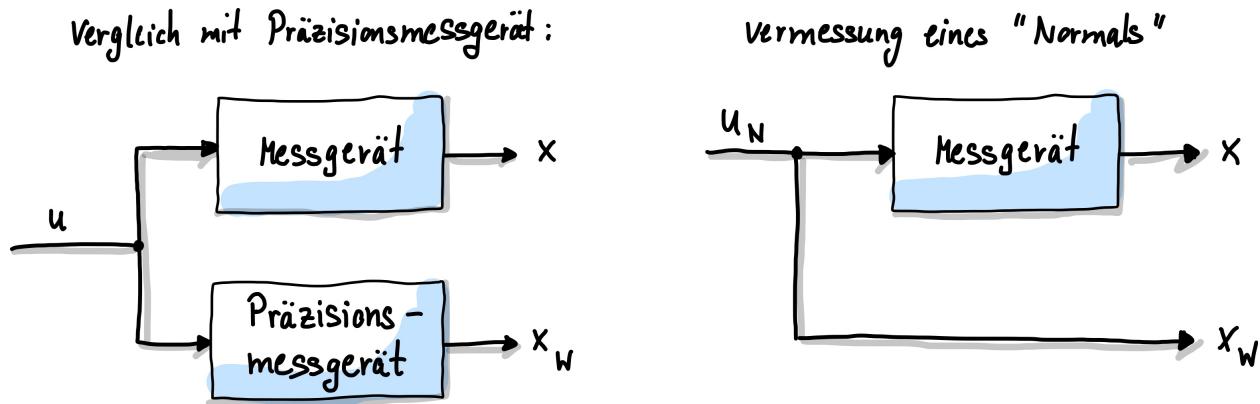
```
Leistung = 1e-06 W = 0.0000010000 W
Pegel in dB: -30.0 (0.001)
```

3.1.3 Kalibrieren und Eichen (und Prüfen)

Es gibt verschiedene Sprachlichkeiten in der Messtechnik, die im folgenden einmal kurz definiert werden, da es hier in der Umgangssprache häufig zu Unkorrektheiten kommt. Der Unterschied zwischen *messen* und *prüfen* ist nicht immer klar. Im technischen Bereich versteht man unter **prüfen**, ob ein Prüfgegenstand bestimmte Vorgaben erfüllt. Diese werden dann auch in Form von Prüfbedingungen spezifiziert. Zum Beispiel kann man auf elektromagnetische Verträglichkeit (EMV) prüfen., diese werden in Normen festgehalten, welche wiederum ganz konkrete Randbedingungen für Messaufbauten - zur Messung von elektromagnetischen Störungen - bei Messgeräten gegeben sein müssen. Messgeräte können also darauf geprüft werden, ob diese Normen eingehalten werden und die Messungen entsprechend durchgeführt und ausgeführt werden. Prüfen ist also etwas mehr, als nur das Messen einer Größe.

Kalibrieren und Eichen

Messsysteme haben bekannterweise Messgenauigkeiten (darauf kommen wir später noch mal zurück), welche eine Messung limitieren. Die erzielbare Messgenauigkeit kann werksseitig während des Herstellprozesses oder später, in der gewünschten Testumgebung, verbessert werden. Hierzu benötigt man eine *bekannte* Referenz, die an das Messgerät angeschlossen werden kann. Nun kann das Messsystem entweder eingestellt werden, sodass der angezeigte Messwert möglichst genau dem *bekannten* Referenzwert entspricht. Dieses Verfahren nennen man auch **Justieren** oder **Kalibrieren**.



Unter dem Begriff **Eichen** hingegen versteht man die Prüfung und Stempelung eines Messgeräts, welches nach gesetzlichen Eichvorschriften erfolgt ist. Geeicht werden müssen Messsysteme, die im gewerblichen Verkehr oder Handel eingesetzt werden sollen, wie z.B. eine Obst- und Gemüsewaage an der Kasse eines Supermarktes. Dem Verbraucher wird damit eine bestimmte Sicherheit gegeben, dass die Waage - innerhalb bestimmter Grenzen - genau arbeitet. In Deutschland existiert dafür die sogenannten *Eichordnung*. Da das Eichen ein hoheitlicher Akt ist, kann es nur in vom Staat autorisierten Behörden durchgeführt werden, in den sogenannten Eichämtern, und muss in bestimmten Abständen wiederholt werden. Um generell eine möglichst gute Genauigkeit und hohe Manipulationssicherheit sicherzustellen, werden die meisten Messgeräte bereits während des Herstellungsprozesses kalibriert oder geeicht. In Deutschland macht dies häufig die Physikalisch-Technische Bundesanstalt (PTB) in Braunschweig und Berlin.

Normale

Normale sind Maßverkörperungen, welche einfach handhabbar sind und von Basisgrößen abzuleiten sind. Wir wissen, dass die Basiseinheiten - bzw. die von ihnen ableitende Einheiten - über atomare Naturkonstanten definiert sind. Dies ist in der Praxis allerdings sehr unpraktisch und in Betrieben nicht realisierbar. Im BIPM (Internationales Büro für Maß und Gewicht) werden praktisch anwendbare **Primärnormale** direkt von Basisgrößen abgeleitet und hergestellt. Diese werden ständig überwacht um deren Genauigkeit sicherzustellen. Für so ziemlich jede Messgröße existieren solche Primärnormale, wie z.B. für Ohm, Volt, Henry, Farad usw. Alle Staaten, die damals bei der Generalkonferenz für Maß und Gewicht den Vertrag unterzeichnet haben, erhalten jeweils ein solches Primärnormal. Von diesen werden dann sogenannte **Sekundärnormale** innerhalb der Staaten abgeleitet, welche dann wiederum zur Eichung von betrieblichen Arbeitsnormalen in Eichlaboren (oder Behörden oder Ämtern) zur Verfügung stehen. **Arbeitsnormale**, abgeleitet von den Sekundärnormalen, werden in Firmen verwendet, um ihre betrieblichen Messmittel eigenhändig kalibrieren zu können.



MESSDATEN ANALYSIEREN

Viele Studierende sammeln im Praktikum erstmals Erfahrung im Umgang mit Messdaten, nehmen eigene Messreihen auf und müssen diese begründbar und nachvollziehbar auswerten und darstellen. Im Studium, z.B. im Rahmen von Semester-, Abschluss- oder sogar Promotionsarbeiten müssen Analyse und Präsentation wissenschaftlich und sachgerecht sein. Auf den folgenden Seiten findet ihr das absolute Minimum an notwendigen Hilfsmitteln, Grundideen und Praktiken, die ihr bei der Auswertung von Messdaten im Praktikum berücksichtigen solltet!

- *Quellen und Ursachen von Messunsicherheiten*
- *Statistische Größen*
- *Fortpflanzung von Messunsicherheiten*
- *Kurvenanpassung*

4.1 Zusammenfassung

Begriff	Beschreibung
Messgröße	die spezielle Größe der Messung, x
Wahrer Wert	tatsächlich vorhandener Wert einer Messgröße, dessen Wert niemals bekannt sein wird.
Messergebnis	(Schätz-)Wert, den die Messgröße durch Auswertung einer Messung bekommt
arithmetischer Mittelwert \bar{x}	Schätzwert für den wahren Wert einer Messgröße aus einer Messreihe mit den Messwerten x_j und der Anzahl der Messwerte m : $\bar{x} = \frac{1}{m} \sum_{j=1}^m x_j$
Messunsicherheit $u(x)$	Wichtiger Bestandteil zur Angabe eines Messergebnisses. Die Unsicherheit charakterisiert einen Vertrauensbereich, der der Messgröße zugeschrieben wird: $\bar{x} \pm u(x)$
relative Messunsicherheit	Messunsicherheit dividiert durch den Betrag des Mittelwerts: $A_r = \frac{u(x)}{\bar{x}}$
Varianz: mittlere quadratische Abweichung $s^2(x)$	Ein Maß für die Messunsicherheit. Abweichung der Messwerte zum Mittelwert werden quadriert und gemittelt: $s^2 = \frac{1}{m-1} \sum_{j=1}^m (x_j - \bar{x})^2$
Standardabweichung	Wurzel aus der mittleren quadratischen Abweichung: $s = \sqrt{\frac{1}{m-1} \sum_{j=1}^m (x_j - \bar{x})^2}$
empirische Varianz $\sigma(x)^2$	Schätzung der Varianz bezogen auf den <i>wahren</i> Wert (nicht auf den Mittelwert der Messreihe) der Messgröße: $\sigma^2 = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{j=1}^m (x_j - \mu)^2$
empirische Standardabweichung $\sigma(x)$	Wurzel aus der empirischen Varianz: $\sigma = \sqrt{\frac{1}{m} \sum_{j=1}^m (x_j - \mu)^2}$
Wahrscheinlichkeitsverteilung $dP(x)$	eine Funktion, die die Wahrscheinlichkeit angibt, dass eine Messgröße x durch Messung einen bestimmten Wert $x_j + dx$ annehmen wird.

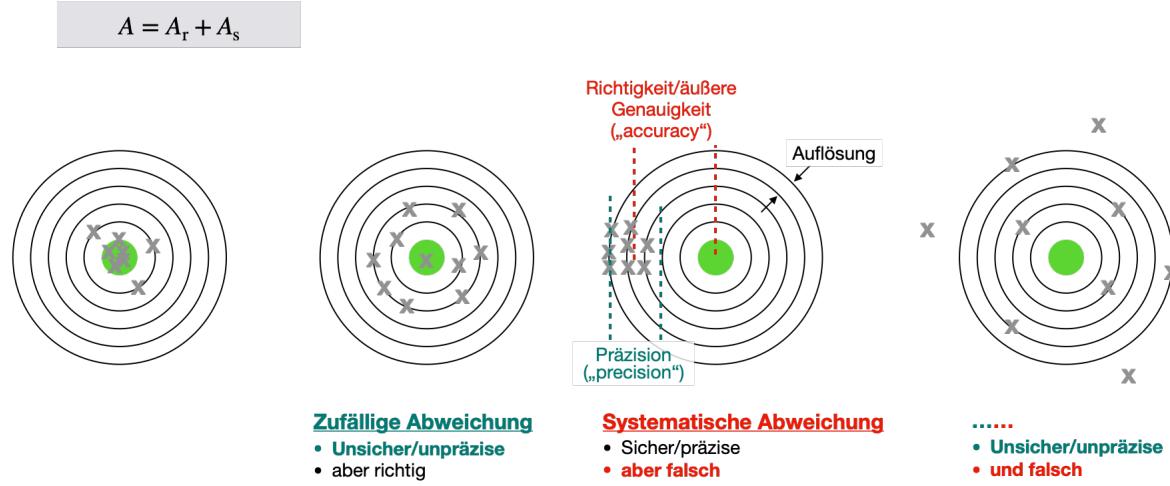
4.2 Quellen und Ursachen von Messunsicherheiten

Früher hat man statt *Abweichung* noch den Begriff *Messfehler* verwendet. Man dachte, dass man mit genügend Aufwand, Sorgfalt und bestmöglicher Technologie den Fehler vollständig eliminieren können. Spätestens seit der Theorie der *Quantenphysik* ist uns allerdings bekannt, dass zufällige Einflüsse auf die beobachteten Messgrößen unvermeidlich sind und auch nicht vorhergesagt werden können. Statt eines einzigen *wahren* Wertes werden in der Quantenphysik Messgrößen durch deren Erwartungswerte vorhergesagt. Diesen Messgrößen liegt eine Wahrscheinlichkeitsdichte zu Grunde, dessen Varianz (bzw. Standardabweichung) nicht verschwindet! Somit werden für ein und dieselbe physikalische Messgröße verschiedene Ergebnisse gemessen, obwohl nahezu identische Bedingungen herrschen. Das Eintreten eines bestimmten Messergebnisses ist an eine bestimmte Wahrscheinlichkeit gekoppelt, mit der dieses Ergebnis eintritt.

Jede Messung einer physikalischen Größe x ist abhängig von den verwendeten Messgeräten, dem Messverfahren, dem Messobjekt, von Umwelteinflüssen (Temperatur, Feuchtigkeit, elektromagnetische Felder) und schließlich auch vom Beobachter (Müdigkeit, Sehschärfe, Übung). Messungen liefern daher lediglich Schätzwerte für die *wahren* Werte einer Größe. Es gibt prinzipiell keine Möglichkeit, den wahren Wert einer Messgröße zu messen. Im Rahmen internationaler Anstrengungen für eine einheitliche Bewertung von Einflussgrößen auf eine Messung werden zwei Kategorien von Methoden der Berechnung von Unsicherheiten unterschieden [GUM]:

- **Typ A (“Zufällige Abweichung”):** Berechnung der Messunsicherheit durch statistische Analyse der Messungen
- **Typ B (“Systematische Abweichung”):** Berechnung der Messunsicherheit mit anderen Mitteln als der statistischen Analyse

Wir können sehr präzise „falsch“ messen...



4.2.1 Typ A-Unsicherheiten (“Zufällige Fehler”)

Dies sind Messunsicherheiten, die nicht einseitig gerichtet sind, sondern einer zufälligen Streuung der Messwerte zugrunde liegen. Zur Behandlung dieser Messunsicherheiten nutzt man die Stochastik (Wahrscheinlichkeitslehre und Statistik). Zufällige Fehler kennt man nicht, er ist folglich nicht korrigierbar und lässt sich auch nicht reproduzieren unter gleichen Messbedingungen. Hier sind wiederholte Messungen und statistische Analysen notwendig, wodurch Mittelwert und Standardabweichung von sogenannten *Stichproben* ermittelt wird. Wie gewinnt man aus einer Messreihe x_j den besten Schätzwert, der mit maximaler Wahrscheinlichkeit am nächsten am *wahren* Wert, x_w , liegt? Mit welcher Wahrscheinlichkeit liegt das Messergebnis innerhalb eines bestimmten Intervalls um den *wahren* Wert, $\$x = x_w + \Delta x\$$?

4.2.2 Typ B-Unsicherheiten (“Systematische Fehler”)

Hierbei handelt es sich um reproduzierbare Messunsicherheiten. Sie kann durch Aufwand und Kalibration verbessert werden, was *nicht* für zufällige Messabweichungen gilt. Systematische Messabweichungen (z.B. Kennlinienfehler) sollten in aller Regel am besten korrigiert werden, wenn dies möglich ist. Ansonsten sollte mindestens eine Angabe der Messabweichung erfolgen. Es gibt keine allgemeingültige Definition oder allgemeine Verfahren zur Korrektur. Das heißt für jeden Fall müssen neue Verfahren entwickelt werden. Hier kommen ein paar Beispiele, wie man mit solchen *systematischen* Messunsicherheiten umgehen kann:

Anzeigefehler von Messgeräten

Messgeräte werden anhand ihrer Genauigkeit in Klassen eingruppiert. Die Klasse entspricht der relativen Messabweichung. Präzisionsmessgeräte besitzen somit Abweichungen die zwischen 0,001% und 0,05% liegen. Die Genauigkeitsklasse K 2,5 (Angabe auf der Messskala nach DIN EN 60051 Abb. 1) bedeutet: Ist der Endwert des eingestellten Messbereichs U_{end} , dann beträgt die Typ B-Unsicherheit über den gesamten Messbereich $u(U) = 0,0025 \cdot U_{\text{end}}$. Für $U_{\text{end}} = 15 \text{ V}$ erhält man also:

```
U_end = 15
u = 0.0025
print('eine absolute Unsicherheit von ', u*U_end, 'V')
```

eine absolute Unsicherheit von 0.0375 V

Dieser Wert von 0,375V gilt unabhängig davon, wie groß der Zeigerausschlag beim Messgerät ist. Um die relative Unsicherheit gering zu halten, sollte der Messbereich möglichst so gewählt werden, dass der Messwert am Skalenende abgelesen wird.

Digitalstellenfehler

Messgerätekategorie	Genauigkeitsklasse
Präzisionsmessgeräte	0.001
	0.002
	0.005
	0.01
	0.05
Feinmessgeräte	0.1
	0.2
	0.5
Betriebsmessgeräte	1.0
	1.5
	2.5
	5.0

Herstellerangabe z.B.:
Maximalabweichung bei Referenzbedingung:
 $\pm (0,5\% \text{ vom Messwert} + 9 \text{ Digit})$



Das Gerät im Bild zeigt den Messwert 5.847V an. Laut Hersteller ist die Maximalabweichung (unter Referenzbedingungen) $a = \pm (0,5\% \text{ vom Messwert} + 9 \text{ Digit})$. Die Anzahl der Nachkommastellen (also der Digits) ist in diesem Falle 3, also 0,001V. Genauer kann das Messgerät keine Spannung angeben. Die Messabweichung setzt sich also wie folgt

zusammen (zwei signifikante Stellen reichen hierbei, da der Messwert selber nicht genauer angezeigt wird):

$$a = \pm(0,5\% \cdot 5,847 \text{ V} + 9 \cdot 0,001 \text{ V}) \approx \pm 0,038 \text{ V} = \pm 38,235 \text{ mV}$$

Innerhalb dieses \pm Bereiches der Breite $2a$ unterstellt man eine Gleichverteilung der Messwerte und bekommt damit die Standardunsicherheit:

$$u(U) = \frac{a}{\sqrt{3}} = \frac{38,235 \text{ mV}}{\sqrt{3}} \approx 22 \text{ mV}$$

Ist nichts weiter bekannt, schätzt man die Unsicherheit über einen Mindestfehler von $a = 1$ Digit ab.

```
Messwert = 5.847 # in V
Nachkommastellen = 5
A_rel = 0.005 # = 0.5%
Digit = 0.001 # in V
A_total = A_rel * Messwert + 9 * Digit
print('Die Messtoleranz beträgt: +-',round(A_total,Nachkommastellen), 'V = +-',_
      round(A_total*1000,Nachkommastellen), 'mV')
print('Die Unsicherheit beträgt: +-',round(A_total/np.sqrt(3),Nachkommastellen), 'V ='
      +-', round(A_total*1000/np.sqrt(3),Nachkommastellen), 'mV')
```

Die Messtoleranz beträgt: +- 0.03824 V = +- 38.235 mV

```
NameError                                                 Traceback (most recent call last)
Input In [2], in <cell line: 7>()
      5 A_total = A_rel * Messwert + 9 * Digit
      6 print('Die Messtoleranz beträgt: +-',round(A_total,Nachkommastellen), 'V ='
      7      +-', round(A_total*1000,Nachkommastellen), 'mV')
----> 7 print('Die Unsicherheit beträgt: +-',round(A_total/np.sqrt(3),
      8 Nachkommastellen), 'V = +-', round(A_total*1000/np.sqrt(3),Nachkommastellen), 'mV
      9')

NameError: name 'np' is not defined
```

Man merkt, dass systematische Fehler sehr unangenehm sein können, da Gegenmaßnahmen fallabhängig entwickelt werden müssen. In einigen Fällen gelangt man zu einer brauchbaren Abschätzung der Unsicherheit, wenn man den „worst case“ annimmt.

4.2.3 Weitere Unsicherheiten

Dann gibt es noch der Vollständigkeitshalber die **dynamischen Fehler**. Jedes Messsystem braucht eine bestimmte Zeit, um sich auf einen bestimmten Messwert einzupedeln. Dies nennt man auch die Latenzzeit, oder die Trägheit, eines Messsystems. Es besteht also die Möglichkeit, dass man *zu früh misst*, und das Messsystem den zu messenden Wert noch gar nicht richtig anzeigen kann.

Quantisierungsfehler entstehen beim Prozess des Digitalisierens der Messwerte. Da nur eine endliche Anzahl von Möglichkeiten besteht einen analogen Messwert darzustellen (abhängig von der Anzahl der verwendet Bits) gehen bei diesem Prozess *immer* Informationen verloren.

4.2.4 Schreibweise eines Messwertes mit Messabweichung

Die **Gesamt-Messabweichung** einer Messgröße x setzt sich also wie folgt zusammen:

$$A = A_r + A_s$$

wobei A_r zufällige und A_s systematische Messabweichungen sind. Ein Messwert setzt sich also zusammen aus dem *wahren* oder *richtigen* Wert, den wir niemals kennen werden, und der Messabweichung. Es gelten folgende Zusammenhänge:

- Der **ermittelte Messwert** lässt sich wie folgt schreiben, wobei x_w der *wahre*, aber uns unbekannte, Wert ist. A ist die Messabweichung:

$$x = x_w + A$$

- Die **absolute Messabweichung** ergibt sich aus Umstellen der Gleichung:

$$A = x - x_w = \Delta x$$

- Bei der Angabe **relativer Messabweichung** wird die Messabweichung auf einen Referenzwert, r , bezogen, der entweder der Messwert selber ist ($r = x$), oder manchmal auch die Spanne ($r = x_{\max} - x_{\min}$) oder Maximalwert/Messbereichsendwert ($r = x_{\max}$):

$$A_{\text{rel}} = \frac{A}{r} = \frac{\Delta x}{r}$$

Signifikante Stellen

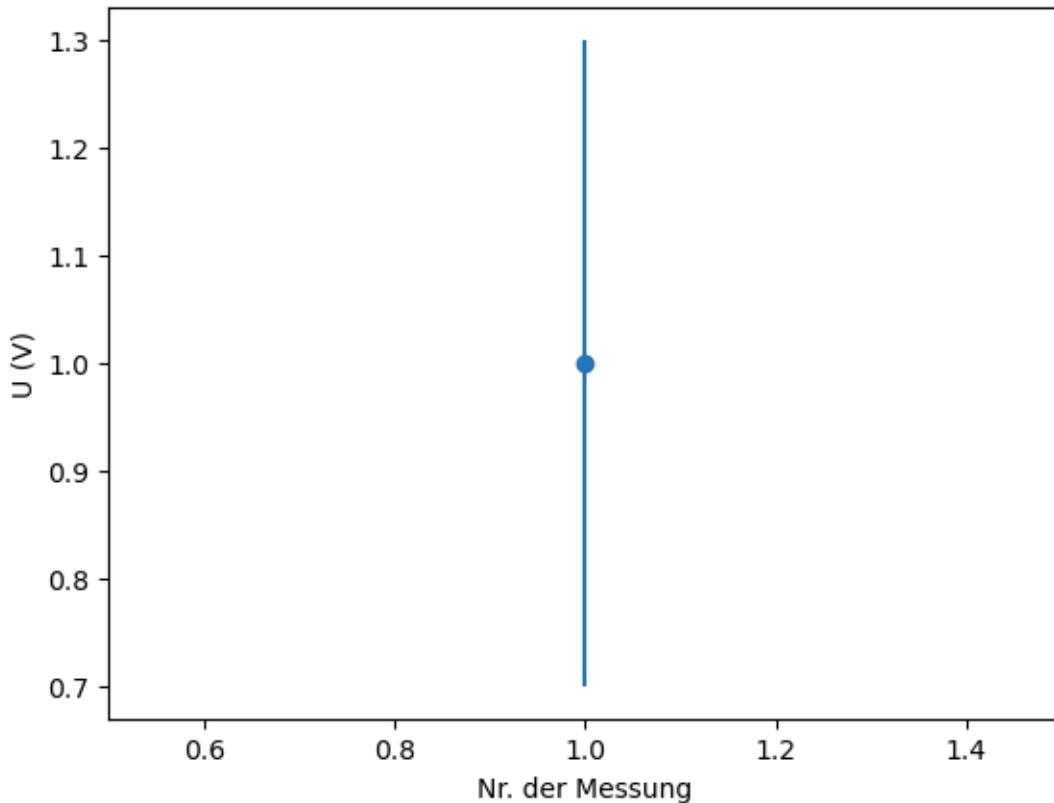
Die **Anzahl der Nachkommastellen** eines Messwertes ist niemals größer als die der angegebenen Messabweichung oder Unsicherheit. Die Anzahl der Nachkommastellen der Messabweichung wird über **signifikante Stellen** (= angegebene Ziffern ohne führende Nullen) definiert. Je mehr signifikante Stellen angegeben werden, desto größer ist die Genuigkeit, die reklamiert wird. Es gelten folgende Rechenoperationen nach DIN1333:

- Bei **Addition von Größen** bekommt das Ergebnis genauso viele Nachkommastellen wie die Zahl mit den *wenigsten* Nachkommastellen.
- Bei **Multiplikation von Größen** bekommst das Ergebnis genauso viele signifikante Stellen wie der Zahl mit den *wenigsten* signifikanten Stellen.
- **Messunsicherheiten** werden auf eine signifikante Stelle gerundet. Eine Ausnahme existiert, wenn die erste Ziffer eine "1" ist, weil sonst Rundungsfehler schnell zu groß werden. Beispiel: $u(g) = 0,1562 \text{ m/s}^2 = 0,16 \text{ m/s}^2$. Die Darstellung $g = (9,81 \pm 0,03562) \text{ m/s}^2$ wäre unsinnig, da die Genuigkeit auf zwei Nachkommastellen durch den Messwert beschränkt ist.
- **Messwerte** werden so angegeben, dass die letzte signifikante Stelle die gleiche Größenpräzision hat, wie die Messunsicherheit: Die Angabe $H = (13,13 \pm 1) \text{ m}$ ist sinnlos, richtig wäre $H = (13 \pm 1) \text{ m}$.

Um Rundungsfehler zu reduzieren, führen Sie in den Berechnungen soviel signifikante Stellen der Messunsicherheit mit, wie nötig.

Grafische Darstellung eines Messwertes mit Messabweichung

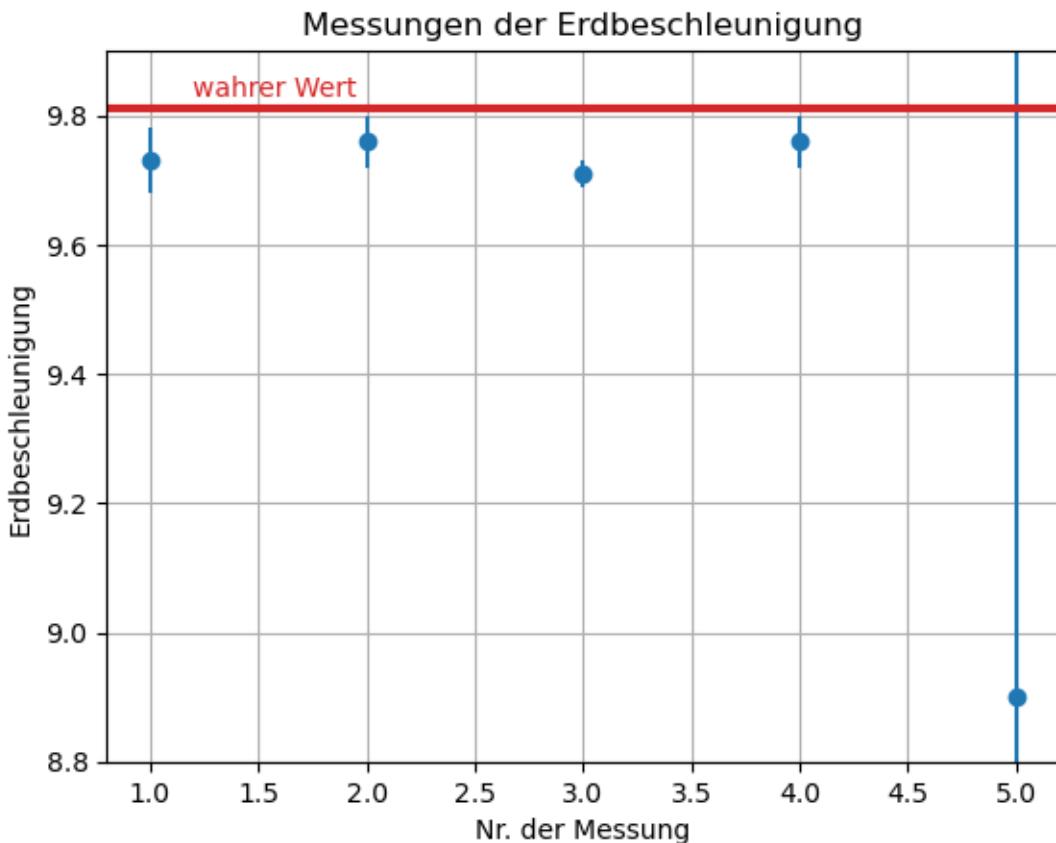
Die grafische Darstellung eines solchen Messwertes in einem Diagramm kann im folgenden Code-Block ausgeführt und angesehen werden. Prinzipiell gilt, dass für jeden Messwert in der Regel ein solcher **Fehlerbalken** stets anzugeben ist.



Diskrepanz und Konsistenz

Die Diskrepanz zweier Messwerte derselben Größe ist der Absolutbetrag ihrer Differenz. Zwei Messungen sind konsistent, wenn ihre Diskrepanz geringer ist, als die (kleinere der) Messunsicherheiten:

- $g = (9,73 \pm 0,05) \text{ m/s}^2$ und $g = (9,76 \pm 0,04) \text{ m/s}^2$ sind *konsistent*, nicht jedoch $g = (9,71 \pm 0,02) \text{ m/s}^2$ und $g = (9,76 \pm 0,04) \text{ m/s}^2$
- Ist $g = (8,9 \pm 1,5) \text{ m/s}^2$ das Ergebnis einer Messung des Ortsfaktors, dann ist die Messung zwar nicht sonderlich präzise, aber mit dem Literaturwert von $g = 9,81 \text{ m/s}^2$ vereinbar.



Nur einer der Messwerte überlappt mit dem *wahren* Wert der Erdbeschleunigung. Die Fehlerbalken der zweiten Messung hingegen sind davon entfernt, in den *wahren* Bereich überzulappen. D.h. es existiert hier ein Widerspruch zu vorherigen Messungen, die den wahren Wert kennzeichnete. Würde es sich hierbei nicht um die Messung der Erdbeschleunigung, sondern um die einer Naturkonstante handeln, gäbe es sogar einen Widerspruch zum SI-Einheitenystem.

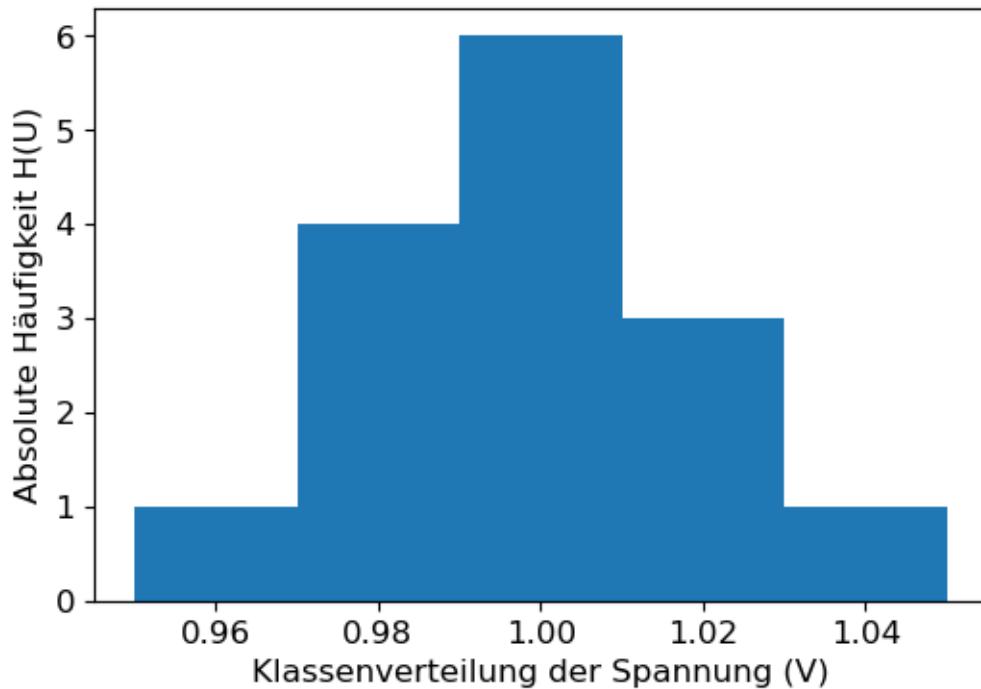
4.3 Statistische Größen

4.3.1 Statistische Messunsicherheit

Statistische, oder zufällige, Einflüsse auf einen Messwert lassen sich durch Wiederholungen der eigentlichen Messung bestimmen. Dies nennt man auch Messreihe und bedeutet, dass m Messungen für ein und denselben Messwert durchgeführt werden. Die einzelnen Messwerte x_j unterscheiden sich, da der *wahre* Wert, x_w , immer mit einer zufälligen Abweichung, A_j , versehen wird:

$$x_j = x_R + A_j$$

Bei genügend vielen Wiederholungen der Messung kann in vielen Fällen beobachtet werden, dass sich die Messwerte x_j um einen zentralen Werte, \bar{x} scharen. Die Häufigkeit, einen Messwert in einem bestimmten Abstand zu diesem zentralen Werte zu finden, $|x_j - \bar{x}|$, ist umso kleiner, je größer der Abstand ist. Man spricht hierbei von einer Häufigkeitsverteilung der x_j . Die grafische Darstellung einer solchen Messreihe erfolgt in einem **Histogramm**. Diese Darstellung wird insbesondere dann häufig benutzt, wenn Messreihen mit vielen Messwerten vorliegen. An einer solchen Darstellung erkennt häufig schon die Art der zugrundeliegenden Verteilung der Messwerten. In nachfolgendem Code-Block wird für die Messreihe die **absolute Häufigkeit** der einzelnen Werte in einer bestimmten Klasse grafisch dargestellt.

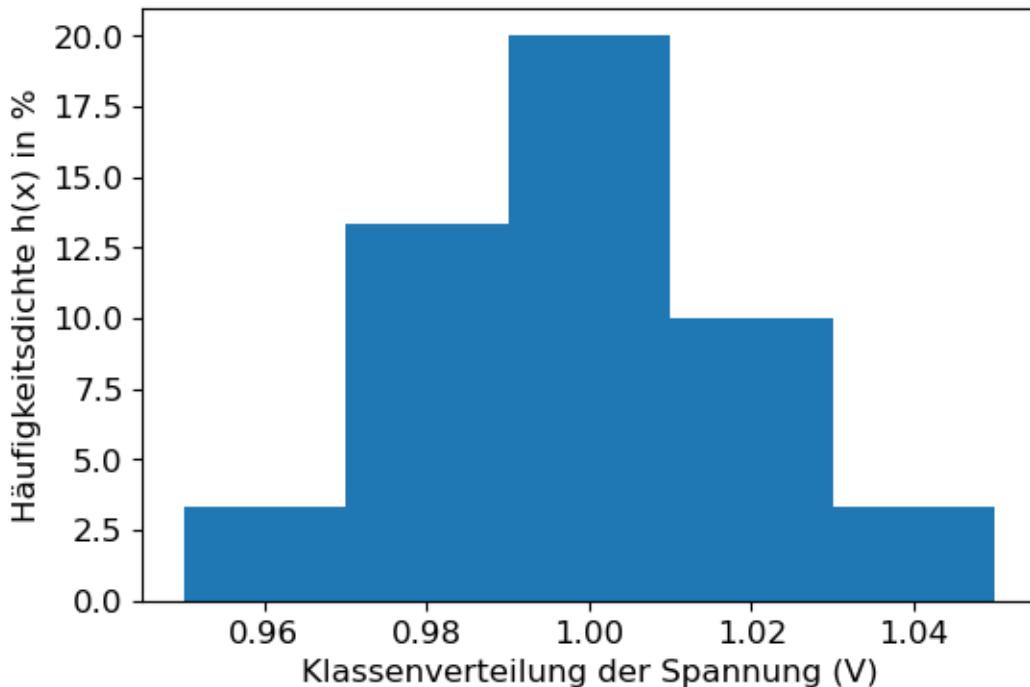


Die **relative Häufigkeit** berechnet sich aus der absoluten Häufigkeit dividiert durch die Gesamtanzahl der vorgenommenen Messungen. Aus der Häufigkeit lässt sich auch die sogenannte **Häufigkeitsdichte** berechnen. Die Häufigkeitsdichte gibt bei einem Histogramm die Höhe des Rechtecks an. Mit ihr kann man den Vergleich verschiedener Klassen erst vornehmen. Anders ausgedrückt heißt dass, dass die Häufigkeitsdichte einer Klasse das Verhältnis der absoluten oder der relativen Häufigkeit einer Klasse zur entsprechenden Klassenbreite ist. Genauso lässt sich die Häufigkeitsdichte auch berechnen:

$$h(x) = \frac{\text{relative oder absolute Häufigkeit}}{\text{obere Grenze der Klasse } i - \text{untere Grenze der Klasse } i}$$

Das Integral über die relative Häufigkeitsdichte ist immer auf 1 normiert, bzw auf 100% im Falle der absoluten Häufigkeitsdichte.

Im nachstehenden Code-Block wird die relative Häufigkeitsdichte für die obenstehende Messreihe geplottet und das Integral berechnet.



Integral über die Häufigkeitsdichte: 1.0

Mit größerer mathematischer Schärfe formuliert man: Die Wahrscheinlichkeit, einen Messwert innerhalb eines kleinen Intervalls Δx um den Wert \bar{x} zu finden, ist $dP(X) = p(x) \cdot \Delta x$. Je größer die Stichprobe m (Anzahl der Messungen), desto eher erkennt man die zugrundeliegende Verteilung, die der Messreihe unterliegt. Häufig handelt es sich in der Praxis um eine *Normalverteilung* (oder auch Gaußverteilung genannt):

$$P(x) = \frac{1}{s\sqrt{2\pi}} \int_{x_1}^{x_2} \exp\left(-\frac{(x-\bar{x})^2}{2s^2}\right) dx$$

Dies ist verblüffender Weise auch immer noch dann der Fall, wenn sehr viele externe Störungen (evtl. mit unterschiedlichen Verteilungsfunktionen) zu einer gemeinsamen Störgröße kombiniert werden. Die zusammengefasste Störung ist trotzdem fast immer gaußverteilt, egal die Einzelverteilungen aussehen (Poisson oder anderes). Dies wird auch als der **zentrale Grenzwertsatz der Wahrscheinlichkeitstheorie** bezeichnet.

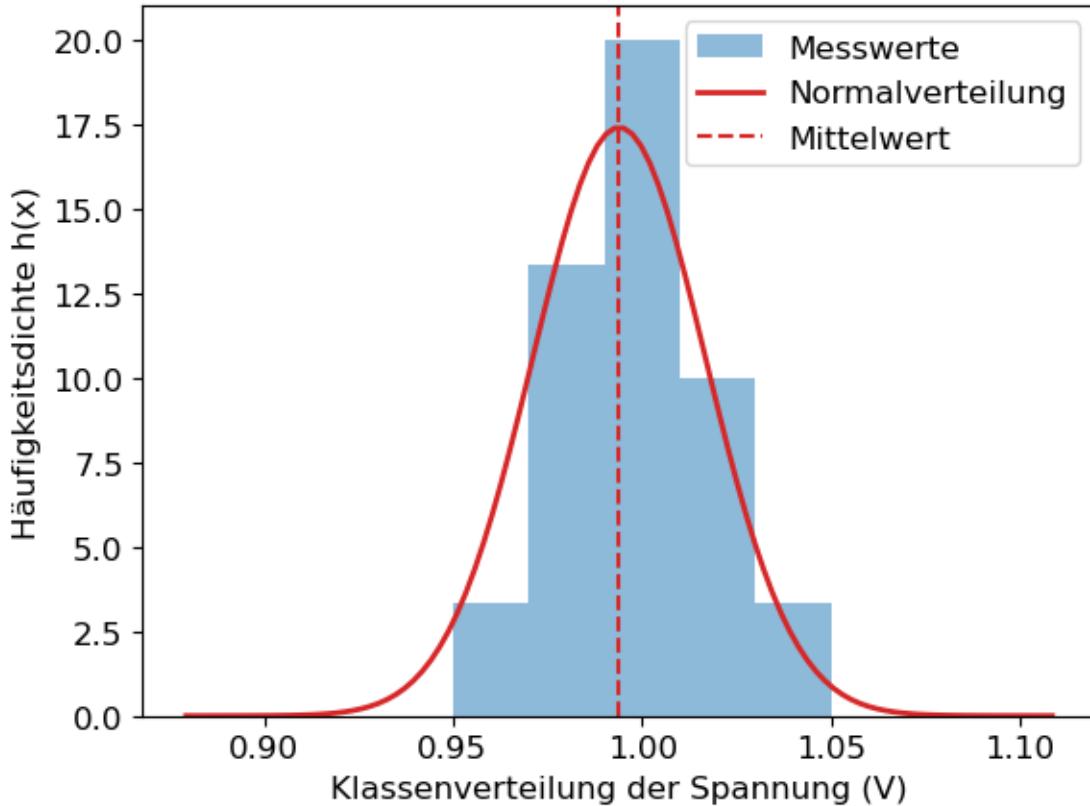
Zentraler Grenzwertsatz der Wahrscheinlichkeitstheorie: Der Durchschnitt einer großen Anzahl von Zufallsvariablen aus derselben Verteilung sind annähernd normalverteilt, unabhängig von der Verteilungsfunktion aus der sie herausgenommen wurden.

- Normalverteilte Zufallsgrößen werden also von den beiden Parametern \bar{x} und s beschrieben. Der **arithmetische Mittelwert**, der das **arithmetische Mittel** aus m Beobachtungen ist: $\bar{x} = \frac{1}{m} \sum_{j=1}^m x_j$. Erwartungswert findet man in der Literatur unterschiedliche Bezeichnungen, unter anderem zum Beispiel $x = E(x) = \langle x \rangle = \mu$.
- Der *Erwartungswert der quadratischen Abweichung der Einzelmessungen vom Mittelwert*, die **Varianz** s^2 , lässt sich allgemein wie folgt schreiben: $s^2 = \frac{1}{m} \sum_{j=1}^m (x_j - \mu)^2$. Der arithmetische Mittelwert zeichnet sich dadurch aus, dass für diesen Wert die Summe der Abweichungsquadrate minimal ist. Die Varianz hängt nicht von der Anzahl der Messungen ab. Die Streuung kann allein durch ein besseres Messverfahren verkleinert werden. Anschaulich ist das direkt nachvollziehbar: Die „Punktwolke“ der Messergebnisse um den Mittelwert schmiegt sich nicht enger an den Mittelwert, nur weil häufiger gemessen wurde. Ein stark streuendes Messverfahren streut durch seine Wiederholung nicht weniger.

- Unabhängig von der zugrundeliegenden Verteilung der Messwerte kann nun ein Maß für die Abweichung definiert werden, welche als **empirische Standardabweichung der Einzelmessungen** bekannt ist und sich aus der Quadratwurzel der Varianz berechnen lässt: $\sigma = \sqrt{\frac{1}{m} \sum_{j=1}^m (x_j - \mu)^2}$

Man nehme beispielhaft die Messung einer Spannung. Die Messreihe ist im nachstehenden Bode-Block gegeben. Es wurden 15 wiederholte Messungen durchgeführt in denen 15x der Wert 1V gemessen werden sollte. Mittelwert, Standardabweichung der Einzelmessungen und Unsicherheit des Mittelwertes werden berechnet.

Mittelwert der Messreihe: 0.994 V
 Standardabweichung der Messungen: 0.02293 V



- Du wirst bei deinen Messungen in der Regel weniger an der Streuung um den Mittelwert sondern mehr an der (geschätzten) Streuung der Messwerte um den (unbekannten) *wahren* Wert interessiert sein. Man schätzt diese Unsicherheit durch die empirische Varianz $s^2(x)$ der Messwerte der x_j ab. Diese ist etwas größer, um den Faktor $m/(m-1)$: $s^2 = \frac{1}{m-1} \sum_{j=1}^m (x_j - \bar{x})^2$
- Die empirische Standardabweichung $s(x)$ der Messwerte ist die Wurzel aus der empirischen Varianz: $s = \sqrt{\frac{1}{m-1} \sum_{j=1}^m (x_j - \bar{x})^2}$

Welche der Größen s oder σ du sinnvoll verwendest, hängt vom Einzelfall ab – wichtig ist, dass du dazu schreibst, welcher Wert verwendet wurde, damit die Leser die Argumentation nachvollziehen kann. Ein wenig spricht für die Verwendung s , da damit auch der etwas seltsame Fall einer Einzelmessung abgedeckt wird. Für $m = 1$ wäre $\sigma = 0$, s dagegen nicht definiert. Die zweite Aussage ist, bezogen auf die statistische Interpretation, sicher sinnvoller. Deshalb wird in diesem Skript s verwendet.

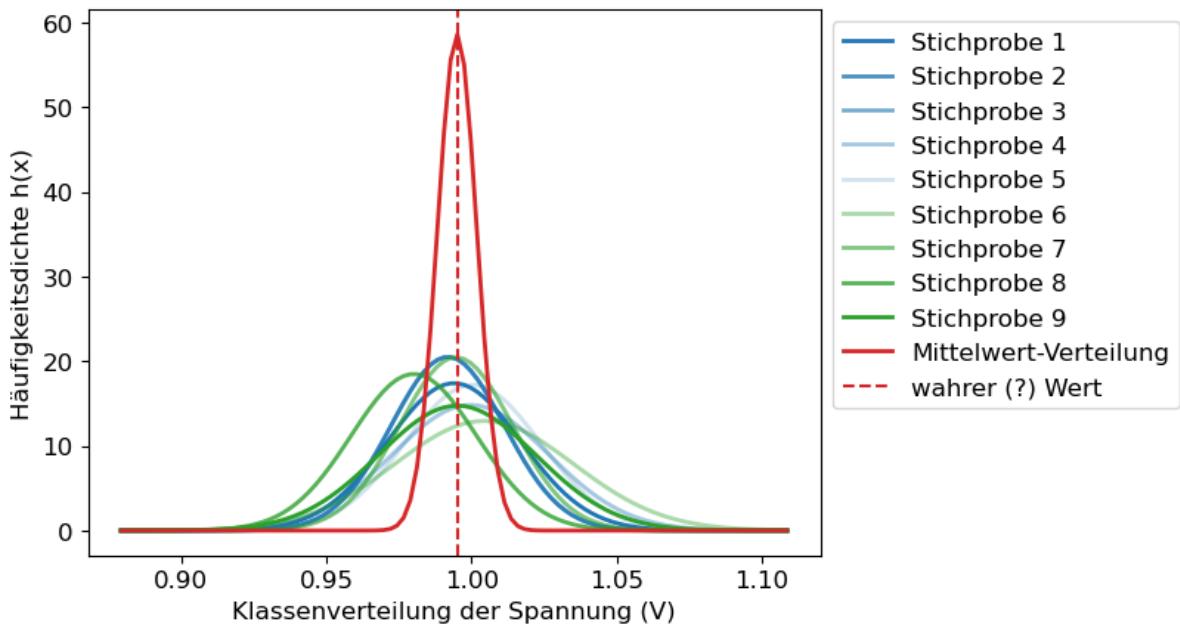
Auch die Messunsicherheit des Mittelwertes selbst, $u(\bar{x})$, kann natürlich kritisch bewertet werden: Wie wirkt sich die zu erwartende Messunsicherheit der einzelnen Messwerte $u(x)$ auf die Unsicherheit des Mittelwerts $u(\bar{x})$ der Messreihe aus? Laut *Grenzsetsatz* sind folglich auch die Mittelwerte (sollte man mehrere Stichproben aufnehmen) normalverteilt. Das

heißt aus den verschiedenen Mittelwerten von k Stichproben könnte theoretisch wieder ein Mittelwert berechnet werden. Außerdem kann analog die Standardabweichung der Mittelwerte berechnet werden:

$$s(\bar{x}) = \frac{s}{\sqrt{m}} = u_{\bar{x}} = \sqrt{\frac{1}{m(m-1)} \sum_{j=1}^m (x_j - \bar{x})^2}$$

Der Grenzwertsatz sagt aus, wenn man eine ausreichend große Stichprobe aus einer *Grundgesamtheit* nimmt, so wird der **Mittelwert der Stichprobe** näherungsweise dem **Mittelwert der Grundgesamtheit** entsprechen und sich somit dem *wahren* Wert annähern. Die **Unsicherheit des Mittelwertes** einer Messreihe ist gegenüber der Abweichung der Einzelmessungen um den Faktor \sqrt{m} reduziert. Der Grund dafür ist, dass sich statistisch ermittelte Unsicherheiten teilweise herausmitteln.

Bei einer großen Anzahl Messungen (>30) werden Sie stets finden, dass etwa 68% der Messungen im Intervall $\pm s(x)$ um den Mittelwert der Messreihe liegen. Würden Sie den wahren Wert kennen, könnten Sie weiter herausfinden, dass etwa 68% der Messwerte im Intervall $\pm s(x)$ um den wahren Wert und 68 % der Mittelwerte im Intervall $\pm s(\bar{x})$ um den wahren Wert liegen.



4.3.2 Vertrauensintervalle

Wir haben eben bereits erwähnt, dass 68% der Messwerte innerhalb des Intervalls $\pm s(x)$ liegen. Bei bekannter Wahrscheinlichkeitsverteilung $p(x)$ der Messwerte x um den Mittelwert \bar{x} , lässt sich die Wahrscheinlichkeit dafür berechnen, einen Messwert im Intervall $[x_1, x_2]$ um den Mittelwert zu finden. Wir nehmen im Folgenden eine Normalverteilung, mit Standardabweichung σ , der Messwerte an, dann ist die Wahrscheinlichkeit für

- einen Messwert $x \pm dx$:
$$h(x)dx = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right) dx$$
- irgendeinen Messwert zwischen $\pm\infty$:
$$P(-\infty < x < \infty) = \int_{-\infty}^{\infty} h(x)dx = 1$$
- einen Messwert im Intervall $[x_1, x_2]$:
$$P(x_1 < x < x_2) = \int_{x_1}^{x_2} h(x)dx = \frac{1}{\sigma\sqrt{2\pi}} \int_{x_1}^{x_2} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right) dx$$

Das hier auftretende Integral ist nicht elementar berechenbar und man findet statt dessen Tabellen, die man hierfür benutzen kann. Mit $x_1 = \mu - r \cdot \sigma$ und $x_2 = \mu + r \cdot \sigma$ findet man die Wahrscheinlichkeit $P(\mu - r \cdot \sigma \leq x \leq \mu + r \cdot \sigma)$ dafür, dass der Messwert innerhalb einer $r \cdot \sigma$ -Umgebung um den Mittelwert liegt. Bei einer echten Normalverteilung gilt folgendes:

- 68,3% aller Messwerte liegen im Bereich $\pm\sigma$
- 95,5% aller Messwerte liegen im Bereich $\pm 2\sigma$
- 99,7% aller Messwerte liegen im Bereich $\pm 3\sigma$
- Im Abstand $\pm\sigma$ sind die Wendestellen
- Die Normalverteilung reicht von $-\infty$ bis $+\infty$

Durch die Intervallgrenzen dieser Verteilung werden Güteklassen von Messeinrichtungen definiert. Andersherum können auch Anforderungen an Messgeräte gestellt werden: Die Anforderungen werden umso höher, je höher die Wahrscheinlichkeit sein soll, dass sich die Messwerte dem *richtigen* Wert annähern. Der Messtechniker kann somit mit der gegebenen Wahrscheinlichkeit abschätzen, ob ermittelte Messwerte innerhalb einer durch Fertigungsunterlagen zugelassenen Toleranz für die Maße eines Werkstücks liegen. Übliche Werte für Intervallgrenzen (meist symmetrisch, also Abweichungen sowohl nach unten als auch nach oben im gleichen Maße) und die zugehörigen Wahrscheinlichkeiten sind der nachfolgenden Tabelle angegeben:

Intervallgrenzen	Transformation	$P(x) = \phi(z)$	Bezeichnung	Wahrscheinlichkeit
$\mu \pm 1 \cdot \sigma$	1	0.6827	Orientierende Messung	68.27 %
$\mu \pm 1,96 \cdot \sigma$	1.96	0.95	Betriebsmessung	95.00 %
$\mu \pm 2 \cdot \sigma$	2	0.9545		95.45 %
$\mu \pm 2,58 \cdot \sigma$	2.58	0.99		99.00 %
$\mu \pm 3 \cdot \sigma$	3	0.9973	Präzisionsmessung	99.73 %
$\mu \pm 4 \cdot \sigma$	4	0.9999		99.99 %

Als Messtechniker gehen wir immer daher davon aus, dass innerhalb $\pm 3\sigma$ alle Messwerte liegen. Auf dieser Basis wird entsprechend auch die Messabweichung berechnet, also $A = \pm 3\sigma$.

Das **Endergebnis** der oben dargestellten Messreihe von $m = 15$ Messwerten wird in der Regel wiefolgt angegeben. Als Messwert wird nicht das Ergebnis einer Einzelmessung angegeben, sondern stets der Mittelwert der Messreihe inkl. seiner Unsicherheit: $s(\bar{x}) = \frac{s}{\sqrt{m}} = u_{\bar{x}}$

Der Mittelwert ist der beste Schätzwert, den wir für den *wahren* Wert ermitteln können. Die Angabe des Messergebnisses erfolgt also wiefolgt:

$$x = \bar{x} \pm r \cdot u_{\bar{x}}$$

wobei r ein Maß für den **Vertrauensbereich** v ist:

$$v = \pm \frac{r}{\sqrt{m}} \cdot s = u_v$$

Der Vertrauensbereich beschreibt die Aufspreizung des Mittelwertes einer Messreihe zu einem Vertrauensband, das umso breiter ist, je weniger Messwerte zur Auswertung zur Verfügung stehen und je größer das geforderte Vertrauenniveau ist. Der Vertrauensbereich gilt als Qualitätsmaß für die Genauigkeit einer durchgeführten Messung. Für $r = 1$ wählen wir also die $\pm 1\sigma$ -Umgebung, in der 68% der Messwerte liegen. Für $r = 2$, also die $\pm 2\sigma$ -Umgebung, werden schon 95% aller Messwerte in diesem Bereich erwartet. Die Messabweichung ist dadurch erhöht, das Vertrauen allerdings auch.

Mittelwert der Messreihe: 0.994 V
 Standardabweichung der Messungen: 0.023 V
 Abweichung des Mittelwertes: 0.006 V

 Messergebnis (95%): (0.994 + - 0.012) V

4.3.3 Korrektur bei kleinen Stichproben: Student-t

Ist die Zahl der Messwerte nur klein ($m \leq 25$) werden die aus der Normalverteilung berechneten Parameter ziemlich unsicher. Anfang des 20. Jh. veröffentlichte WILLIAM SEALY GOSSET unter dem Pseudonym „Student“ eine Verteilungsfunktion, die eine zuverlässigere Parameterschätzung auch für kleine Stichprobengrößen erlaubt. Für die Messwertanalyse ist folgender Teilaспект von Bedeutung: Aus der Zahl m der Messwerte und einem vorgegebenen Vertrauensbereich berechnen Sie einen Faktor $t(s = m - 1, p = 1 - \alpha/2)$, der die Unsicherheitsintervalle aus der Normalverteilung korrigiert. Die unten stehende Quantil-Tabelle zeigt die zugehörigen Werte von t in Abhängigkeit von der Messwertanzahl, und dem gewählten Vertrauensniveau, dass die geforderte statistische Sicherheit beschreibt:

s	p	0,9	0,95	0,975	0,98	0,99	0,995
1		3,078	6,314	12,706	15,895	31,821	63,657
2		1,886	2,920	4,303	4,849	6,965	9,925
3		1,638	2,353	3,182	3,482	4,541	5,841
4		1,533	2,132	2,776	2,999	3,747	4,604
5		1,476	2,015	2,571	2,757	3,365	4,032
6		1,440	1,943	2,447	2,612	3,143	3,707
7		1,415	1,895	2,365	2,517	2,998	3,499
8		1,397	1,860	2,306	2,449	2,896	3,355
9		1,383	1,833	2,262	2,398	2,821	3,250
10		1,372	1,812	2,228	2,359	2,764	3,169
11		1,363	1,796	2,201	2,328	2,718	3,106
12		1,356	1,782	2,179	2,303	2,681	3,055
13		1,350	1,771	2,160	2,282	2,650	3,012
14		1,345	1,761	2,145	2,264	2,624	2,977
15		1,341	1,753	2,131	2,249	2,602	2,947
16		1,337	1,746	2,120	2,235	2,583	2,921
17		1,333	1,740	2,110	2,224	2,567	2,898
18		1,330	1,734	2,101	2,214	2,552	2,878
19		1,328	1,729	2,093	2,205	2,539	2,861
20		1,325	1,725	2,086	2,197	2,528	2,845
21		1,323	1,721	2,080	2,189	2,518	2,831

Vertrauensbereich:
 $v = \pm \frac{t}{\sqrt{m}} \cdot s = u_v$

**Umrechnung von
Vertrauensintervallen:**
 $\frac{u_{\alpha 1}}{t_{m-1;1-\frac{\alpha_1}{2}}} = \frac{u_{\alpha 2}}{t_{m-1;1-\frac{\alpha_2}{2}}}$

Die Interpretation der Quantil-Tabelle der Student-t Verteilung kann verwirrend sein. Statt der Anzahl der Messwerte m wird die Anzahl der Freiheitsgrade $s = m - 1$ angegeben. Und statt des zweiseitigen Vertrauensbereich $P = 1 - \alpha$ wird der halbseitige Vertrauensbereich $p = 1 - \alpha/2$ gewählt. Beide *Quantile* können aber über die eben angegebenen Formel einfach bestimmt werden.

Aus der empirischen Standardabweichung des Mittelwertes $s(\bar{x})$ berechnet man beispielsweise: $u(\bar{x}) = t(s, p) \cdot s(\bar{x}) = t(s, p) \cdot \frac{s(\bar{x})}{\sqrt{m}}$. Des Weiteren kennt man den Vertrauensbereich für unterschiedliche Wahrscheinlichkeiten in einer Tabelle. Umgerechnet werden die Formeln:

Hierbei ist α das Signifikanzniveau, also die Irrtumswahrscheinlichkeit $\alpha = 1 - P$, wobei P die Wahrscheinlichkeit bzw. die geforderte statistische Sicherheit ist.

Beispiel: Student-t Verteilung

Für die Spannung-Messreihe aus dem oben aufgeführten Beispiel waren folgende Parameter gegeben:

- Anzahl der Messwerte $m = 15$
- Mittelwert: $\bar{U} = 0,994$ V
- Standardabweichung des Mittelwertes: $u_{\bar{U}} = 0,006$ V

Oben hatten wir das Ergebnis, in dem 95% der Messwerte zu finden sind, wie folgt angegeben gehabt, in dem wir die Gaußverteilung und den zugehörigen 2σ -Vertrauensbereich benutzt haben:

$$U = (0,994 \pm 2 \cdot 0,006) \text{ V} = (0,994 \pm 0,012) \text{ V} \quad (95\%)$$

In Anbetracht der sehr kleinen Stichprobe von lediglich $m = 15$ Messwerten sollte jedoch die Student-t Verteilung hinzugezogen werden und der Vertrauensbereich für 95% korrigiert werden. Es gilt also:

$$P = 1 - \alpha = 0,95. \text{ Daraus folgt:}$$

$$\Rightarrow \alpha = 1 - P = 1 - 0,95 = 0,05$$

$$\Rightarrow \alpha/2 = 0,025$$

$$\Rightarrow p = 1 - \alpha/2 = 1 - 0,025 = 0,975$$

Für die Berechnung des s-Quantils gilt:

$$s = m - 1 = 15 - 1 = 14$$

Der t -Wert des korrigierten Vertrauebnsbereichs wird aus der Tabelle abgelesen:

$$t_{s;p} = t_{m-1;1-\alpha/2} = t_{14;0,975} = 2,145$$

Der Unterschied zum Vertrauensbereich, der aus der Normalverteilung hervorgeht, ist ein über 7% höherer Fehler.

```
print((2.145-2)/2*100, '%')
```

```
7.250000000000001 %
```

Die Umrechnung dieses Vertrauensbereich in einen Vertrauensbereich mit einem anderen Vertrauenniveau, z.B. von 95% zu 99%, wird im folgenden anhand dieses Beispiels verdeutlicht. Die Ergebnisse für 95% sind bekannt. Nun muss das Quantil für 99% (also $\alpha = 1\%$) bestimmt werden. Analog werden die $s = m - 1 = 14$ und $p = 1 - \alpha/2 = 1 - 0,005 = 0,995$ Quantile bestimmt und der t -Wert aus der Tabelle abgelesen:

$$t_{s;p} = t_{m-1;1-\alpha/2} = t_{14;0,995} = 2,977$$

Hieraus kann nun nach obiger Gleichung der Vertrauensbereich für 99% berechnet werden:

$$\begin{aligned} \frac{u_{\alpha 1}}{t_{m-1;1-\frac{\alpha_1}{2}}} &= \frac{u_{\alpha 2}}{t_{m-1;1-\frac{\alpha_2}{2}}} \\ \Rightarrow u_{\alpha 2} &= u_{\alpha 1} \cdot \frac{t_{m-1;1-\frac{\alpha_2}{2}}}{t_{m-1;1-\frac{\alpha_1}{2}}} = 0,013 \text{ V} \cdot 2,977 / 2,145 = 0,018 \text{ V} \end{aligned}$$

Die Angabe des Messergebnisses lautet somit:

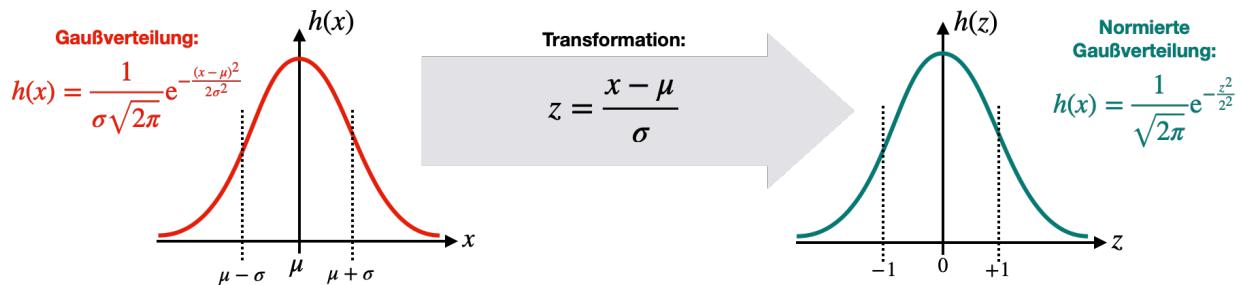
$$U = (0,994 \pm 0,018) \text{ V} \quad (99\%)$$

```
print(0.013*2.977/2.145)
```

```
0.01804242424242424
```

4.3.4 Normierte Normalverteilung

Jede Normalverteilung kann in eine normierte Gaußverteilung transformiert werden. Der Übergang erfolgt mit einer Transformation, die die Messwerte normiert. Dadurch können die Verteilungen verschiedener Messwerte miteinander verglichen werden, unabhängig von deren physikalischen Einheit.



Die Differenz $x - \mu$ wird auf die Standardabweichung σ normiert. Dadurch erhält man folgende Funktion für $\mu = 0$ und $\sigma = 1$ und -1 :

$$h(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

Der Flächeninhalt unterhalb dieser Verteilungsfunktion ist wieder auf 100% normiert, wenn von $\pm\infty$ integriert wird. Die Maximale Amplitude ist jetzt $1/\sqrt{2\pi}$. Die z -Achse (vorher Messwerte x) ist jetzt dimensionslos (einheitenlos). Somit können mit normierten Gaußverteilungen verschiedene physikalische Größen auch von verschiedenen Messgeräten bezüglich ihrer Streuung verglichen werden. Folgende Tabelle zeigt die Wahrscheinlichkeiten für verschiedene z -Werte. Die Spalten geben die erste Nachkommastelle von z an, die Zeilen die zweite:

z	0	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,50000	0,50399	0,50798	0,51197	0,51595	0,51994	0,52392	0,52790	0,53188	0,53586
0,1	0,53983	0,54380	0,54776	0,55172	0,55567	0,55962	0,56356	0,56749	0,57142	0,57535
0,2	0,57926	0,58317	0,58706	0,59095	0,59483	0,59871	0,60257	0,60642	0,61026	0,61409
0,3	0,61791	0,62172	0,62552	0,62930	0,63307	0,63683	0,64058	0,64431	0,64803	0,65173
0,4	0,65542	0,65910	0,66276	0,66640	0,67003	0,67364	0,67724	0,68082	0,68439	0,68793
0,5	0,69146	0,69497	0,69847	0,70194	0,70540	0,70884	0,71226	0,71566	0,71904	0,72240
0,6	0,72575	0,72907	0,73237	0,73565	0,73891	0,74215	0,74537	0,74857	0,75175	0,75490
0,7	0,75804	0,76115	0,76424	0,76730	0,77035	0,77337	0,77637	0,77935	0,78230	0,78524
0,8	0,78814	0,79103	0,79389	0,79673	0,79955	0,80234	0,80511	0,80785	0,81057	0,81327
0,9	0,81594	0,81859	0,82121	0,82381	0,82639	0,82894	0,83147	0,83398	0,83646	0,83891
1,0	0,84134	0,84375	0,84614	0,84849	0,85083	0,85314	0,85543	0,85769	0,85993	0,86214
1,1	0,86433	0,86650	0,86864	0,87076	0,87286	0,87493	0,87698	0,87900	0,88100	0,88298
1,2	0,88493	0,88686	0,88877	0,89065	0,89251	0,89435	0,89617	0,89796	0,89973	0,90147
1,3	0,90320	0,90490	0,90658	0,90824	0,90988	0,91149	0,91309	0,91466	0,91621	0,91774
1,4	0,91924	0,92073	0,92220	0,92364	0,92507	0,92647	0,92785	0,92922	0,93056	0,93189
1,5	0,93319	0,93448	0,93574	0,93699	0,93822	0,93943	0,94062	0,94179	0,94295	0,94408
1,6	0,94520	0,94630	0,94738	0,94845	0,94950	0,95053	0,95154	0,95254	0,95352	0,95449
1,7	0,95543	0,95637	0,95728	0,95818	0,95907	0,95994	0,96080	0,96164	0,96246	0,96327
1,8	0,96407	0,96485	0,96562	0,96638	0,96712	0,96784	0,96856	0,96926	0,96995	0,97062
1,9	0,97128	0,97193	0,97257	0,97320	0,97381	0,97441	0,97500	0,97558	0,97615	0,97670
2,0	0,97725	0,97778	0,97831	0,97882	0,97932	0,97982	0,98030	0,98077	0,98124	0,98169
2,1	0,98214	0,98257	0,98300	0,98341	0,98382	0,98422	0,98461	0,98500	0,98537	0,98574
2,2	0,98610	0,98645	0,98679	0,98713	0,98745	0,98778	0,98809	0,98840	0,98870	0,98899
2,3	0,98928	0,98956	0,98983	0,99010	0,99036	0,99061	0,99086	0,99111	0,99134	0,99158
2,4	0,99180	0,99202	0,99224	0,99245	0,99266	0,99286	0,99305	0,99324	0,99343	0,99361
2,5	0,99379	0,99396	0,99413	0,99430	0,99446	0,99461	0,99477	0,99492	0,99506	0,99520
2,6	0,99534	0,99547	0,99560	0,99573	0,99585	0,99598	0,99609	0,99621	0,99632	0,99643
2,7	0,99653	0,99664	0,99674	0,99683	0,99693	0,99702	0,99711	0,99720	0,99728	0,99736
2,8	0,99744	0,99752	0,99760	0,99767	0,99774	0,99781	0,99788	0,99795	0,99801	0,99807
2,9	0,99813	0,99819	0,99825	0,99831	0,99836	0,99841	0,99846	0,99851	0,99856	0,99861
3,0	0,99865	0,99869	0,99874	0,99878	0,99882	0,99886	0,99889	0,99893	0,99896	0,99900
3,1	0,99903	0,99906	0,99910	0,99913	0,99916	0,99918	0,99921	0,99924	0,99926	0,99929
3,2	0,99931	0,99934	0,99936	0,99938	0,99940	0,99942	0,99944	0,99946	0,99948	0,99950
3,3	0,99952	0,99953	0,99955	0,99957	0,99958	0,99960	0,99961	0,99962	0,99964	0,99965
3,4	0,99966	0,99968	0,99969	0,99970	0,99971	0,99972	0,99973	0,99974	0,99975	0,99976
3,5	0,99977	0,99978	0,99978	0,99979	0,99980	0,99981	0,99981	0,99982	0,99983	0,99983
3,6	0,99984	0,99985	0,99985	0,99986	0,99986	0,99987	0,99987	0,99988	0,99988	0,99989
3,7	0,99989	0,99990	0,99990	0,99990	0,99991	0,99991	0,99992	0,99992	0,99992	0,99992
3,8	0,99993	0,99993	0,99993	0,99994	0,99994	0,99994	0,99994	0,99995	0,99995	0,99995
3,9	0,99995	0,99995	0,99996	0,99996	0,99996	0,99996	0,99996	0,99996	0,99997	0,99997
4,0	0,99997	0,99997	0,99997	0,99997	0,99997	0,99997	0,99998	0,99998	0,99998	0,99998

Beispiel: Wahrscheinlichkeit für Zutreffen eines Ereignisses mittels Gauß-Test

Als wichtigste Erkenntnis gilt es festzuhalten, dass zu jeder Aussage zu zufälligen Abweichungen die zugehörige Wahrscheinlichkeit für das Zutreffen dieser Aussage zwingend erforderlich ist. Dies sollte immer mit angegeben werden, wenn die Messabweichung angegeben wird. Messwertangaben ohne Aussage zur Wahrscheinlichkeit bezüglich der zufälligen Abweichungen sind in der betrieblichen Praxis nicht brauchbar!

Es soll im Folgenden ein Messsystem einer Abfüllanlage überprüft werden, die Flaschen sind jeweils mit 0,7 l Saftgetränk befüllt. Aus Kalibrierungen ist bekannt, dass die Messwerte des Messsystems der Abfüllanlage normalverteilt mit einem Erwartungswert $\mu = 0,7$ l und Standardabweichung $\sigma = 5$ ml sind. Wie groß ist die Wahrscheinlichkeit, dass die Saftflaschen mit einem Inhalt zwischen 0,69 l und 0,71 l befüllt werden?

Lösung:

- Obere und untere Grenze werden in den z -Wert umgerechnet:

$$z_o = \frac{x-\mu}{\sigma} = \frac{(0,69-0,7)l}{0,005l} = -2$$

$$z_u = \frac{x-\mu}{\sigma} = \frac{(0,71-0,7)l}{0,005l} = 2$$

- Aus der Tabelle werden die Werte für $z = 2,00$ abgelesen:

$\phi(2) = 0,97725$ (Wahrscheinlichkeit dafür, dass der Messwert zwischen 0 l und 0,71 l liegt)

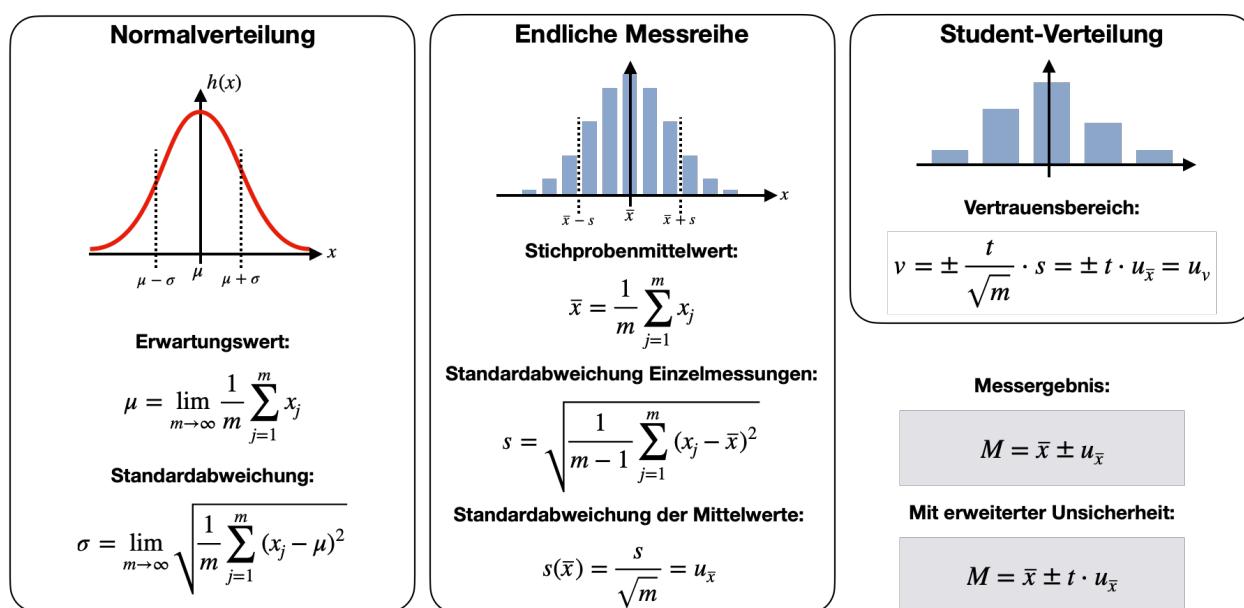
$\phi(-2) = 1 - \phi(2) = 1 - 0,97725 = 0,02275$ (Wahrscheinlichkeit dafür, dass der Messwert NICHT zwischen 0 l und 0,69 l liegt)

- Berechnung der Wahrscheinlichkeit, dass die Saftflaschen mit einem Inhalt zwischen 0,69 l und 0,71 l befüllt werden:

$$\phi(2) - \phi(-2) = 0,97725 - 0,02275 = 0,9545$$

Mit einer Wahrscheinlichkeit von 95,45 % werden die Flaschen in der Abfüllanlage mit einem Inhalt von 0,69 l – 0,71 l befüllt. Dies entspricht auch genau der Wahrscheinlichkeit der 2σ -Umgebung (siehe vorheriges Kapitel), was für $z = \pm 2$ natürlich auch so sein sollte.

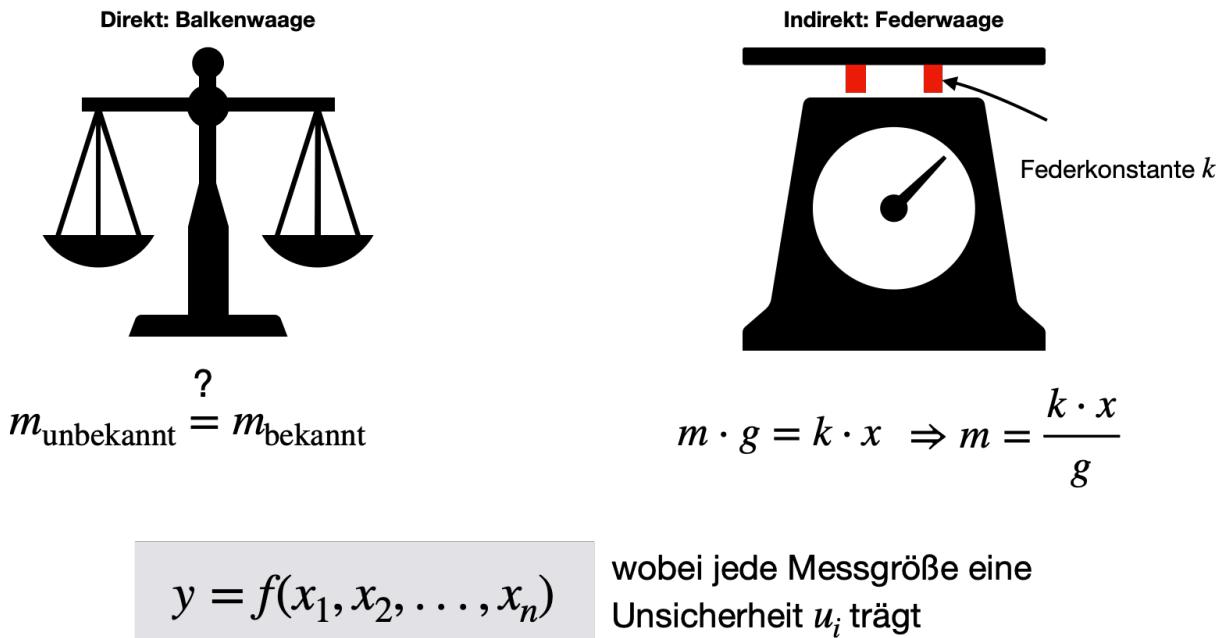
4.3.5 Zusammenfassung zufälliger Unsicherheiten



4.4 Fortpflanzung von Messunsicherheiten

In der Messtechnik gibt es zwei Möglichkeiten eines Messprinzips:

- Das **direkte Messverfahren**: der gesuchte Messwert wird unmittelbar durch den Vergleich mit einem Bezugswert derselben Messgröße gewonnen. Als Beispiel eignet sich hier die Balkenwaage, die die unbekannte Masse m mit der bekannten Masse eines Gewichtssteins vergleicht.
- Das **indirekt Messverfahren**: Die meisten physikalischen Größen werden so ermittelt, da sie nur indirekt ermittelt werden können. Die gesuchte Messgröße wird hierbei über physikalische Zusammenhänge auf andere Größen zurückgeführt und anschließend aus diesen ermittelt. Die Federwaage ist ein Beispiel hierfür, bei der eine unbekannte Masse m über die Auslenkung x einer Feder (Federkonstante k) ermittelt werden soll. Gravitationskraft wird der Rückstellkraft der Feder gleichgesetzt und man erhält folgenden Zusammenhang, wobei g die Schwerebeschleunigung ist.

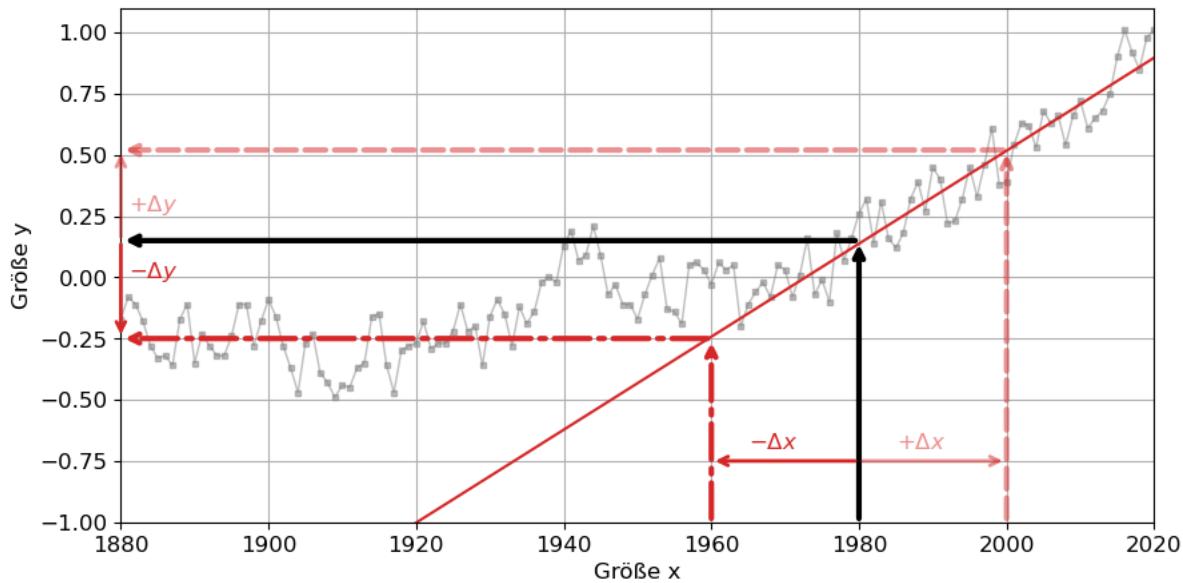


Durch dieses Beispiel wird klar, dass Messgrößen y oft nicht direkt ermittelt werden, sondern über funktionelle Zusammenhänge von n Messgrößen x_i bestimmt werden:

$$y = f(x_1, x_2, \dots, x_n)$$

Achtung, bei x_i handelt es sich nicht um Stichprobengrößen x_j , sondern um Messungen *unterschiedlicher* physikalischer Messgrößen, beispielsweise Federkonstante k und Auslenkung x).

Wie wirken sich Messunsicherheiten der Messgrößen x_i auf das Ergebnis y aus? Im folgenden Plot ist ersichtlich, dass eine Abweichung der Größe x zwangsläufig eine Abweichung der Größe y zur Folge hat:



Das Intervall $\pm \Delta x = 20$ im obigen Plot um 1980 drum herum wird durch *irgendeinen* funktionellen Zusammenhang $f(x)$ auf das Intervall Δy abgebildet. Durch den Messwert $x_0 = 1980$ kann eine lineare Ausgleichsgerade gezogen werden, deren Steigung der Ableitung der Funktion in diesem Punkt entspricht, $df(x_0)/dx$. Anhand der Steigung der linearen Ausgleichsgeraden kann man den Fehler für y direkt ablesen: $\Delta y \approx 0,375$. Man sieht jedoch in der Abbildung auch, dass diese Abschätzung umso schlechter wird, je größer Δx ist. Je nach Funktionstyp müssen also auch höhere Ableitungen berücksichtigt werden (*Taylorentwicklung* von $f(x)$).

4.4.1 Herleitung: Taylorentwicklung

Für eine allgemeine Funktion $f(x) = y(x)$ einer Zufallsgröße x lässt sich die Frage nach der Unsicherheit näherungsweise beantworten, wenn man die Taylor-Entwicklung von y an der Stelle \bar{x} um deren Messabweichung Δx herum entwickelt:

$$y = y(x) \Rightarrow y(x + \Delta x) = y(x) + \frac{1}{1!} \frac{dy(x)}{dx} \cdot \Delta x + \frac{1}{2!} \frac{d^2y(x)}{dx^2} \cdot (\Delta x)^2 + \dots$$

Da die Unsicherheit typischerweise eine kleine Größe ist, wird die Reihenentwicklung nach dem linearen Glied abgebrochen, da höhere Ordnung von $(\Delta x)^2$ recht klein werden. Damit ergibt sich die Näherung:

$$y(x + \Delta x) = y(x) + \frac{1}{1!} \frac{dy(x)}{dx} \cdot \Delta x \Rightarrow y(x + \Delta x) - y(x) = \Delta y = \frac{dy(x)}{dx} \cdot \Delta x$$

Diese Formel gilt, wenn das Messergebnis von einer einzelnen Zufallsgröße abhängt. Kommen weitere Messgrößen dazu, welche voneinander unabhängig sind, wird die entsprechende Reihenentwicklung verwendet, welche ebenfalls bis zum linearen Glied als Näherung für kleine Δx angenommen werden kann:

$$y = y(x_1, x_2, \dots) \Rightarrow \Delta y = \frac{\partial y}{\partial x_1} \cdot \Delta x_1 + \frac{\partial y}{\partial x_2} \cdot \Delta x_2 + \dots$$

wobei hier der Term $\frac{\partial y}{\partial x_1}$ die *partielle* Ableitung der Funktion $y(x_1, x_2, \dots)$ nach der Größe x_1 bedeutet. Wird die partielle Ableitung nach x_1 berechnet, verhalten sich alle anderen Eingangsgrößen (x_2, \dots) wie eine Konstante.

4.4.2 Vorzeichen bekannt (systematische Abweichungen)

Ist das Vorzeichen der Messabweichung bekannt, müssen die Vorzeichen unbedingt berücksichtigt werden. Das Ergebnis wird anschließend um diesen *Offset* korrigiert und nicht mit dem \pm -Symbol, wie für *Messunsicherheiten* üblich versehen.

Die allgemeine Formel aus dem vorangegangenen Kapitel ist gültig und die Messabweichung für y berechnet sich zu:
 $\Rightarrow \Delta y = \frac{\partial y}{\partial x_1} \cdot \Delta x_1 + \frac{\partial y}{\partial x_2} \cdot \Delta x_2 + \dots$

Wichtige Spezialfälle sind Summen, Differenzen, Produkte oder Quotienten von zwei Größen. Beachte hier, dass wir in diesem Kapitel Fehler **mit Vorzeichen** betrachten, das heißt wir wissen, in welche Richtung der Messwert abweicht.

- Setzt sich die gesuchte Größe y aus der **Addition** zweier unabhängigen Messwerte zusammen, so wird ihre Abweichung aus der Addition der Abweichungen der Einzelmessungen berechnet:

$$y = x_1 + x_2 \Rightarrow \Delta y = \Delta x_1 + \Delta x_2$$

- Setzt sich die gesuchte Größe y aus der **Subtraktion** zweier unabhängigen Messwerte zusammen, so wird ihre Abweichung aus der Subtraktion der Abweichungen der Einzelmessungen berechnet:

$$y = x_1 - x_2 \Rightarrow \Delta y = \Delta x_1 - \Delta x_2$$

- Setzt sich die gesuchte Größe y aus der **Multiplikation** zweier unabhängigen Messwerte zusammen, so wird ihre relative Abweichung aus der Addition der relativen Abweichungen der Einzelmessungen berechnet:

$$y = x_1 \cdot x_2 \Rightarrow \frac{\Delta y}{y} = \frac{\Delta x_1}{x_1} + \frac{\Delta x_2}{x_2}$$

- Setzt sich die gesuchte Größe y aus der **Division** zweier unabhängigen Messwerte zusammen, so wird ihre relative Abweichung aus der Subtraktion der relativen Abweichungen der Einzelmessungen berechnet:

$$y = x_1/x_2 \Rightarrow \frac{\Delta y}{y} = \frac{\Delta x_1}{x_1} - \frac{\Delta x_2}{x_2}$$

Anmerkung: Hierbei werden keine Fehlergrenzen (mit \pm) angegeben, sondern systematische Messabweichungen mit bekanntem Vorzeichen. Bei Fehlergrenzen und statischen Unsicherheiten, gelten andere Sachverhalte (siehe nächsten Abschnitt)! Die Formeln gelten *nur*, wenn das Vorzeichen des Fehlers bekannt ist. Bei dieser Fehlerfortpflanzung können sich also Abweichungen ergänzen oder sogar *aufheben*, wie das folgende Beispiel zeigt.

Beispiel: Eine Messgröße x_1 wird um 2% zu klein gemessen und Messgröße x_2 um 3% zu groß.

Bei der Multiplikation $y = x_1 \cdot x_2$ wird y um 5% zu groß gemessen ($2\% + 3\% = 5\%$).

Bei der Division $y = x_1/x_2$ wird y um 1% zu klein gemessen ($2\% - 3\% = -1\%$).

4.4.3 Vorzeichen unbekannt (systematische unabhängige Abweichungen)

Die Größe der Messabweichung eines Messgerätes ist zwar betragsmäßig bekannt, das Vorzeichen jedoch nicht. Dies war auch der Fall in unserem obigen Beispiel der *Messgeräteabweichung*. Somit sind nur die Grenzen dieser Abweichung bekannt. Die gesuchte Abweichung Δy der Messgröße y kann aber über denselben mathematischen Ansatz wie eben ermittelt werden, wobei wir annehmen, dass sich die Abweichungen im schlimmsten Fall bei ungünstigen Vorzeichenkombinationen zu einem **Maximalfehler** addieren:

$$\Delta y = \left| \frac{\partial y}{\partial x_1} \right| \cdot \Delta x_1 + \left| \frac{\partial y}{\partial x_2} \right| \cdot \Delta x_2 + \dots$$

Die Spezialfälle vereinfachen sich aufgrund der Beträge zu folgenden Sachverhalten:

- Setzt sich die gesuchte Größe y aus der **Addition oder Subtraktion** zweier unabhängigen Messwerte zusammen, so wird ihre Abweichung aus der Addition der Abweichungen der Einzelmessungen berechnet:

$$y = x_1 \pm x_2 \Rightarrow \Delta y = \Delta x_1 + \Delta x_2$$

- Setzt sich die gesuchte Größe y aus der **Multiplikation oder Division** zweier unabhängigen Messwerte zusammen, so wird ihre relative Abweichung aus der Addition der relativen Abweichungen der Einzelmessungen berechnet:

$$y = x_1 \cdot x_2 \quad \text{oder} \quad y = x_1/x_2 \Rightarrow \frac{\Delta y}{y} = \frac{\Delta x_1}{x_1} + \frac{\Delta x_2}{x_2}$$

4.4.4 Zufällige, unabhängigen Variablen

Nun gehen wir über von Fehlerfortpflanzungsgesetzen für systematische Abweichungen zu Gesetzmäßigkeiten, die für rein statistische Messgrößen x gelten. Die Messgröße x wird wie oben eingeführt über statistische Verfahren ermittelt und im Rahmen einer Messreihe (mit m Messwerten) aufgenommen. Mittelwert und Unsicherheit werden also wie folgt berechnet:

$$\bar{x} = \frac{1}{m} \sum_{j=1}^m x_j$$

$$s(\bar{x}) = \frac{s}{\sqrt{m}} = \sqrt{\frac{1}{m(m-1)} \sum_{j=1}^m (x_j - \bar{x})^2} = u_{\bar{x}}$$

Für große Stichproben wird die Unsicherheit des Mittelwertes immer kleiner und strebt gegen Null, und bei Abwesenheit systematischer Fehler nähert sich der Mittelwert somit dem *wahren* Wert an.

Bei der Fehlerfortpflanzung wird als Eingangsgröße häufig der Mittelwert benutzt, und seine entsprechend kleine Unsicherheit benutzt, um die Unsicherheit für die Ausgangsgröße y zu bestimmen. Die Taylorreihe von $y(x)$ wird entsprechend an der Stelle des Mittelwertes $x = \bar{x}$ entwickelt und es folgt:

$$u_y = \left| \frac{dy}{dx} \right| \cdot u_{\bar{x}}$$

Da es sich bei statistischen Unsicherheiten um Grenzen handelt, also das Vorzeichen der Abweichung nicht existiert, müssen wieder deren Beträge berücksichtigt werden.

Haben wir nun den Fall, dass sich die gesuchte Größe y aus mehreren voneinander unabhängigen Eingangsgrößen $\bar{x}_1, \bar{x}_2, \dots$ und deren Unsicherheiten u_1, u_2, \dots zusammensetzt, gelten folgenden Regeln:

- Das Messergebnis y berechnet sich aus den Mittelwerten $\bar{x}_1, \bar{x}_2, \dots$
- Um die Unsicherheit u_y zu bestimmen, wird wieder mit der linearen Näherung bei mehreren unabhängigen Variablen begonnen (partielle Ableitungen bilden!), allerdings müssen jetzt, wie bei der Berechnung der Standardunsicherheit, die *quadrierten* Beiträge der Einzelunsicherheiten addiert werden:

$$u_y = \sqrt{\left(\frac{\partial y}{\partial x_1} \cdot u_1 \right)^2 + \left(\frac{\partial y}{\partial x_2} \cdot u_2 \right)^2 + \dots}$$

Dies nennt sich auch das **Gauß'sche Fehlerfortpflanzungsgesetz** und wird bei Unsicherheiten, aber nicht bei systematischen Messabweichungen / Fehlern verwendet.

Für die Spezialfälle gilt nun:

- Setzt sich die gesuchte Größe y aus der **Addition oder Subtraktion** zweier unabhängigen Messwerte zusammen, so wird ihre quadrierte Unsicherheit aus der Addition der quadrierten Unsicherheiten der Einzelmessungen berechnet:

$$y = \bar{x}_1 \pm \bar{x}_2 \Rightarrow u_y^2 = u_1^2 + u_2^2$$

- Setzt sich die gesuchte Größe y aus der **Multiplikation oder Division** zweier unabhängigen Messwerte zusammen, so wird ihre relative quadrierte Unsicherheit aus der Addition der relativen quadrierten Unsicherheiten der Einzelmessungen berechnet:

$$y = \bar{x}_1 \cdot \bar{x}_2 \quad \text{oder} \quad y = \bar{x}_1 / \bar{x}_2 \Rightarrow \left(\frac{u_y}{y} \right)^2 = \left(\frac{u_1}{\bar{x}_1} \right)^2 + \left(\frac{u_2}{\bar{x}_2} \right)^2$$

Hinweis: Diese Gesetze sind nur bei hinreichender Linearität anwendbar, d.h. wenn sich die Funktion $y(x_1, x_2, \dots)$ bei Änderung einer der Eingangsgrößen x_i im Bereich ihrer eigenen Unsicherheit u_i nur hinreichen linear verändert. Andernfalls wird die Fehlerfortpflanzung aufwendiger (DIN1319).

4.4.5 Zufällige, abhängige (korrelierte) Variablen

Korrelation und Kovarianz-Matrix

Häufig liegt in der Messtechnik der Fall vor, dass wir voneinander unabhängige Messungen betrachten und diese zu unserer gesuchten Messgröße kombinieren. Teilweise können aber auch Messungen beobachtet werden, welche eine Abhängigkeit voneinander aufweisen. In diesem Fall spricht man von **Korrelationen** zwischen Messgrößen und ein *Kovarianz-Term* muss berücksichtigt werden!

- Bei **unabhängigen (nicht-korrelierten)** Messungen wird der Kovarianz-Term auf Null gesetzt:

$$\text{cov}(x_1, x_2) = 0$$

- Bei **abhängigen (korrelierten)** Messungen besitzen die Messungen *gemeinsame* Unsicherheiten, wenn z.B. alle Messungen eine gemeinsame systematische Unsicherheit besitzen:

$$\text{cov}(x_1, x_2) = \underbrace{\langle x_1 x_2 \rangle}_{\substack{\text{zuerst Multiplikation,} \\ \text{dann Mittelwertbildung}}} - \underbrace{\langle x_1 \rangle \langle x_2 \rangle}_{\substack{\text{zuerst Mittelwertbildung,} \\ \text{dann Multiplikation}}}$$

Diese *gemeinsame* Unsicherheit resultiert darin, dass Abhängigkeiten zwischen den gemessenen Messwerten x_1 und x_2 entstehen. Ist zum Beispiel die eine Messgröße x_1 zu groß, so ist die Wahrscheinlichkeit, dass auch die andere Größe x_2 einen zu großen Wert hat, erhöht. Diesen Effekt, dass beide Größen zu große Werte ausgeben, nennt man auch **positive Korrelation**. Entsprechend würde eine **negative Korrelation** bedeuten, dass der zweite Wert einer Messgröße mit erhöhter Wahrscheinlichkeit zu klein ausgegeben wird, obwohl die erste Größe einen zu großen Wert hat. Ein Beispiel für eine negative Korrelation ist die Ausgleichsgerade $y = m \cdot x + b$, welche an Messwerte angepasst werden kann. Wenn sich der Wert von m vergrößert, zu muss zwangsläufig der y -Achsenabschnitt b kleiner werden, damit die Gleichung weiterhin erfüllt ist. Die Größen m und b sind also negativ korreliert. Sind hingegen zwei Variablen *unabhängig*, so kann man aus dem Wert der ersten *keine* Information über die zweite Variable gewinnen. In diesem Fall ist die Korrelation exakt Null.

- Im Prinzip kann die Kovarianz, und somit auch die Werte in der Kovarianzmatrix, jeden beliebigen Wert annehmen, was zu einem sehr großen Wertebereich führt. Um die Darstellung zu vereinfachen wird die Kovarianz typischerweise normiert, sodass die Kovarianz Werte zwischen -1 und $+1$ annimmt und sich der **Korrelationskoeffizient** ergibt:

$$(r_{ij}) = \left(\frac{s_{ij}^2}{s_i s_j} \right)$$

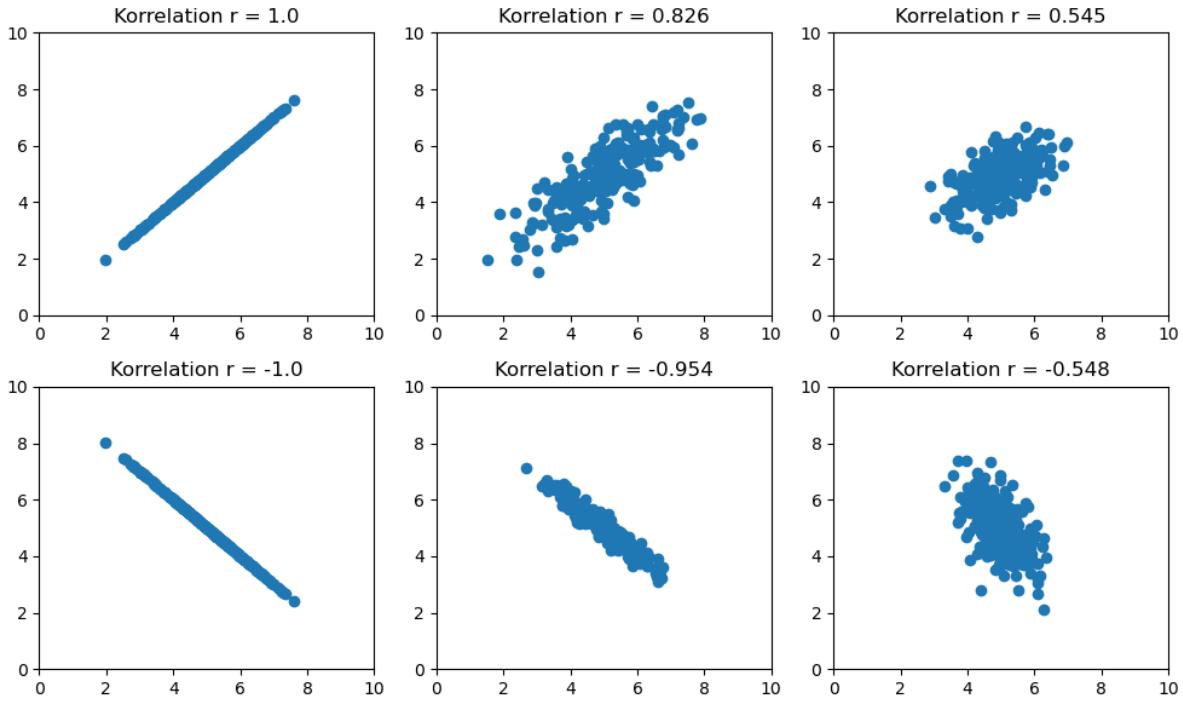
- Der Korrelationskoeffizient für nur zwei Variablen sieht in vereinfachter Schreibweise wie folgt aus:

$$r = \frac{\text{cov}_{x_1 x_2}}{s_{x_1} \cdot s_{x_2}} = \frac{\sum_{i=1}^N (x_{1,i} - \bar{x}_1) \cdot (x_{2,i} - \bar{x}_2)}{\sqrt{\sum_{i=1}^N (x_{1,i} - \bar{x}_1)^2} \cdot \sqrt{\sum_{i=1}^N (x_{2,i} - \bar{x}_2)^2}}$$

Wir haben nun verschiedene Kenngrößen der Statistik kennengelernt, um Beziehungen zwischen zwei Variablen zu beschreiben. Die **Kovarianz** ist eher ungeeignet um Größen miteinander zu vergleichen, da sie von der Skalierung abhängt. Der **Korrelationskoeffizient** hingegen ist eine normierte Größe und kann als Maß für Korrelationen benutzt werden. Allerdings nur für lineare Korrelationen! Außerdem muss auch hierbei Vorsicht gewahrt werden: Eine hoher Korrelationskoeffizient bedeutet nicht immer eine hohe Korrelation der Variablen. Es kann sich auch um Ausreißer handeln! Außerdem wird die Bedeutung der Korrelation

auch häufig überinterpretiert. Ein kleiner Korrelationskoeffizient bedeutet nämlich auch nicht notwendigerweise, dass es keinerlei Beziehung zwischen zwei Variablen gibt. Der Zusammenhang könnte ja ein anderer als linear sein, z.B. quadratisch. Dann würde die Korrelation trotz quadratischem Zusammenhang den Wert 0 ausgeben.

Zum besseren Verständnis mag ein Beispiel helfen. Wir erzeugen dazu Zufallsgrößen mit unabhängigen und gemeinsamen Fehlern.



Fehlerfortpflanzung

Das **Gauß'sche Fehlerfortpflanzungsgesetz** muss für diesen Fall erweitert werden, indem die Kovarianzen oder die Korrelationskoeffizienten zwischen jeweils zwei Größen, für alle N Messgrößen, berücksichtigt werden. Setzt sich eine Messgröße y aus N fehlerbehafteten Größen $x_1 \pm u_1, x_2 \pm u_2, \dots, x_N \pm u_N$ zusammen (die Mittelwertschreibweise \bar{x}_i wurde hier wegen der Übersichtlichkeit weggelassen), so gilt für ihre Unsicherheit:

$$u_y = \sqrt{\underbrace{\sum_{i=1}^N \left(\frac{\partial y}{\partial x_i} \cdot u_i \right)^2}_{\text{wie oben}} + 2 \sum_{i=1}^{N-1} \sum_{k=i+1}^N \frac{\partial y}{\partial x_i} \frac{\partial y}{\partial x_k} \cdot \text{cov}(x_i, x_k)}$$

Für die Spezialfälle gilt nun:

- Addition und Subtraktion zweier korrelierter Messwerte führt zu folgendem Fehlerfortpflanzungsgesetz:

$$y = x_1 \pm x_2 \Rightarrow s_y^2 = s_1^2 + s_2^2 \pm 2 \text{cov}(x_1, x_2)$$

- Multiplikation oder Division zweier korrelierter Messwerte führt zu folgendem Fehlerfortpflanzungsgesetz:

$$y = x_1 \cdot x_2 \text{ oder } y = \frac{x_1}{x_2} \Rightarrow \left(\frac{s_y}{y} \right)^2 \simeq \left(\frac{s_1}{x_1} \right)^2 + \left(\frac{s_2}{x_2} \right)^2 \pm 2 \frac{\text{cov}(x_1, x_2)}{x_1 x_2}$$

- der spezielle Fall für $y = f(x_1, x_2, \dots, x_n)$ und **100% abhängige (100% korrelierte)** Variablen:

$$u_y = \sum_{i=1}^n \left| \frac{\partial y}{\partial x_i} \right| u_i$$

Beispiel: Man nehme zwei Messwerte x_1 und x_2 deren Erwartungswerte und Unsicherheiten jeweils identisch sind, z.B. $\bar{x}_1 = \bar{x}_2 = 10$ mit $u_1 = u_2 = 1$. Die zu ermittelnde Messgröße wird über folgende Relation berechnet: $y = \bar{x}_1 - \bar{x}_2$

- Das Ergebnis für $r = 0$, d.h. **unabhängige** Messwerte, der kombinierten Messunsicherheit für y beträgt $u_y = 1.4$
- Das Ergebnis für $r = 1$, d.h. **korrelierte** Messwerte, der kombinierten Messunsicherheit für y beträgt $u_y = 0.0$

Dieses Beispiel findet häufig Anwendung im Bereich die Signalübertragung und ist unter der Bezeichnung **differentielle Signalübertragung** bekannt. Neben dem eigentlichen Signal, x_1 wird hierbei ein zweites, invertiertes Signal mitübertragen, $x_2 = -x_1$. Da alle Störungen innerhalb der gleichen Übertragungsstrecke fast vollständig korreliert sind, heben sich diese am Ende auf. Das Nutzsignal erhält man zurück, indem man folgende Signalkombination berechnet: $y = 0.5 \cdot (x_1 - x_2)$. Aufgrund des hohen Korrelationsgrades der einzelnen Signalkomponenten ist die Unsicherheit des extrahierten Signal sehr klein: $u_y \simeq 0$.

4.5 Kurvenanpassung

Bei der Kurvenanpassung handelt es sich um ein statistisches Analyseverfahren zur Feststellung funktionaler Beziehungen zwischen einer abhängigen und einer oder mehreren unabhängigen Variablen. Der Begriff **lineare Regression** ist weit verbreitet, doch dies ist nur der einfachste Fall eines Modells, nämlich der einer Geraden: $y = a \cdot x + b$. Grundsätzlich sollte man den Typ der Fit-Funktion $y = f(x)$ immer vorher festlegen und auch begründen können. Es ist keine wissenschaftlicher oder messtechnische Vorgehensweise alle möglichen Funktionen nur auf Verdacht *auszuprobieren* und sich für die besten entscheiden. Hierbei wäre es möglich, dass unbrauchbare Näherungen oder sogar falsche (unsinnige) und nicht-wissenschaftlicher Ergebnisinterpretationen auftreten könnten, was es zu vermeiden gilt.

Zusammengefasst suchen wir nun also ein bestimmtes Modell für ein bestimmtes Set an Daten und wollen die Modellparameter bestimmen. Dabei soll das Modell möglichst gut die Messdaten vorhersagen. Die Modellanpassung wird häufig über die Methode der kleinsten Quadrate verwendet, mit welcher sich fast alle Messdaten modellieren lassen (auch kompliziertere Situationen wie beispielsweise korrelierte Unsicherheiten).

Hinweis zur Begrifflichkeit:

- **Regression:** Untersuchung der *Korrelation* von Datenpunkten ohne Messfehler mit angenommenen Zusammenhang
- **Fit/Anpassung:** wie die Regression, allerdings unter Berücksichtigung von Messfehlern.
- **Interpolation:** Hierbei handelt es sich nicht um eine Regression bzw. Approximation. Anstelle eines funktionalen Zusammenhangs, der an die Messwerte angenähert wird, verwendet man Polynome hohen Grades, um eine analytische Kennlinie zu beschreiben, die *exakt* durch alle Messpunkte geht. Für eine große Anzahl von Messwerten wird die Interpolationsfunktion sehr schnell unhandlich.

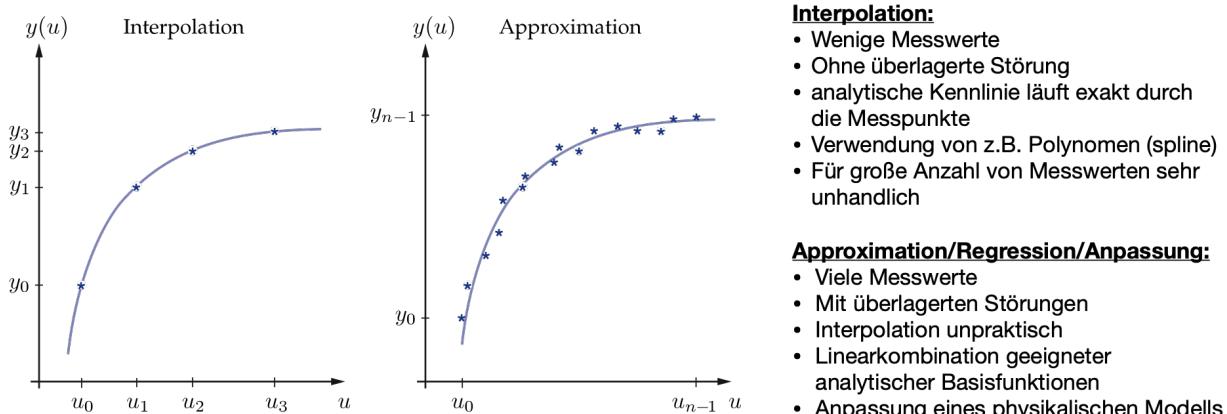


Abbildung 2.1. Kennlinie in Form von n Messpunkten und Ergebnis der Kurvenanpassung.

Tutorial

Python-Beispiele für Kurvenanpassungen findet ihr hier:

- [Vergleich verschiedener Fit-Routinen in Python](#)
- [Fitten mit Fehlerbalken in Python](#)
- [Fitten von echten Klimadaten](#)

4.5.1 Modellanpassung

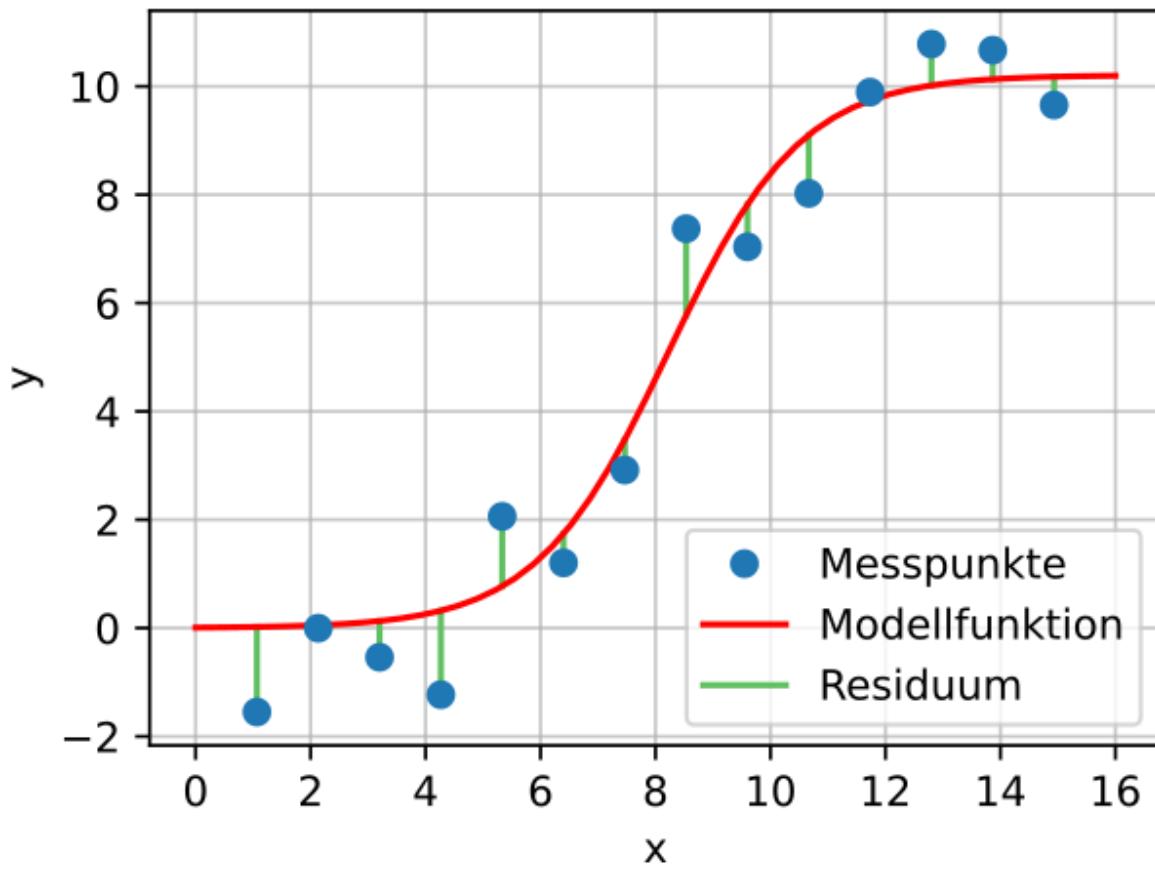
Um ein Regressionsmodell zu berechnen, benötigen wir ein objektives Maß um die Zuverlässigkeit und Güte unserer Modellfunktion zu bestimmen. Dies nennt man auch das **Bestimmtheitsmaß**, bzw. auf englisch **coefficient of determination** oder **goodness of fit**. Dieses Maß

- bestimmt die Verkleinerung des Vorhersagefehlers der Ausgangsgröße y
- definiert die Größe der Streuung von y
- zeigt die Qualität der linearen Regression, aber nicht ob das Modell richtig ist
- sagt nichts über die statistische Signifikanz des ermittelten Zusammenhangs der einzelnen Regressoren aus (Signifikanztest notwendig)

Als erstes soll überprüft werden, inwiefern die Funktion oder das Modell mit den Messdaten übereinstimmt. Ausgangspunkt ist also unsere Messreihe mit N Messpunkten (x_i, y_i) und wir haben eine Funktion $f(x_i)$ definiert, die die Messwerte y_i möglichst gut vorhersagen soll. In der unteren Grafik (geborgt von [Wikipedia](#)), sind Messpunkte in blau und eine Modellfunktion in rot aufgezeichnet. Die Parameter der gesuchten Modellfunktion werden nun so bestimmt, dass die Modellfunktion möglichst wenig von den Messwerten abweicht, d.h. das Residuum

$$\epsilon = (f(x_i) - y_i)$$

soll möglichst klein werden.



Least-Squares: Methode der kleinsten Quadrate

Laut Carl Friedrich Gauß kann also der Abstand der Funktionswerte zu den Messwerten wie folgt definiert werden:

$$\text{Methode der Gauß'schen Fehlerquadrate (Gütfunktion): } Q := \sum_{i=1}^N (f(x_i) - y_i)^2 := \sum_{i=1}^N \epsilon_i^2 = \min! \$$$

Die Gütfunktion, die gleichbedeutend mit den SQT ist, ist wieder so gewählt, dass sich Abweichungen aufgrund unterschiedliche Vorzeichen nicht aufheben können und dass größere Abweichungen stärker gewichtet werden (durch die Quadrierung). Die Modellfunktion $f(x)$ wird bestimmt, indem das Minimierungsproblem gelöst wird. Dies kann entweder analytisch geschehen (siehe *Lineare Modellanpassung*) oder man lässt sich die Regression mittels Software berechnen.

Es handelt sich um ein *Minimierungsproblem* welches je nach Art der Modellfunktion unterschiedlich gelöst wird. Man sollte stets die Regressionsgerade zusammen mit den Datenpunkten in ein Diagramm zeichnen, um zu sehen, wie "gut" die Messdaten durch die Kurvenanpassung beschrieben werden. Je enger die Datenpunkte um die Regressionsgerade herum konzentriert sind, d. h. je kleiner also die Residuenquadrate sind, desto „besser“. Die Residuenquadrate sind meistens relativ klein, insbesondere dann, wenn die abhängige Variable sehr konstant ist. Das heißt man möchte eigentlich auch die Streuung der abhängigen Variablen mit ins Spiel bringen. Sowohl die Streuung der Messwerte zum Mittelwert, als auch die der geschätzten Werte, sollte in Relation zueinander gebracht werden. Das heißt wir definieren im Folgenden zwei Summen der Abweichungsquadrate. Die **Summe der Abweichungsquadrate (Sum of Squares) SQT oder SST** gibt die Streuung der Messwerte um ihren Mittelwert an. Das *mittlere* Abweichungsquadrat bestimmt deren Varianz.

- Die **Summe der Quadrate der Erklärten Abweichungen (Sum of Squares Explained) SQE oder SSE** gibt die Streuung der Schätzwerte $f(x_i)$ des Fits um den Mittelwert $\bar{f} = \bar{y}$ der gemessenen Messwerte an:

$$SQE = SSE = \sum_{i=1}^N (f(x_i) - \bar{y})^2$$

- Die **totale Quadratsumme (Summe der Quadrate der Totalen Abweichungen bzw. Sum of Squares Total) SQT oder SST** gibt die Streuung der Messwerte y_i um deren Mittelwert \bar{y} an:

$$SQT = SST = \sum_{i=1}^N (y_i - \bar{y})^2$$

- Die **Restabweichungen** (nicht erklärte Abweichungen), welche nach der Regression übrig bleiben sind ein Maß für die Abweichung der Datenpunkte von der Regressionskurve und werden durch die Residuenquadratsumme (**Summe der Quadrate der Restabweichungen** (oder: „Residuen“) bzw. englisch Sum of Squares Residual) SQR oder SSR erfasst:

$$SQR = SSR = \sum_{i=1}^N (y_i - f(x_i))^2$$

- Man kann beweisen, dass folgendes gilt:

$$SQT = SQR + SQE$$

Bestimmtheitsmaß

Für das **Bestimmtheitsmaß** gelten folgenden Punkte:

- Es beschreibt den Anteil der Varianz einer abhängigen Variablen y durch ein statistisches Modell
- Es ist nur für lineare Regressionen eindeutig definiert: $R^2 = r^2 (= \text{Korrelation}^2)$
- Es kann bedingt zur Beschreibung der Güte einer Regression verwendet werden.
- Das Verhältnis der beiden Größen SQE und SQT wird das **Bestimmtheitsmaß** der Regression genannt und gibt an, wie gut die gefundene geschätzte Modelfunktion zu den Messdaten passt, oder wie gut sich die Regression an die Daten annähert.

$$R^2 \equiv \frac{SQE}{SQT} = \frac{\sum_{i=1}^N (f(x_i) - \bar{y})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} = 1 - \frac{SQR}{SQT} = 1 - \frac{\sum_{i=1}^N (y_i - f(x_i))^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

wobei:

- $f(x_i)$ der Funktionswert der Regression ist
- x_i der Datenwert ist
- und \bar{y} der Mittelwert y_i der Messwerte ist

Das Bestimmtheitsmaß lässt sich mit 100 multiplizieren, um es in Prozent anzugeben, dies entspricht dann dem prozentualen Anteil der Streuung in y , der durch das lineare Modell beschrieben wird und liegt somit zwischen 0% und 100%:

- 0%: es existiert kein linearer Zusammenhang
- 100%: perfekter linearer Zusammenhang

Allgemein gilt für das Bestimmtheitsmaß:

- je näher R^2 an 1 liegt, desto höher ist die Güte der Kurvenanpassung
- für $R^2 = 0$ ist der Schätzer im Modell völlig unbrauchbar für irgendeine Vorhersage eines Zusammenhangs zwischen x_i und y_i .

- für $R^2 = 1$ lässt sich y vollständig durch ein lineares Modell beschreiben und alle Messpunkte liegen auf einer nicht-horizontalen Geraden. In diesem Falle würde man davon sprechen, dass ein deterministischer Zusammenhang besteht, kein stochastischer.

Nachteile des Bestimmtheitsmaß: Für immer mehr Messwerte steigt R^2 an, ohne dass die Korrelation oder die Regression besser wird. Dies könnte durch ein korrigiertes R^2 behoben werden:

$$\bar{R}^2 = 1 - (1 - R^2) \cdot \frac{n - 1}{n - p - 1}$$

mit

- n Anzahl der Messwerte und
- p Anzahl der Variablen im Regressionsmodell

Ein weiterer Nachteil ist, dass keine Aussage darüber geliefert werden kann, ob ein *korrektes* Regressionsmodell verwendet wurde.

Modellanpassung mit Unsicherheiten

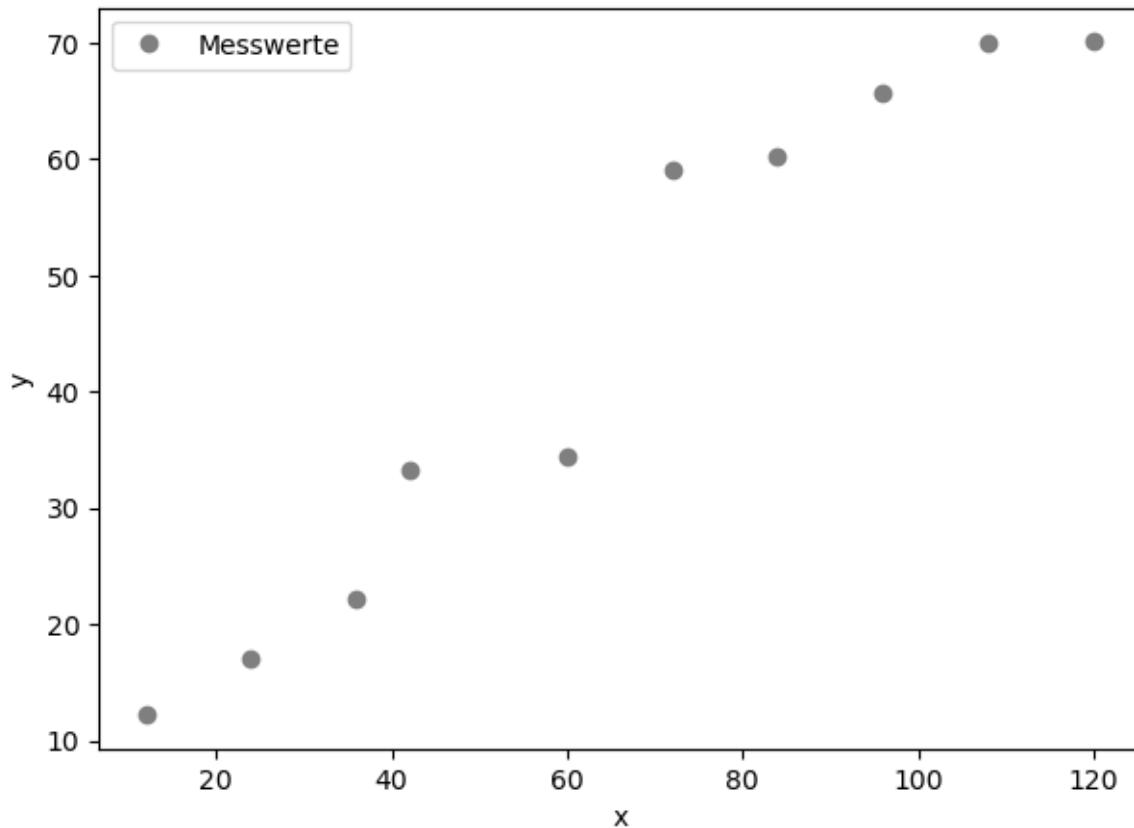
Im allgemeinen Fall, d.h. wenn die Messwerte y_i mit Unsicherheiten s_i behaftet sind, lässt sich die Residuensumme wie folgt definieren:

$$\chi^2 = \sum_{i=1}^N \left(\frac{f(x_i) - y_i}{s_i} \right)^2$$

Für die obigen Berechnungen, und auch im Falle von konstanten Unsicherheiten, d.h. wenn für alle Werte von y_i die gleiche absolute Unsicherheit existiert, ändert sich nichts. Denn es gilt $s_i = s = \text{const}$ und beim "Nullsetzen" werden diese einfach eliminiert. Gelten für die N Messwerte allerdings unterschiedliche Unsicherheiten, so müssen diese miteinbezogen werden.

Lineare Modellanpassung

Da wir als Messtechniker immer danach streben möglichst lineare Kennlinien zu erreichen, ist die Gerade eine häufig auftretende Kurve, die angepasst werden soll. Daher wollen wir uns in diesem Abschnitt mit der Herleitung der linearen Regression befassen. Die Herleitung für andere Modelfunktionen, welche quadratische Terme, noch höhere Terme oder ganz andere Zusammenhänge beinhalten, ist auch deutlich schwieriger.



Unser Ausgangspunkt ist also eine Gerade der Form

$$f(x) = mx + b$$

Die Parameter m und b werden nun durch das Minimierungsproblem bestimmt mithilfe der Gütfunktion, die nun wie folgt geschrieben werden kann:

$$Q = \sum_{i=1}^N (y_i - f(x_i))^2 = \sum_{i=1}^N (y_i - mx_i - b)^2 = \min!$$

wobei y_i und x_i die Messwerte (Datenpunkte) sind.

Durch Differentiation nach den Parametern und gleichsetzen auf Null können die Parameter bestimmt werden:

$$\frac{dQ}{dm} = -2 \sum_{i=1}^N x_i (y_i - mx_i - b) = 0$$

$$\frac{dQ}{db} = -2 \sum_{i=1}^N (y_i - mx_i - b) = 0$$

Nach Umstellen der beiden Ableitungen gelangt man zu folgenden Gleichungssystem:

$$m \sum_{i=1}^N x_i^2 + b \sum_{i=1}^N x_i = \sum_{i=1}^N x_i y_i$$

$$m \sum_{i=1}^N x_i + Nb = \sum_{i=1}^N y_i$$

Auflösen nach den gesuchten Parameter erhält man folgende Gleichungen für die gesuchten besten Schätzparameter der Regressionsgeraden, auch **Regressionskoeffizienten** genannt:

$$m = \frac{\sum_{i=1}^N (x_i y_i) - b \sum_{i=1}^N x_i}{\sum_{i=1}^N x_i^2} = \frac{S_{xy}}{S_x^2} = \frac{\bar{x} \cdot \bar{y} - \bar{x} \cdot \bar{y}}{\bar{x}^2 - (\bar{x})^2}$$

$$b = \frac{\sum_{i=1}^N y_i - m \sum_{i=1}^N x_i}{N} = \bar{y} - m \cdot \bar{x}$$

mit folgenden Definitionen:

$$\bar{x} = \sum_{i=1}^N x_i$$

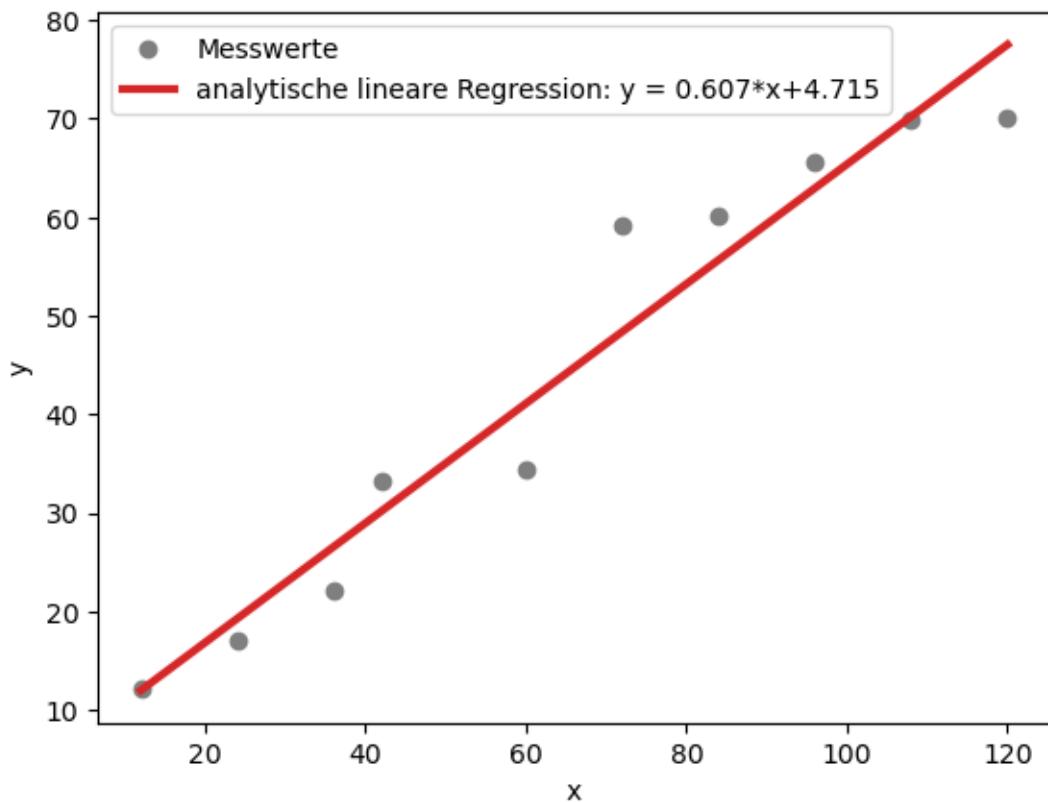
$$\bar{y} = \sum_{i=1}^N y_i$$

$$S_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

$$S_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

Wir sind hier in der verrückten Situation, dass tatsächlich Mittelwerte für x und y bestimmt werden müssen, obwohl die x -Werte absichtlich während der Versuchsreihe verändert werden, sich also die Größen x und y laufend ändern.

Die Steigung ist	$m = 0.6067 \text{ s/m}$
Der Ordinatenabschnitt ist	$b = 4.7148 \text{ s}$



Nun sind die Schätzwerte allerdings zusätzlich fehlerbehaftet (wie sollte es auch anders sein). Mithilfe der Gleichung der Größtfehlers/Maximalfehlers kann man zeigen (den Beweis überspringen wir hier), dass für den Fehler von y folgendes gilt:

$$s_y = \sqrt{\frac{1}{N-2} \sum (y_i - mx_i - b)^2}$$

Die Unsicherheit von y ist $s_y = 5.8831 \text{ s}$

Die ist auch die Standardabweichung der Einzelmessung aber *nicht* der Fehlerbalken, der im Diagramm als Fehlerbalken eingezeichnet wird. Die Abweichung der Einzelmessung wurde bisher mit $N - 1$ definiert, damals hat es sich aber um die Abweichung vom *Mittelwert* gehandelt. Nun betrachten wir die Abweichung zu einem linearen Modell, welches 2 offene Parameter, m und b , hat, und somit einen Freiheitsgrad mehr besitzt. Erst ab 3 Messwertpaaren können also Fehler für Steigung und Achsenabschnitt berechnet werden. Die besten Schätzwerte für die Abweichungen von m und b können nun wie folgt berechnet werden. Der Fehler der Geradensteigung beträgt:

$$s_m = s_y \cdot \sqrt{\frac{N}{N \cdot \sum x_i^2 - (\sum x_i)^2}} = s_y \cdot \sqrt{\frac{1}{\sum x_i^2 - N \cdot \bar{x}^2}} = s_y \cdot \sqrt{\frac{1}{\sum (x_i - \bar{x})^2}} = s_y \cdot \sqrt{\frac{1}{N \cdot (\bar{x}^2 - (\bar{x})^2)}}$$

Die Unsicherheit von m ist $s_m = 0.0534 \text{ s/m}$

Der Fehler des Ordinatenabschnitts beträgt:

$$s_b = s_y \cdot \sqrt{\frac{\sum x_i^2}{N \cdot \sum x_i^2 - (\sum x_i)^2}} = s_y \cdot \sqrt{\frac{1}{N \sum x_i^2 - N \cdot \bar{x}^2}} = s_y \cdot \sqrt{\frac{1}{N \sum (x_i - \bar{x})^2}} = s_m \cdot \sqrt{\bar{x}^2}$$

Die Unsicherheit von b ist $s_b = 3.9581 \text{ s}$

Korrelationskoeffizient

Für lineare Zusammenhänge ist es häufig sinnvoll den Korrelationskoeffizient zu berechnen:

$$r = \frac{\bar{x} \cdot \bar{t} - \bar{x} \cdot \bar{t}}{\sqrt{\bar{x}^2 - (\bar{x})^2} \cdot \sqrt{\bar{t}^2 - (\bar{t})^2}}$$

Der Korrelationskoeffizient zwischen x und t beträgt: 0.97035603

Die Korrelationsmatrix zwischen x und t mittels numpy-Paket lautet:

```
[[1.          0.97035603]
 [0.97035603 1.        ]]
```

Der Korrelationskoeffizient von +0,97035 zeigt mit positivem Vorzeichen eine direkte Proportionalität zwischen x und t . Die geringfügige Abweichung zu +1 zeigt, dass die Messwerte leicht von dem erwarteten linearen Zusammenhang dennoch abweichen.

Part III

Messsignale und -systeme

STATIONÄRE MESSSYSTEME

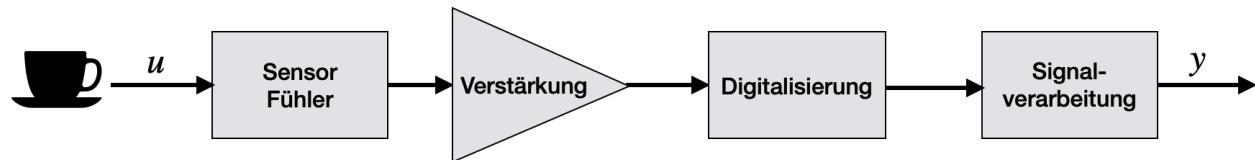
Wir benutzen Messsysteme, um eine Messgröße in einen Messwert umzuführen. Hierbei werden wir den realen, echten Wert der Messgröße jedoch nie erfahren (Damit befassen wir uns im Kapitel *Messunsicherheiten*). Ein Messwert kann direkt vom Messgerät angezeigt werden, beispielsweise über ein Display, oder er steht als Datenwert in analoger oder digitaler Form zur Verfügung, welcher mit entsprechenden Geräten oder Algorithmen weiterverarbeitet werden muss.

In diesem Kapitel wollen wir uns mit den Begrifflichkeiten und Kenngrößen eines Messsystems befassen. Hierbei nehmen wir vorerst an, dass sich die Messwerte über die Zeit während des Messprozesses nicht ändern. Diese Art von Messsystemen werden *statische Messsysteme* genannt.

- *Grundstruktur eines Messsystems*
- *Ideale Kennlinie*
- *Reale Kennlinie*

5.1 Grundstruktur eines Messsystems

Die generelle Struktur eines Messsystems kann wie folgt dargestellt werden:



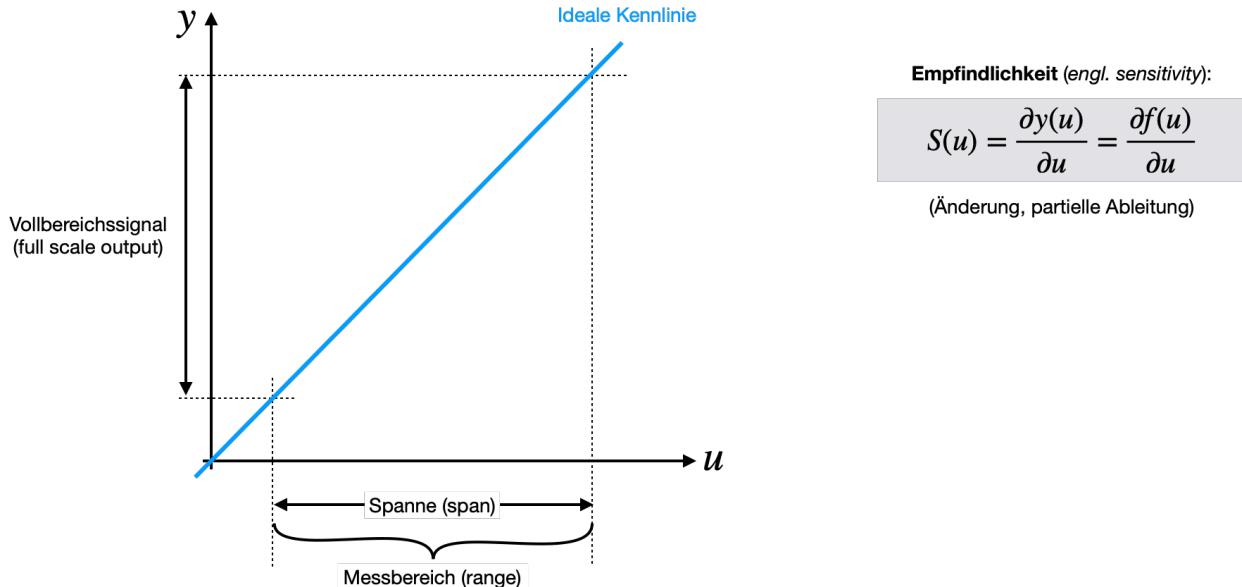
Dies ist einer der ersten Messketten, die wir kennenlernen. Allgemein findet man immer wieder die gleichen Komponenten in solchen Ketten, die im folgenden aufgelistet werden:

- **Aufnehmer/Sensor** (auch Messgrößenaufnehmer genannt): Die Erfassung der physikalischen Messgröße, u , wird mit einem entsprechend geeigneten *Sensor* realisiert. Mit konkreten Sensoren werden wir uns noch am Ende der Vorlesung genauer beschäftigen. Einige Beispiele werden uns aber während der kompletten Veranstaltung immer mal wieder begegnen. Ein Sensor nimmt eine Messgröße auf, z.B. die Umgebungstemperatur, wandelt diese beispielsweise in eine Widerstandänderung um, welche wiederum in ein weiterverarbeitungsfähigen Signals (hier elektrisch) mit einer geeigneten Schaltung umgewandelt wird.
- **Verstärkung**: An dieser Stelle startet die Messsignalverarbeitung. Da das elektrische Primärsignal mitunter sehr klein sein kann, muss es deshalb vielleicht noch verstärkt werden, bevor es einer Digitalisierung zugeführt werden kann.
- **Anpassung** (optional): Die meist elektrische Größe wird in einen darstellbaren Messwert umgewandelt. Hierfür werden Messschaltungen mit Messverstärken oder Computern verwendet.

- **Messwertausgabe** (diverse): Anzeige, Registrierung, Speicherung, Dokumentation in analoger oder digitaler Form können an dieser Stelle in die Messkette implementiert werden.
- **Digitalisierung:** Dies ist die häufigste Art der *Messwertausgabe*. Das analog vorliegende elektrische Signal wird in ein Digitalwort umgewandelt.
- **Digitale Signalverarbeitung:** Durch Algorithmik wird der Messwert digital weiterverarbeitet. Häufig können an dieser Stelle auch Korrekturen vorgenommen werden, um beispielsweise Kennenlinienfehler zu minimieren und zu korrigieren. Dies werden wir gleich noch genauer betrachten. Ausgegeben wird schließlich ein Messwert y .

5.2 Ideale Kennlinie

Jede einzelne der Komponenten führt die an ihr anliegenden Eingangssignal in Ausgangssignale über. Wie diese Überführung genau aussieht beschreibt die sogenannte **Kennlinie**, die für jede Komponente unterschiedlich aussehen kann. Im folgenden Bild ist beispielhaft eine lineare Kennlinie dargestellt:



5.2.1 Statische Kenngrößen

Für jede Eingangsgröße u wird anhand der Kennlinie eine Ausgangsgröße y definiert. Beispielsweise wird anhand der Kennlinie eines Temperatursensors ein bestimmter Temperaturwert in eine Spannung umgewandelt.

- Der Bereich, indem der Sensor beispielsweise noch korrekt arbeitet (meist vom Hersteller garantiert) wird **Messbereich (engl. range)** genannt und wird durch den minimal und maximal möglichen Wert, der noch als Eingang angelegt werden kann oder darf, angegeben. Im Falle eines Temperatursensors wäre ein typischer Messbereich z.B. -40°C bis 120°C.
- Die Differenz zwischen diesen Maximal- und Minimalwert nennt man **Spanne (engl. span)** (bezogen auf unser Beispiel also 160°C).
- Die Spanne ist mit einem Bereich auf der y-Achse korreliert, nämlich dem sogenannten **Vollbereichssignal (engl. full scale output = FSO)**.

- Der **Übertragungsfaktor** k oder auch oft *Verstärkung* genannt beschreibt die Überführung des Eingangssignals der Messeinrichtung, also der Messgröße, in ein Ausgangssignal:

$$y = k \cdot u$$

- Die **Statische Kennlinie** ist im Prinzip der *Übertragungsfaktor*, aber dieses Mal definiert für alle möglichen Eingangssignale.

$$y(u) = k \cdot u$$

5.2.2 Empfindlichkeit

Die Kennlinie sollte immer eine gewisse Steigung aufweisen. Oder man könnte auch sagen, dass wenn sich die Eingangsgröße u ändert, dass dies auch immer eine Änderung in der Ausgangsgröße y mit sich ziehen sollte. Im Falle einer Verstärkungseinheit könnte diese Änderung sogar vergrößert werden und das System reagiert somit *empfindlicher* auf Änderungen der Eingangsgröße. Mathematisch betrachtet bedeutet die Steilheit nichts anderes als die Steigung bzw. die Ableitung der Ausgangsgröße $y(u)$ an einer bestimmten Stelle u :

$$S(u) = \frac{\partial y(u)}{\partial u} = \frac{\partial f(u)}{\partial u}$$

Dies wird auch die Empfindlichkeit $S(u)$ genannt und entspricht der Tangente im Punkt u .

5.2.3 Beispiele

Kennlinie von Widerstandsthermometern

Reine Metalle sind **Kalteiter**. Platin hat beispielsweise einen Widerstandswert von $R_0 = R(0) = 100 \text{ Ohm}$ bei 0°C , daher der Name PT100. Sie können bei geeigneter Ausrüstung bis 850°C eingesetzt werden. Sie haben eine fast lineare Kennlinie bei einer relativen Widerstandsänderung von knapp $0,4\%$ pro $^\circ\text{C}$:

$$R(T) = R_0 \cdot (1 + AT + BT^2)$$

Folgende Parameter gelten für bestimmte Temperaturbereiche:

- $0\text{-}850^\circ\text{C}$: $A = 3,9\text{e-}3/\text{ }^\circ\text{C}$, $B = -5,7\text{e-}7/\text{ }^\circ\text{C}^2$
- $-200\text{-}0^\circ\text{C}$: $C = -4,2\text{e-}1/\text{ }^\circ\text{C}^4$

Thermistoren mit negativen Temperaturkoeffizienten (NTC = Negative thermal coefficient, Heißleiter) weisen aufgrund des zugrundeliegenden Halbleiter-Effektes eine hohe Abhängigkeit von Fehlstellen, wie der Dotierung der Grundstoffe auf. **Heißleiter** sind bis etwa 150°C einsetzbar. Sie weisen gegenüber Platin-Messwiderständen eine deutlich höhere Empfindlichkeit auf:

$$R(T) = R_{25} \cdot e^{B(\frac{1}{T+273} - \frac{1}{298})}$$

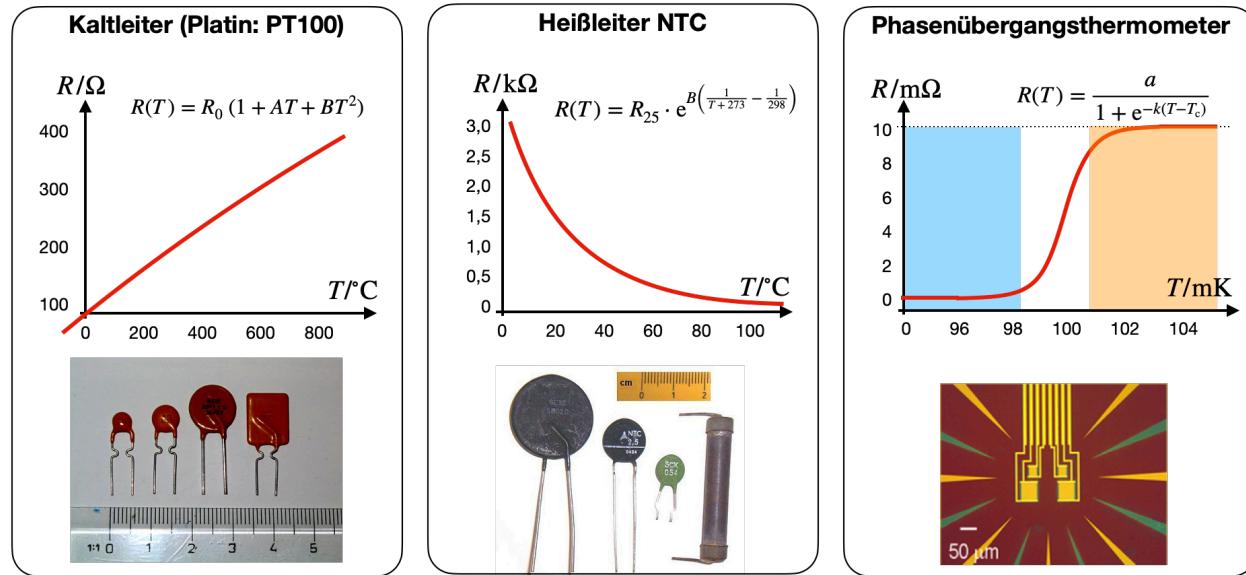
mit $R_{25} = 1000 \text{ Ohm}$ und $B = 3528$.

Ein Transition Edge Sensor (TES), bzw. auf deutsch Phasenübergangsthermometer, ist ein supraleitender Temperatursensor, der in der Lage ist, kleinste Temperaturänderungen im Bereich von wenigen $100 \mu\text{K}$ zu messen. Seine Umgebungstemperatur wird je nach Material auf wenige mK runtergekühlt, um das Material in einen supraleitenden Zustand (d.h. sein Widerstand verschwindet) zu bringen. Dies ist die sogenannte kritische Temperatur, T_c , aber der Materiale supraleitende Eigenschaften aufzuweisen. Die Widerstand wird über Anlegen einer Stromstärke minimal aufgeheizt, sodass sein

Arbeitspunkt in den Übergang (die steile Flanke) geschoben wird. An diesem Punkt ist die Ableitung, und somit die Empfindlichkeit, maximal.

$$R(T) = \frac{a}{1 + e^{-k(T-T_c)}}$$

mit z.B. $k = 1.86/\text{mK}$, $a = 0.224 \text{ Ohm}$ und $T_c = 103.2 \text{ mK}$



```
#Benötigte Libraries:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import time
import warnings
warnings.filterwarnings('ignore')

# Matplotlib Settings:
plt.style.use('default') # Matplotlib Style wählen
plt.figure(figsize=(10,5)) # Plot-Größe
plt.rcParams['font.size'] = 10; # Schriftgröße

----- Kaltleiter PT100 -----
T_PT = np.linspace(0, 800, num=800)
R_0 = 100 # in Ohm
A = 3.93e-3 # in 1/°C
B = -5.7e-7 # in 1/^C^2
def R_PT(T_PT):
    return R_0 * (1 + A*T_PT + B * T_PT**2)

----- Heißleiter NTC -----
T_NTC = np.linspace(0+273.15, 120+273.15, num=50)
R_25 = 1000 # in Ohm
T_25 = 25.0 + 273.15 # in Kelvin
```

(continues on next page)

(continued from previous page)

```

B = 3528
def R_NTC(T_NTC, B):
    return R_25 * np.exp(B * (1 / (T_NTC) - 1 / T_25))

----- Transition Edge Sensor -----
k = 1.86 # in 1/mK
a = 0.01 # in Ohm
Tk = 100 # in mK
T = np.linspace(94, 106, num=50)
def R(T):
    return a / (1 + np.exp(-k * (T-Tk)))

----- Diagramme -----
f, axs = plt.subplots(1,3,figsize=(10,4))

axs[0].plot(T_PT,R_PT(T_PT))
axs[0].set_xlabel('T/°C')
axs[0].set_ylabel(r'R/k$\Omega')
axs[0].set_title('Kaltleiter PT100')

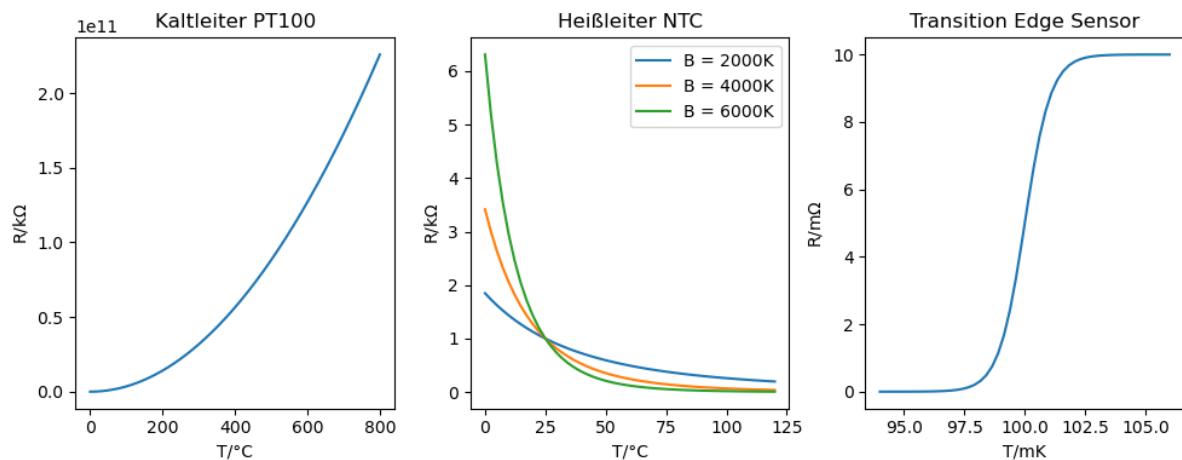
axs[1].plot(T_NTC-273.15,R_NTC(T_NTC, 2000)/1000, label = "B = 2000K")
axs[1].plot(T_NTC-273.15,R_NTC(T_NTC, 4000)/1000, label = "B = 4000K")
axs[1].plot(T_NTC-273.15,R_NTC(T_NTC, 6000)/1000, label = "B = 6000K")
axs[1].set_xlabel('T/°C')
axs[1].set_ylabel(r'R/k$\Omega')
axs[1].set_title('Heißleiter NTC')
axs[1].legend()

axs[2].plot(T,R(T)*1000)
axs[2].set_xlabel('T/mK')
axs[2].set_ylabel(r'R/m$\Omega')
axs[2].set_title('Transition Edge Sensor')

plt.tight_layout()
plt.show()

```

<Figure size 1000x500 with 0 Axes>



Wunder an Empfindlichkeit

LIGO (Laser Interferometer Gravitational-Wave Observatory) ist ein Gravitationswellendetektor. Mit diesem experimentellen Aufbau werden kleinste Störungen in der Raumzeit, so genannten Gravitationswellen, gemessen, welche damals von Albert Einstein innerhalb seiner allgemeinen Relativitätstheorie vorhergesagt wurden. Gravitationswellen entstehen, wenn sich sehr schwere Massen in der Raumzeit bewegen, wie z.B. zwei schwarze Löcher oder Neutronensterne, die umeinander kreisen. Die direkte Nachweis von Gravitationswellen wird durch die wirklich verschwindend kleinen Effekte der Wellen auf den Detektor erheblich erschwert. Die Amplitude einer Gravitationswelle ist zudem umgekehrt proportional zur Entfernung der Quelle. Dadurch klingen sogar Wellen, welche von Extremsystemen wie das von zwei verschmelzenden schwarzen Löchern, auf dem Weg zur Erde zu einer kleinen Amplitude ab.

LIGO, VIRGO und KAGRA sind ultra-empfindliche Detektoren und nutzen Laserinterferometrie, um die Bewegung von *freien* Massen zu messen, die durch eben diese Gravitationswellen in der Raumzeit ausgelöst wurde. Mit Laserinterferometern werden wir uns später noch genauer befassen. Es hilft auf jeden Fall, dass Interferometer so groß wie möglich zu machen. Die *Arme* des Interferometers sind mehrere Kilometer lang (LIGO z.B. 4 km). Die stärkste Gravitationswelle hat diese Armlänge um ca. 10^{-19} m geändert. Das bedeutet eine relative Längenänderung von lediglich:

$$h = \frac{dL}{L} = 2,5 \cdot 10^{-23}$$

```
L = 4000 # Armlänge in m
dL = 1e-19 # Spiegelbewegung in m
h = dL / L
print('relative Armlängenänderung bei einer Gravitationswelle: ', h)
```

relative Armlängenänderung bei einer Gravitationswelle: 2.499999999999998e-23

Nimmt man den Abstand von Erde und Sonne (150 000 000 km), würde sich dieser beim Einfall einer Gravitationswelle um weniger als einen Atomdurchmesser ändern:

```
d_Erde_Sonne = 150e9 # Abstand Erde Sonne in m
dL_Erde_Sonne = d_Erde_Sonne * h
print('Abstandsänderung Erde-Sonne verursacht durch Gravitationswellen: ', dL_Erde_-
    ↪Sonne, ' = ', dL_Erde_Sonne*1e12, ' pm' )
```

Abstandsänderung Erde-Sonne verursacht durch Gravitationswellen: 3.75e-12 = 3.
75 pm

Die Größe eines Atoms beträgt um die 100 pm = 1 Å (Angstrom = 10^{-10} m).

Rechts im nachfolgenden Bild ist die Empfindlichkeitskurve eines Laserinterferometers gegeben. Die Phasenverschiebung ist ein Maß für die Verschiebung der Spiegel. Phase kann über die Wellenlänge in eine Längenänderung umgerechnet werden. Das werden wir später noch mal genauer betrachten ($2\pi \hat{=} \lambda = 1064\text{nm}$).

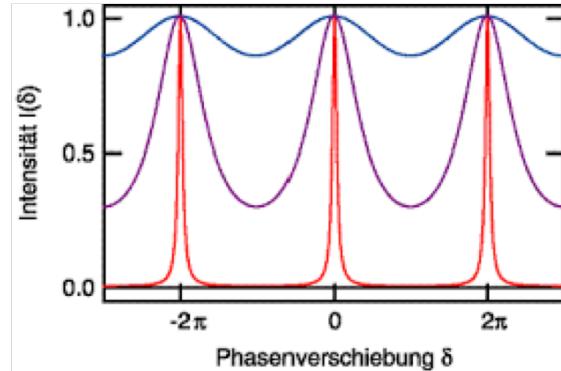
- Ein klassisches Laserinterferometer überlagert zwei elektromagnetische Wellen und produziert ein sinusförmiges Intensitätsprofil im Ausgangsbereich (blaue Kurve). Die maximale Empfindlichkeit für Gravitationswellen (oder Spiegelbewegungen) wird an den Umkehrpunkten erreicht. An den Extrempunkten erhält man für kleine Spiegelbewegungen gar kein Ausgangssignal.
- Gravitationswellendetektoren benutzen in ihren Interferometerarmen zusätzliche Spiegel, um optische Resonatoren einzubauen. Optische Resonatoren *speichern* das Licht in den Armen, d.h. es hin und her reflektiert, bis es am Ende des Interferometers wieder verlässt. Dadurch verlängert sich künstlich die Armlänge (von 4km). Das Interferometer wird empfindlicher, da Spiegelbewegungen aufaddiert werden. Im Abhängigkeit von der *Güte* des optischen Resonators können sehr starke Überhöhungen erreicht werden (lila und rote Kurve im Diagramm). Eine kleine Änderung der Spiegelbewegung verursacht nun eine drastische Änderung im Ausgangssignal. Dadurch werden kleine Signale stark überhöht und deutlich messbar. Um dies jedoch in der Praxis zu erreichen, muss das Interferometer

an sich extrem stabil und robust sein, sodass Erdbeben oder der Hase, der über das Feld hoppelt, keine Messsignale verursachen.

**Laser Interferometer Gravitational-Wave Observatory
(LIGO) in Livingston, Louisiana, USA**



Positionsänderung mit einer Empfindlichkeit von 10^{-18} m
über 4km messbar [<https://www.ligo.caltech.edu/>]



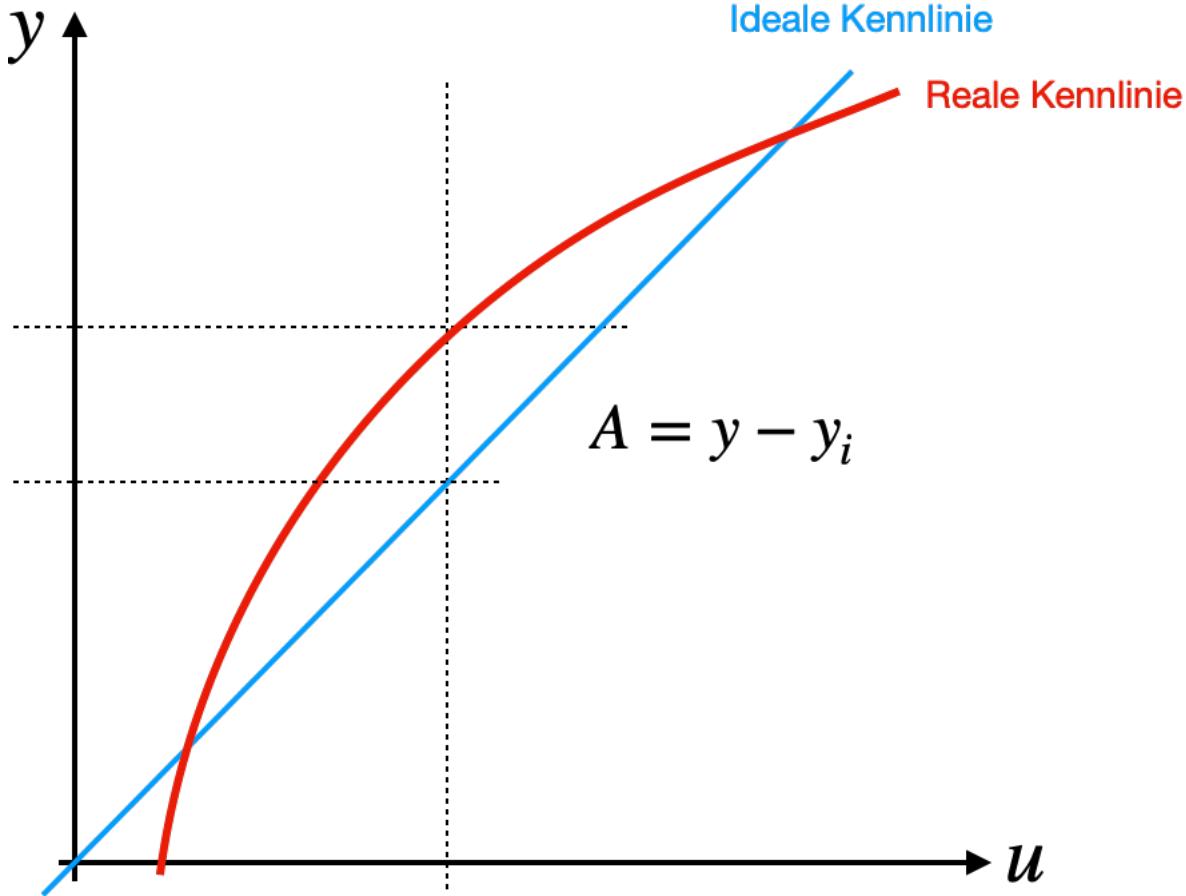
Kennlinien eines Fabry-Perot-Interferometers für
verschiedene *Finessen* [www.physik.uni-muenchen.de]

5.3 Reale Kennlinie

Nun hängt allerdings der Verlauf einer Kennlinie nicht nur von der Herstellung ab, sondern auch von (äußereren) Einflussefekten während des Messprozesses. Es hängt folglich immer von der Herstellung *und* Anwendung eines Messsystems ab, wie genau man wirklich messen kann. Womit wir beim einem anderen Aspekt unserer Betrachtungen wären, den *Messabweichungen*, die im nächsten Jupyter Notebook erläutert werden.

Im nebenstehenden Bild ist eine ideale und eine reale Kennlinie gezeigt. Zu jeder Messgröße existiert eine Messabweichung. In der bereits zitierten einschlägigen deutschen Norm DIN 1319 wurde die früher hierfür enthaltene Bezeichnung *Fehler* bereits vor längerer Zeit durch *Abweichung* ersetzt. Der Grund liegt darin, dass im üblichen Sprachgebrauch unter einem *Fehler* eine Fehlfunktion eines technischen Systems verstanden wird, was hier jedoch nicht gemeint ist. Für eine Messeinrichtung mit linearer Kennlinie sind demzufolge Empfindlichkeit S und Übertragungsfunktion k identisch. Zu jeder Messgröße, y , existiert eine bestimmte Messabweichung, A , in Bezug auf ihren *realen* oder *idealen* Wert, y_i :

$$A = y - y_i$$



5.3.1 Kennlinienkorrektur

Die Korrektur von systematischen Messabweichung erfolgt über Kalibrierung von Kennlinien. Am häufigste und am einfachsten kann eine **Nullpunktakorrektur** vorgenommen werden. Die Abweichung vom Nullpunkt wird hierbei korrigiert, wobei zum Zeitpunkt der Kalibrierung bestimmte Umgebungsbedingungen herrschen müssen. Störgrößen, wie Temperatur und Feuchte, müssen den allgemeinen Betriebsbedingungen folgen. Was man mit dieser Nullpunktakorrektur besonders gut korrigieren kann sind Nullpunktabweichungen, die durch Streuungen im Fertigungsprozess entstanden sind.

Grundsätzlich gilt, dass eine Referenz benötigt, entweder in Form einer definierten Messgröße, oder in Form eines Referenz-Messgeräts, welches seinerseits vorher kalibriert wurde. Bei der Erstinbetriebnahme an einem Kalibrierpunkt wird einmalig also beispielsweise eine wohlbekannte Messgröße angelegt. Das Messgerät wird nun mit einem Messwert antworten, der von der aktuell herrschenden realen Kennlinie bestimmt wird. Er wird vermutlich leicht über oder unter der idealen (gestrichelten) Kennlinie verschoben sein. Im nachfolgenden Bild ist der Kalibrierpunkt am Nullpunkt des Messbereichs. Für viele Messgeräte ist der Nullpunkt ein geeigneter Kalibrierpunkt, Beispiele sind:

- Wägesystem: hier wird schlichtweg einfach kein Wägegut aufgebracht. Auch Leergewichte von Wägebehältern können so *wegkalibriert* werden.
- Abstandsmessungen: ein abstand von Null ist meist relativ einfach einstellbar
- elektrische Größen: auch bei Spannung, Strom oder Widerstand ist die Nullpunktakalibrierung einfach realisierbar.
- Beschleunigungssensoren: diese werden typischerweise parallel zur Erdoberfläche auf einer Ebene gelagert, sodass nicht einmal die Erdbeschleunigung auf diesen Sensor wirkt

- Temperaturmessungen: hier ist es tatsächlich schwierig. Für 0°C müssten gefrierendes Wasser oder eine Klimakammer genutzt werden. Wurde die Nullpunktabweichung einmal bestimmt, müssen alle nachfolgenden Messungen vorzeichenrichtig korrigiert werden. Fällt die Nullpunktabweichung positiv aus (es wird immer ein zu hoher Messwert ausgegeben), muss der Betrag später vom Messwert abgezogen werden. In der Regel verfügt das Messgerät über eine eingebaute Funktion, sodass die Kalibrierung nicht in der Nachverarbeitung berücksichtigt werden muss. Sollten sich Betriebsbedingungen ändern, ist eine Rekalibrierung nötig.

Eine Erweiterung der Nullpunktkorrekt ist die **Toleranzbandjustierung**, die den Fehler um einen Faktor 2 gegenüber der Nullpunktcorrekt reduziert, indem die Kennlinie einfach noch weiter additiv verschoben wird. Trotz der Fehlerreduktion hat die Methode den Nachteil, dass die Kennlinie nicht mehr durch den Nullpunkt geht.

Die Nullpunktcorrekt kann auch mit einer sogenannten **Steigungskorrektur** vorgenommen werden, wie es im nachfolgenden Bild dargestellt ist. Für die Steigungskorrektur sind zwei Kalibermessungen notwendig, d.h. es werden zwei Datenpunkte benötigt. Häufig ist der eine Datenpunkt der Messwert der Nullpunktakalibrierung. Der zweite Datenpunkt sollte möglichst nah am Messbereichsendwert liegen, sodass eine große Spanne abgedeckt wird. Die reale Kennlinie wird nun wieder unter Betriebsbedingungen in zwei Schritten korrigiert: Sie wird einerseits vertikal verschoben und zusätzlich um ihren Nullpunkt gedreht, sodass in beiden Kaliberpunkten keine Messabweichung mehr besteht (siehe Bild). Anschaulich kann man sich Hilfsgeraden durch die Kaliberpunkte vorstellen. Die Steigung einer Hilfsgerade durch die Kaliberpunkte weicht von der Steigung der idealen Kennlinie ab (im Bild ist sie steiler). Mittels Korrektur werden die beiden Steigungen einander angepasst.

Am ersten Kaliberpunkt, dem Nullpunkt $x_0 = 0$ wird folgender Wert gemessen:

$$y_0 = y(x_0 = 0)$$

Dann beträgt die Messabweichung an diesem Punkt:

$$\Delta y(x_0) = y_0 - 0 = y_0$$

Bei einer einfachen Nullpunktcorrekt müsste die reale Kennlinie folglich um diesen Wert verschoben werden, damit am Nullpunkt die Abweichung verschwindet. Am zweiten Kaliberpunkt, an der Stelle x_1 , gilt das gleiche. Wir messen den folgenden Wert:

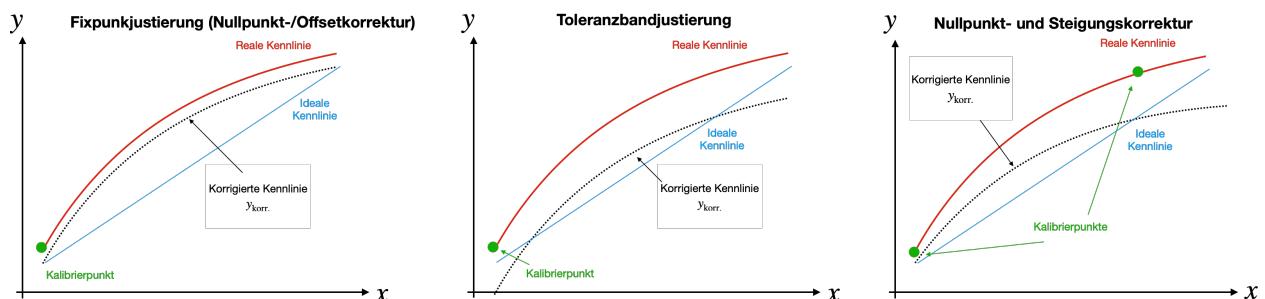
$$y_1 = y(x_1)$$

und berechnen die Messabweichung, bzw. den Korrekturwert, wiefolgt, wobei wir den *richtigen* Wert y_r an der Stelle miteinbeziehen:

$$\Delta y(x_1) = y_1 - y_r$$

Dieser Werte müsste also von allen nachfolgenden Messungen abgezogen werden. Für alle Messwerte dazwischen, ändert sich die Messabweichung linear. Wir berechnen also eine Gerade durch die beiden Kaliberpunkte und können alle anderen Messabweichungen interpolieren:

$$\Delta y(x) = \Delta y(x_0) + \frac{(\Delta y(x_1) - \Delta y(x_0))}{x_1} x = y_0 + \frac{y_1 - y_r - y_0}{x_1} x$$



Zusammenfassung:

- **Fixpunktjustierung (auch Nullpunktjustierung oder Offsetkorrektur genannt):**
 - Nach der Fixpunktjustierung geht die Kennlinie durch den Anfangspunkt und durch den Endpunkt.
 - Der Messbereich wird auf den Anzeigebereich abgebildet.
 - Im Messanfang und Messende ist damit der Fehler null.
- **Toleranzbandjustierung:**
 - Die Toleranzbandjustierung entsteht durch eine zusätzliche additive Verschiebung der Fixpunktjustierung.
 - Ziel ist es, den maximalen Fehler im Messbereich möglichst klein zu gestalten.
 - Der maximale Fehler wird im Vergleich zur Fixpunktjustierung auf die Hälfte reduziert.
 - Kennlinie geht allerdings nicht mehr zwangsläufig durch Anfangs- und den Endpunkt.
- **Nullpunkt und Steigungskorrektur:**
 - Nullpunkt und Steigungskorrektur ist sehr aufwendig: Es werden zwei Kalibriermessungen benötigt.
 - Zweiter Kalibrierpunkt ist meist der Messbereichsendwert (MBE), da hier die größte Spanne erreicht wird.
 - Die reale Kennlinie geht durch Nullpunkt und wird anschließend noch rotiert
 - Korrigierte Kennlinie hat eine kleinere Steigung und ist somit weniger empfindlich!

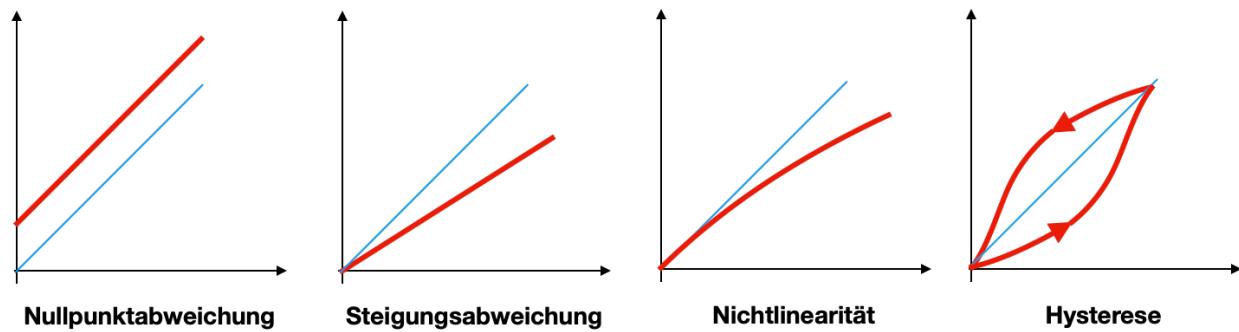
5.3.2 Kennlinienfehler

Auf den grundsätzlichen Kennlinienverlauf können durch verschiedene Einflusseffekte im Prinzip vier elementare Auswirkungen beobachtet werden. Es hängt dabei sehr stark von der konkreten Situation ab, ob ein einzelner Einflusseffekt sich primär in nur einer Art der Kennlinienabweichung zeigt oder in mehreren, d.h. es entstehen Abhängigkeiten zwischen den Komponenten. Auch wirken derartige Einflusseffekte meist auf jede einzelne Komponente eines Messsystems mit ihrer zugehörigen Einzelkennlinie. Summiert man alle diese Einflusseffekte auf alle Teilkomponenten auf, dann ergibt sich für ein konkretes Messsystem unter einer bestimmten Kombination und Anzahl von Störungen, eine ganz bestimmte *reale* Kennlinie.

Nach der Justierung sind alle systematischen Fehler Kennlinienfehler. Hierzu gehören Nichtlinearitäten (Abweichungen von der idealen Kennlinie) und Einfluss von Störgrößen. In Summe aller Einflüsse auf alle Teilkomponenten ergibt sich eine ganz bestimmte reale Kennlinie.

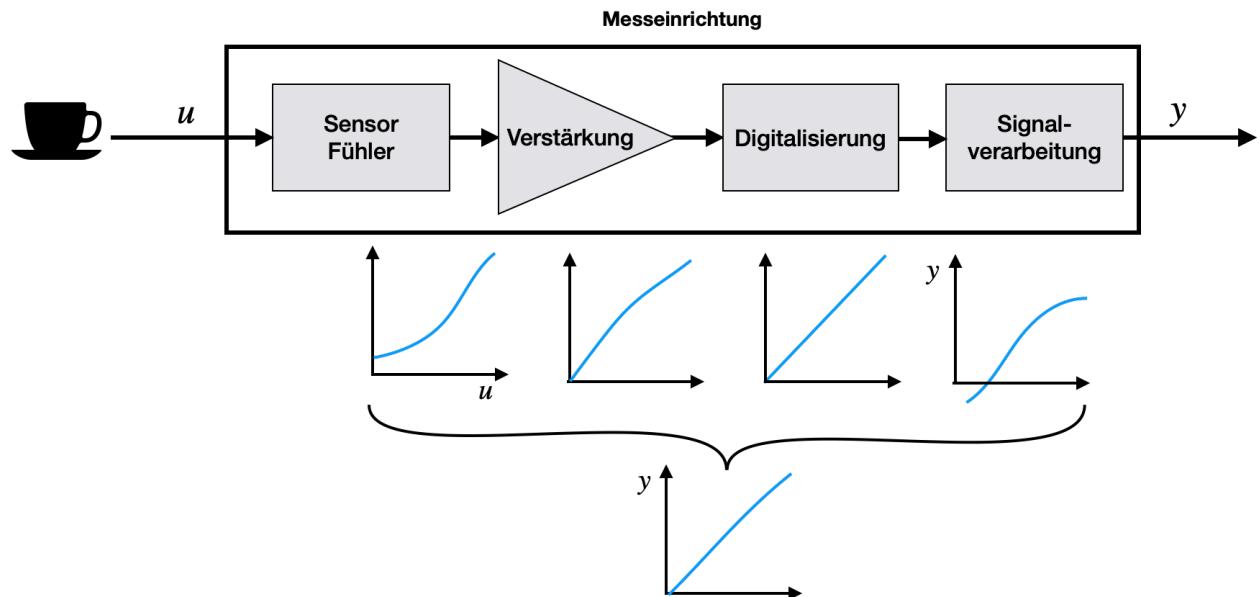
- **Nullpunktabweichung:** Dieser Kennlinienfehler ist additiv und somit absolute Messgeräteabweichung unabhängig von der Aussteuerung einer Messeinrichtung und wird auch Offset (-fehler) genannt. An jeder Stelle des Messbereiches wird eine Abweichung mit gleichem Betrag und Vorzeichen sowohl durch systematische, als auch durch zufällige Fehlerwirkungen verursacht. Die Beschreibung der *idealen* Übertragungsfunktion wird durch den additiven Fehler verändert. Man erkennt, dass der *relativer* Fehler für kleine Messwerte steigt, d.h. man möchte diese Messeinrichtungen möglichst groß aussteuern.
- **Multiplikativ:** Hierbei handelt es sich um eine absolute Abweichung der Anzeigegröße als Funktion ihrer Ausssteuerung. Technisch wird dieser Fehlertyp auch als Verstärkungsfehler bezeichnet, d.h. man beobachtet unerwünschte Veränderung des Übertragungsfaktors, also die Verstärkung einer Messeinrichtung ändert sich! Auch der multiplikative Fehler kann systematische und zufällige Ursachen besitzen. Diese Art von Abweichungen verlaufen aber immer durch den Nullpunkt und sind daher eher tolerierbar, auch bei kleinen Aussteuerungen.
- **Nichtlinearität:** Die oben genannten zwei Fehlertypen werden bei realen Messeinrichtungen fast immer gleichzeitig auftreten. Überlagerung der beiden Kurven führt immer zu unerwünschten Nichtlinearitäten im System.
- **Hysterese:** Die Kennlinie unterscheidet sich, je nachdem ob die Messgröße ansteigt oder abfällt. Die Komponente hat damit gewissermaßen ein Gedächtnis. Hier sollte man sich einmal vor Augen führen, dass dies *nicht* bedeutet,

dass das System zwei Kennlinien besitzt. Es hat vielmehr unendlich viele Kennlinien, da die Kennlinie nicht nur von der Richtung, in die sich die Messgröße ändert, variiert, sondern sie hängt auch von der aktuellen Position der Messgröße ab. Je nachdem, von welchem Umkehrpunkt aus sich die Messgröße in die jeweils andere Richtung weiter bewegt, muss man eine andere Kennlinie erwarten. Dieses Verhalten beobachtet man häufig bei mechanischen Sensorkonstruktionen (Beispiel: Druckmembran in einem Drucksensor) oder wenn magnetische Werkstoffe verbaut sind. In reinen Analogelektroniken ist die Hysterese meist nicht relevant bzw. eher relativ gering ausgeprägt.



5.3.3 Gesamtkennlinie

Nachdem wir uns einige Beispiele von Kennlinien, also Empfindlichkeitskurven, angesehen haben, können wir diese natürlich auch hintereinander schalten. Dies ist insbesondere wichtig, da Messsysteme eigentlich immer aus mehreren Komponenten bestehen, wie wir am Anfang des Kapitels in der *Grundstruktur* bereits gesehen haben. Jede einzelner Komponenten hat ihre eigene Kennlinie und wandelt eine Eingangsgröße in eine Ausgangsgröße um. Diese Kennlinien werden nun hintereinander geschaltet, sodass daraus eine Kennlinie resultiert, die das gesamte Messsystem beschreibt. Die einzelnen Kennlinien werden hierfür einfach aneinander *multipliziert*.



Optimalerweise möchte man erreichen, dass die Gesamtkennlinie eines Systems über einen möglichst großen Eingangsbereich für u linear ist, d.h. dass die einer Geraden entspricht. Dafür müssen die individuellen Kennlinien der Komponenten nicht zwangsläufig alle linear sein, sondern können sich am Ende kompensieren. Dies ist die große Kunst des Herstellens von Messsystemen.

Nun hängt allerdings der Verlauf einer Kennlinie nicht nur von der Herstellung ab, sondern auch von (äußereren) Einflussefekten während des Messprozesses. Es hängt folglich immer von der Herstellung und Anwendung eines Messsystems ab, wie genau man wirklich messen kann. Womit wir beim einem anderen Aspekt unserer Betrachtungen wären, den Messabweichungen, die im nächsten Jupyter Notebook erläutert werden.

Aufgabe

Aufgabe: Versuche im nächsten Code-Block eine ideale Gesamtkennlinienfunktion in einem bestimmten Bereich zu erhalten, ohne dass alle einzelnen Kennlinien linear sind. Verändere hierfür die Funktionen f1, f2, f3 und deren Parameter a_i, b_i, c_i.

```
#Benötigte Libraries:  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import plotly.offline as py  
py.init_notebook_mode(connected=True)  
import plotly.graph_objs as go  
import plotly.tools as tls  
import seaborn as sns  
import time  
import warnings  
warnings.filterwarnings('ignore')  
  
# Matplotlib Settings:  
plt.style.use('default') # Matplotlib Style wählen  
plt.rcParams['font.size'] = 10; # Schriftgröße  
  
def f1(x):  
    a_1 = .0  
    b_1 = 1.0  
    return b_1 * x + a_1  
  
def f2(x):  
    a_2 = 0.0  
    b_2 = 1.0  
    c_2 = 1.0  
    return c_2 * x**2 + b_2 * x + a_2  
  
def f3(x):  
    a_3 = 1.0  
    b_3 = 1.0  
    return b_3 * x + a_3  
  
start = 0.0  
stop = 10.0  
u = np.linspace(start, stop, num=50)  
  
## Hintereinanderschaltung:  
x1 = f1(u)  
x2 = f2(x1)  
x3 = f3(x2)  
  
f, axs = plt.subplots(1,3,figsize=(10,3))  
axs[0].plot(u, x1)  
axs[0].set_xlabel('u')
```

(continues on next page)

(continued from previous page)

```

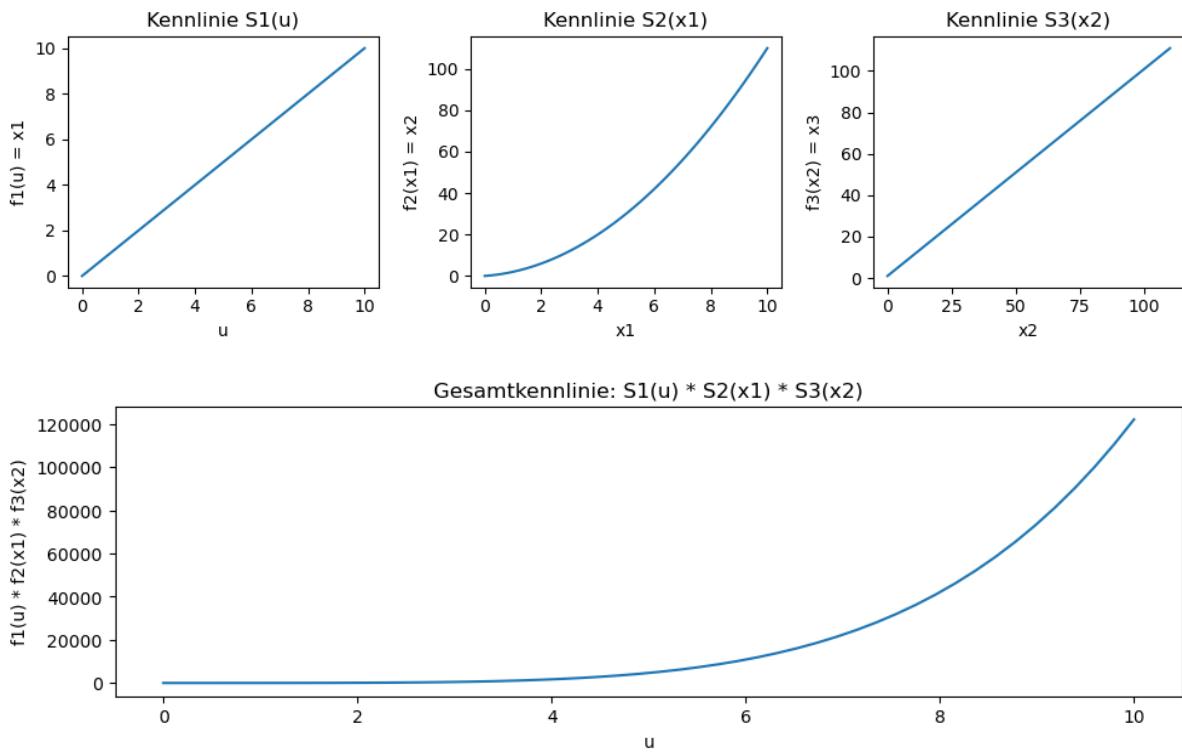
axs[0].set_ylabel('f1(u) = x1')
axs[0].set_title('Kennlinie S1(u)')

axs[1].plot(x1, x2)
axs[1].set_xlabel('x1')
axs[1].set_ylabel('f2(x1) = x2')
axs[1].set_title('Kennlinie S2(x1)')

axs[2].plot(x2, x3)
axs[2].set_xlabel('x2')
axs[2].set_ylabel('f3(x2) = x3')
axs[2].set_title('Kennlinie S3(x2)')
plt.tight_layout()
plt.show()

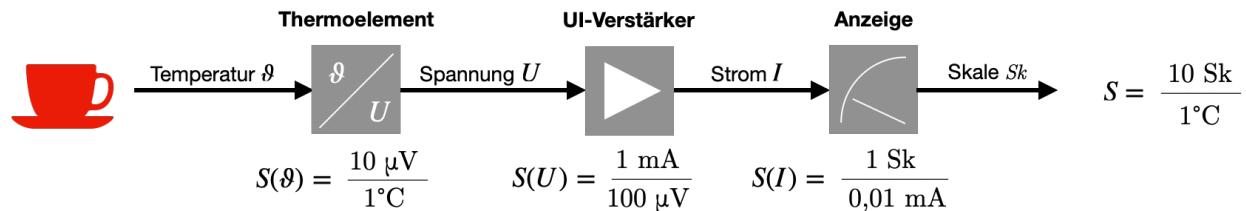
## Gesamtkennlinie:
f, axs = plt.subplots(1,1,figsize=(11,3))
axs.plot(u, x1*x2*x3)
axs.set_xlabel('u')
axs.set_ylabel('f1(u) * f2(x1) * f3(x2)')
axs.set_title('Gesamtkennlinie: S1(u) * S2(x1) * S3(x2)')
plt.show()

```



5.3.4 Messketten-Empfindlichkeit und Auflösung

Die wichtigste Kenngröße, die Empfindlichkeit der Messkette, kann analog einfach über Multiplikation berechnet werden. Das folgende Beispiel, im Bild dargestellt, zeigt, wie eine Temperatur über die Umwandlung in zuerst eine Spannung und dann in Stromstärke am Ende in eine analoge Anzeige gewandelt wird. Die Messketten-Empfindlichkeit ist, dass sich der Zeiger um 10 Skaleneinheiten weiter bewegt, wenn sich die Temperatur um 1°C verändert. Die **Auflösung** des Messsystems, also 0,5 Skaleneinheiten (was man typischerweise per Auge noch ablesen könnte) beträgt somit $0,05^{\circ}\text{C} = 50\text{mK}$.

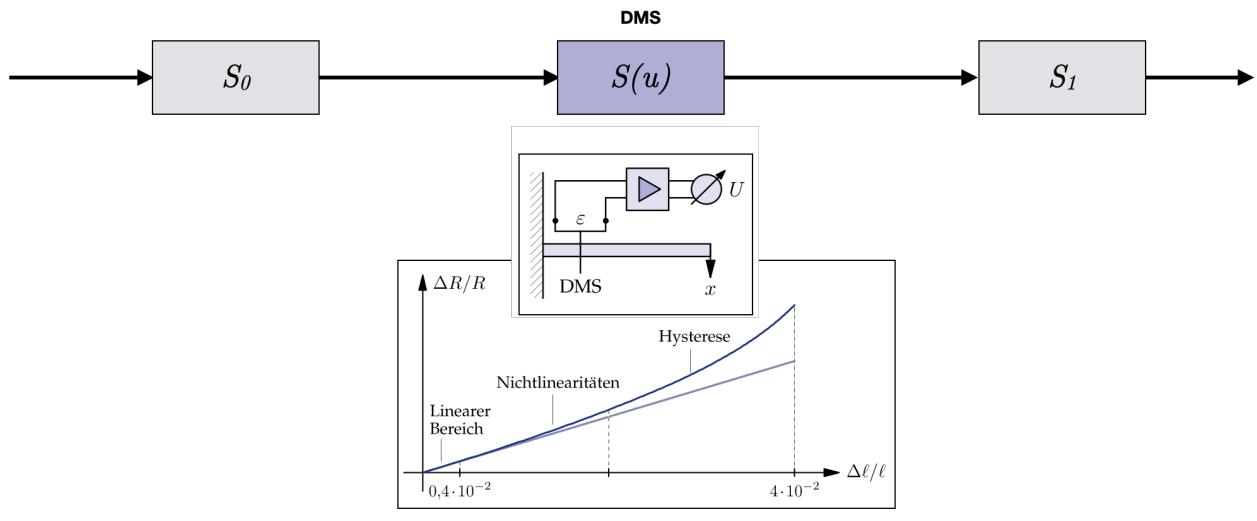


Messketten-Empfindlichkeit:

$$S = S_1 \cdot S_2 \cdot \dots \cdot S_n = \prod_{k=1}^n S_k \quad (n \text{ Anzahl der Messglieder})$$

5.3.5 Herabsetzen des Messbereichs

Ziel ist es, eine möglichst optimale Kennlinie zu erhalten, welche einer Geraden entspricht. Dadurch entstehen auch die wenigsten Kennenlinienfehler. Am Beispiel des Dehnungsmessstreifens (DMSs) soll dies einmal erläutert werden. Ein DMS ist nur für kleine Auslenkungen linear, das heißt er zeigt ein stark nicht-lineares Verhalten wenn er über 1 mm ausgelenkt wird. Die Idee ist nun, den DMS ausschließlich in diesem Bereich zu benutzen. Dafür wird eine weitere Komponente in der Messkette benötigt, die an den DMS angelegte Eingangsgröße auf einem bestimmten Bereich limitiert. Sollen auch größere Auslenkungen als 1 mm gemessen werden, so wird diese Komponente außerdem dafür sorgen, dass eine Verminderung der Auslenkung statt findet. Dies kann beispielsweise über eine Blattfeder realisiert werden. Diese nimmt große Auslenkungen auf und projiziert sie auf kleine Auslenkungen (in nachfolgender Grafik verkörper durch eine Komponenten mit Empfindlichkeit $S_0 \ll 1$), die dann mittels DMS gemessen werden können. Damit die Messgröße unverändert bleibt, dürfen wir am Ende die Verstärkung ($S_1 \gg 1$) nicht vergessen, sodass $S_0 \cdot S_1 = 1$. Dies nennt man auch *Kompensations-Bedingung*.



Kennlinie eines Dehnungsmessstreifens [Puente León, F. 2019]

MESSSIGNALE

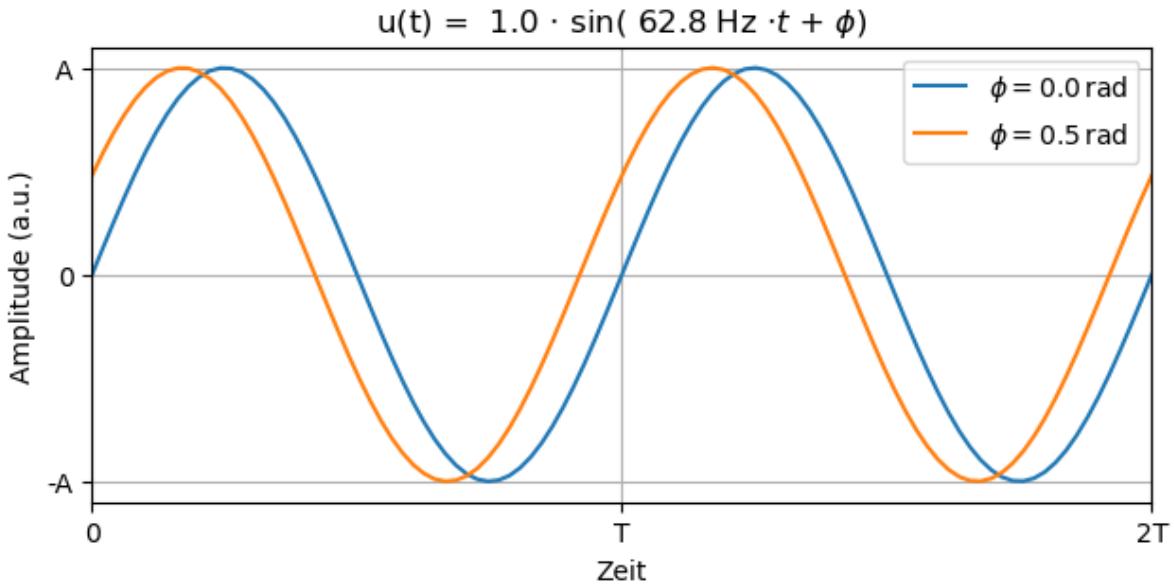
Bisher haben wir uns mit dem stationären Messungen einer physikalischen Größe befasst. Bis zu diesem Punkt sollte jedem bekannt sein was eine Kennlinie $y(x)$ ist und was die Sensitivität bedeutet. Außerdem haben wir uns mit Störungen und Unsicherheiten befasst, die während Messungen auftreten können und wie wir diese anhand von Fehlerabschätzungen und Fehlerfortpflanzungen abschätzen können. Außerdem sollte jede/r in der Lage eine Kurvenanpassung inkl. Fehlerbalken und Fehlerfortpflanzung auf gemessene Werte anwenden zu können.

- *Grundlagen*
- *Kenngrößen*
- *Modulation von Messsignalen*
- *Digitalisierung*
- *Fourier-Analyse*
- *Rauschen und spektrale Dichte*

6.1 Grundlagen

Wir wissen also bereits, dass Messwerte die gesuchten Informationen über eine physikalische Größe beinhalten. Häufig findet die Übertragung dieser Information in Form eines Messsignals statt. Die Frage ist nun: Was ist denn überhaupt ein *Signal*? Dieser Begriff wird umgangssprachlich häufig mehrdeutig benutzt. In unserem Falle, also im technischen Gebrauch, sprechen wir von einem *Zeitverlauf* einer physikalischen Größe. Damit es eindeutig wird, wollen wir statt des Begriffs *Signal* einfach das Wort *Messsignal* einführen. Das bedeutet, dass ein Signal nicht an eine bestimmte physikalische Größe gebunden ist. Ein Signal besteht nämlich in der Regel aus mehreren Parametern (*Informationsparameter*), die die *Träger* von physikalischen Informationen sein können. Das einfachste Beispiel kann ein Messsignal in Form einer sinusförmigen Spannung sein, z.B.:

$$u(t) = A \cdot \sin(\omega t + \varphi)$$



Die Funktion hängt von drei Parametern ab:

- der Amplitude A
- der Frequenz f , bzw. Kreisfrequenz $\omega = 2\pi f$
- der Phase (Phasenwinkel) ϕ

Je nach Aufgabe können wir das Messsignal auswerten indem wir entweder die Amplitude messen, oder die Frequenz, oder den Phasenwinkel (oder alle zusammen).

Vorteil der Nutzung von Messsignalen ist, dass physikalische Größen mittels dieser Umwandlung relativ einfach übertragen und weiterverarbeitet werden können. Dies ist insbesondere für Steuer- oder Regelungsprozesse von besonderem Interesse.

6.1.1 Wandlung von physikalischen Größen

Schon bei sehr einfachen Messeinrichtungen erfolgt im Allgemeinen eine Wandlung des aus der Umwelt gewonnenen Messsignals in einer für den Menschen interpretierbare, bzw. zur Weiterverarbeitung geeignete Signalform. Ganz wichtig ist hierbei, dass die Messinformation auf gar keinen Fall verändert werden darf. Bzw. wenn die Messinformation verändert werden sollte, dann sollte dies immer kontrolliert und bewusst geschehen, sodass die gewünschte Information immer noch zurück gewonnen werden kann. Dies ist in der Praxis aber wieder nur bedingt möglich.

Wir können auf jeden Fall folgendes zusammenfassen, wenn wir die Wandlung einer physikalischen Größe betrachten:

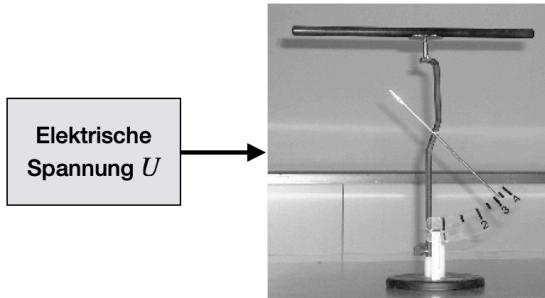
- Jede physikalische Größe wird in der Regel in ein Signal umgewandelt, da die wenigsten Größen in der Natur *direkt* messbar sind.
- Selbst elektrische Spannungen müssen irgendwie angezeigt werden können:
 - Die Amplitude könnte beispielsweise über einen Zeigerausschlag transformiert werden. Hierfür kann ein *Elektroskop* benutzt werden, welches über die Spannung, bzw. Spannungsdifferenzen, aufgeladen wird. Gleichnamige Ladungen stoßen sich ab und es wird eine elektrostatische Kraft erzeugt: $F = q \cdot E$, wobei q die Ladung und E die Feldstärke ist. D.h. zunächst einmal kann das Elektroskop Ladungen messen. Da es sich aber um ein Kondensator handelt (zwei voneinander isolierte Elektroden) können wir mittels $Q = CU$ auch Spannungen messen. C ist hierbei die Kapazität des Kondensators (konstant).

- Für die Temperatur könnte ein temperaturabhängiger Widerstand benutzt werden, der einen Spannungsabfall erzeugt.

Für die Wandlung einer Signalform in eine andere sollte folgendes gelten:

- Die physikalische Größe der Informationsparameter darf nicht verändert werden, sie wird lediglich in eine neue Signalform umgewandelt.
- Beispiel: Ein Messverstärker, der eine Amplitude eines Messsignals für die Anzeige vorbereitet, wandelt eine *Amplitude* in eine *Amplitude*.

Zur Informationsübertragung werden häufig **elektromagnetische Wellen** oder **elektrische Signale** benutzt. In einigen Bereichen sind aber auch **mechanische, pneumatische und hydraulische** Signale üblich, vor allem wenn man sich technische Regeleinrichtungen ansieht, spielen die letztgenannten Signale eine wichtige Rolle. Die nachfolgenden Beschreibungen kann aber auch jede Art von Messsignal angewendet werden.



Zeigerelektroskop [Wikipedia]



Heinrich-Hertz-Turm in Hamburg [Wikipedia]

6.1.2 Elektromagnetische Wellen als Informationsträger

Radiowellen schwingen bis zu 300 Millionen-mal pro Sekunde (= 300 MHz). Je nach Frequenz, bzw. Wellenlänge, werden Radiowellen in Mittel-, Kurz- und Ultrakurzwellen klassifiziert. Deren Ausbreitungsgeschwindigkeit beträgt Lichtgeschwindigkeit (es sind ja elektromagnetische Wellen) und liegt somit bei 299.792.458 m/s (also ca. 300.000km/s). Die Wellenlänge kann mittels folgender Formel aus Frequenz f und Geschwindigkeit c berechnet werden:

$$\lambda = \frac{c}{f}$$

Die Wellenlängen von Radiowellen sind somit über 1m lang.

```
f = 300e6 # Frequenz in Hz
c = 299792458 # Lichtgeschwindigkeit in m/s
wellenlaenge = c/f
print('Die Wellenlänge einer Schwingung mit Frequenz f = %5.2f MHz beträgt %5.2f m'
      %(f/1e6,wellenlaenge))
```

Die Wellenlänge einer Schwingung mit Frequenz $f = 300.00$ MHz beträgt 1.00 m

Da die Wellenlänge sehr groß ist, wird diese Art von Strahlung kaum durch Teilchen in unserer Atmosphäre oder Ionosphäre absorbiert und abgeschwächt, sondern tatsächlich reflektiert. Dadurch eignet sie sich hervorragend für die Signalausbreitung von langen Distanzen. Die Ionosphäre agiert quasi als riesengroßer Spiegel für die Radiowelle, wodurch die Strahlung zurück zur Erde reflektiert wird. (Tatsächlich wird dank Strahlung aus dem Weltall dafür gesorgt, dass Elektronen aus Atomen in der Ionosphäre gelöst werden. Dadurch bilden sich positiv geladene Atomreste, welche zusammen mit den freien Elektronen einen Spiegel für Radiowellen bilden.)

- Radiowellen mit Frequenzen **unter 30 MHz** können von der Ionosphäre vollständig reflektiert werden. Einige Telekommunikations- und Radarsysteme nutzen die reflektierenden Eigenschaften der Ionosphäre für Radiowellen unterhalb von etwa 30 MHz zur Überbrückung großer Entfernung.
- Radiowellen mit Frequenzen **oberhalb von 30 MHz** können die Ionosphäre durchdringen, so dass eine Kommunikation mit Satelliten möglich wird. (Die Radiowellen regen die freien Elektronen zum Schwingen an, wodurch ein Dipolstrahler entsteht. Dieser erzeugt Strahlung mit einer Phasenverschiebung, aber der gleichen Frequenz. Als Resultat sieht es so aus, als würde die einfallende Strahlung von Erde vom Lot webgebrochen werden (Die Ionosphäre ist also ein optisch dünneres Medium. Je steiler der Einfallsinkel (oder je höher die Frequenz), umso tiefer dringt die Radiowelle in die ionisierte Schicht ein).
- Selbst im **GHz-Bereich** macht sich ein Brechungseinfluss noch störend bemerkbar, wodurch Messfehler bei der Satellitennavigation entstehen.

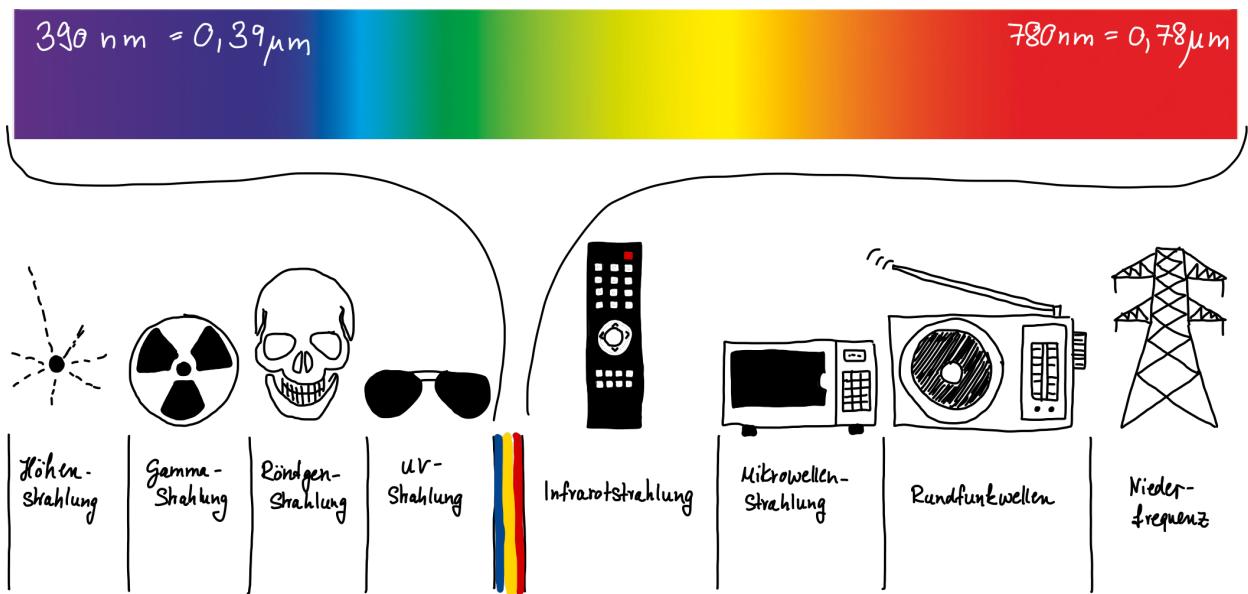
Die Reichweite von Signale zur Übertragung von Informationen hängt also von der Wellenlänge bzw. Frequenz der Strahlung ab und den Umgebungsparametern (wie z.B. die Ionosphäre). Weitere elektromagnetische Wellen, welche gerne zur Signalausbreitung benutzt werden, sind z.B.:

- Funkwellen (~1m - ~1km)
- Mikrowellen (~1mm - ~1m)
- Infrarot-Strahlung (~1um - ~1mm)
- UV-Strahlung (~100nm - ~390nm)
- Röntgenstrahlung (~30pm - ~1nm)

Das sichtbare Licht, ebenfalls eine elektromagnetische Welle, ist nur in einem verhältnismäßig kleinem Bereich zu finden.~ Es gilt allgemein, je kleiner die Wellenlänge, desto höher die Frequenz, desto höher ist die Energie, die transportiert wird:

$$E = h \cdot f$$

mit $h = 6.626 \cdot 10^{-34}$ Js dem Planck'schen Wirkungsquantum.



Note: Exkurs: Funkwellen in der Ionosphäre

Radiowellen unterschiedlicher Frequenz und Einfallswinkel werden an der Ionosphäre reflektiert und auf die Erde zurück geworden, oder durchgelassen, wodurch Satellitenkommunikation ermöglicht wird. Ob und wie Funkwellen reflektiert werden, hängt von der Dichte und Höhe der Ionosphärenschicht ab, und von der Frequenz der einfallenden Funkwelle.

Durch Sonneneinstrahlung im ultravioletten Bereich werden Elektronen in der Ionosphäre von ihren Luftatomen getrennt. Dadurch entstehen freie Elektronen und positiv geladene Ionenrumpfe bleiben zurück. Je stärker die Sonneneinstrahlung (je mehr UV-Lichtteilchen), desto mehr Elektronen frei gesetzt und die Elektronendichte, N_e , steigt an.

- 0-90km: hier gibt es keine freien Elektronen, keine Ionosphäre!
- 50-90km: D-Schicht: keine Bedeutung für unsere Funkwellen (sie existiert auch nur am Tag wenn die Sonneneinstrahlung am stärksten ist)
- ~100km: E-Schicht
- ~200km: F1-Schicht
- 250-400km: F2-Schicht: größte Elektronen-/Ionendichte mit etwa $N_e = 1 \text{ Million/cm}^3$ freie Elektronen
- noch höher: Luftmoleküle werden seltener, wodurch die Elektronendichte trotz stärkerer Strahlung wieder abnimmt

Ionosphäre ist tagüber, bei maximaler Sonneneinstrahlung, am stärksten mit freien Elektronen versetzt. Nachts nimmt die Ionosphäre ab, da sich die freien Elektronen wieder mit den Ionen verbinden. Dies dauert je nach Höhe der Schicht unterschiedlich lang. Die F2-Schicht beispielsweise ist auch nachts noch so stark ausgeprägt, dass sie zur Übertragung von Funkwellen benutzt werden kann.

Wenn Funkwellen in die Ionosphäre eintreten, ändern sie ihre Richtung, ähnlich wie Lichtstrahlen an einem Prisma. Niedrigere Frequenzen brechen sich hierbei leichter, also hohe Frequenzen. Schräge Wellen berechnen sich leichter als senkrecht eintreffende Strahlen.

Das Verhalten von Funkwellen beim Auftreffen auf freie Elektronen kann mittels Plasmatheorie beschrieben werden. In einem Plasma treten lokal verdichtete Bereiche von freien Elektronen auf. Das sogenannten Elektronengas. An diesen Stellen ist die Dichte der Elektronen so hoch, dass die Coulombkraft abstoßend zwischen den Elektronen in Vorschau tritt. Dadurch werden die Elektronen wieder auseinander getrieben. Aufgrund der Trägheit der Elektronen treten hierdurch allerdings an anderen Stellen wieder lokale Verdichtungen auf und der Effekt wiederholt sich. Dieses Hin und Her von lokalen Ladungsüberschüssen resultiert in einer periodischen Schwingung, die sogenannte Plasmafrequenz, mit der die

Elektronendichte um ihre mittlere Dichte oszilliert:

$$f_p = \frac{1}{2\pi} \sqrt{\frac{N_e e^2}{\epsilon_e m_e}} \approx 8,978 \text{ Hz} \cdot \sqrt{\frac{N_e}{m^3}}$$

Hier ist $\epsilon_0 = 8,854 \cdot 10^{-12} \frac{\text{As}}{\text{Vm}}$ die elektrische Feldkonstante, N_e die Elektronendichte, m_e die Elektronenmasse, $e = 1,602 \cdot 10^{-19} \text{ C}$ die Elementarladung. Abhängig von der Höhe haben wir in der Ionosphäre unterschiedliche Elektronendichten und können folgende Plasmafrequenzen berechnen:

- 100km Höhe: $f_p = 1,5 \text{ MHz}$
- 800km Höhe: $f_p = 7 \text{ MHz}$

Der Brechungsindex n der Ionosphäre lässt sich mittels folgender Formel berechnen:

$$n = \sqrt{1 - \frac{f_p^2}{f^2}}$$

Hierbei wurde die Zyklotronfrequenz der Elektronen, die durch das Magnetfeld der Erde hervorgerufen wird, vernachlässigt. Unter Berücksichtigung der Kreisbewegung der Elektronen um die Feldlinien im Erdmagnetfeld würde ansonsten ein zusätzlicher Frequenzterm von etwa 1,3MHz zu der Gleichung hinzugefügt werden und je nach Polrisation der einfallenden Funkwelle (links-zirkular oder recht-zirkular) würden sich leicht unterschiedliche Brechungsindizes ergeben, abhängig davon, ob die Drehrichtung mit der Elektronenbewegung übereinstimmt.

Anhang dieser Formel sind zwei Fälle zu berücksichtigen:

- $f < f_p$: der Brechungsindex wird imaginär. D.h. es findet vollständige Reflexion statt und Lang- und Mittelwellen kommen wieder zurück zur Erden
- $f > f_p$: der Brechungsindex ist real. D.h. die Funkwellen können die Schicht durchdringen (bei senkrechtem Einfall der Strahlung)

Das bedeutet für unsere beiden Fälle von oben, dass unter senkrechtem Einfall:

- in 100km Höhe Funkwellen mit einer Frequenz von $<1,5\text{MHz}$ vollständig reflektiert werden. Wellen höherer Frequenz können diese Ionosphärenschicht durchdringen.
- in 800km Höhe Funkwellen mit einer Frequenz von $<7,0\text{MHz}$ vollständig reflektiert werden. Wellen höherer Frequenz können diese Ionosphärenschicht durchdringen.
- Funkwellen mit Frequenzen $>7\text{MHz}$ werden die Erde nicht wieder erreichen.

Die Plasmafrequenz ist in dieser Annäherung also die kritische Frequenz, bis zu welcher Funkwellen unter einem senkrechten Einfallswinkel, vollständig reflektiert werden.

Des Weiteren wird beobachtet, dass die Durchlässigkeit der Ionosphäre für Funkwellen vom Eintreffwinkel abhängt und näherungsweise über folgende Formel beschrieben werden kann:

$$f_{\text{MUF}} \approx \frac{f_{\text{krit}}}{\sin(\alpha)} = f_{\text{krit}} \cdot \sqrt{1 + \left(\frac{d}{2h_{\text{krit}}}\right)^2}$$

wobei α der Winkel ist, der die Strahlung zum Horizont bildet, f_{krit} die kritische Frequenz ist, unter welcher die Strahlung vollständig reflektiert werden würde, d ist die Entfernung zwischen Sender und Empfänger und h_{krit} die virtuelle Höhe der Reflektion. Hier ist f_{MUF} die **Grenzfrequenz** der Funkwelle, welche gerade noch von der Schicht reflektiert wird. MUF steht hierbei für *maximum usable frequency*. Analog gibt es auch die LUF-Frequenz, die *lowest usable frequency*.

6.2 Kenngrößen

Eine Wechselgröße, z.B. eine Wechselspannung, liegt in der allgemein Form als Sinusschwingung vor. Die Form wollen wir im folgenden wie folgt bescheiben: $u(t) = \hat{U} \cdot \sin(\omega t)$

In elektronischen Schaltungen hat man dazu eine Stromstärke, der dieser Wechselspannung zugehörig ist, und sich allgemein wie folgt schreiben lässt:

$$i(t) = \hat{I} \cdot \sin(\omega t + \varphi)$$

Der Stromstärke haben wir die Phase φ hinzugefügt, da durch besondere Komponenten in der Schaltung, wie z.B. ein Kondensator oder eine Spule, eine Verzögerung zwischen Spannung und Stromstärke auftritt. Im klassischen Stromkreis, mit lediglich einem Widerstand, ist so eine Verzögerung meist vernachlässigbar.

- Mit $\omega = 2\pi f$ wird die **Kreisfrequenz** bezeichnet, welche ebenfalls die Einheit Hz hat. Genauso wie die Frequenz f .
- Die **Frequenz** $f = \frac{1}{T}$ ist der Kehrwert der **Periodendauer** T .
- Die **Phase** φ gibt die Phasenverschiebung oder Verzögerung des Messsignals an

Bei einem Wechselsignal interessiert man sich häufig nicht für den Zeitverlauf, oder den einzelnen Wert zu einem bestimmten Zeitpunkt, den sogenannten Momentanwert. Dieser hat nämlich keine große Aussagekraft, da er sich gemäß der zugrundeliegenden Signalform ständig ändert. Für viele Fragestellungen sind Kenngrößen relevant, die sich aus der Periode des Wechselsignals ermitteln lassen.

6.2.1 Scheitelfaktor / Amplitude

Mit \hat{U} bezeichnen wir den **Scheitelfaktor** oder die **Amplitude**. Dies ist der maximale Ausschlag eines Messsignals innerhalb einer Periode (oder Messzeit) T .

6.2.2 Gleichanteil

Der **Gleichanteil** einer Wechselgröße wird über den arithmetischen Mittelwert über eine Periode T berechnet. Es ist nur dann aussagekräftig, wenn nicht gerade eine Signal vorliegt was keinen Gleichanteil hat. Andernfalls ist der Wert immer 0. $\bar{u} = \frac{1}{T} \int_t^{t+T} u(t) dt$

6.2.3 Gleichrichtwert

Wenn wir uns jetzt einen Fön oder Lampe, angeschlossen an eine normale Steckdose mit Wechselstrom vorstellen, so ist der Gleichanteil der Wechselspannung 0, aber trotzdem wirkt die Spannung, da Fön oder Lampe arbeiten. Das heißt es kommt offensichtlich nicht auf die Polarität der Spannung an, sondern nur darum, dass im Durchschnitt irgendeine Art von beliebig gepolter Spannung anliegt. Mathematisch gesehen entspricht dies einer Betragsbildung und wird durch den Gleichrichtwert definiert. Das Signal wird mittels analoger Schaltung *gleichgerichtet* (das sehen wir uns später noch mal genauer an), bevor es zur Mittelwertbildung kommt. Daher auch der Name. Der Gleichrichtwert hat auch für Signale ohne Gleichanteil eine Aussagekraft. Der **Gleichrichtwert** einer Wechselgröße ist der Mittelwert des Absolutwertes einer Wechselgröße berechnet sich zu: $\overline{|u|} = \frac{1}{T} \int_t^{t+T} |u(t)| dt$

6.2.4 Effektivwert

Aus der Physik ist bekannt, dass die Berechnung der elektrischen Leistung *nicht* über den Einbezug von Gleichrichtwerten erfolgt. Vielmehr muss der **Effektivwert** (auch RMS genannt = root mean square) berechnet werden: $U = u_{\text{eff}} = \sqrt{\bar{u^2}} = \sqrt{\frac{1}{T} \int_t^{t+T} u(t)^2 dt}$

6.2.5 Scheitelfaktor und Formfaktor

Mittelwert, Gleichrichtwert und Effektivwert lassen sich für jede beliebige periodische Signalform berechnen und sind *unabhängig* von der Frequenz. Zwischen Ihnen sind Umrechnungsfaktoren definiert, Scheitelfaktoren oder Formfaktoren, welche das Verhältnis zwischen Effektivwert und anderen Kenngrößen angeben. Wir werden später noch sehen, dass die Messung von Effektivwerten sehr aufwendig ist (die mathematischen Operationen im Integral zeigen dies auch). Es ist einfacher den Scheitel- oder Gleichrichtwert zu messen und mittels dieser Umrechnungs-Faktoren den Effektivwert anschließend zu berechnen. Die Faktoren sind jedoch für jede Signalform anders!

```
* **Scheitelfaktor (Crest-Faktor)**:  
$$k_S = \frac{\hat{u}}{u_{\text{eff}}}$$  
  
* **Formfaktor**:  
$$k_F = \frac{u_{\text{eff}}}{\overline{|u|}}
```

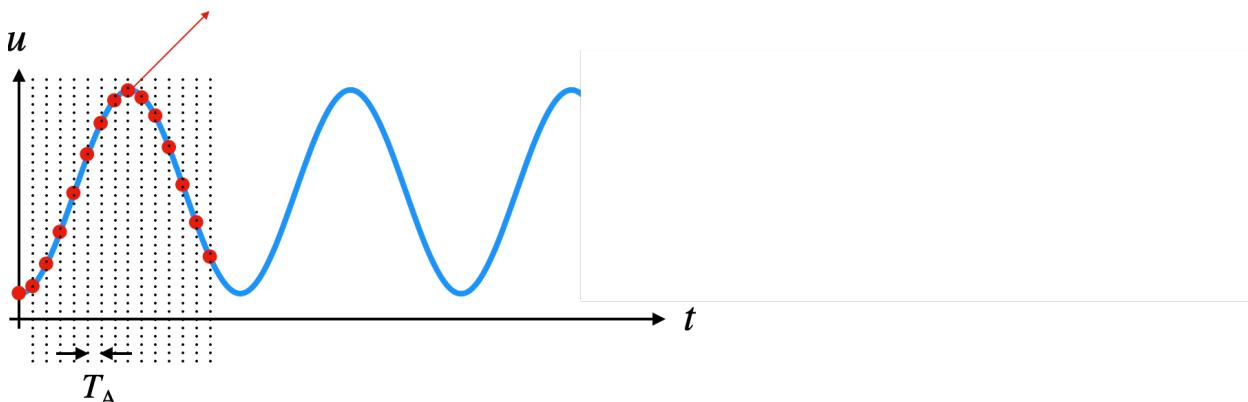
Allgemein kann man sagen, dass je größer der Scheitelfaktor, desto mehr übersteigt der Spitzenwert eines Signals den Effektivwert. Für die Praxis bedeutet dies, dass bei der Ermittlung eines Effektivwertes die Messeinrichtung aufgrund sehr hoher Spitzenwerte bereits übersteuert, obwohl der Effektivwert eigentlich noch ausreichend Aussteuerungsreserven aufweist. Besonders kritisch ist dieses Verhalten bei Pulsfolgen mit sehr kleinem Tastverhältnis. Hier kann der Scheitelfaktor unter Umständen Werte von 10 oder mehr annehmen. Messgeräte geben daher häufig Scheitelfaktoren vor, welche nicht überschritten werden sollten.

6.2.6 Kenngrößen bei der Digitalisierung

Bei der digitalen Signalverarbeitung können die oben genannten Kenngrößen ebenfalls mathematisch ermittelt werden, indem die Integrale durch diskrete Summen ausgetauscht werden. Der Effektivwert berechnet sich dann wie folgt:

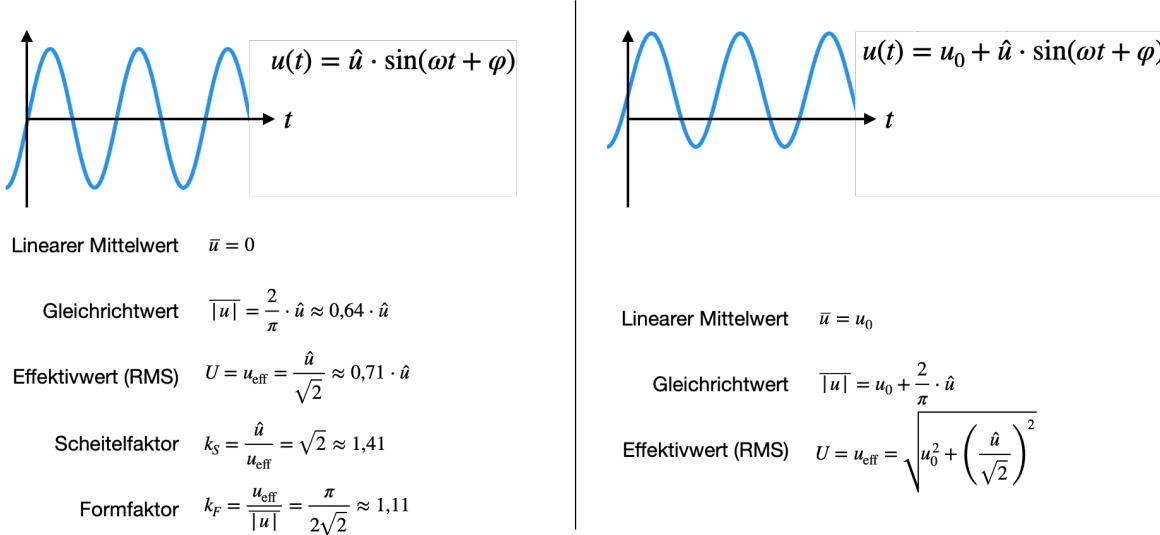
$$U = u_{\text{eff}} = \sqrt{\bar{u^2}} = \sqrt{\frac{1}{T} \sum_{k=1}^n T_A \cdot u_k^2}$$

wobei n die Abtastwerte sind, T_A der Abstand zwischen zwei Messungen und u_k der einzelne Messwert zum Zeitpunkt $k \cdot T_A$.



6.2.7 Sinusschwingung

Beispiel eines Sinussignals mit und ohne Gleichanteil ist im folgenden Bild einmal dargestellt. Für beide Fälle wurden die Integrale von oben gelöst und aus den entsprechenden Ergebnissen die Formeln für Scheitelfaktor und Formfaktor abgeleitet.



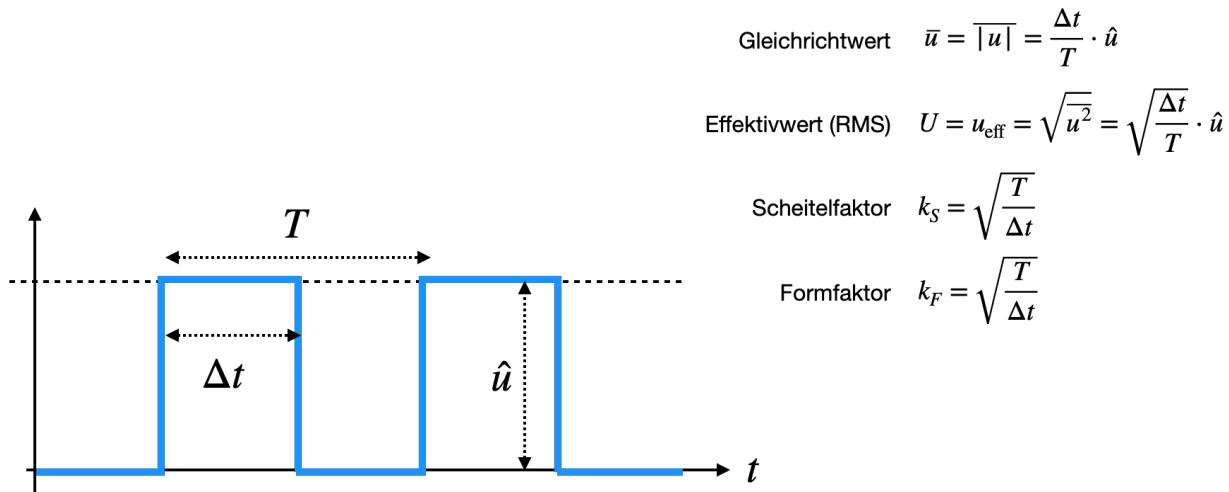
Die Angabe von Scheitel- und Formfaktor haben für Signale mit Gleichanteil keine größere Bedeutung und müssen nicht berechnet werden. Im Vergleich der beiden Beispiele zeigt sich, dass der Effektivwert eines Sinussignals mit Gleichanteil der quadratische Mittelwert der Effektivwerte des Gleichanteils und dem Wert des Gleichanteils u_0 entspricht.

6.2.8 Pulsweitenmodulation

Ein weiteres Beispiel ist für eine Pulsweitenmodulation berechnet und die Formeln können aus folgendem Bild abgelesen werden. Er handelt sich um ein unsymmetrisches Rechtecksignal. In diesem Fall sprechen wir von einem Pulsweitenmodulierten Signal, da die ‘An’-Zeit innerhalb einer bestimmten Anwendung nicht immer gleich groß sein muss. Das Verhältnis

$$\tau = \frac{\Delta t}{T}$$

wird auch Tastverhältnis genannt. Hierüber kann ein Messwert analog codiert und übertragen werden, während das Signal selber digital ist.



6.3 Modulation von Messsignalen

Im ersten Abschnitt haben wir schon angedeutet, dass nicht nur physikalische Größen in Messsignale umgewandelt werden, sondern auch Signale in andere Signalformen. Sehr häufiges Beispiel aus der Messtechnik ist der Informationsparameter Amplitude der in Frequenz umgewandelt wird. Dies hat den einfachen Grund, dass die Frequenz eines Messsignals extrem unempfindlich gegenüber Störeinflüssen aus der Umgebung ist. Eine Amplitude wird viel schneller verfälscht, oder abgeschwächt, als die Frequenz. Ein weiterer Vorteil ist, dass die Frequenz eines Messsignals sehr einfach digitalisiert werden kann, wodurch auch die physikalische Größe schnell digital zur Verfügung steht. Man könnte beispielsweise eine einfache Triggereinrichtung nutzen, welche die Frequenz eines Messsignals in Pulse umwandelt, die mit einer bestimmten Wiederholrate (also mit der Frequenz des Signals) zur Verfügung gestellt werden. Diese Pulse werden dann innerhalb eines definierten Zeitfenster ausgezählt. Mit solchen *Frequenzzählern* werden wir uns später noch ausführlicher beschäftigen. Für solch eine Art von Messsignal wird eine **Modulation** des Messsignals benötigt, welche die physikalische Größe (z.B. ein Audiosignal) auf den Träger (z.B. eine Radiowelle), aufbringt.

6.3.1 Träger- und Nutzsignale

Klassischerweise benutzt man eine sinusförmige Schwingung konstanter Frequenz und Amplitude als **Trägerschwingung**. Durch die Modulation können folgende Parameter der Trägerschwingung beeinflusst werden:

- die Amplitude. Dies ist bekannt als Amplitudenmodulation (AM)
- die Frequenz. Dies ist bekannt als Frequenzmodulation (FM)
- die Phase. Hierbei wird der Nullphasenwinkel verschoben. Dies ist bekannt als Phasenmodulation (PM)

Im Anschluss muss mittels **Demodulation** der originale Zeitverlauf der physikalischen Größe zurückgewonnen werden.

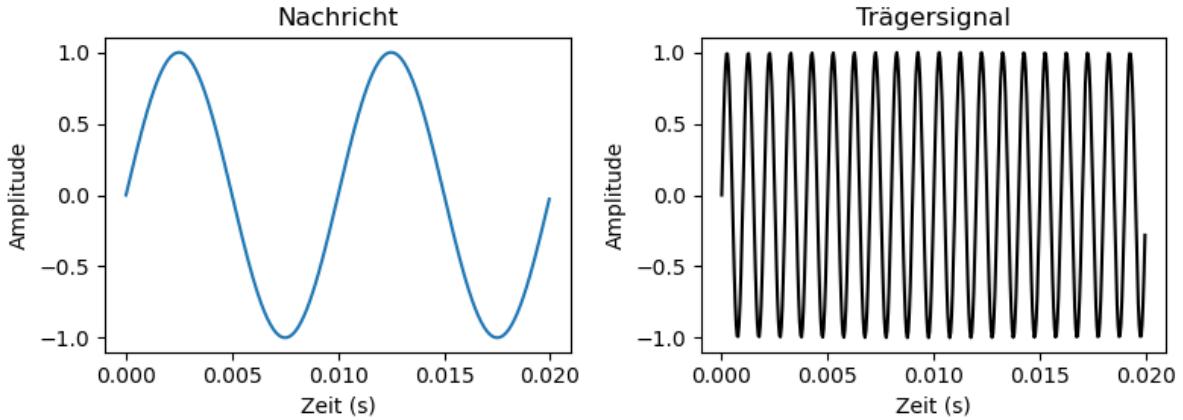
Nutzsignale wie Sprache oder Musik können häufig nicht direkt über gewünschte Übertragungsmedien wie beispielsweise einen Funkkanal übertragen werden. Hierfür wird ein Trägersignal benötigt, mittels welchem das Nutzsignal übertragen werden kann. Wir nehmen ein sinusförmiges Nutzsignal (*message*) an:

$$m(t) = A_m \cdot \sin(2\pi f_m t)$$

mit der Amplitude A_m und Frequenz $f_m = 100 \text{ Hz}$. Außerdem widmen wir uns zunächst einem *sinusförmigen* Trägersignal (*carrier*)

$$s(t) = A_c \cdot \sin(2\pi f_c t + p_0)$$

mit der Amplitude A_c und der Trägerfrequenz $f_c = 1 \text{ kHz}$. p_0 ist eine Konstante, die Phase zum Zeitpunkt $t = 0$. Beide Signale sind den folgenden Diagrammen für $f_m = 100 \text{ Hz}$ und $f_c = 1 \text{ kHz}$ dargestellt.



Es gibt nun verschiedene Methoden um die Nachricht auf das Trägersignal zu modulieren und so zu übertragen:

6.3.2 Amplitudenmodulation

Zur Übertragung wird das Nutzsignal in einen anderen Frequenzbereich verschoben, was beispielsweise durch eine **Amplitudenmodulation (AM)** geschehen kann. Durch das Verschieben können auch mehrere Nutzsignale gleichzeitig und ohne gegenseitige Störung übertragen werden.

$$y_{\text{AM}}(t) = [A_c + M_a \cdot m(t)] \cdot \sin(2\pi f_c t)$$

M_a ist die Modulationsstärke und $m(t)$ die modulierende Funktion oder kurz der Modulator. Hier wäre es entsprechend die Nachricht, die auf das Trägersignal moduliert wird.

Setzt man ein sinusförmiges Nutzsignal ein, so erhält man:

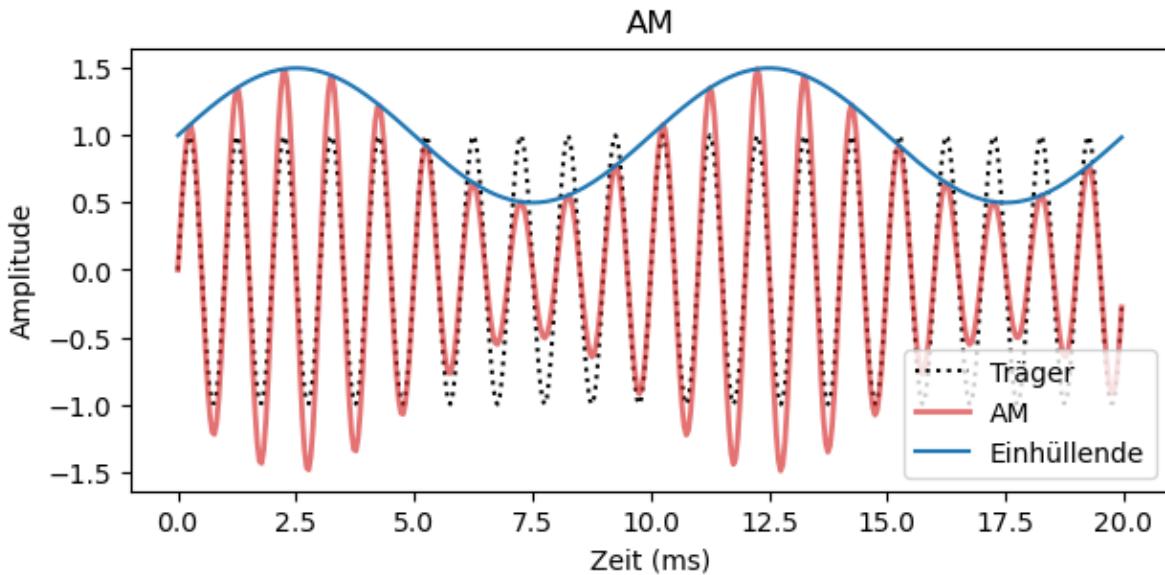
$$\begin{aligned} y_{\text{AM}}(t) &= [A_c + M_a \cdot A_m \cdot \sin(2\pi f_m t)] \cdot \sin(2\pi f_c t) \\ &= A_c \sin(2\pi f_c t) + \frac{M_a A_m}{2} [\cos\{2\pi(f_m - f_c)t\} - \cos\{2\pi(f_m + f_c)t\}] \end{aligned} \quad (6.1)$$

Aufgabe

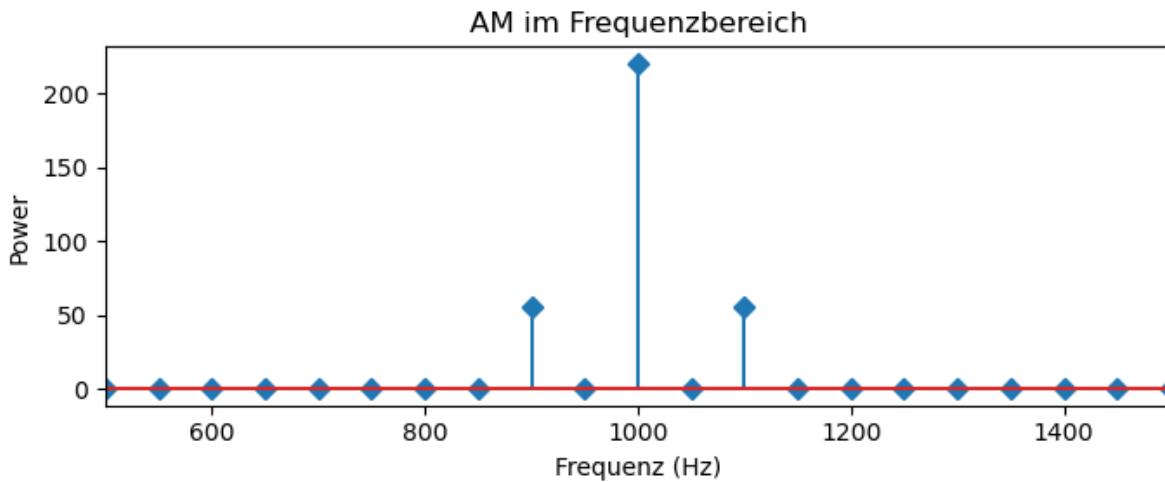
Beweise die obige Umformung mithilfe der trigonometrischen Umrechnungsformel $\sin \alpha \sin \beta = \frac{1}{2}(\cos(\alpha - \beta) - \cos(\alpha + \beta))$!

Aus Formel kann man das entstandene Frequenzspektrum ablesen (siehe *Fourier-Analyse*). Vorne in der Formel steht das Trägersignal mit der Frequenz f_c und hinten stehen zwei Terme, die sogenannten Seitenbänder, mit den Frequenzen $f_m - f_c$ und $f_m + f_c$. Diese einfachste Modulationsart der AM nennt man deshalb auch **Zweiseitenbandmodulation (ZSB oder englisch DSB)** mit Träger. Hier steckt die Information in den Seitenbändern, während der Träger selbst bei der Übertragung nur unnötigen Ballast darstellt. Wenn sich die Amplitude des Nutzsignals ändert, ändert sich auch die Amplitude der Seitenfrequenzen. Wenn sich die Frequenz des modulierenden Signals ändert, ändern sich auch die Frequenzen der Seitenbänder.

Eine Amplitudenmodulation für einen Modulationsgrad von 0,5 sieht wie folgt für die beiden oben definierten Nutz- und Trägersignale aus:



Alternativ zur Berechnung des modulierten Signals im Zeitbereich kann dies auch über die Fourier-Transformation im Frequenzbereich geschehen. Die dazu inverse Fourier-Transformation führt wieder in den Zeitbereich (siehe [Fourier-Analyse](#)).



Demodulation

Es gibt prinzipiell zwei Möglichkeiten die Modulation rückgängig zu machen um das ursprüngliche Nutzsignal zu rekonstruieren:

Bei schwachen oder stark gestörten Signalen wird auf der Empfängerseite ein lokaler Träger benutzt, der in Phase zu Träger des empfangenen Signals ist. Beide Träger sind synchron und damit kohärent zu einander. Dieses Verfahren ist technisch gesehen aufwendiger als die zweite Option. Zunächst wird das empfangene Signal, $y_{AM}(t)$, bestehend aus den beiden Seitenfrequenzen und dem Träger, mit dem lokalen Träger multipliziert und anschließend werden die hohen Frequenzanteile $2f_c$ mit einem Tiefpass und der Gleichanteil mit einem Hochpass heraus gefiltert. Das gesuchte Nutzsignal bleibt übrig, jedoch nur mit halber Amplitude: \$\$

:::{admonition} Aufgabe

:class: tip Beweise die obige Umformung unter Vernachlässigung der hohen Frequenzanteile $2f_c$ und des Gleichanteils. \$\$

Diese einfachste Form der Demodulation kommt ohne die aufwendige Erzeugung eines lokalen Trägers aus und ermöglichte deshalb vor hundert Jahren die Verbreitung von Rundfunksendern. Das Verfahren ist allerdings nur bei ausreichend starken Signalen anwendbar und erfordert im Regelfall einen vorhergehenden Verstärker. Hier wird das gesuchte Frequenzband mit einem Bandpass herausgefiltert, anschließend mit einer Diode gleichgerichtet und zum Schluss mit einem Tiefpass geglättet. Der enthaltene Gleichanteil wird gegebenenfalls mit einem Hochpass entfernt.

Auf Grund der Einfachheit dieses Verfahrens kann das empfangene Signal leicht durch Verzerrungen gestört werden.

6.3.3 Phasenmodulation

Frequenzmodulation und Phasenmodulation hängen mathematisch eng zusammen. Eine Phasenmodulation eines sinusförmigen Trägers kann man sehr einfach ausdrücken. Zunächst der unmodulierte Träger:

$$s(t) = A_c \cdot \sin(2\pi f_c t + p_0)$$

Wir können die momentane Phase als Funktion der Zeit schreiben:

$$p(t) = (\omega_c t + p_0)$$

mit der Kreisfrequenz $\omega_c = 2\pi f_c$.

Nun wird die momentane Phase durch Addition eines Modulators verändert, dadurch entsteht der Ausdruck für eine Phasenmodulation:

$$y_{PM}(t) = A_c \sin(2\pi f_c t + p_0 + M_p m(t))$$

Dabei ist M_p die Modulationsstärke und $m(t)$ die modulierende Funktion oder kurz der Modulator, hier die entsprechende Nachricht, $y_{Nachricht}(t)$. Entsprechend:

$$p(t) = 2\pi f_c t + p_0 + M_p m(t)$$

Man sieht, dass eine Phasenmodulation mathematisch sehr einfach auszudrücken ist.

6.3.4 Frequenzmodulation

Vom heutigen Stand der Technik betrachtet ist die AM allerdings überholt, weil die Qualitätsansprüche gestiegen sind und mit modernen Bauelementen die **Frequenzmodulation (FM)** erheblich einfacher, billiger und leistungssparender gebaut werden können. Außerdem haben die FM einen höheren Dynamikumfang des Informationssignals und ist weniger anfällig gegenüber Störungen. Eine Frequenzmodulation setzt voraus, dass sich die Frequenz ständig ändert. Dieses lässt sich nicht mehr durch einen Term der Form $\omega_c t = 2\pi f_c t$ ausdrücken, sondern wir müssen den Begriff der momentanen Kreisfrequenz einführen:

$$\omega(t) = \frac{d}{dt} p(t)$$

wobei die $p(t)$ eine sich verändernde Phase ist. Die **momentane Frequenz** ist also ganz allgemein die zeitliche Ableitung der Phasenfunktion (vgl. in diesem Zusammenhang die Beziehung zwischen Geschwindigkeit und Ort und die Analogie

zwischen Phase und Ort bzw. Frequenz und Geschwindigkeit). Eine Frequenzmodulation fordert nun, dass sich die momentane Frequenz verändert:

$$\omega(t) = \omega_c + M_f m(t)$$

wobei M_f wieder die Modulationstärke ist und $m(t)$ die modulierende Funktion, bzw. *message*.

Für die Berechnung der Kurvenform zu jedem Zeitpunkt benötigen wir jedoch nicht die momentane Frequenz, sondern die Phasenfunktion. Wenn die Frequenz die Ableitung der Phase ist, so ist umgekehrt die Phase das Integral der Frequenz:

$$p(t) = \int \omega(t) dt$$

Setzen wir nun die Gleichung für momentane Frequenz ein, erhalten wir folglich:

$$p(t) = \int \omega_c + M_f m(t) dt = \omega_c t + p_0 + M_f \int m(t) dt$$

Damit erhält man für die Frequenzmodulation folgenden Ausdruck:

$$y_{\text{FM}}(t) = A_c \cdot \sin \left(2\pi f_c t + p_0 + M_f \int m(t) dt \right)$$

Die Frequenzänderung ist durch die Amplitude des Nachrichtensignals gegeben. Angenommen $m(t)$ ist die einheitssnormierte Nachricht und f_{dev} ist die maximal mögliche Abweichung von der Trägerfrequenz f_c .

6.3.5 Vergleich zwischen PM und FM

Wir setzen als Beispiel das oben eingeführte sinusförmige Nachrichtensignal, $m(t) = A_m \cdot \sin(2\pi f_m t)$, mit $A_m = 1$ und $A_c = 1$ in die PM und FM Formeln ein:

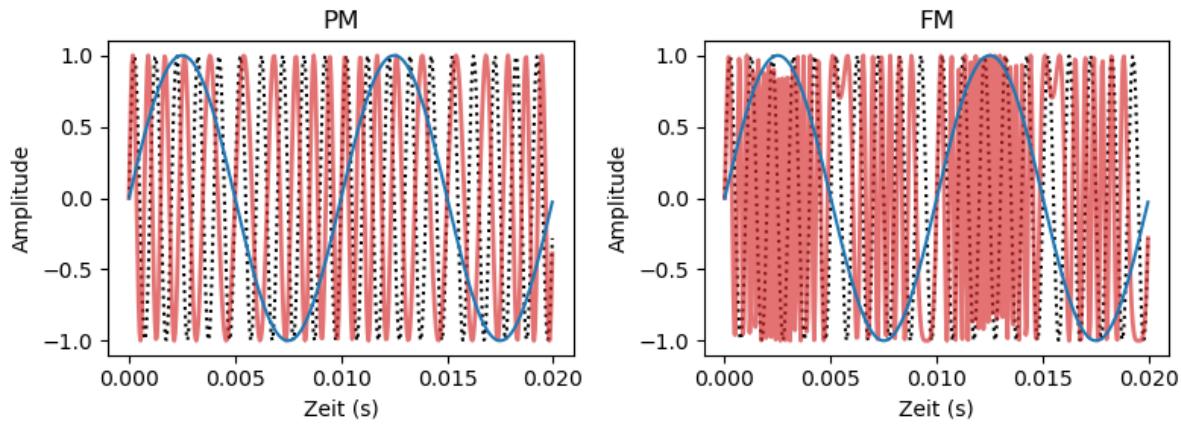
$$y_{\text{PM}}(t) = \sin \left(2\pi f_c t + p_0 + M_p \sin[2\pi f_m t] \right)$$

$$y_{\text{FM}}(t) = \sin \left(2\pi f_c t + p_0 - \frac{M_f}{2\pi f_m} \cos[2\pi f_m t] \right)$$

Folgende Eigenschaften können beobachtet werden:

- Der Phasenhub für die Phasenmodulation ist immer noch M_p
- Der Phasenschub für die Frequenzmodulation ist $\frac{M_f}{\omega_m}$.
- Die momentane Frequenz für die Phasenmodulation ist $\omega_c + M_p \omega_m \cos(\omega_m t + p_m)$
- Der Phasenschub für die Frequenzmodulation ist $\omega_c + M_f \sin(\omega_m t + p_m)$

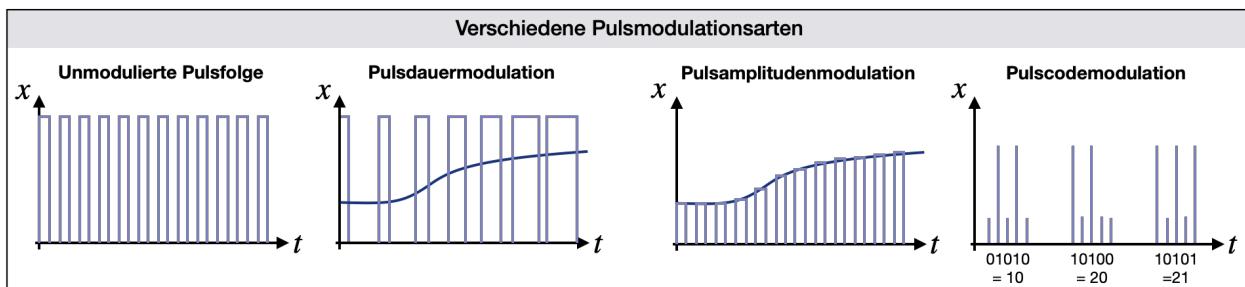
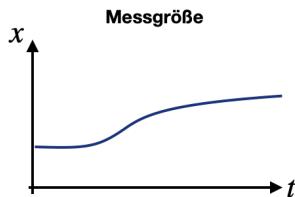
In beiden Fällen findet eine Modulation der Phase statt. Allerdings wirkt bei der Frequenzmodulation nicht der Modulator direkt auf die Phase ein, sondern es ist erst das Integral des Modulators zu rechnen. Das Integral hat eine Tiefpasswirkung. Der Phasenhub wird also bei der Frequenzmodulation mit zunehmender Frequenz des Modulators geringer. Umgekehrt wird der Frequenzhub bei der Phasenmodulation mit niedriger werdender Modulatorfrequenz immer geringer.



6.3.6 Puls-Modulation

Häufig werden auch Rechteckpulsfolgen als Trägerschwingung verwendet. Ähnlich wie bei der Sinusschwingung können auch hier verschiedene Modulationen vorgenommen werden, die größtenteils selbsterklärend sind (wie Pulsamplitudenmodulation oder Pulsdauermodulation). Nur die Pulscode-Modulation ist etwas anders und besonders. Hierbei wird das Messsignal kodiert und besteht aus einer Folge von Pulsgruppen. Diese Aneinanderreihung von Pulsgruppen entspricht einem bestimmten Alphabet. Vorteil dieser Variante ist, dass nur noch zwei Zustände, nämlich 0 und 1, in einer bestimmten Reihenfolge verarbeitet werden müssen. D.h. diese Art von Signalen ist direkt im Computer weiterverarbeitbar, ohne vorangegangen Analog-Digital-Wandlung.

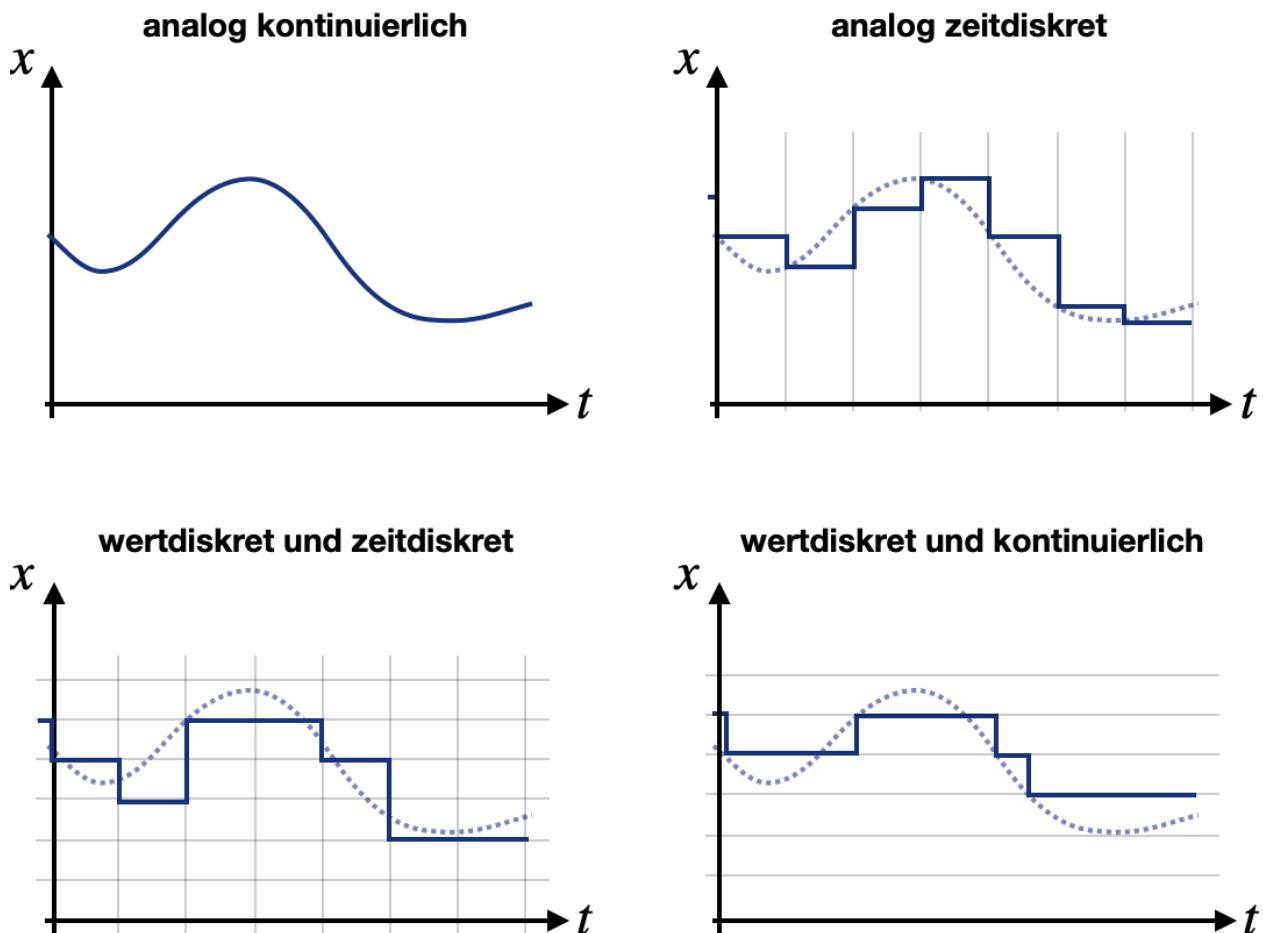
Auch hier gilt natürlich, dass eine entsprechende Demodulation nachgeschaltet werden muss, um das eigentliche Signal zurück zu gewinnen.



6.3.7 Klassifizierung von Messsignalen

Für eine ausreichende Charakterisierung von Messsignalen werden diese noch nach dem Wertevorrat der Informationsparameter (analog oder diskret) und dem zeitlichen Auftreten (kontinuierlich oder diskontinuierlich/diskret) unterschieden. Oft kommt dazu noch eine weitere Signalart, nämlich *stochastische Signale*, auch bekannt als *Rauschen*, bei welchem die Werte zufallsbehaftet sind. In diesem Fall können erst bei einer großen Anzahl von Ereignissen gesicherte Aussagen über die Qualität einer Messung geliefert werden (damit haben wir uns ja schon innerhalb der Messunsicherheiten und der Statistik befasst). Determinierte Signale liefern hingegen zu jedem Zeitpunkt einen festen Zusammenhang zwischen zu analysierenden und den erhaltenen Messwerten. Allgemein finden wir in technischen Messsystemen eigentlich immer deterministische Signale. Das heißt es gilt Determiniertheit.

Das folgende Bild zeigt, wie der Wertevorrat und deren zeitliche Verfügbarkeit aussehen könnte. Hierbei können kontinuierliche und diskrete sowohl im Wertebereich, also auch im Zeitbereich, kombiniert werden.



Folgende Klassifizierungen von Messsignalen existieren:

- **analog:** Der Wertebereich (also die Werte auf der y-Achse) ist kontinuierlich. Theoretisch kann der Informationsparameter y jeden beliebigen Wert innerhalb seines Wertebereichs annehmen. Dadurch erhält man eine proportional Abbildung zwischen Messsignal und Parameter. Allerdings ist dieses System sehr störanfällig und reagiert auf Störsignale, Rauschen oder Temperaturdrifts.
- **diskret (wertdiskret):** Diskret kann man sowohl der Werte- als auch den Zeitbereich bezeichnen. Für die Eindeutigkeit können wir den Begriff wertdiskret benutzen. Hierbei kann der Informationsparameter nur endlich viele Werte annehmen und wird nicht mehr vollständig proportional abgebildet. Dadurch ist das System zwar etwas weniger empfindlich gegenüber Störeinflüssen (erst nach Überschreiten von Grenzwerten wird der nächste *diskrete*

Messwert angezeigt), doch bei der Abbildung analoger Messwerte auf einen diskreten Informationsparameter gehen Informationen verloren.

- **kontinuierlich:** Dies ist das Analogon auf der Zeitachse zum *analogen* Wertebereich. Kontinuierliche Messwerte bedeutet, dass die Informationparameter zu jedem beliebigen *Zeitpunkt* seinen Wert ändern kann. D.h. der zeitliche Verlauf ist immer verfolgbar, jedoch können auch hier Störungen einwirken, und zwar jederzeit, die die Messung beeinflussen. Außerdem wird die Datenmenge oft unnötig groß.
- **diskontinuierlich (zeitdiskret):** Nun befassen wir uns mit diskreten Messwerten im Zeitbereich. Dies nennt zur Unterscheidung auch diskontinuierlich. Hierbei stehen Informationsparameter nur zu bestimmten Zeitpunkten zur Verfügung, wodurch Störungen, die zwischen zwei Zeitpunkten wirken, keine Rolle mehr spielen. Nachteil ist jedoch, dass die Informationen nur noch zu bestimmten Zeitpunkten zur Verfügung stehen.
- **stochastisch:** Jetzt steckt der Informationsparameter in einer stochastischen Größe, also im Rauschen. Da Störungen häufig eine charakteristische Verteilung haben, sind diese, wenn überhaupt, nur stark reduziert bemerkbar. Sie werden quasi über die Messzeit hinweg mitgemessen, integrieren sich auf, und können weggemittelt werden (Mittelwert ist häufig bei Null). Nachteil dieses Verfahrens ist, dass die Information erst nach mehrmaligen Messungen zur Verfügung steht, was einen großen Zeitbereich benötigen kann.
- **determiniert:** Im Unterschied zu stochastischen Signalen kann hier die Information mit nur einer einmaligen Messung gewonnen werden. Nachteil ist, dass diese einmalige Messung durch Störungen unbrauchbar gemacht werden können.

Bei *digitalen* Signalen sind die diskreten Werte einem bestimmten Alphabet zugeordnet. Im *binären* System kann der Informationsparameter dann nur noch zwei Werte annehmen.

Beispiele zum obigen Bild sind:

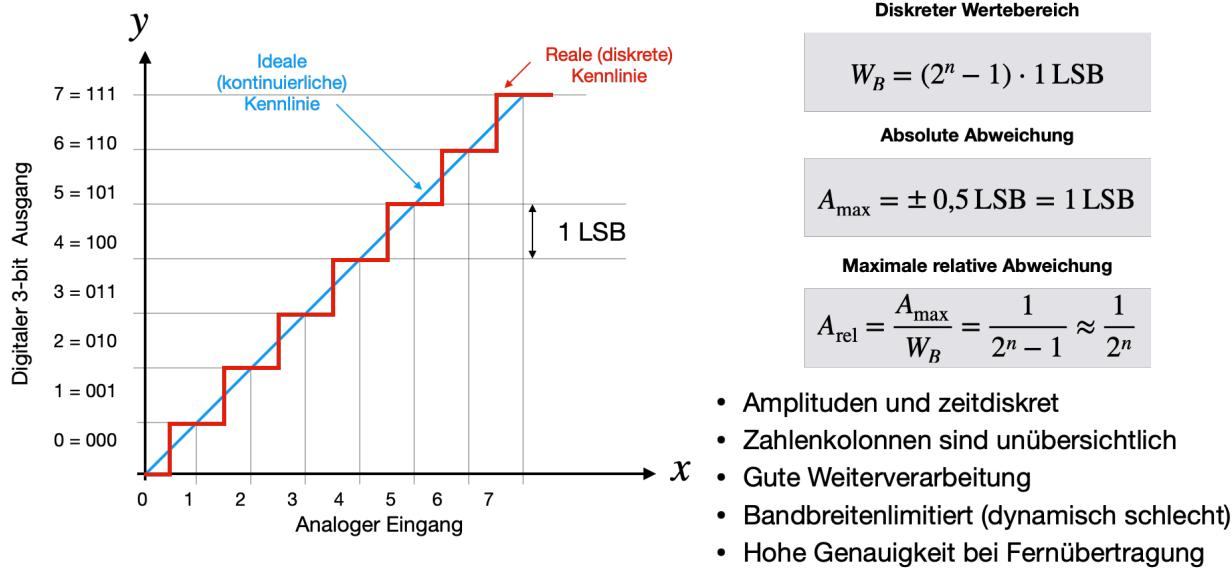
- **analog kontinuierlich:** Manometer am Druckkessel, Aufgesetzte Messuhr (Wärmeausdehnung), Analoge Spannungs- oder Strommessung
- **analog zeitdiskret:** Abgetastetes System. Informationen gehen verloren.
- **wertdiskret und kontinuierlich:** Lagerbestand (Es können nur ganze Bauelemente dem Lager entnommen werden, sodass der Lagerbestand wertdiskret ist. Es ist aber zu jedem Zeitpunkt bekannt, wie viele Bauelemente eines bestimmten Typs vorhanden sind. Das Signal ist zeitkontinuierlich.)
- **wertdiskret und zeitdiskret:** Digitalisierung

Nur bei analogen kontinuierlichen Signalen sind informationsverlustfreie Signalübertragungen möglich.

6.4 Digitalisierung

6.4.1 Kennlinie

In Folge der Digitalisierung wird der unendliche Wertevorrat einer analogen Größen, z.B. eine elektrischen Spannung, auf einen endlichen Wertevorrat abgebildet. Dadurch erhält man eine stufenförmige Kennlinien. Infolge dieser Quantisierung kann man sich vorstellen, dass Informationen verloren gegangen werden. Ein beliebig genauer analoger Spannungswert, mit unendlich vielen Ziffern, kann nur mit einem endlichen Speichervorrat im Computer verarbeitet und dargestellt werden. Erst wenn ein Grenzwert überschritten wird, wird die nächste *Stufe* erreicht und der Computer zeigt einen neuen Spannungswert an. Nicht aber die inkrementalen Zwischenschritte.



Die gestufte (Treppen-)Kurve im dargestellten Bild ist die reale Übertragungskurve eines 3-Bit-Analog-Digital-Wandlers (ADW). Das digitale Ausgangssignal ist so lange konstant bei einem festen Wert, wie sich das analoge Eingangssignal innerhalb eines Inkrement, also 1 LSB (least significant bit), verändert. An diesem Beispiel hier stellen wir Eingangssignale zwischen 0V und 7V mit einem 3-Bit-Wandler dar. Variiert die Eingangsspannung nur minimal, also beispielsweise zwischen 1,1V und 1,9V, so würde sich der digitale Ausgangswert tatsächlich nicht ändern und konstant 001 ausgeben.

Je feiner die Stufen (also je mehr Bits), desto besser wird die Auflösung. In der Praxis wird man die Stufung und damit die Auflösung eines ADW nur so fein wie für die zu lösende (Mess-) Aufgabe notwendig wählen, da der technische Aufwand sonst enorm steigt und manchmal gar nicht notwendig ist, da die Auflösung auch von der Genauigkeit der Referenzinformation limitiert sein könnte (i. Allg. der Referenzspannung zur Darstellung des LSB-Intervalls).

Der diskrete Wertebereich kann wie folgt angegeben werden:

$$W_B = (2^n - 1) \cdot 1 \text{ LSB}$$

6.4.2 Quantisierungsabweichung

Aufgrund des begrenzten (diskreten) Wertebereichs ergibt sich eine absolute Messabweichung infolge der Quantisierung, die oft als absoluter Quantisierungsfehler bezeichnet wird. Diese bildet sich aus der Differenz zwischen dem digitalen Ist-Wert und dem idealisierten Kennenlinienwert (linear verlaufende ideale Kennlinie, im Bild blau gekennzeichnet).

Die maximale Abweichung beträgt hier 1LSB, wodurch die relative Messabweichung, bezogen auf den Wertebereich, wie folgt abgeschätzt werden kann:

$$A_{\text{rel}} = \frac{A_{\max}}{W_B} = \frac{1 \text{ LSB}}{(2^n - 1) \cdot 1 \text{ LSB}} \approx \frac{1}{2^n}$$

Beispiel: Für einen A-D-Wandler mit einer Wortbreite von 10 Bit ist die relative Abweichung infolge der Quantisierung anzugeben:

$$A_{\text{rel}} = \frac{1}{2^{10}} = 0,00097 = 0,001 = 0,1\%$$

6.4.3 Nyquist-Shannon-Abtasttheorem (Aliasing)

Im zeitlichen Verlauf weist der ADW ebenfalls eine Diskretisierung auf. Das heißt, dass nicht nur der Wertebereich eingeschränkt wird, sondern auch die Zeitpunkte, zu welchen Informationen abgerufen werden können. Dem kontinuierlichen Eingangssignal wird also nur eine endliche Zahl von Proben (Samples) entnommen. Jeder ADW benötigt für die Wandlung eines analogen Signals in ein digitales eine bestimmte Zeit und reagiert relativ spät. Diese Zeit ist endlich und kann nicht beliebig klein werden. Aus dem kontinuierlichen, analogen Signal wird also nur eine Schnittmenge von Messwerten zu bestimmten Zeitpunkten entnommen. Was zwischen zwei Zeitpunkten (Samples) mit dem analogen Signal passiert, wird nicht mitaufgezeichnet. Diese Zeitdiskretisierung muss allerdings nicht zwangsläufig mit einem Informationsverlust einhergehen.

Unter bestimmten Voraussetzungen, die im Abtasttheorem festgelegt sind, kann man die Signalwerte zwischen den festen Abtastzeitpunkten wieder vollständig rekonstruieren. Das Shannon'sche Abtasttheorem gibt an, wie oft eine Sinusschwingung mit Frequenz f abgetastet werden muss, damit sie aus dem digitalisierten Signal extrahiert werden kann:

$$f_{ab} > 2f$$

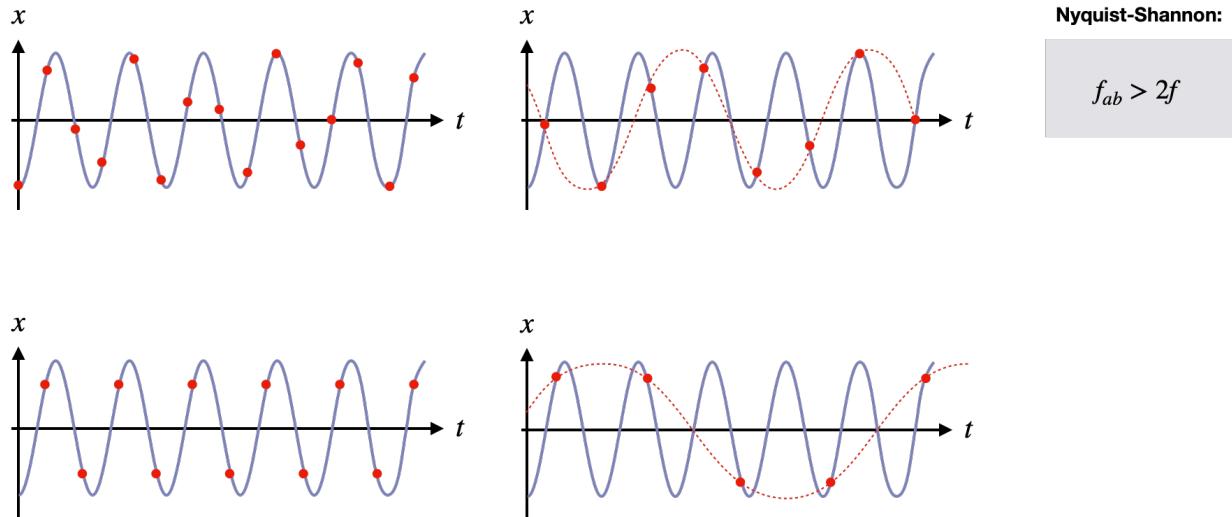
Eine Schwingung muss innerhalb einer Periode mindestens 2-mal abgetastet werden. Nur wenn diese Bedingung erfüllt ist, entstehen durch die Abtastung keine Signalfehler. Diese Bedingung ist ebenfalls auf beliebige periodische Signale anwendbar. Hierbei betrachtet man die höchste Frequenz, die noch in dem Signal vorkommt, und wählt entsprechend die *Nyquist-Frequenz*.

Bestimmte Signale, wie z.B. Rechteckpulse oder Einzelpulse, haben eine unendlich hohe Flanke. Mathematisch werden wir im nächsten Kapitel dieser Vorlesung noch sehen, dass das bedeutet, dass deren Frequenzanteile bis ins Unendliche gehen. Das heißt eine Rechteckspannung hat unendliche hohe Frequenzen. Diese können durch Digitalisierung *nicht* mehr berücksichtigt werden, da keine Abtastfrequenzen von $2 \cdot \infty$, also Zeitintervallen von 0 Sekunden, erreichbar sind. In diesem Falle findet ein Informationsverlust statt.

Andersherum kann man analysieren, welche Art von Signalen mit bestimmten ADWs noch analysiert werden können. Hat mein Messsystem eine Abtastfrequenz (auch Sampling-Frequenz oder Bandbreite genannt) von $f_{ab} = 100\text{ MHz}$, so können nur Signale mit Frequenzanteilen bis zu $f = 50\text{ MHz}$ verlustfrei analysiert werden.

Außerdem sollte man als Messtechniker:in stets vermeiden, dass höhere Frequenzen als diese Grenzfrequenz in den ADW gelangen. Dadurch können nämlich hässliche Effekte entstehen, die Mehrdeutigkeiten des gewonnenen Ausgangssignals zulassen. Diesen Effekt nennt man auch **Aliasing**. Angenommen man speist hohe Frequenzen in einen ADW ein, für die der ADW nicht mehr ausgelegt ist. Dies hat zu Folge, dass die Kurve nicht mehr mit mehr als 2 Punkten innerhalb einer Periode abgetastet wird (Nyquist-Shannon ist nicht erfüllt). Dadurch kann der ursprüngliche Kurvenverlauf nicht reproduziert werden. Man erhält irgendeinen anderen Kurvenverlauf (siehe Bild), der keinerlei Rückschlüsse auf den wahren Verlauf aufweist. Um diese Effekte zu vermeiden, werden Filter, so genannten *Anti-Aliasing-Filter* verwendet. Diese sind im Prinzip nur sehr steilflankige Tiefpass-Filter, welche Signalanteile bei hohen Frequenzen ($> f_{ab}/2$) rausfiltern und unterdrücken. Das Signal sollte hierbei allerdings möglichst nicht in seinem Frequenzverlauf beeinflusst oder gestört werden, weshalb man bei dem Filterdesign sehr sorgfältig vorgehen muss.

Im Bild sieht man das Beispiel für 2,6 Abtastungen pro Periode (oben links), 1,4 Abtastungen (oben rechts), 2 Abtastungen (unten links) und 0,8 Abtastungen (unten rechts). Abtastungspunkte, die 2 pro Periode unterschreiten, zeigen eine falsch-rekonstruierte Wellenform auf, was durch die rot gestrichelte Linie verdeutlicht werden soll.



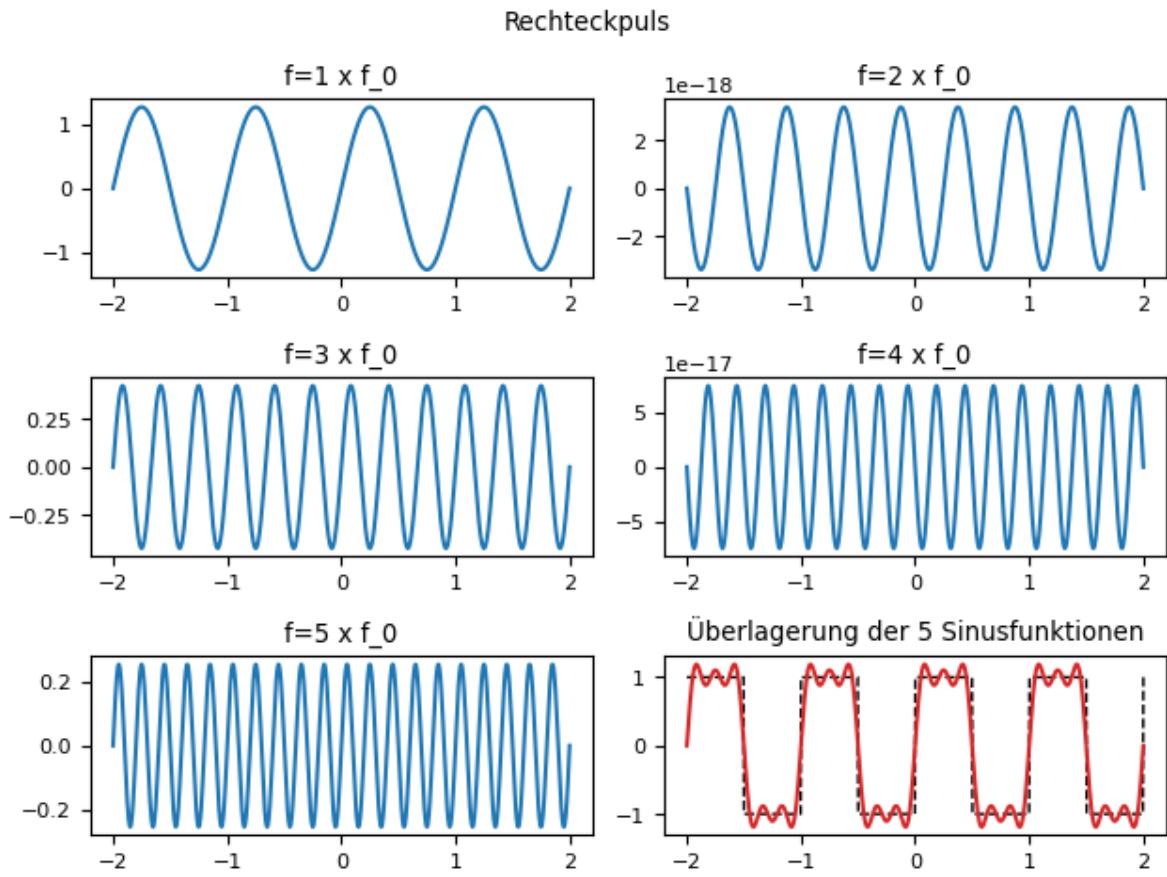
Digitale Methoden haben den großen Vorteil, dass die Genaugkeit erhalten bleibt, auch wenn die Daten über große Entfernungen übertragen werden und digitale Messwerte sind einfacher weiterverarbeitbar. Die Güte der Übertragung von elektrischen Spannungen oder allgemeinen Analogsignalen ist hingegen stark von äußeren Einwirkungen abhängig, wie Drift- und Rauscheigenschaften von elektronischen Komponenten. Allerdings sind analoge Messwertausgaben häufig angenehmer und übersichtlicher, als digitale Zahlenkolonnen und analoge Verfahren arbeiten kontinuierlich, simultan und haben daher auch häufig bessere dynamische Eigenschaften, wenn man sich zeitlich veränderliche Größen ansieht.

6.5 Fourier-Analyse

6.5.1 Fourierreihen

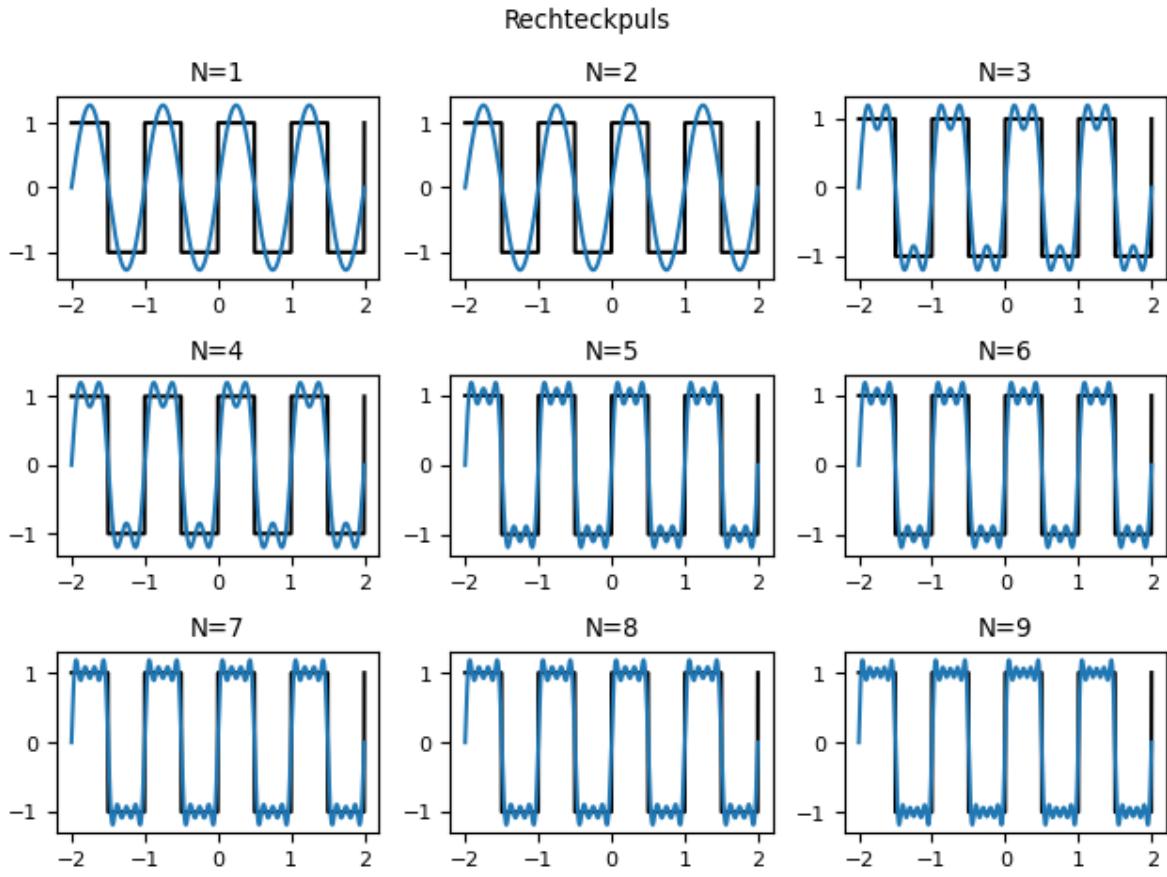
Jedes periodische Signal kann als *Summe von Sinus- und Cosinusfunktionen* mit Frequenzen von ganzzahligen Vielfachen der Grundfrequenz des Signals beschrieben werden. Dies ist die sogenannten **Fourierreihe**, Fourierreihen-Entwicklung/oder -Zerlegung.

<Figure size 1000x500 with 0 Axes>



Im folgenden Bild seht ihr einen Rechteckpuls und die zugehörige Fourierreihe unter Einbeziehung verschiedener Anzahl von Sinus- bzw. Cosinusfunktionen. Je mehr Sinusfunktionen bei ganzzahligen Vielfachen der Grundfrequenz des Rechteckpulses miteinbezogen werden, desto genauer können Rechtecksignale rekonstruiert werden.

```
<Figure size 1000x500 with 0 Axes>
```



Reelle Fourierreihe

Die **reelle Darstellung der Fourierreihe** für eine Funktion $x(t)$ sieht wie folgt aus:

$$x(t) = x_0 + \sum_{k=1}^{\infty} a_k \cos(2\pi k f_0 t) + \sum_{k=1}^{\infty} b_k \sin(2\pi k f_0 t)$$

x_0 ist hierbei der Gleichanteil (Mittelwert) des Signals, der sich wieder über den arithmetischen Mittelwert berechnet:

$$x_0 = \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt = \frac{a_0}{2}$$

Die (reellen) Koeffizienten a_k und b_k nehmen für jedes Messsignal eine anderen Wert an und berechnen sich über:

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos(2\pi k f_0 t) dt$$

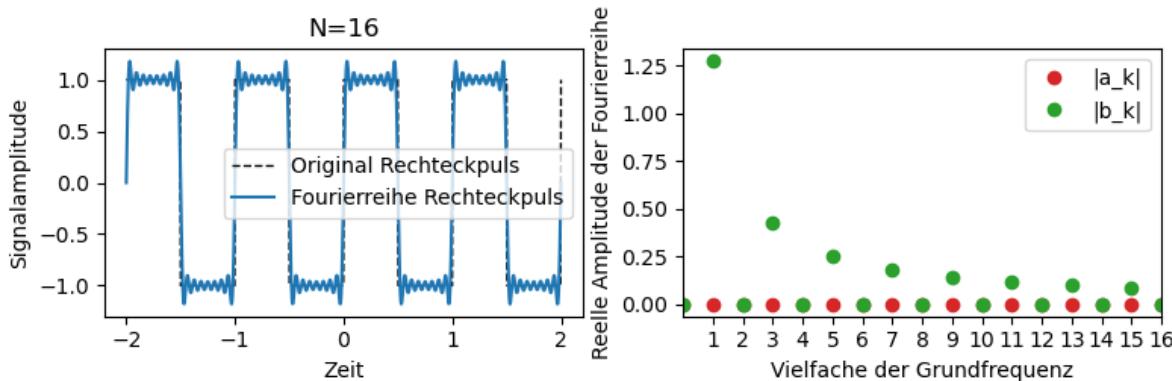
und

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin(2\pi k f_0 t) dt$$

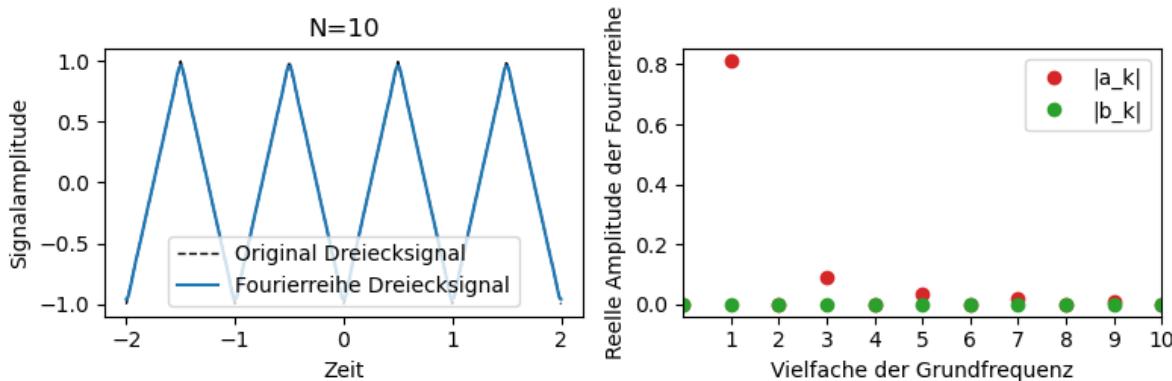
Jedes Integral muss immer über eine Periode ausgeführt werden. Ob hier die Grenzen $\pm T/2$ gewählt werden, oder von 0 bis T integriert wird, ist jedem selber überlassen.

Die Koeffizienten geben die Amplitude der Sinus- bzw. Cosinusfunktionen an, aus denen ein periodisches Signal rekonstruiert wird. Die Koeffizienten werden für die einzelnen Frequenzen berechnet und über die Vielzahl der Grundfrequenz in den folgenden Plots dargestellt. Um eine Rechteckfunktion in einer Fourierreihe zu entwickeln, werden viele höhere Harmonische benötigt. Bei einem Dreiecksignal sieht das anders aus.

Reelle Fourierkoeffizienten für einen Rechteckpuls



Reelle Fourierkoeffizienten für ein Dreicksignal



Aufgabe

Woran könnte es liegen, dass das Rechtecksignal höhere Harmonische benötigt als das Dreiecksignal? Im Python-Code könnt ihr die Flanke des Dreiecksignals ändern indem ihr in $f = \lambda t: \text{signal.sawtooth}(2 * \text{np.pi} * 1/T * t, 0.5)$ den letzten Parameter, die 0.5, entfernt. Was ändert sich jetzt?

Es kann übrigens folgendes gezeigt werden, was für die Praxis oft sehr hilfreich ist, da es die Anzahl von Integralberechnungen reduziert:

- für **gerade** Funktionen, also wenn $x(t) = x(-t)$ gilt, dann sind alle $b_k = 0$ (es existieren nur noch Cosinus-Terme)
- für **ungerade** Funktionen, also wenn $x(t) = -x(-t)$ gilt, dann sind alle $a_k = 0$ (es existieren nur noch Sinus-Terme)
- einen Gleichanteil x_0 kann es folglich bei ungeraden Funktionen *nicht* geben.

Das ist in den obigen Darstellungen bereits zu sehen. Für den Rechteckpuls sind die $a_k = 0$, während für das Dreiecksignal die $b_k = 0$ sind.

Komplexe Fourierreihe

Eine alternative Schreibweise ist die **komplexe Darstellung**. Hierbei wird eine periodische Funktion als eine Überlagerung von komplexen Exponentialfunktionen (anstelle von Sinus- und Cosinusfunktionen) dargestellt:

$$x(t) = \sum_{k=-\infty}^{\infty} \underline{c}_k e^{j2\pi k f_0 t}$$

Diese liefert den Vorteil, dass nur eine Art von Koeffizienten berechnet werden muss:

$$\underline{c}_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j2\pi k f_0 t} dt$$

Trotz der Rechnung mit komplexen Funktionen handelt es sich immer noch um eine reelle Funktion. Für $k = 0$ erhält man wieder den Gleichanteil:

$$\underline{c}_0 = x_0$$

Außerdem sieht man, dass die Werte für \underline{c}_{-k} und \underline{c}_k zueinander komplex konjugiert sind:

$$\underline{c}_{-k} = \underline{c}_k^*$$

Umrechnung zwischen reellen und komplexen Fourier-Koeffizienten

Mittels der Euler-Formel

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$$

lassen sich die Koeffizienten aus reeller Fourierreihen-Entwicklung und komplexer Darstellung ineinander umformen. Durch die Addition eines zueinander komplex konjugierten Koeffizientenpaares lässt sich der reelle Koeffizient a_k bestimmen:

$$a_k = \underline{c}_k + \underline{c}_{-k}$$

und analog fällt bei der Subtraktion der Realteil weg, sodass nach zusätzliche Multiplikation mit $j b_k$ berechnet wird:

$$b_k = j(\underline{c}_k - \underline{c}_{-k})$$

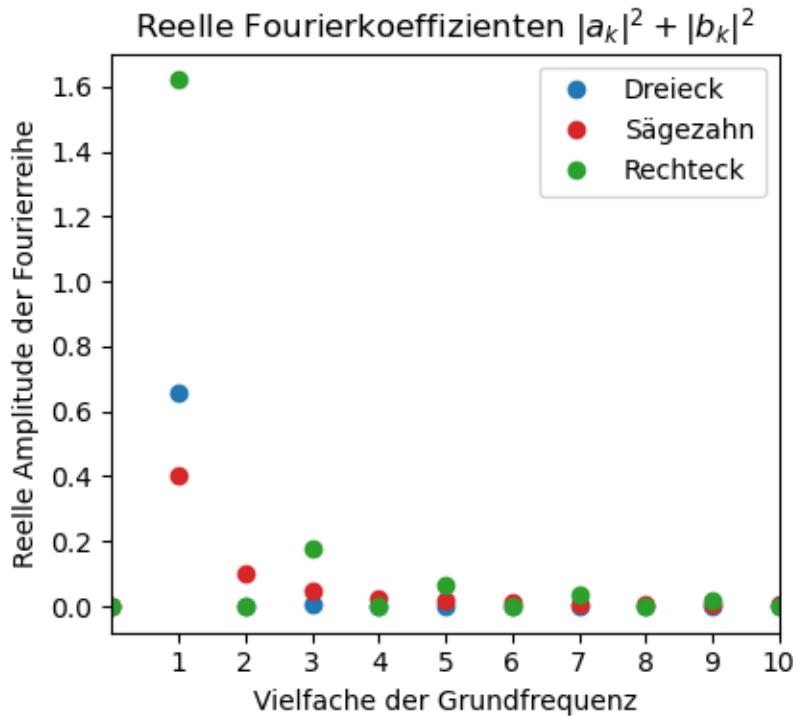
Andersherum können aus den reellen Koeffizienten auch die komplexen Koeffizienten berechnet werden:

$$\underline{c}_k = \frac{1}{2}(a_k - jb_k)$$

$$\underline{c}_{-k} = \frac{1}{2}(a_k + jb_k) = \underline{c}_k^*$$

An dieser Stelle wollen wir noch mal festhalten, dass die Koeffizienten der Fourierreihe eine Schwingung oder ein Messsignal im Frequenzbereich eindeutig beschreibt. In Ihrer Gesamtheit stellen diese Koeffizienten das **Spektrum** des Signals dar. Dies ist zumindest wahr für die hier dargestellte mathematische Betrachtung mittels Fourier-Transformation. Ein Spektrumanalysator wertet hingegen bei der jeder Einzelmessung in einem begrenzten Bereich Frequenzbereich das Signal aus, was häufig noch durch einen Bandpassfilter geschleust wurde. Dabei gehen Informationen über die Phasenlage verloren.

<Figure size 800x300 with 0 Axes>



6.5.2 Fourier-Transformation

Die Fourier-Transformation ist Teil der Spektralanalyse in der Messtechnik. Sie basiert auf der Grundidee, dass, wie wir eben gesehen haben, sich jede periodische Funktion aus Sinus- und Cosinusfunktionen schreiben lässt. Das Ziel ist es, die Anteile dieser Schwingungen sichtbar zu machen. Die Fourier-Transformation ist eine mathematische Methode mit der nun auch aperiodische Signale in ein kontinuierliches Spektrum zerlegt werden. Die Fourier-Transformation ist ein Werkzeug, mit dem man ein Signal (siehe [Messsignale](#)) nehmen und die Leistung jeder einzelnen Frequenz darin sehen kann.

Anwendung

Die Fourier-Transformation ist in vielen Anwendungen nützlich.

Ein Spektralanalyse, wie sie die Fouriertransformation durchführt, eignet sich besonders gut zur Zustandsüberwachung. Hier können Motoren, Turbinen, Sägen, Kugellager uvm, im Prinzip alles was rotiert, überwacht werden. Die spezifischen Frequenz jedes Kugellagers kann beispielsweise überwacht werden. Sollte sich die Amplitude über die Zeit verändert, kann dies ein Indiz dafür sein, dass eine Kugel ins Lager gefallen ist oder das Lager einen Schaden bekommen hat. Verschlechtert sich das Verhalten kann frühzeitig gegengewirkt werden, indem das Kugellager ausgetauscht wird. Das heißt auch Fehlerfrüherkennung, Fehlerdiagnose und Trendanalysen (“predictive maintenance”) werden häufig im Frequenzraum durchgeführt.

Shazam und andere Musikerkennungsdienste verwenden beispielsweise die Fourier-Transformation, um Lieder zu erkennen. Bei der JPEG-Komprimierung wird eine Variante der Fourier-Transformation verwendet, um die hochfrequenten Komponenten von Bildern zu entfernen. Bei der Spracherkennung werden die Fourier-Transformation und verwandte Transformationen verwendet, um die gesprochenen Wörter aus dem Audiomaterial wiederherzustellen.

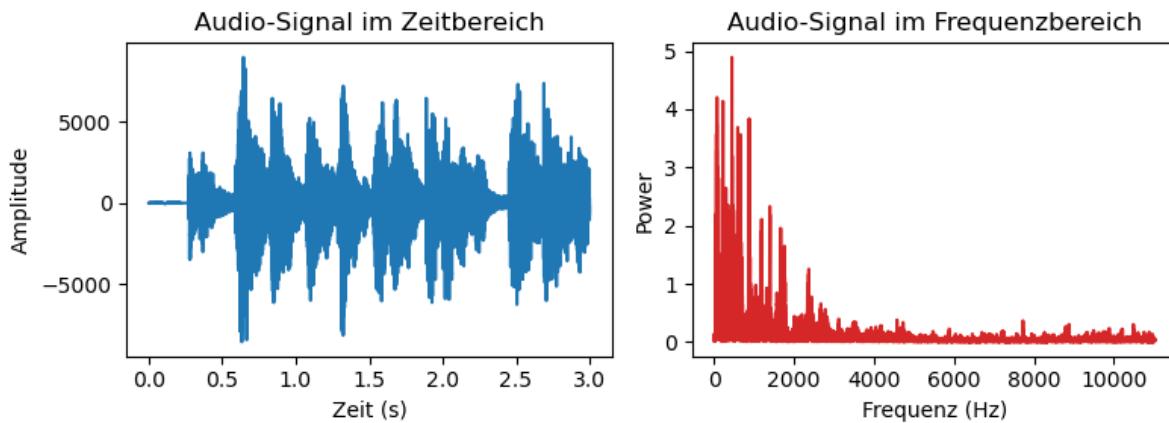
Im Allgemeinen benötigst du die Fourier-Transformation, wenn du die Frequenzen in einem Signal betrachten musst. Wenn die Arbeit mit einem Signal im Zeitbereich schwierig ist, lohnt es sich, die Fourier-Transformation zu verwenden, um es in den Frequenzbereich zu übertragen. Im nächsten Abschnitt werden Sie die Unterschiede zwischen dem Zeit- und dem Frequenzbereich kennen lernen.

Zeit- vs. Frequenzbereich

Du wirst im Folgenden immer wieder auf die Begriffe Zeitbereich und Frequenzbereich stoßen. Diese beiden Begriffe beziehen sich auf zwei verschiedene Arten der Betrachtung eines Signals, entweder als seine Frequenzkomponenten oder als Information, die sich über die Zeit verändert (siehe [Messsignale](#)).

Im Zeitbereich ist ein Signal eine Welle, deren Amplitude (y-Achse) über die Zeit (x-Achse) variiert. Sie sind wahrscheinlich daran gewöhnt, Diagramme im Zeitbereich zu sehen, wie z. B. dieses hier:

```
<IPython.lib.display.Audio object>
```



Dies ist ein Bild eines Audiosignals, also eines Signals im Zeitbereich. Die horizontale Achse steht für die Zeit, die vertikale Achse für die Amplitude.

Im Frequenzbereich wird ein Signal als eine Reihe von Frequenzen (x-Achse) dargestellt, denen jeweils eine Leistung (y-Achse) zugeordnet ist. Das Bild rechts zeigt das linke Audiosignal nach der Fourier-Transformation

Hier wird das Audiosignal von vorher durch seine einzelnen Frequenzen dargestellt. Jeder Frequenz entlang der Unterseite ist eine Leistung zugeordnet, wodurch das Spektrum entsteht.

Typen von Fourier-Transformationen

Die Fourier-Transformation kann in verschiedene Arten von Transformationen unterteilt werden. Die grundlegendste Unterteilung basiert auf der Art der Daten, mit denen die Transformation arbeitet: kontinuierliche Funktionen oder diskrete Funktionen. Die Begriffe DFT und FFT werden oft synonym verwendet. Sie sind jedoch nicht ganz dasselbe. Die **fast (kontinuierliche) Fourier-Transformation (FFT)** ist ein Algorithmus zur Berechnung der **diskreten Fourier-Transformation (DFT)**, während die DFT die Transformation selbst ist.

Die **diskrete Fourier-Transformation (DFT)** (z.B. auf digitalisierte, abgetastete Messwerte angewendet) entspricht der Fourierreihen:

$$X_d(k\Delta f) = \sum_{i=0}^{N-1} x(i\Delta t) e^{-j2\pi k\Delta f i \Delta t}$$

wobei $\Delta f = 1/T$ mit der Periode $T = N \cdot \Delta T$, N ist die Anzahl der Samples.

Die **kontinuierliche Fourier-Transformation** ist für beliebige Funktionen $f(t)$ definiert, d.h. die Periode kann unendlich lang werden und die Funktion kann aperiodisches Verhalten aufweisen:

$$\mathcal{F}(x(t)) = X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

Die Rücktransformation ist wie folgt definiert:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega$$

Der Vollständigkeitshalber soll an dieser Stelle auch noch die **Laplace-Transformation** erwähnt werden, die sich wie folgt berechnen lässt:

$$\mathcal{L}(x(t)) = X(s) = \int_0^{\infty} x(t) e^{-st} dt$$

mit der Rücktransformation

$$x(t) = \frac{1}{2\pi} \int_0^{\infty} X(s) e^{st} ds$$

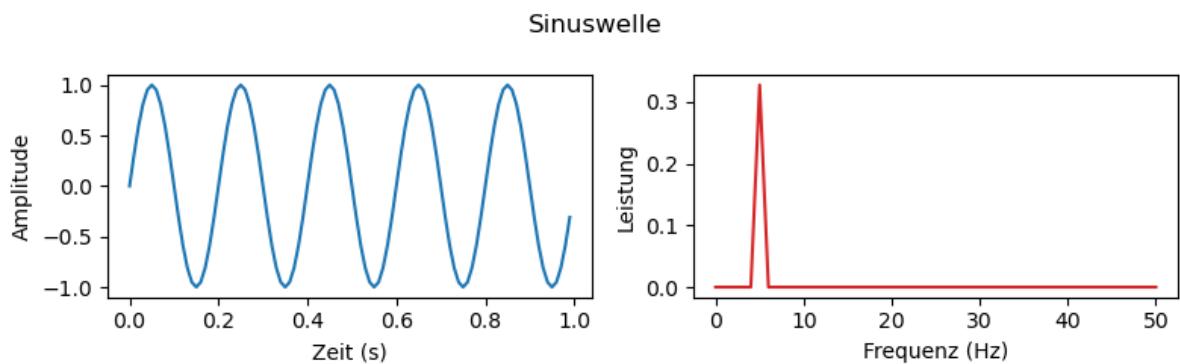
Hierbei ist $s = \sigma + j\omega$ eine komplexe Zahl (anstelle von ω) und wird für dynamische Messsysteme wichtig werden.

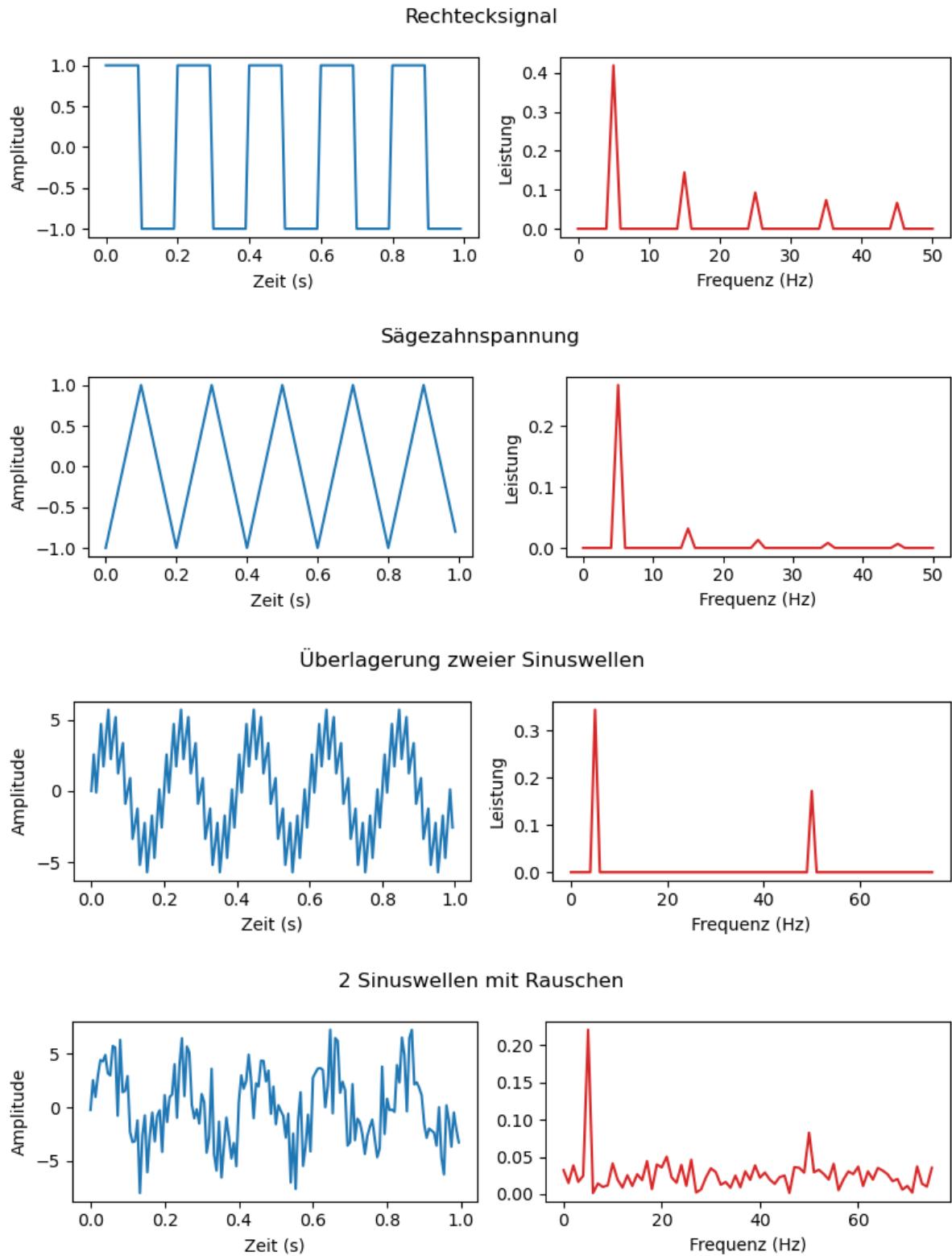
Eigenschaften

Jeder Fourier-Transformation hat folgende wichtige **Eigenschaften**, die das Leben und Rechnen im Frequenzraum erheblich erleichtern können:

- **Linearität:** $\mathcal{F}(ax_1 + bx_2) = a\mathcal{F}(x_1) + b\mathcal{F}(x_2)$
- **Ableitung:** $\mathcal{F}(\dot{x}) = j\omega \cdot \mathcal{F}(x)$
- **Faltung:** $\mathcal{F}(x_1 * x_2) = \mathcal{F}(x_1) \cdot \mathcal{F}(x_2)$
 - Faltung im Zeichbereich ist zum Vergleich sehr kompliziert: $(x_1 * x_2)(t) = \int_{-\infty}^{\infty} x_1(\tau) x_2(t - \tau) d\tau$
- **Zeitverschiebung:** $\mathcal{F}(x(t - \tau)) = \mathcal{F}(x(t)) \cdot e^{-j\omega\tau}$

6.5.3 Beispiele von FFTs und Fourier-Reihen-Koeffizienten





6.6 Rauschen und spektrale Dichte

DYNAMISCHE MESSSYSTEME

Einleitung

- *Linear zeitinvariante (LZI) Systeme*
- *Differentialgleichung*
- *Impuls- und Sprungantwort*
- *Übertragungsfunktion*
- *Faltung und Korrelation*

7.1 Linear zeitinvariante (LZI) Systeme

LTI Systeme sind wichtig, da sie zwar sehr vereinfacht sind, aber gut zu lösen sind. Sie sind in der Tat die einzigen Systeme, die wir lösen können (mittels Differentialgleichungen).

Für ein LTI System können Ausgangssignale vorhergesagt werden, die man für ein bestimmtes Eingangssignal erwarten würde. Das gleiche gilt auch für *Veränderungen* von Eingangssignalen.

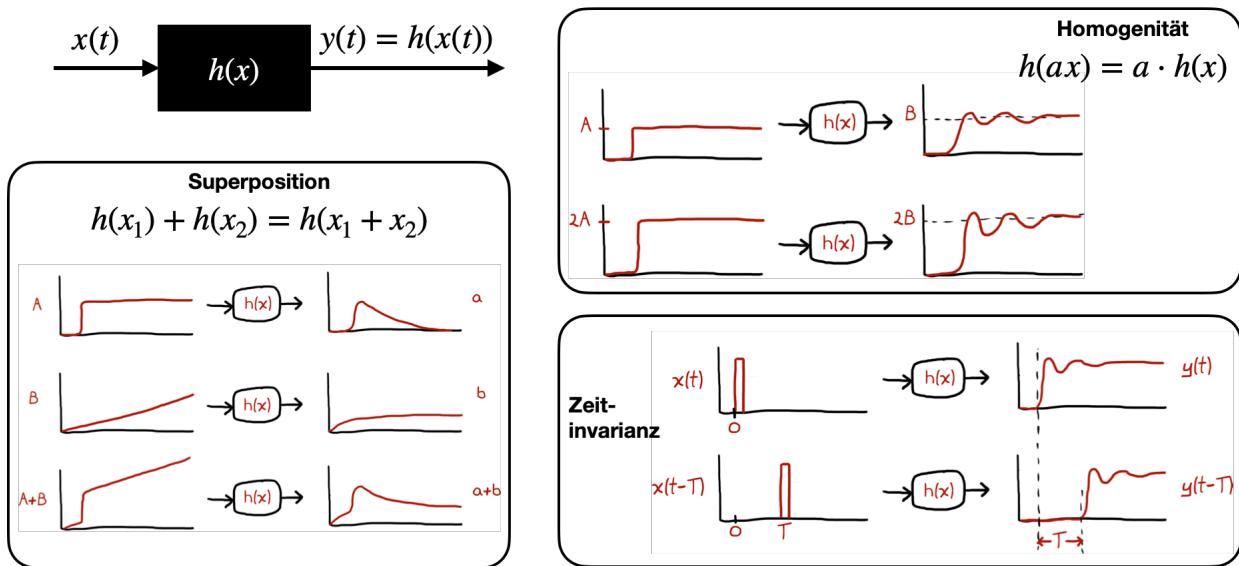
7.1.1 Eigenschaften

Das liegt daran, dass LTIs bestimmte Eigenschaften haben: *Superposition, Homogenität und Zeitinvarianz*. Im Folgenden sei h ein Operator, der eine lineare Abbildung zwischen Eingangsvektor $x(t)$ und dem Ausgang eines Systems, Vektor $y(t)$, darstellt. Für ein **lineares System** gelten allgemein folgende Eigenschaften:

- **Homogenität:** Wenn $x(t)$ mit Faktor a skaliert wird, dann wird $y(t)$ ebenso skaliert
 - Ein Sprung der Größe A produziert eine Schwingung mit Größe B
 - Da $h(x)$ linear ist, wird ein verdoppelter Sprung am Eingang, also $2A$ zu einer Verdopplung am Ausgang, $2B$, führen
- **Superposition** bedeutet Additivität: Addiere zwei Eingangssignale, A und B zusammen, sprich $A + B$, dann ist die Antwort des linearen Systems auch die Summe der individuellen Ausgänge (a und b), also $a + b$.
- **Zeitinvariant:** Das System verhält sich immer gleich, egal wann in der Zeit es durchgeführt wird.
 - Wird das Eingangssignal $x(t)$ um T verschoben, dann ist auch das Ausgangssignal um T verschoben.

In Realität gibt es fast immer Abweichungen von diesen Idealvorstellungen. D.h. wir können mit LTIs Eingänge eines Signals skalieren, verschieben und summieren, aber sie entsprechen niemals einem realen System. Warum sind sie dann so wichtig? Richard Feynman sagte damals: „Linear Systems are important, because we can solve them“. Selbst wenn

LTI nicht real ist, so können wir reale Probleme sehr gut *annähern* indem wir LTI Analysen durchführen. Das heißt, stell immer sicher, dass dein System für einen bestimmten Bereich linear funktioniert.



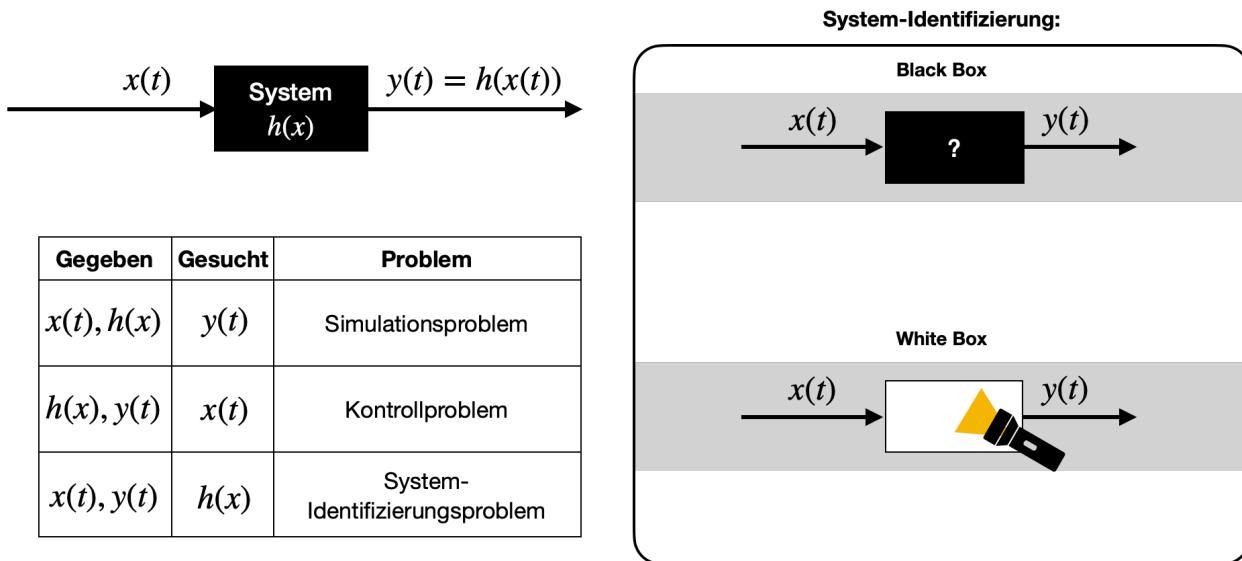
7.1.2 System-Identifizierung

Jedes Messsystem hat eine spezifische Übertragungsfunktion $h(t)$, welche, angewendet auf ein Eingangssignal $x(t)$, ein Ausgangssignal $y(t)$ liefert. Im Allgemeinen kann man drei Probleme in der Praxis erkennen, jenachdem welche Informationen einem vorliegen:

1. $x(t)$ und $h(t)$ sind gegeben $\rightarrow y(t)$ ist gesucht: Dies ist das **Simulationsproblem**
 - Anhand von Simulationen kann der Ausgang eines Systems vorhergesagt werden.
2. $h(t)$ und $y(t)$ sind gegeben $\rightarrow x(t)$ ist gesucht: Dies ist das **Kontrollproblem**
 - Dieses Problem wird man in der Regelungstechnik häufig antreffen, wo eine Regelgröße am Eingang des Mess- bzw. Regelsystems gesucht wird um einen stabilen Zustand zu erreichen.
3. $u(t)$ und $y(t)$ gegeben $\rightarrow h(t)$ ist gesucht: Dies ist das **System-Identifizierungsproblem**
 - Hier wollen wir das Messsystem charakterisieren, was unter anderem durch 2 Methoden möglich ist: Mittels Messungen der Eingangs- und Ausgangsgröße, oder mittels Aufstellen von Differentialgleichungen (Systemtheorie).

Während Problem 1 und 2 Bestandteil anderer Vorlesungen oder Themengebiete ist, wollen wir uns im Folgenden mit dem Problem 3 befassen. Hier wollen wir im Folgenden zwei Möglichkeiten betrachten, mittels welcher $h(t)$ bestimmt werden kann. Die beiden Methoden werden im folgenden mit **black box** und **white box** bezeichnet.

In allen Kapiteln wird der Inhalt am Beispiel eines Tiefpasses 1. Ordnung noch einmal konkreter erläutert.



7.2 Differentialgleichung

Mit dem *white box*-Verfahren lösen wir das System-Identifizierungsproblem indem wir genau analysieren, aus welchen Komponenten unser System besteht. Anhand eines Tiefpasses 1. Ordnung wollen wir uns die Vorgehensweise einmal ansehen.

7.2.1 Aufstellen der Differentialgleichung

In diesem Falle gucken wir in unser System rein. Wir sehen die elektronische Schaltung bestehend aus einer Reihenschaltung von Widerstand mit Wert R und Kondensator mit Kapazität C . Am Eingang liegt die Spannung $u_e(t)$ an. Über C kann die Ausgangsspannung $u_a(t)$ gemessen werden. Der Strom, mit dem der Kondensator aufgeladen ist bekanntlich

$$i_C(t) = C \cdot \frac{du_a}{dt}$$

wobei $u_a(t)$ im allgemeinen die Spannung ist, die am Kondensator anfällt und in diesem Falle gleich der Ausgangsspannung ist.

Außerdem gilt $i_R(t) = i_C(t)$. Damit können wir die Spannung, die über den Widerstand R abfällt, wie folgt schreiben:

$$u_R(t) = R \cdot i_R(t) = R \cdot C \cdot \frac{du_a}{dt}$$

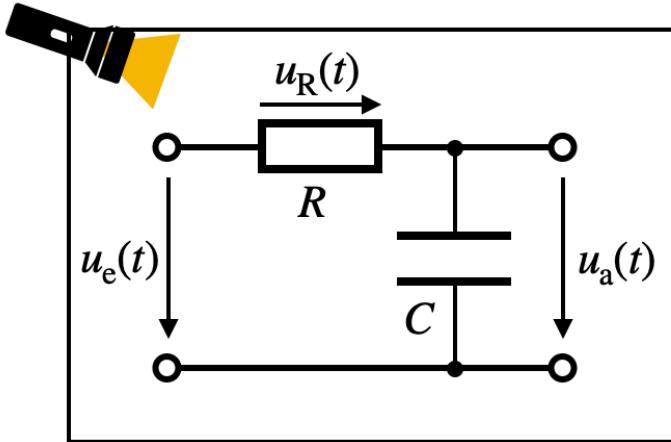
Aus der Maschenregel, hier $u_a(t) + u_R(t) = u_e(t)$ folgt nach Einsetzen von $u_R(t)$ folgendes:

$$RC \frac{du_a(t)}{dt} + u_a(t) = u_e(t)$$

Dies ist eine **Differentialgleichung (DGL) 1. Ordnung**. Die Differentialgleichung erkennt man daran, dass auch eine zeitliche Ableitung der Ausgangsgröße (also ein Differential) in der Formel auftritt. Da allerdings nur die 1. zeitliche Ableitung auftritt, handelt es sich um eine DGL 1. Ordnung. Würde Ausgangs oder Eingangsgröße zweimal zeitliche abgeleitet werden, würde es sich um eine DGL 2. Ordnung handeln usw.

Systeme 1. Ordnung beschreiben *Energiespeicher*, also z.B. Temperaturmesser, die warm werden, oder auch Operationsverstärker, Kapazitäten und Elektromagneten. Im Prinzip ist jedes System, was sich erwärmt, ein Energiespeicher

und wird somit ein System 1. Ordnung sein. Systeme 2. Ordnung haben einen zusätzlichen Dämpfungsterm. In diesen Systemen findet auch eine Umlagerung von Energien statt, wie es z.B. in einem LC-Schwingkreis passiert. Kondensator und Spule tauschen ständig Energien aus, wodurch es zu einer periodischen Schwingung kommt. Dieses System wird durch eine DGL 2. Ordnung beschrieben.



Tiefpass DGL 1. Ordnung:

$$RC \frac{du_a(t)}{dt} + u_a(t) = u_e(t)$$

7.2.2 Lösen der Differentialgleichung

Ziel ist es jetzt, eine Funktion für Eingangs- und Ausgangsspannung zu finden, die diese Gleichung erfüllt. Hierfür geht man im Allgemeinen in 5 Schritten vor. Wie werden später in diesem Kapitel noch sehen, dass DGL im Frequenz- bzw. Laplace Raum viel einfacher gelöst werden können als wir es hier, im Zeitraum, jetzt machen.

1. Homogener Ansatz: Unter homogenen Ansatz verstehen wir das Eingangssignal auf Null zu setzen: $u_e(t) = 0$. Das heißt das System wird nicht belastet. Die DGL vereinfacht sich zu

$$RC \frac{du_a(t)}{dt} + u_a(t) = 0$$

und kann umgeformt werden zu

$$RC \frac{du_a(t)}{dt} = -u_a(t)$$

Das bedeutet, wird suchen eine Funktion für $u_a(t)$ die nach der Zeit abgeleitet wieder sie selber ist. Eine Exponentialfunktion erfüllt genau diese Bedingung und wir nutzen sie für unseren homogenen Ansatz:

$$u_{a,\text{homogen}}(t) = K \cdot e^{-\gamma t} \quad \Rightarrow \quad \dot{u}_{a,\text{homogen}}(t) = -\gamma K \cdot e^{-\gamma t}$$

mit den Konstanten K und γ . Dies wird in die homogene DGL eingesetzt und es folgt:

$$-RCK\gamma e^{-\gamma t} + Ke^{-\gamma t} = 0 \quad \Rightarrow \quad -RC\gamma + 1 = 0$$

Daraus folgt für die erste Konstante

$$\gamma = \frac{1}{RC} := \frac{1}{\tau}$$

2. Spezieller Ansatz für Anfangsbedingung: Jetzt legen wir statt Null einen Sprung an den Eingang $u_e(t)$ an: $u_e(t) = u_0$ für $t = 0$. Für lange Zeiten wird sich der Kondensator komplett aufgeladen haben, sodass der Ausgang ein konstantes Signal liefert. Das heißt unsere Lösung für ein spezielles Eingangssignal ist

$$u_{a,\text{speziell}}(t) = u_0 \quad \text{für } t \rightarrow \infty$$

3. Allgemeiner Ansatz: Nun kann der allgemeine Ansatz formuliert werden. Diese ist einfach die Addition von homogener und spezieller Ansatz:

$$u_a(t) = u_{a,\text{homogen}}(t) + u_{a,\text{speziell}}(t) = u_0 + K \cdot e^{-t/\tau}$$

4. Konstante bestimmen: Eine Konstante war schon bestimmt, nämlich $\gamma = \frac{1}{RC} := \frac{1}{\tau}$. Diese Zeitkonstante wurde bereits in den allgemeinen Ansatz unter 3. eingefügt. Um K zu bestimmen, setzen wir unsere Anfangsbedingung von Schritt 2. ein: Zum Zeitpunkt $t = 0$ ist das System zwar am Eingang mit u_0 belastet, der Ausgang ist jedoch noch Null, solange der Kondensator nicht vollständig aufgeladen ist. Wir setzen also $t = 0$, $u_e(t = 0) = u_0$ und $u_a(t = 0) = 0$ in den allgemein Ansatz ein:

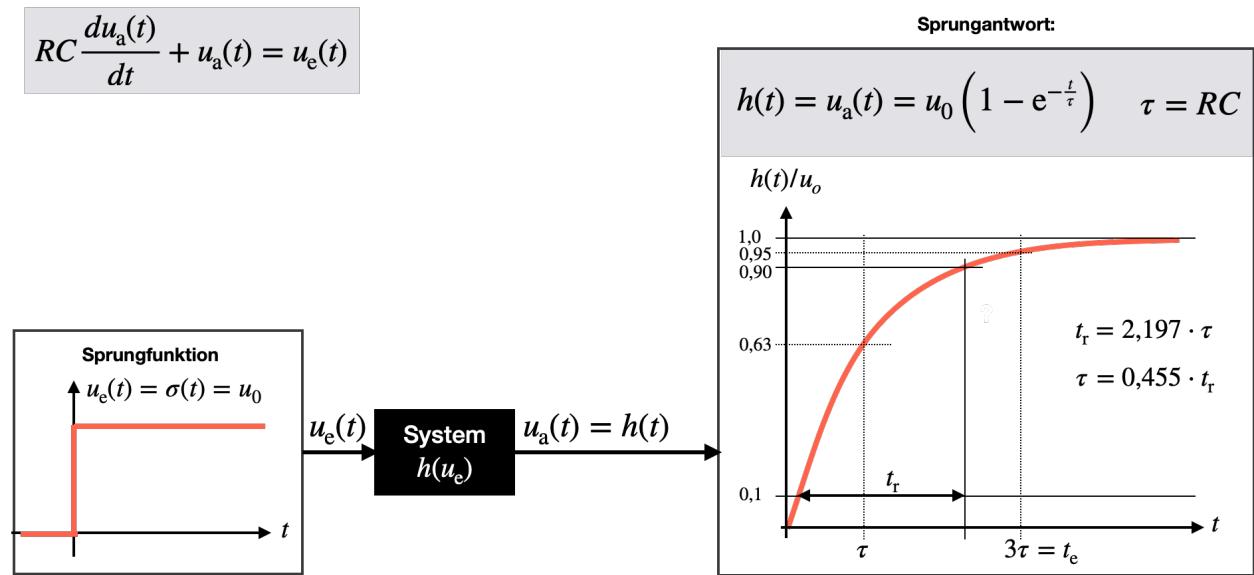
$$u_a(0) = u_0 + K \cdot e^{-0/\tau}$$

$$0 = u_0 + K \cdot 1 \Rightarrow K = -u_0$$

5. Lösung hinschreiben: Für die Lösung setzen wir alle unsere bestimmten Konstanten in den allgemeinen Ansatz ein und erhalten:

$$u_a(t) = u_0 \left(1 - e^{-\frac{t}{RC}} \right)$$

Dies ist auch die sogenannte **Sprungantwort** eines Systems 1. Ordnung, da wir in Schritt 2. einen *Sprung* angelegt haben. Hätten wir uns als Eingangssignal einen Impuls (eine Delta-Funktion) ausgesucht, hätten wir die **Impulsantwort** des Systems bestimmt.

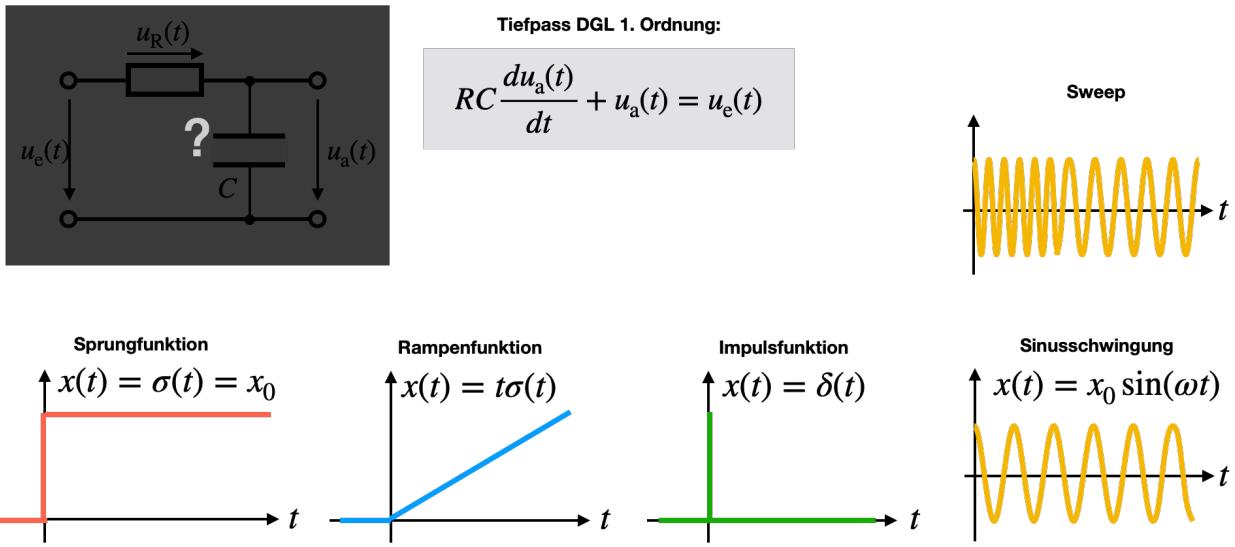


7.3 Impuls- und Sprungantwort

Bei dem Prinzip der black box kenne ich das Innenleben meines Messsystems nicht, und ich kann lediglich mittels Messungen darauf zurück schließen. Je nachdem, welches Signal an den Eingang angelegt wird, erhalte ich ein andere Ausgangssignal. Dies ist in den beiden Bildern aus dem vorausgegangenen Kapitel deutlich zu erkennen. In einem Fall wurde ein Sprung angelegt, im anderen Fall ein Impuls.

In der Messtechnik ist es hingegen häufig viel aussagekräftiger das Übertragungsverhalten einer Messeinrichtung mittels Testfunktionen zu überprüfen. Sprung und Impuls gehören zu des Testfunktionen, doch auch eine Rampenfunktion (ein

sich kontinuierlich erhöhendes Eingangssignal) oder ein Sweep (hier werden verschiedene Frequenzen direkt nacheinander durchgefahren) haben sich bewährt.

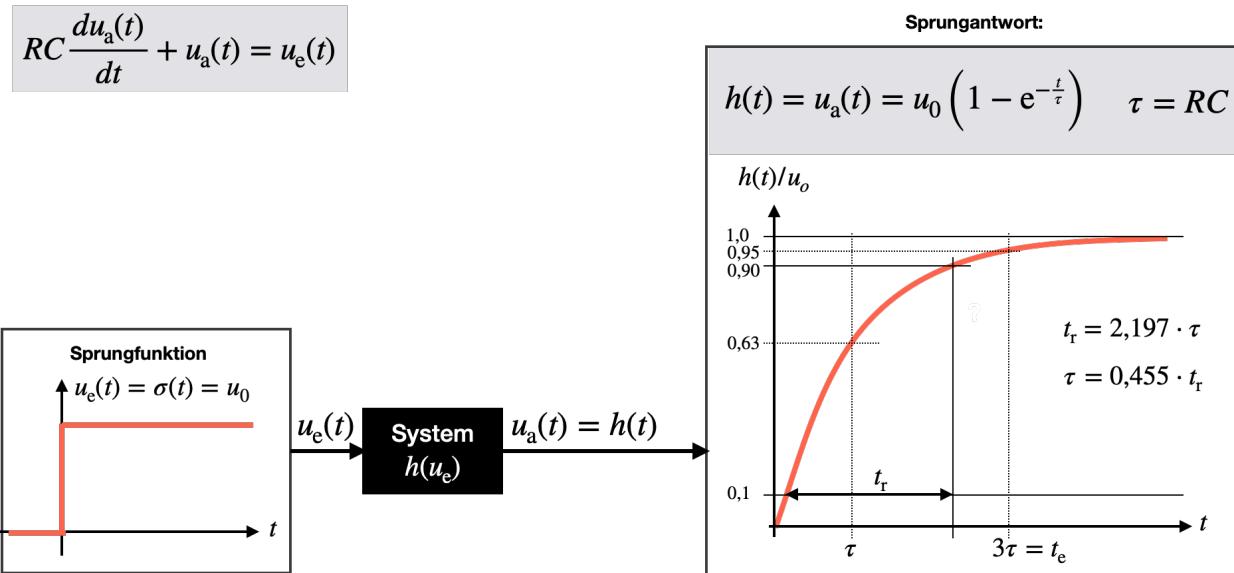


Das Vorgehen ist wie folgt: Es wird eine Testfunktion angelegt und gemessen. Der Ausgang des Systems wird abgegriffen und ebenfalls gemessen. Ein Vergleich aus angelegtem Eingangssignal und gemessener Systemantwort (Ausgangssignal) können Rückschlüsse für das System getroffen werden.

7.3.1 Interpretation der Sprungantwort

An dieser Stelle wollen wir uns ansehen, was man aus der Sprungantwort eines Tiefpasses 1. Ordnung im Labor ablesen kann.

- Geplottet im nachfolgenden Bild ist die normalisierte Sprungantwort, d.h. das Ausgangssignal wurde durch u_0 dividiert. Nach langer Einpendelzeit folgt der Ausgang dem Eingangssignal.
- τ ist die Zeitkonstante, die die *Trägheit* eines Systems bestimmt.
 - Aus der Theorie ist bekannt (siehe vorheriges Kapitel): $\tau = RC$
 - Nach der Zeit τ ist das Ausgangssignal auf 63% seines maximal möglichen Wertes angestiegen. Es ist noch kein stationärer Zustand erreicht.
- Einstellzeit t_e : Dies ist das *95%-Kriterium*:
 - Für ein System 1. Ordnung muss die Sprungantwort $h(t)$ nach der Zeit $t = 3\tau$ 95% des Endwertes erreicht haben.
 - Nur wenn dies der Fall ist, handelt es sich um ein System mit Ausgleich 1. Ordnung
 - Für ein System 2. Ordnung ist häufig ein Überschwingen oder Einpendeln zu beobachten. Dann ist t_e Zeit vom Anlegen der Sprungfunktion bis zum Erreichen des Toleranzbandes innerhalb von 0,95 und 1,05.
- Anstiegszeit t_r : Dies ist die Dauer für einen Signalanstieg von 10% auf 90% am Ausgang des Systems.



7.3.2 Impulsantwort und Faltung im Zeitraum

In diesem Abschnitt wollen wir uns der Genialität der Impulsantwort in Kombination mit der Faltung widmen. Hierfür müssen wir uns zuerst angucken, was *Faltung* überhaupt bedeutet.

Die **Faltung** ist eine mathematische Operation, welche zwischen zwei Funktionen f und g ausgeführt werden kann. Die Kurzschreibweise ist das Sternchen zwischen den beiden Funktionen. Allgemein berechnet wird sie über ein Integral, welches die beiden Funktionen beinhaltet, wobei g hierbei zeitlich gespiegelt und verschoben wird.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

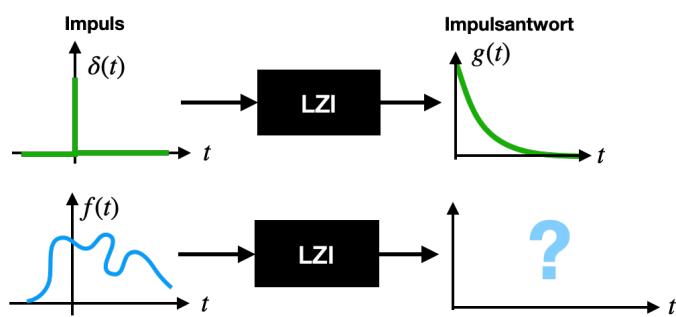
Die resultierende „Überlagerung“ zwischen f und gespiegelten und verschobenen Versionen von g (man spricht auch von einer „Verschmierung“ von f) kann z. B. verwendet werden, um einen gleitenden Durchschnitt zu bilden. Das Faltungsintegral kann in drei Schritten verstanden werden:

1. der Eingang g wird in der Zeit umgekehrt und zeitverschoben
2. Dieses umgekehrte und verschobene g wird nun an f multipliziert
3. Und dann wird das Produkt über alle Zeiten summiert (Integralbildung).

Für viele Funktionen wurde das Faltungsintegral bereits bestimmt, da es sehr aufwendig ist, dieses i.Allg. zu lösen. Das Integral zu lösen hilft außerdem kaum dabei genau zu verstehen, was das Integral, bzw. die Faltung, an sich überhaupt bedeutet. Die Faltung kann auch grafisch bestimmt werden, was wir uns in einer Übung einmal genauer ansehen werden.

Das interessante der Faltung ist jedoch, dass wir durch diese mathematische Operation das Systemverhalten oder Ausgangssignale vorhersagen können, solange die Impulsantwort bekannt ist. Man kann sich das vereinfacht so vorstellen, dass eine beliebige Eingangsfunktion $f(t)$ durch unendliche viele *Impulse* beschrieben werden, mit infinitesimal kleiner Breite dt . Die Impulsantwort ist für einen Impuls (Delta-Peak) bekannt: sie ist die zeitverschobene Antwort $g(t - dt)$. Die Überlagerung vieler einzelner Impulse, eine beliebige Eingangsfunktion $f(t)$ zu modellieren, ist einfach die Aufsummierung aller Impulse. In einem LTI System ist die Impulsantwort ebenfalls die Aufsummerierung der Impulsantworten zu den zugehörigen Impulsen (*Linearitätsbedingung*). Um den *realen* Werteverlauf der Eingangsfunktion sicher zu stellen, muss jeder Delta-Peak für jedes Zeitintervall mit dem Funktionswert skaliert werden. Die *Homogenitätsbedingung* stellt jetzt sicher, dass wir den Ausgang auch entsprechend skalieren dürfen. Mathematisch betrachtet resultiert dies direkt

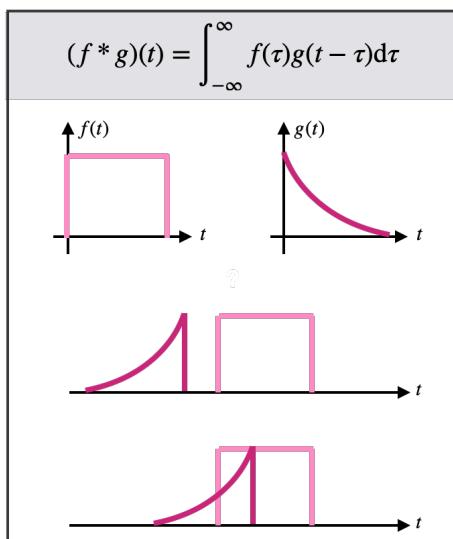
darin, dass wir Antworten von beliebigen Anregungsfunktionen hervorholen können, indem die Anregungsfunktion $f(t)$ mit der Impulsantwort gefaltet wird.



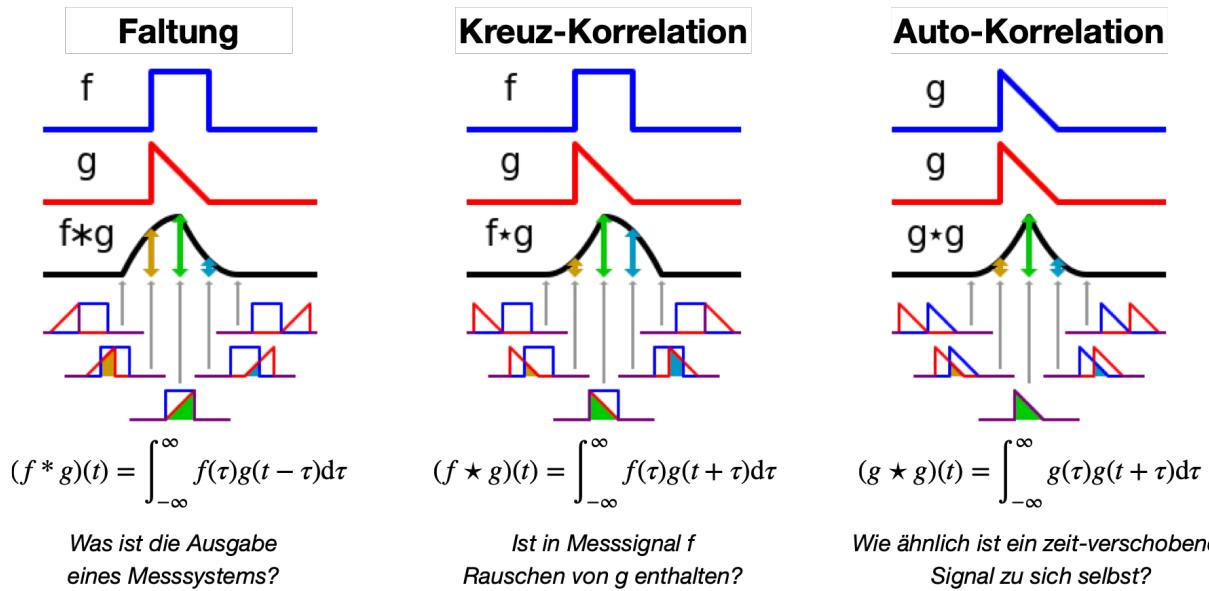
→ Antwort eines LZI mit beliebiger Anregungsfunktion:
= Faltung der Anregungsfunktion mit der Impulsantwort

$$\text{Diskrete Faltung: } (f * g)(t) = \sum_{i=0}^{\infty} f(i \cdot dt) \cdot dt \cdot g(t - idt)$$

Faltung zweier Funktionen f und g im Zeitbereich:



Die Faltung ist nicht zu verwechseln mit der Kross-Korrelation von zwei Messsignalen. Obwohl die Integrale sehr ähnlich aussehen, so besteht ein signifikanter Unterschied zwischen diesen beiden Methoden, der sich im Minuszeichen der Verzögerung aufzeigt. Grafisch bedeutet dies, dass das zu faltende Signal im Falle der Kreuz-Korrelation *nicht* gespiegelt wird, wohingegen es bei der grafischen Faltung zeitlich gespiegelt werden muss. Die *Faltung* berechnet man in der Regel, wenn man die Antwort eines Messsystems berechnen möchte. $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$ Die * Kreuz – Korrelation * berechnet man um zu untersuchen, ob Rauschanteile von Signalen auch in Signalen vorkommen (Stichwort ist hier der * Korrelationsbegriff*, welcher häufig auf zwei * Signale * und nicht Messsysteme angewendet wird). $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t + \tau)d\tau$ Die * Auto – Korrelation * soll hier die Vollständigkeit halber noch einmal als Spezialfall der Kreuz – Korrelation aufgeführt werden. Hier berechnet man, wie ähnlich ein zeit – verschobenes Signal sich selbst ist (*zeitliche Korrelation $g(t) = \int_{-\infty}^{\infty} g(\tau)g(t + \tau)d\tau$)



7.3.3 Interpretation im Frequenzraum

Als nächstes gucken wir uns dynamische Messsysteme im Frequenzraum an. Aus den letzten Vorlesungen sollte bekannt sein, dass periodische Zeitserien in Fourierreihen umgeschrieben werden können. Nicht-periodische Funktionen können mittels Fourier-Transformation in den Frequenzraum transformiert werden. In beiden Fällen erhält man Auskunft darüber, welche Frequenzanteile in dem Signal vorhanden sind. Die Frequenzanteile weisen eine Amplitude und eine Phase auf und können auch in einem Phasordiagram oder Amplitudendiagram eingezeichnet werden.

Nicht nur Signale, sondern natürlich auch das Messsystem selber, kann in den Frequenz- oder eher den Laplace-Raum, transformiert werden. Wie sich das Messsystem im Frequenzraum verhält, wird über die sogenannte **Übertragungsfunktion** definiert. Die Übertragungsfunktion eines Systems ist das Verhältnis von Ausgangs- zu Eingangssignal. Das Eingangssignal kann mittels Fourier- oder Laplace-Transformation transformiert werden. Zur Erinnerung schreiben wir hier noch mal die Integrale, die für die Umrechnung benutzt werden:

- Fourier-Transformation: $\mathcal{F}(x(t)) = X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$
- Laplace-Transformaton: $\mathcal{L}(x(t)) = X(s) = \int_0^{\infty} x(t)e^{-st} dt$

Wenn $x(t)$ das Eingangssignal im Zeit-Raum ist, dann bezeichnen wir mit $X(j\omega)$ das Eingangssignal im Frequenzraum. Hier ist j wieder die komplexe Zahl. Das Ausgangssignal $y(t)$ wird analog dazu mit $Y(j\omega)$ bezeichnet. Es hat sich eingebürgert, dass Zeitsignale mit kleinen Buchstaben, x , bezeichnet werden und Signale im Frequenzbereich mit großen Buchstaben, X .

Die Übertragungsfunktion, $G(j\omega)$, kann also wie folgt ausgedrückt werden:

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)}$$

7.4 Übertragungsfunktion

7.4.1 Interpretation im Frequenzraum

Als nächstes gucken wir uns dynamische Messsysteme im Frequenzraum an. Aus den letzten Vorlesungen sollte bekannt sein, dass periodische Zeitserien in Fourierreihen umgeschrieben werden können. Nicht-periodische Funktionen können mittels Fourier-Transformation in den Frequenzraum transformiert werden. In beiden Fällen erhält man Auskunft darüber, welche Frequenzanteile in dem Signal vorhanden sind. Die Frequenzanteile weisen eine Amplitude und eine Phase auf und können auch in einem Phasordiagram oder Amplitudendiagram eingezeichnet werden.

Nicht nur Signale, sondern natürlich auch das Messsystem selber, kann in den Frequenz- oder eher den Laplace-Raum, transformiert werden. Wie sich das Messsystem im Frequenzraum verhält, wird über die sogenannte **Übertragungsfunktion** definiert. Die Übertragungsfunktion eines Systems ist das Verhältnis von Ausgangs- zu Eingangssignal. Das Eingangssignal kann mittels Fourier- oder Laplace-Transformation transformiert werden. Zur Erinnerung schreiben wir hier noch mal die Integrale, die für die Umrechnung benutzt werden:

- Fourier-Transformation: $\mathcal{F}(x(t)) = X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$
- Laplace-Transformaton: $\mathcal{L}(x(t)) = X(s) = \int_0^{\infty} x(t)e^{-st} dt$

Wenn $x(t)$ das Eingangssignal im Zeit-Raum ist, dann bezeichnen wir mit $X(j\omega)$ das Eingangssignal im Frequenzraum. Hier ist j wieder die komplexe Zahl. Das Ausgangssignal $y(t)$ wird analog dazu mit $Y(j\omega)$ bezeichnet. Es hat sich eingebürgert, dass Zeitsignale mit kleinen Buchstaben, x , bezeichnet werden und Signale im Frequenzbereich mit großen Buchstaben, X .

Die Übertragungsfunktion, $G(j\omega)$, kann also wie folgt ausgedrückt werden:

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)}$$

7.4.2 Herleitung der Übertragungsfunktion

Anhand unseres Beispiels, dem Tiefpass 1. Ordnung, wollen wir die Übertragungsfunktion einmal herleiten. Dazu gibt es verschiedene Ansätze.

1. Methode: Eine Möglichkeit die Übertragungsfunktion zu bestimmen, ist es die DGL aus dem vorherigen Kapitel in den Frequenzraum zu transformieren. Die DGL des Tiefpasses 1. Ordnung war gegeben durch:

$$\tau \frac{du_a(t)}{dt} + u_a(t) = u_e(t)$$

- Transformiere die Signale in den Frequenzraum. Der letzte Punkt resultiert aus den Eigenschaften von Fourier-Transformationen für zeitliche Ableitungen von Zeitsignalen.
 - $u_e(t) \rightarrow U_e(j\omega)$
 - $u_a(t) \rightarrow U_a(j\omega)$
 - $\dot{u}_a(t) \rightarrow j\omega U_a(j\omega)$
- In der DGL werden die Zeitsignale durch die Fourier-Transformierten ersetzt:

$$\tau j\omega U_a(j\omega) + U_a(j\omega) = U_e(j\omega)$$

- Die DGL wird nach $U_a(j\omega)/U_e(j\omega)$ umgestellt, um die Übertragungsfunktion zu erhalten.

$$G(j\omega) = \frac{U_a(j\omega)}{U_e(j\omega)} = \frac{1}{1 + \tau j\omega}$$

2. Methode: Sollte die DGL (noch) nicht bekannt sein, kann die Übertragungsfunktion auch direkt über die komplexen Widerstände bestimmt werden. Bei dem Tiefpass 1. Ordnung handelt es sich um die Reihenschaltung von Widerstand und Kondensator.

- Die komplexe Ausgangsspannung wird über dem Kondensator abgegriffen, das heißt es gilt das ohm'sche Gesetz für komplexe Zahlen. $\underline{Z}_C = \frac{1}{j\omega C}$ ist die Impedanz des Kondensators mit Kapazität C und \underline{I} der Strom.

$$\underline{U}_a = \underline{Z}_C \cdot \underline{I}$$

- Die komplexe Eingangsspannung liegt an kompletten Messsystem, also der Reihenschaltung an, d.h. es gilt

$$\underline{U}_a = (R + \underline{Z}_C) \cdot \underline{I}$$

- Die Division der beiden Spannungen führt abermals zur gesuchten Übertragungsfunktion:

$$G(j\omega) = \frac{\underline{U}_a}{\underline{U}_e} = \frac{1/(j\omega C)}{R + 1/(j\omega C)} = \frac{1}{1 + RCj\omega}$$

- Die Zeitkonstante $\tau = RC$ könnte nun noch in die Gleichung eingesetzt werden.

7.4.3 Bode Diagramm

Nachdem wir nun die Übertragungsfunktion hergeleitet haben, wollen wir wissen, was wir aus dieser Funktion ableiten, bzw. von ihr lernen können. Zunächst einmal sehen wir, dass es sich um eine komplexe Zahl handelt. Wir für jede andere komplexe Zahl können wir also hier Amplitude und Phase bestimmen. Dazu formen wir G in die typische Schreibweise einer komplexen Zahl um, sodass Real- (Re) und Imaginärteil (Im) direkt abgelesen werden können. Hierfür erweitern wir G typischer mit dem komplex Konjugierten:

$$G(j\omega) = \frac{1}{1 + RCj\omega} = \frac{1}{1 + RCj\omega} \cdot \frac{1 - RCj\omega}{1 - RCj\omega} = \frac{1 - RCj\omega}{1 - (RC\omega)^2} = \frac{1}{1 - (RC\omega)^2} - j \frac{RC\omega}{1 - (RC\omega)^2}$$

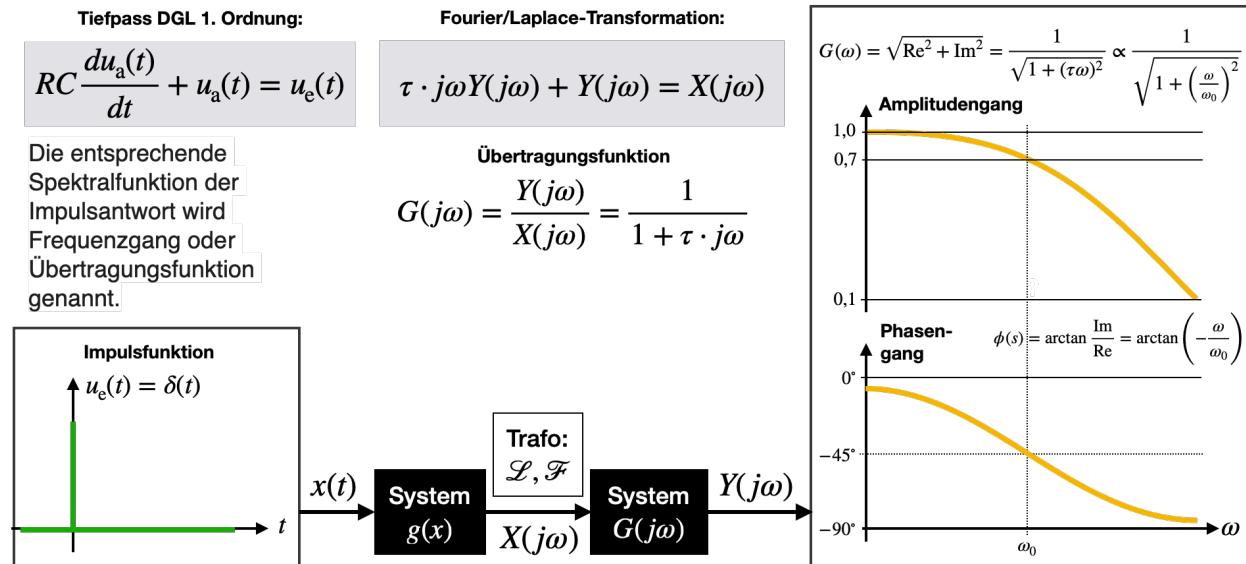
- Die Amplitude wird wie folgt berechnet, wobei $\tau = RC =: 1/\omega_0$

$$G(\omega) = \sqrt{\text{Re}^2 + \text{Im}^2} = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_0}\right)^2}}$$

- Die Phase wird wie folgt berechnet, wobei $\tau = RC =: 1/\omega_0$

$$\phi(\omega) = \arctan\left(\frac{\text{Im}}{\text{Re}}\right) = \arctan\left(-\frac{\omega}{\omega_0}\right)$$

Sowohl Amplitude als auch Phase hängen von der Frequenz ω des eingehenden Signals ab! Daher nennt man die Funktionen für Amplitude und Phase auch **Amplitudengang** bzw. **Phasengang**. Beide zusammengenommen bilden den **Frequenzgang** eines Systems und werden häufig zusammen geplottet, im sogenannten **Bode-Diagramm**. Eine solche Darstellung ist im folgenden Bild gezeigt:

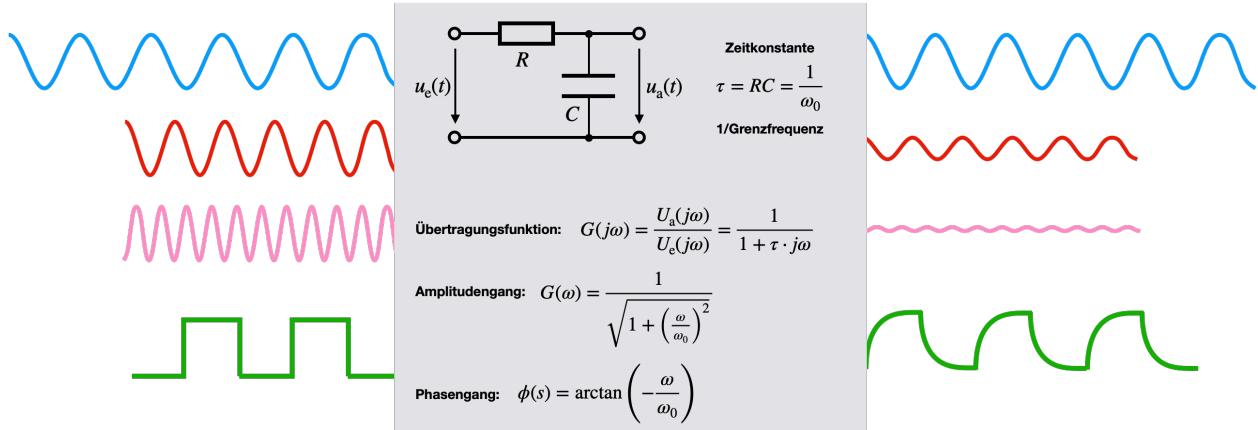


Auch hier kann man, wie schon bei der Interpretation der Sprungantwort, Kenngrößen ablesen. Bei der **Grenzfrequenz** $\omega_0 = 1/\tau$ fällt die Amplitude auf $1/\sqrt{2} = 0,707 = -3 \text{ dB}$ ab. Häufig werden Systeme anhand der Grenzfrequenz charakterisiert. Man sollte aber nicht vergessen, dass bei Signalen mit diesen Frequenzanteilen bereits signifikante Verluste in Höhe von 29% zu erwarten sind, die eigentlich nicht tolerierbar sind. Je höher die Frequenz, desto höher die Verluste (bei dem hier dargestellten Tiefpassfilter!). Das heißt der Fehler, der bei einer Messung gemacht wird, ist frequenzabhängig!

In der Akustik kann damit leben, kann man kaum hören den Unterschied. In der Messtechnik bei der Überwachung von schwingenden Maschinenteilen oder der Ermittlung von Rundlaufabweichungen von drehenden Wellen ist solch ein Amplitudenabfall meist nicht zu akzeptieren. Üblicherweise sollte man andere Grenzfrequenzen separat angeben, die 90% oder 99% der Signalstärke durchlassen.

Sind die zeitbestimmenden Glieder (R und C) des Tiefpasses bestimmt, können wir daraus die Grenzfrequenz bestimmen und somit den Verlust abschätzen.

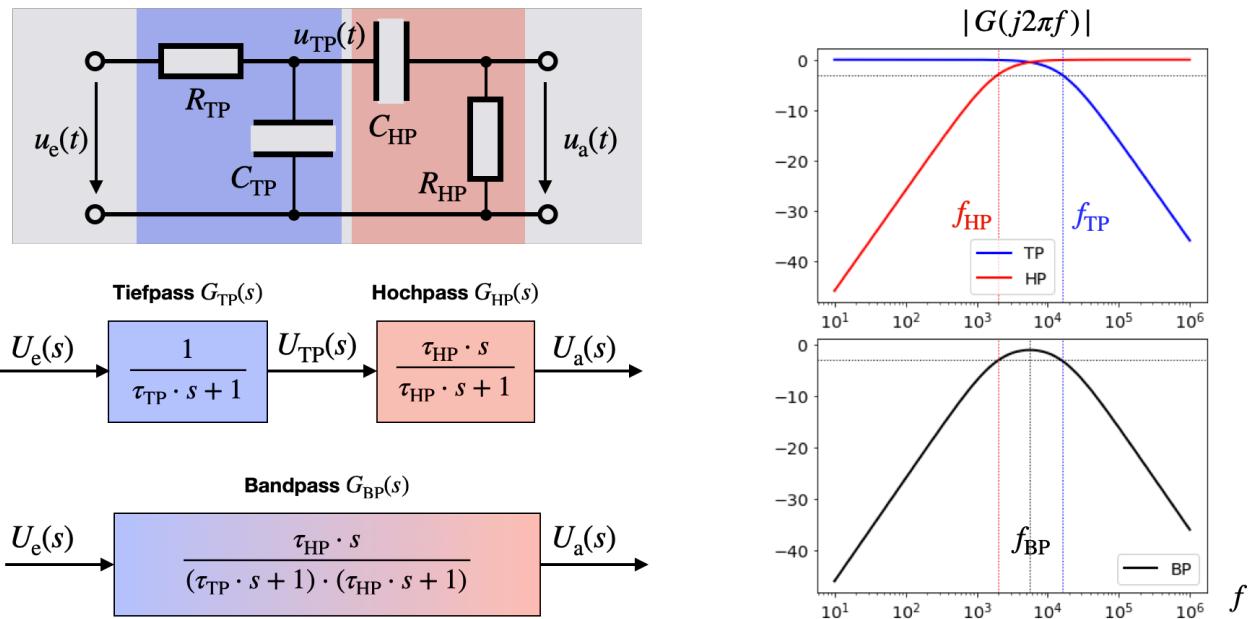
Im Folgenden Bild grafisch dargestellt, wie verschiedene Signale durch einen Tiefpass verfälscht werden können, in dem hohe Frequenzanteile abgeschwächt werden. Auch hier erkennt man wieder einen Zusammenhang zu den Fourierreihen, wenn man sich das Rechtecksignal ansieht. Eine Reihe von Rechteckpulsen benötigt eine hohe Anzahl von Sinusfunktionen bei höheren harmonischen der Grundfrequenz, um möglichst steile Flankenübergänge zu erhalten. Eine Filterung dieser hohen Frequenzanteile sorgt für eine deutliche Verzerrung des Signals. Bei sinusförmigen Signalen hingegen wird nur die Amplitude abgeschwächt und es findet zusätzlich, je nach Frequenz, eine zeitlich Verzögerung statt, d.h. die Signale sind phasenverschoben um bis zu -90° .



7.4.4 Kombination von Übertragungsfunktionen: Faltung im Frequenzraum

Auch das Hintereinanderschalten von Messsystemen ist im Frequenzraum viel einfacher zu berechnen als im Zeitraum. Eine Hintereinanderschaltung bedeutet allgemein nichts anderes, als die Faltung von Übertragungsfunktionen der jeweiligen Messsysteme. Im Zeitbereich müsste man hierzu das Faltungsintegral lösen. Im Frequenzraum ist die Faltung lediglich eine Multiplikation der Übertragungsfunktionen.

Als Beispiel soll uns ein Bandpass dienen, der aus der Hintereinanderschaltung eines Hoch- und Tiefpasses realisiert werden kann. Im Folgenden Bild sind die Komponenten des Tiefpasses (TP) blau dargestellt, die des Hochpasses (HP) rot dargestellt. Der Tiefpass lässt tiefe Frequenzen bis zu seiner Grenzfrequenz passieren (bis auf die 71% Signalverlust) und der Hoch lässt hohe Frequenzen bis zu seiner Grenzfrequenz passieren. Wir wählen C und R der beiden elektronischen Schaltungen so, dass die Grenzfrequenz der Hochpasses unterhalb der des Tiefpasses liegt, also $f_H P < f_{TP}$. Der Amplitudengang ist rechts im folgenden Bild geplottet:



Wie eben schon beschrieben, können im Zeitraum die Übertragungsfunktionen der Einzel-Systeme (hier also Hochpass und Tiefpass) einfach multipliziert werden und man erhält die kombinierte Übertragungsfunktion des resultierenden Bandpasses. Wird die Übertragungsfunktionen in Einheiten von dB gezeichnet, also logarithmisch aufgetragen, so ergibt sich eine weitere grafische Vereinfachung bei der Kombination: In logarithmischen Einheiten können die einzelnen Übertragungsfunktionen in einem Amplitudengangs-Plot addiert (!) statt multipliziert werden.

7.4.5 Anlegen von Testfunktionen

Nicht nur im Zeitraum können Sprünge oder Impulse angelegt werden. Für diese Testfunktionen können auch die Laplace, bzw. Fourier-Transformierten berechnet werden. Auch dies ist im Frequenzraum häufig einfacher, da die Testfunktionen, wie es in der Tabelle im folgenden Bild zu erkennen ist, sehr einfach sind.

Um das Verhalten unseres eben diskutierten Bandpasses auf verschiedene Eingangssignale zu untersuchen, können wir die Gleichung der Übertragungsfunktion einfach nach \$U_a\$ auflösen. Das liefert uns im allgemeinen Fall eine Gleichung für das zu erwartende Ausgangssignal:

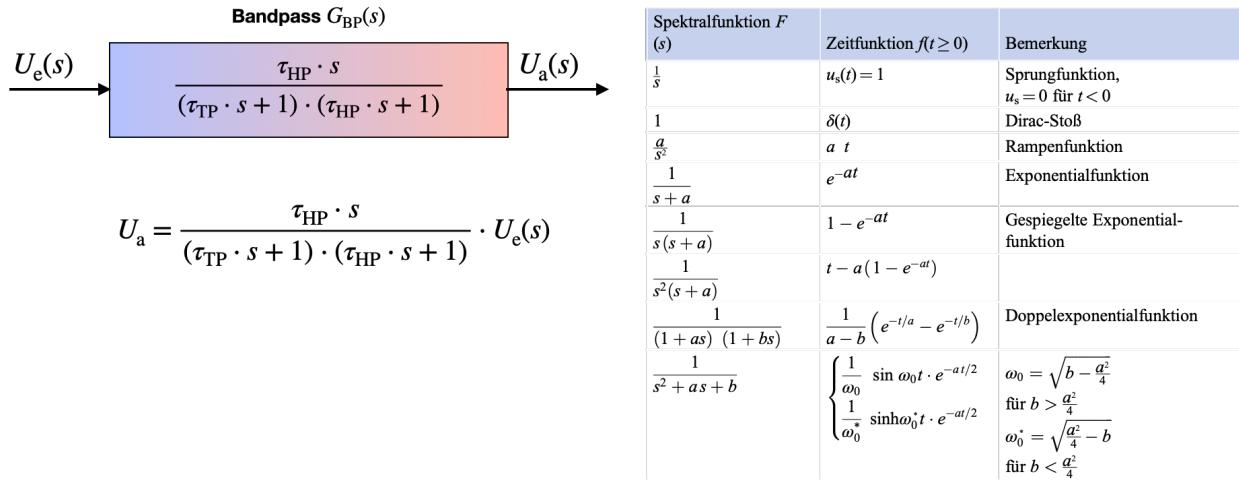
$$U_a = G(s) \cdot U_e(s)$$

Wählen wir als Eingangssignal einen Dirac-Puls, \$\delta(t)\$, um die Impulsantwort zu berechnen, so erhalten wir diese in dem wir für \$U_e(s) = 1\$. Dies ist einfach die Laplace-Transformierte eines Delta-Peaks.

Wählen wir als Eingangssignal einen Sprung, möchten also die Sprungantwort bestimmen, so setzen wir in die Gleichung \$U_e(s) = 1/s\$, die Laplace-Transformierte einer Sprungfunktion.

Auch hierbei handelt es sich wieder um Faltungen im Frequenzraum, deren Berechnungen im Zeitbereich sehr viel komplizierter wären.

Aus Übertragungsfunktionen können noch weitere Eigenschaften von Messsystemen abgeleitet werden, auf die wir hier nicht näher eingehen können. Aus den Nullstellen und Polstellen kann aber abgelesen werden, ob das System stabil ist, sprunghfähig ist oder eher ein integrales Verhalten aufweist.



Einige Korrespondenzen der Laplace-Transformation

7.4.6 Verhalten von Systemen 2. Ordnung

Der Vollständigkeitshalber wollen wir uns noch ganz kurz die Bode-Diagramme von Systemen 2. Ordnung ansehen. Auf eine mathematische Beschreibung wollen wir an dieser Stelle aber verzichten.

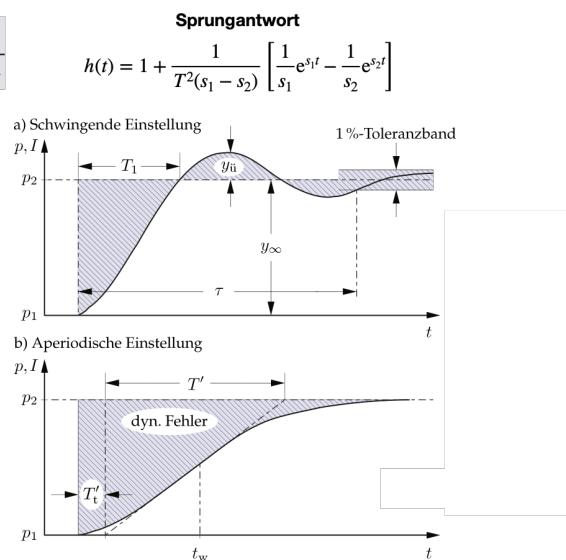
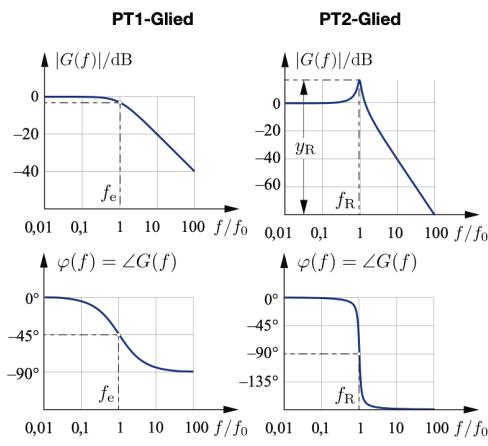
Bei Systemen 1. Ordnung handelt es sich um Systeme mit Energiespeicher, also alle Systeme die irgendwie warm werden. Bei Systemen 2. Ordnung hat man zwei gekoppelte Energiespeicher, die Energie unter Umständen periodisch austauschen können. Hier findet man dann immer einen zusätzlichen Term in der DGL der die Dämpfung des Systems beschreibt.

Um ein System 1. Ordnung von einem System 2. Ordnung zu unterscheiden, kann man sich das Bode-Diagramm (links im nachfolgenden Bild) ansehen. Bei Systemen 1. Ordnung fällt die Amplitude innerhalb einer Frequenzdekade (also ein Faktor 10) um -20 dB ab, bei Systemen 2. Ordnung um -40 dB (*Übung: Warum?*). Auch die zeitliche Verzögerung, also die Phase des Eingangssignals, erfährt ebenfalls einen steileren Abfall.

Rechts im Bild ist das Zeitliche Verhalten einer Sprungantwort dargestellt. Je nach Dämpfung erhält man ein Überschwingen, ein langsames Annähern (aperiodische Einstellung in b)) oder sogar ein oszillierendes Verhalten (schwingende Einstellung in a)) des Ausgangssignals um das Endsignal. Aufgrund dieser Dynamik entstehen Fehler und es ist ratsam eine gewisse Zeit zu warten, bis der Endwert auch hier ein 1% Toleranzband erreicht, ähnlich wie bei Systemen 1. Ordnung. Hieraus kann man außerdem schlussfolgern, dass man in der Tat immer eine gewisse Dämpfung haben möchte, damit die Oszillationen frühzeitig abklingen. Eine zu hohe Dämpfung verursacht jedoch lange Wartezeiten, bis das Ausgangssignal sich dem endgültigem Wert endlich angenähert hat.

1. Ordnung: Energiespeicher
 2. Ordnung: Zwei gekoppelte Energiespeicher, Dämpfung

$$G(s) = \frac{1}{T^2 s^2 + 2\delta T s + 1}$$



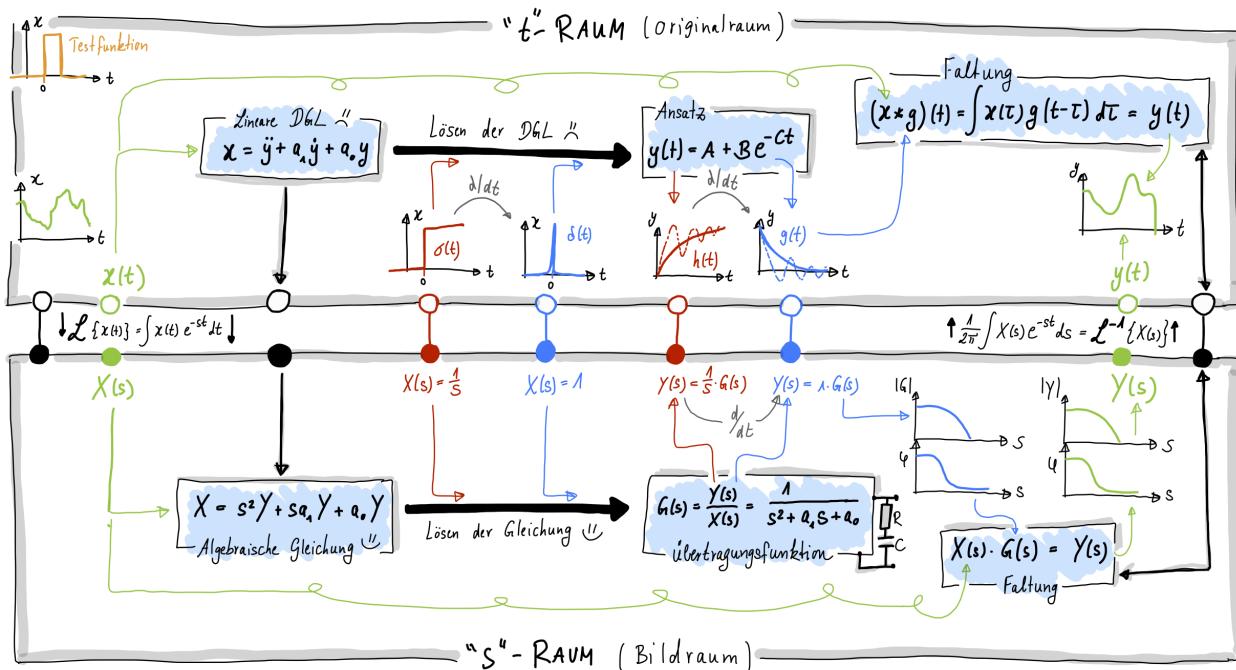
Einige Beispiele zu Systemen mit Verzögerungen, Dämpfungs bzw. auch integrierendem Verhalten sind in nachfolgender Tabelle dargestellt. In der Literatur findet man häufig diese tabellarischen Zusammenfassungen verschiedener Messsysteme inklusive Übertragungsfunktion und Bode-Diagramm, damit die DGL nicht jedes mal neu hergeleitet werden müssen. Die Hintereinanderschaltung einzelner Komponenten kann auch hier wieder ganz einfach im logarithmischen Bode-Diagramm per Addition der Übertragungsfunktionen abgeschätzt werden.

Fachhochschule OOW Fachbereich Ingenieurwissenschaften		Tabelle der wichtigsten Regelkreisglieder Teil 1					Labor für Regelungs- und Steuerungstechnik Dipl. Ing. Dieter Suckert
	Differentialgleichung	Sprungantwort	Frequenzgang F	Amplitudengang F =	Phasengang phi=[rad]	Ortskurve	Bode-Diagramm
P	$x_a = V \cdot x_c$		V	V	0		
PT ₁	$T_1 \cdot \dot{x}_a + x_a = V \cdot x_c$		$\frac{V}{1+T_1 \cdot p}$	$\frac{V}{\sqrt{1+T_1^2 \omega^2}}$	$-\arctan T_1 \cdot \omega$		
PT ₂	$T_2^2 \cdot \ddot{x}_a + T_1 \cdot \dot{x}_a + x_a = V \cdot x_c$		$\frac{V}{1+T_1 \cdot p + T_2^2 \cdot p^2}$	$\frac{V}{\sqrt{(1-T_2^2 \omega^2)^2 + T_1^2 \omega^2}}$	$-\arctan \frac{T_1 \cdot \omega}{1-T_2^2 \cdot \omega^2}$		$D = \frac{T_1}{2 \cdot T_2}$
D	$x_a = K_D \cdot \dot{x}_c$		$K_D \cdot p$	$K_D \cdot \omega$	$\frac{\pi}{2}$		
DT ₁	$T_1 \cdot \dot{x}_a + x_a = K_D \cdot \dot{x}_c$		$\frac{K_D \cdot p}{1+T_1 \cdot p}$	$\frac{K_D \cdot \omega}{\sqrt{1+T_1^2 \omega^2}}$	$\frac{\pi}{2} - \arctan T_1 \cdot \omega$		
I	$x_a = K_I \cdot \int x_c dt$		$\frac{K_I}{p}$	$\frac{K_I}{\omega}$	$-\frac{\pi}{2}$		
IT ₁	$T_1 \cdot \dot{x}_a + x_a = K_I \cdot \int x_c dt$		$\frac{K_I}{p(1+T_1 \cdot p)}$	$\frac{K_I}{\omega \cdot \sqrt{1+T_1^2 \omega^2}}$	$-\frac{\pi}{2} - \arctan T_1 \cdot \omega$		

7.4.7 Zusammenfassung

Bevor wir zu der Zusammenfassung kommen, soll im folgenden Bild noch einmal dargestellt werden, welche Trick und Tips man sich im Frequenzraum noch von Nutzen machen kann, solange es sich um ein LZI System handelt!

- Mittels Fourier-Transformation kann ich jederzeit in den Frequenzraum wechseln und mittels Rücktransformation zu gehen. Egal ob es sich um ein Signal oder ein System (hier wird häufig der Laplace-Raum verwendet) handelt.
- Die Ableitung der Sprunganregung ist eine Dirac-Funktion (Impulsanregung). Die Ableitung der Sprungantwort ergibt die Impulsantwort. Die gilt sowohl im Zeit- wie auch im Frequenzraum. (*Frage: Was sieht die zeitliche Ableitung der Fourier-Transformierten $1/s$?)
- Lösen von DGLs im Laplace-Raum ist eine algebraische Umformung der Gleichung
- Lösen von DGLs im Zeitraum benötigt einen Lösungsansatz, eine homogene Lösung, eine spezielle Lösung und verschiedene Schritte inkl. Anfangsbedingungen um die Lösung zu finden.
- Die Faltung im Laplace-Raum ist eine Multiplikation
- Die Faltung im Zeitraum ist ein Integral
- Die Faltung ist generell ein hilfreiches Werkzeug um...
 - Systeme hintereinanderzuschalten und die Gesamt-Übertragungsfunktion zu berechnen (Faltung der beiden Systeme berechnen)
 - Die Antwort auf ein beliebiges Eingangssignal mittels Impulsantwort zu bestimmten (Faltung von beliebigem Eingangssignal mit Impulsantwort liefert Antwort des Systems auf das beliebige Eingangssignal)
 - Die Faltung hilft bei allen drei Problemen: System-Identifizierungsproblem, Simulationsproblem und Kontrollproblem



7.5 Faltung und Korrelation

Part IV

Sensoren

MESSEN ELEKTRISCHER GRÖßen

Richtig messen

- *Elektronische Minimum*
- *Spannungsmesseingang*
- *Strommesseingang*
- *Operationsverstärker*
- *Leistungsmessung*
- *Widerstandsmessung*
- *Frequenzmessung*

8.1 Elektronische Minimum

8.2 Spannungsmesseingang

8.3 Strommesseingang

8.4 Operationsverstärker

8.5 Leistungsmessung

8.6 Widerstandsmessung

8.7 Frequenzmessung

SENSOREN

Richtig messen

- *Einführung*
- *Dehnungsmessstreifen*
- *Piezoelektrische Sensoren*
- *Kapazitive Sensoren*
- *Drucksensoren*
- *Temperaturmessungen*
- *Messung von Wärmestrahlung*
- *Optische Sensoren*

9.1 Einführung

9.2 Dehnungsmessstreifen

9.3 Piezoelektrische Sensoren

9.4 Kapazitive Sensoren

9.5 Drucksensoren

9.6 Temperaturmessungen

9.7 Messung von Wärmestrahlung

9.8 Optische Sensoren

Part V

Tutorials

JUPYTER NOTEBOOKS

- *Klimadatei (.csv) einlesen und analysieren*
- *Fehlerfortpflanzung*
- *Vergleich verschiedener Fit-Funktionen*
- *Lineare Regression mit Berücksichtigung von Fehlerbalken*
- *Fourier-Transformation: Entfernung von Hintegrundrauschen in einem Audiosignal*

10.1 Klimadatei (.csv) einlesen und analysieren

Zunächst werden die für dieses Jupyter Notebook benötigten Libraries geladen:

```
#Benötigte Libraries:  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import plotly.offline as py  
py.init_notebook_mode(connected=True)  
import plotly.graph_objs as go  
import plotly.tools as tls  
import seaborn as sns  
import time  
import warnings  
warnings.filterwarnings('ignore')  
  
# Matplotlib Settings:  
plt.style.use('default') # Matplotlib Style wählen  
plt.figure(figsize=(10,5)) # Plot-Größe  
plt.rcParams['font.size'] = 10; # Schriftgröße
```

```
<Figure size 1000x500 with 0 Axes>
```

10.1.1 .csv-Datei als *DataFrame* einlesen

Im Folgenden Nutzen wir globale Klimadaten, die auf der Webseite der NASA zu finden sind: <https://data.giss.nasa.gov/gistemp/>. Hierbei handelt es sich um Temperaturdaten, die Anomalien gegenüber dem Mittelwert in den Jahren 1951-1980 aufweisen. Es werden Daten von Dateien (online oder offline) eingelesen mit der Python Bibliothek pandas. Die Daten werden in sogenannten *DataFrames* hier mit dem Namen `global_mean` abgespeichert.

```
#link = "https://data.giss.nasa.gov/gistemp/graphs_v4/graph_data/Global_Mean_
→Estimates_based_on_Land_and_Ocean_Data/graph.csv"
link = 'data/graph.csv'
global_mean = pd.read_csv(link, header = 1)
```

Wir geben das *DataFrame* aus um uns die Messdaten einmal anzusehen:

```
global_mean.head(6) # Ausgabe der ersten 5 Spalten
#global_mean.tail(5) # Ausgabe der letzten 5 Spalten
#global_mean # Ausgabe des DataFrames
```

	Year	No_Smoothing	Lowess(5)
0	1880	-0.17	-0.09
1	1881	-0.08	-0.13
2	1882	-0.11	-0.17
3	1883	-0.18	-0.20
4	1884	-0.28	-0.24
5	1885	-0.33	-0.26

In der ersten Spalte befinden sich lediglich die Indizes der Messungen. Die zweite Spalte beinhaltet das Jahr und die dritte Spalte zeigt den gemessenen globalen Temperaturunterschied im Vergleich zur gemittelten Temperatur der Jahre 1951-1980. Die letzte Spalte zeigt die gleichen Messwerte, jedoch gefiltert.

Einzelne Spalten kann man sich anzeigen lassen, indem den Spalten-Namen des zugehörigen *DataFrames* nutzt:

```
global_mean['Year']
```

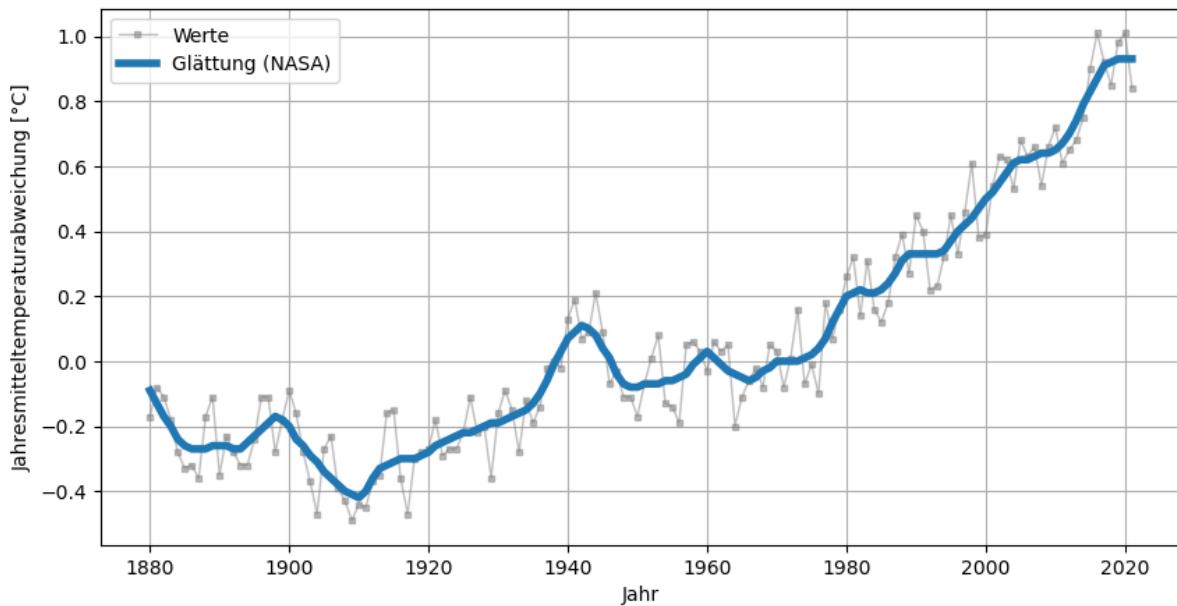
0	1880
1	1881
2	1882
3	1883
4	1884
	...
137	2017
138	2018
139	2019
140	2020
141	2021

Name: Year, Length: 142, dtype: int64

10.1.2 Daten plotten mit 'matplotlib'

Als Beispiel für eine gelungene grafische Darstellung wollen wir die beiden Spalten, *No_Smoothing* und *Lowess(5)* gegenüber der Zeitachse *Year* plotten. Hierfür benutzen wir die Python Library `matplotlib`:

```
import matplotlib.pyplot as plt
plt.style.use('default')
plt.figure(figsize=(10,5))
plt.rcParams['font.size'] = 10;
plt.plot(global_mean["Year"],global_mean["No_Smoothing"], ls="--", lw=1, marker="s", ms=3, color="tab:gray", alpha=0.5, label="Werte");
plt.plot(global_mean["Year"],global_mean["Lowess (5)"], lw=4, color="tab:blue", label="Glättung (NASA)");
plt.xlabel('Jahr')
plt.ylabel("Jahresmitteltemperaturabweichung [°C]")
plt.legend();
plt.grid();
```



10.1.3 Messunsicherheiten als Fehlerbalken hinzufügen

Bei diesem Datenset stehen uns leider keine Messunsicherheiten zur Verfügung. Um Sie jedoch als Fehlerbalken miteinzubeziehen, wollen wir im Folgenden annehmen, dass der Temperaturunterschied auf 0.25K genau messen werden konnte und fügen die unserem Datensatz hinzu:

```
global_mean["uncertainty"] = 0.25
print(global_mean)
```

	Year	No_Smoothing	Lowess (5)	uncertainty
0	1880	-0.17	-0.09	0.25
1	1881	-0.08	-0.13	0.25
2	1882	-0.11	-0.17	0.25
3	1883	-0.18	-0.20	0.25

(continues on next page)

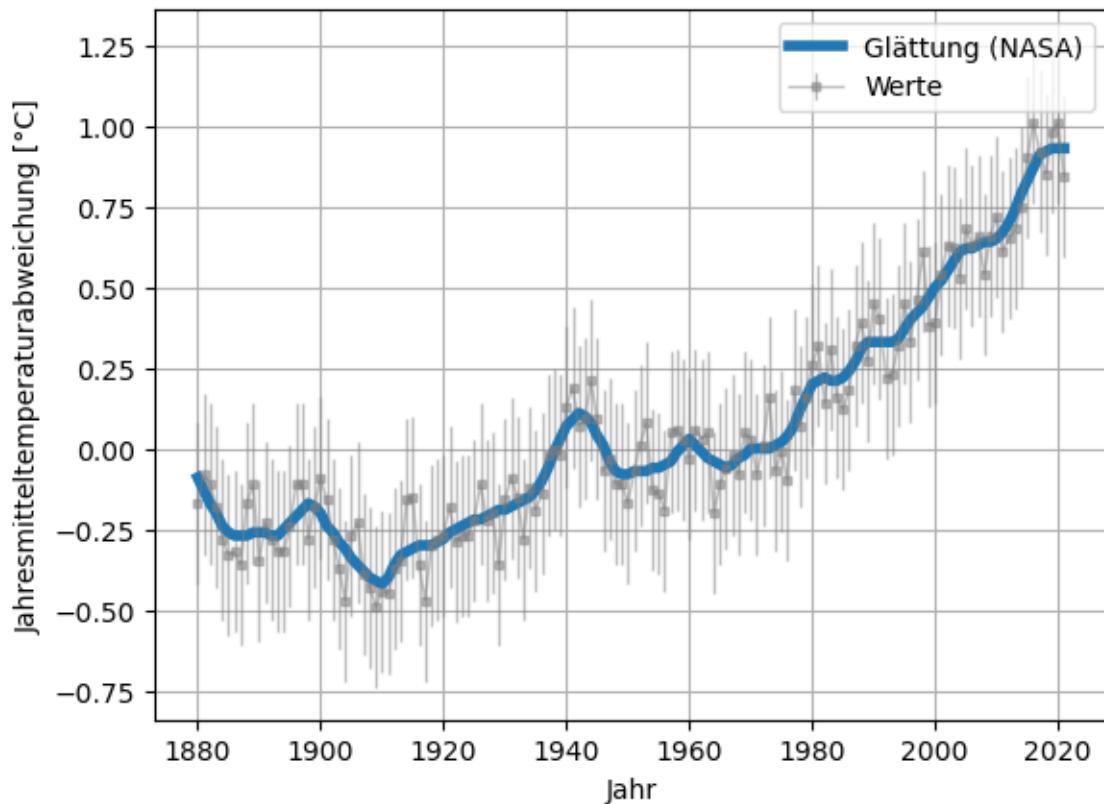
(continued from previous page)

4	1884	-0.28	-0.24	0.25
..
137	2017	0.92	0.91	0.25
138	2018	0.85	0.92	0.25
139	2019	0.98	0.93	0.25
140	2020	1.01	0.93	0.25
141	2021	0.84	0.93	0.25

[142 rows x 4 columns]

Grafisch darstellen tun wir Messunsicherheiten mittels Fehlerbalken und der Matplotlib-Funktion plt.errorbar.

```
plt.errorbar(global_mean["Year"],global_mean["No_Smoothing"], yerr=global_mean[
    "uncertainty"], ls="--", lw=1, marker="s", ms=3, color="tab:gray", alpha=0.5, label=
    "Werte");
plt.plot(global_mean["Year"],global_mean["Lowess (5)"], lw=4, color="tab:blue", label=
    "Glättung (NASA)");
plt.xlabel('Jahr')
plt.ylabel("Jahresmitteltemperaturabweichung [°C]")
plt.legend();
plt.grid();
```



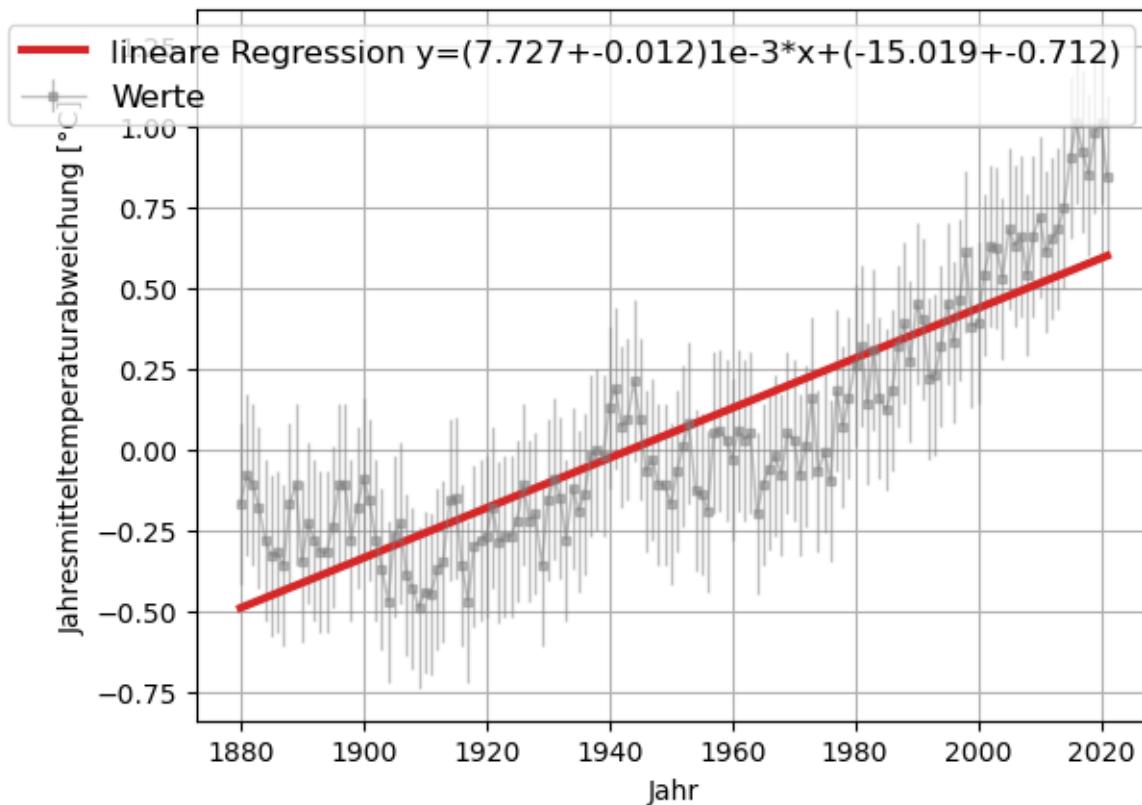
10.1.4 Ausgleichsgerade berechnen und plotten

Mittels linearer Regression kann der Temperaturanstieg aus den Daten berechnet werden. Hierfür wird die Python Library `numpy` benutzt und die Funktion `polyfit` aufgerufen und in als `model` gespeichert. Diese Funktion benutzt die Least-Square Methode für polynomische Modelle. Weitere Informationen zu der Funktion findet ihr [hier](#). Mit der Option `cov=True` wird die Kovarianz-Matrix berechnet, welche die Unsicherheiten für die Fit-Parameter beinhaltet.

```
import numpy as np
import pandas as pd

x=global_mean["Year"]
y=global_mean["No_Smoothing"]
y_err = global_mean["uncertainty"]
model = np.polyfit(x, y, deg=1, w=1/y_err, cov=True) # 1. Wert = Anstieg , 2. Wert = Schnittpunkt mit y-Achse
y_model = model[0][0]*x+model[0][1] # Modell einer linearen Regression

plt.ylabel("Jahresmitteltemperaturabweichung [°C]")
plt.xlabel("Jahr")
plt.errorbar(global_mean["Year"],global_mean["No_Smoothing"], yerr=global_mean["uncertainty"], ls="--", lw=1, marker="s", ms=3, color="tab:gray", alpha=0.5, label="Werte");
plt.plot(x,y_model, ls="--", lw=3, color="tab:red", label=f"lineare Regression y=( {model[0][0]*1000:.3f}+{np.sqrt(model[1][0]*1000):.3f})1e-3*x+({model[0][1]:.3f}+{np.sqrt(model[1][1]):.3f})");
plt.legend(fontsize=12);
plt.grid();
```



Das Modell beinhaltet zwei Matrizen:

```
model
```

```
(array([ 7.72740894e-03, -1.50191421e+01]),
 array([[ 1.33257719e-07, -2.59919181e-04],
       [-2.59919181e-04,  5.07196269e-01]]))
```

Im ersten Array stehen die Fit-Parameter der linearen Ausgleichsgeraden entsprechend der obigen Deklaration: $y_{\text{model}} = \text{model}[0][0] * x + \text{model}[0][1]$. Im zweiten Array, hier eine 2x2 Matrix, sind die Unsicherheiten in Form von der Kovarianz-Matrix dargestellt. Der Temperaturanstieg kann entsprechend ausgegeben werden:

```
print(f"Temperaturanstieg pro Jahr (von 1981 bis 2020): {model[0][0]:.3f}°C/Jahr")
print(f"Temperaturanstieg seit Beginn der Messung: {(y_model.iloc[-1]-y_model.
    ↪iloc[0]):.3f}°C")
```

```
Temperaturanstieg pro Jahr (von 1981 bis 2020): 0.008°C/Jahr
Temperaturanstieg seit Beginn der Messung: 1.090°C
```

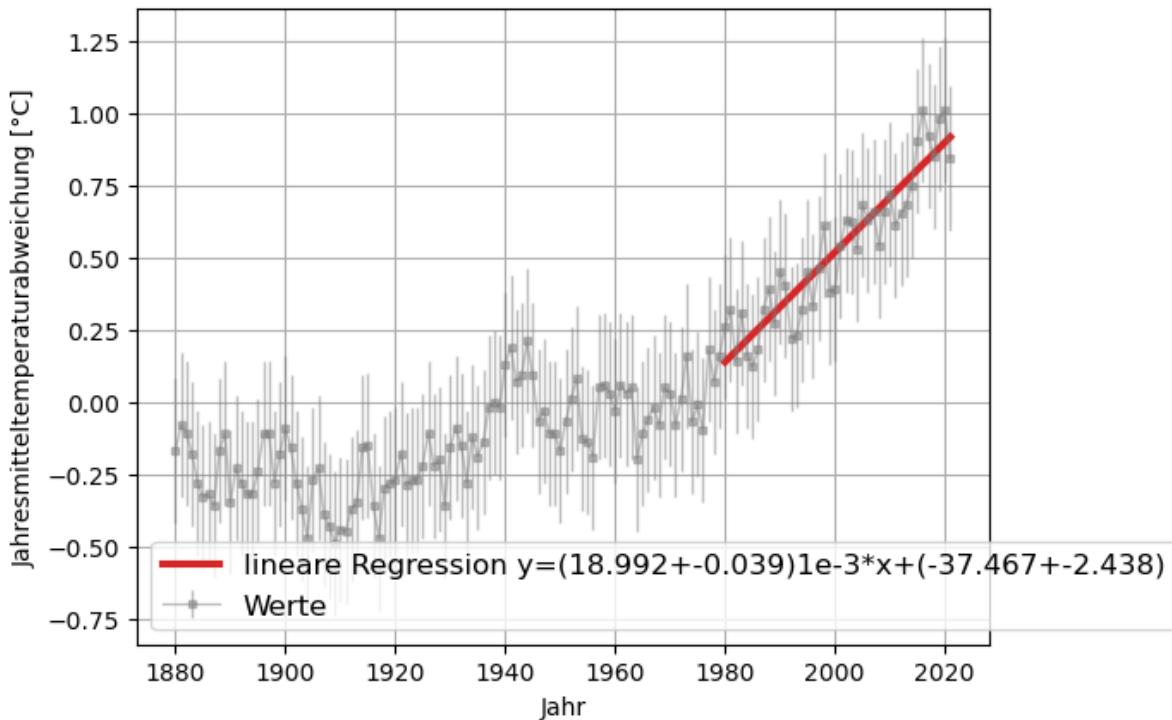
Warning: Die lineare Regression bezieht hier den ganzen Zeitraum mit ein! Im folgenden betrachten wir für den Temperaturgradienten nur die Daten von 1980 bis 2020!

```
x=global_mean.loc[global_mean["Year"] >= 1980,"Year"]
y=global_mean.loc[global_mean["Year"] >= 1980,"No_Smoothing"]
y_err = global_mean.loc[global_mean["Year"] >= 1980,"uncertainty"]

model = np.polyfit(x, y, deg=1, w=1/y_err, cov=True) # 1. Wert = Anstieg , 2. Wert =
    ↪Schnittpunkt mit y-Achse
y_model = model[0][0]*x+model[0][1] # Modell einer linearen Regression
print(f"Temperaturanstieg pro Jahr (von 1980 bis 2020): {model[0][0]:.3f}°C/Jahr")
```

```
Temperaturanstieg pro Jahr (von 1980 bis 2020): 0.019°C/Jahr
```

```
plt.ylabel("Jahresmitteltemperaturabweichung [°C]")
plt.xlabel("Jahr")
plt.errorbar(global_mean["Year"],global_mean["No_Smoothing"], yerr=global_mean[
    ↪"uncertainty"], ls="--", lw=1, marker="s", ms=3, color="tab:gray", alpha=0.5, label=
    ↪"Werte");
plt.plot(x,y_model, ls="--", lw=3, color="tab:red", label=f"lineare Regression y=(
    ↪{model[0][0]*1000:.3f}+{np.sqrt(model[1][0]*1000):.3f})1e-3*x+({model[0][1]:.3f}
    ↪+{np.sqrt(model[1][1]):.3f})");
plt.legend(fontsize=12);
plt.grid();
```



10.2 Fehlerfortpflanzung

10.2.1 Messdaten: Karussell

Wir stellen uns im Folgenden ein Kinderkarussel vor und wir wollen die Geschwindigkeit der Feuerwehrautos ermitteln. Das Auto selbst verfügt über kein Tachometer - es bleibt also eigentlich nur die Möglichkeit, die Zeit eines Umlaufs zu stoppen und aus der Geometrie des Karussells den zurückgelegten Weg zu bestimmen. Daraus lässt sich die Geschwindigkeit leicht berechnen.

Beispielsweise könnte man die Zeit nach jedem Umlauf messen, die näherungsweise konstant sein sollte (bis auf die 1. Runde, wo noch beschleunigt wird). Es könnte sich also ein linearer Zusammenhang zwischen zurückgelegtem Weg und der benötigten Zeit vermuten:

$$y = m \cdot x + b$$

Folgende Messwerte wurden ermittelt:

- Weg x (m): 12, 24, 36, 42, 60, 72, 84, 96, 108, 120
- Zeit t (s): 12.2, 17, 22.1, 33.2, 34.4, 59.1, 60.2, 65.7, 69.9, 70.1

```
import numpy as np
import matplotlib.pyplot as plt
# Matplotlib Settings:
plt.style.use('default') # Matplotlib Style wählen
plt.figure(figsize=(7,5)) # Plot-Größe
plt.rcParams['font.size'] = 10; # Schriftgröße

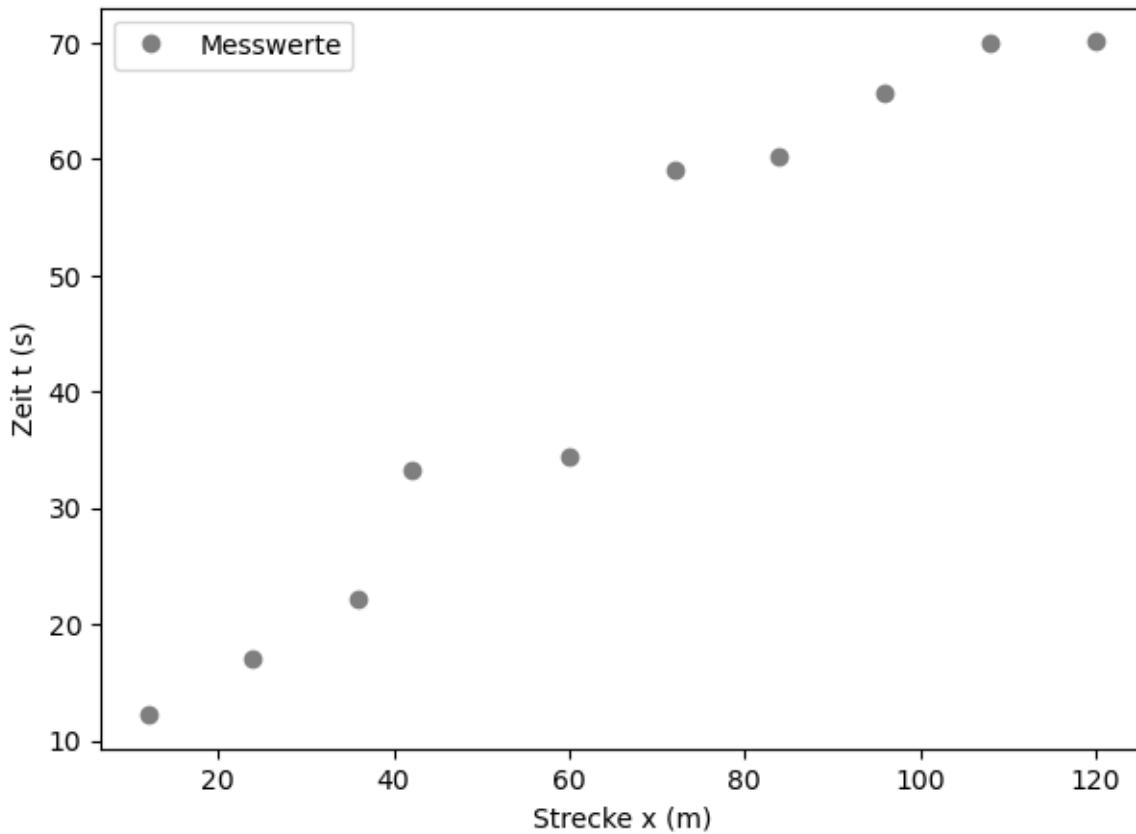
x = [12, 24, 36, 42, 60, 72, 84, 96, 108, 120] # Messwerte der Strecke x in m
```

(continues on next page)

(continued from previous page)

```
t = [12.2, 17, 22.1, 33.2, 34.4, 59.1, 60.2, 65.7, 69.9, 70.1] # Messwerte der Zeit t
# in sek.
x = np.array(x) # konvertiere die Messwerte in ein Numpy-Array
t = np.array(t) # konvertiere die Messwerte in ein Numpy-Array

plt.plot(x,t, 'o', label = 'Messwerte', ms=6, color="tab:gray")
plt.xlabel('Strecke x (m)')
plt.ylabel('Zeit t (s)')
plt.legend()
plt.show()
```



10.2.2 Lineare Regression

Um die Geschwindigkeit des Karussells zu bestimmen benötigen wir die Ausgleichsgerade. Weitere Information dazu findest du [hier](#) im Vorlesungsskript oder im [Tutorial zur Linearen Regression](#). Im folgenden code berechnen wir die Steigung einmal analytisch und einmal mittels .

```
# Analytisch:
m = (np.mean(x*t) - np.mean(x)*np.mean(t)) / (np.mean(x**2) - np.mean(x)**2)
b = np.mean(t) - m * np.mean(x)

# Fit-Funktion:
from scipy.optimize import curve_fit
def fit_lin(x, b, a): # Fit Funktion für eine Gerade mit Steigung b
```

(continues on next page)

(continued from previous page)

```

return b*x + a
fit_out = curve_fit(fit_lin, x, t)

print('Die analytischen Parameter lauten:')
print('Die Steigung ist \t\t m = %5.4f s/m' %(m))
print('Der Ordinatenabschnitt ist \t\t b = %5.4f s' %(b))

print('Die scipy-Fitparameter lauten:')
print('Die Steigung ist \t\t m = (%5.4f +- %5.4f) s/m' %(fit_out[0][0], np.sqrt(fit_
    ↪out[1][0][0])))
print('Der Ordinatenabschnitt ist \t\t b = (%5.4f +- %5.4f) s' %(fit_out[0][1], np.
    ↪sqrt(fit_out[1][1][1])))

plt.plot(x,t, 'o', label = 'Messwerte', ms=6, color="tab:gray")
plt.plot(x,m*x+b, lw=3, color="tab:red", label = 'analytische Lin. Reg.: y = %5.3f*x+
    ↪%5.3f' %(m,b)) # plot Ausgleichsgerade mit m und b
plt.plot(x,fit_out[0][0]*x+fit_out[0][1], '--', lw=3, color="tab:blue", label = 'Fit_
    ↪scipy: y = %5.3f*x+%5.3f' %(fit_out[0][0],fit_out[0][1])) # plot Fitfunktion mit_
    ↪b und a

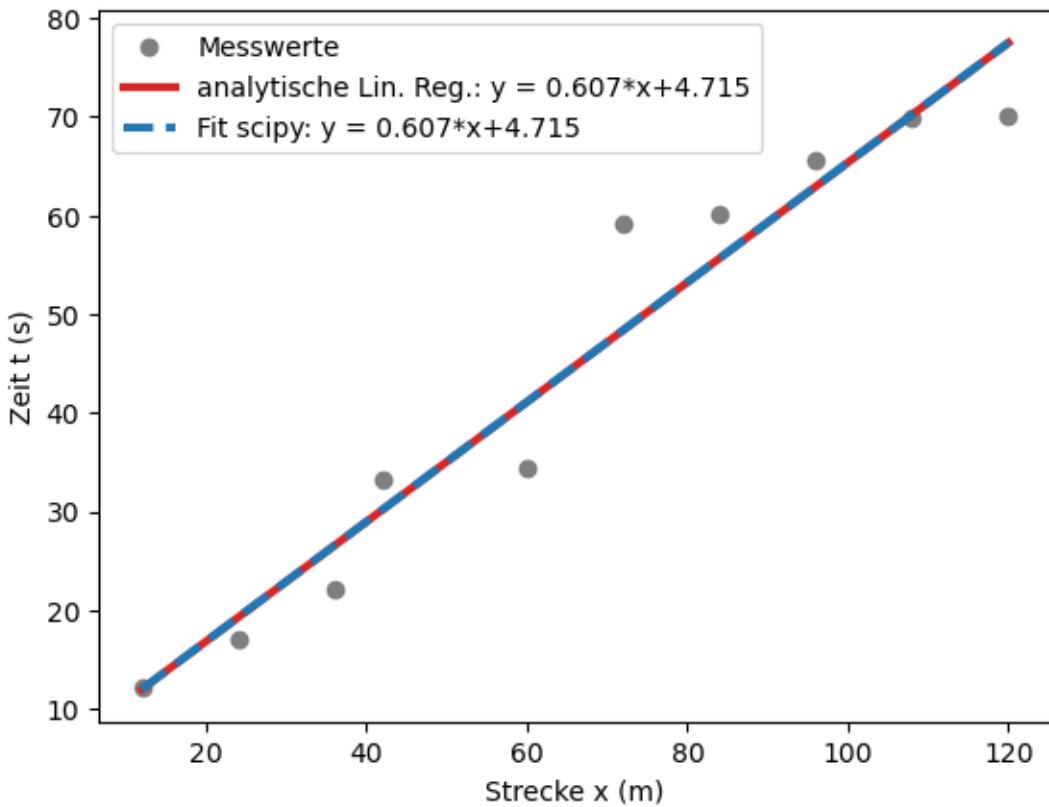
plt.xlabel('Strecke x (m)')
plt.ylabel('Zeit t (s)')
plt.legend()
plt.show()

```

```

Die analytischen Parameter lauten:
Die Steigung ist           m = 0.6067 s/m
Der Ordinatenabschnitt ist   b = 4.7148 s
Die scipy-Fitparameter lauten:
Die Steigung ist           m = (0.6067 +- 0.0534) s/m
Der Ordinatenabschnitt ist   b = (4.7148 +- 3.9581) s

```



10.2.3 Geschwindigkeit

Der Kehrwert der Steigung m liefert uns die Geschwindigkeit des Karussells, wenn wir den Kehrwert berechnen:

```
v = 1/m
v_fit=1/fit_out[0][0]
print('Die Geschwindigkeit ermittelt mittels Analytik ist: v = %5.4f m/s = %5.4f km/h
      ↪ % (v, v/1000*3600)')
print('Die Geschwindigkeit ermittelt mittels Fit ist:           v = %5.4f m/s = %5.4f km/h
      ↪ % (v_fit, v_fit/1000*3600)')
```

Die Geschwindigkeit ermittelt mittels Analytik ist: $v = 1.6484 \text{ m/s} = 5.9342 \text{ km/h}$
 Die Geschwindigkeit ermittelt mittels Fit ist: $v = 1.6484 \text{ m/s} = 5.9342 \text{ km/h}$

10.2.4 Unsicherheit der Steigung

Wie bereits oben schon erwähnt, hat die Regressionen eine Abweichung. Daher müssen wir uns jetzt fragen, was der Fehler der Geschwindigkeit ist. Diesen bekommen wir aus der Steigung m . Das heißt wir benötigen zuerst den Fehler von m :

$$s_m = s_t \cdot \sqrt{\frac{1}{N \cdot (\bar{x}^2 - (\bar{x})^2)}} \quad \text{mit} \quad s_t = \sqrt{\frac{1}{N-2} \sum (t_i - mx_i - b)^2}$$

Der Vollständigkeit halber berechnen wir ebenfalls den Fehler für b :

$$s_b = s_m \cdot \sqrt{\bar{x}^2}$$

```

N = len(t)
diff_t = 0
for i in range(N):
    diff_t += (t[i] - m * x[i] - b)**2

streuung_t = 1/(N-2)*diff_t
s_t = np.sqrt(streuung_t)
s_m = s_t * np.sqrt(1 / (N*(np.mean(x**2) - np.mean(x)**2)))
s_b = s_m * np.sqrt(np.mean(x**2))

print('Die Unsicherheit von t ist \t s_t = %5.4f s' %(s_t))
print('Die Unsicherheit von m ist \t s_m = %5.4f s/m' %(s_m))
print('Die Unsicherheit von b ist \t s_b = %5.4f s' %(s_b))

```

Die Unsicherheit von t ist	s_t = 5.8831 s
Die Unsicherheit von m ist	s_m = 0.0534 s/m
Die Unsicherheit von b ist	s_b = 3.9581 s

Die Fitroutine in Python gibt uns ebenfalls Unsicherheiten in Form von der Kovarianz-Matrix aus. Auf der Diagonale stehen die Varianzen, s^2 , auf den Nicht-Diagonalelementen stehen die Kovarianzen (Korrelationsterme zwischen m und b):

$$\text{cov}(m, b) = \begin{pmatrix} s_m^2 & s_{mb} \\ s_{bm} & s_b^2 \end{pmatrix}$$

Die Unsicherheiten für m und b erhalten wir also aus der Wurzel von den Diagonalelementen.

```

print('Die Kovarianzmatrix hat die folgende Form: \n', fit_out[1])
print('')
print('Die Unsicherheit von m ist \t s_m = %5.4f s/m' %(np.sqrt(fit_out[1][0][0])))
print('Die Unsicherheit von b ist \t s_b = %5.4f s' %(np.sqrt(fit_out[1][1][1])))

```

Die Kovarianzmatrix hat die folgende Form:	
[[2.85371544e-03 -1.86632993e-01]	
[-1.86632993e-01 1.56668987e+01]]	
Die Unsicherheit von m ist	s_m = 0.0534 s/m
Die Unsicherheit von b ist	s_b = 3.9581 s

10.2.5 Unsicherheit der Geschwindigkeit mittels Fehlerfortpflanzung

Auch die Unsicherheiten stimmen für beide Methoden perfekt überein.

Nun interessiert uns allerdings der absolute Fehler der geschätzten Geschwindigkeit. Da die Geschwindigkeit der Kehrwert der Steigung ist ($v = \frac{1}{m}$), müssen wir Fehlerfortpflanzung anwenden. Der Fehler wirkt sich wie folgt auf den Kehrwert aus:

$$\Delta v = \left| \frac{\partial v}{\partial m} \right| \cdot \Delta m = \left| -\frac{1}{m^2} \right| \cdot s_m = \frac{1}{m^2} \cdot s_m = s_v$$

Da beide Methoden die gleichen Werte für Schätzungen und Unsicherheiten ausgeben, ersparen wir uns ab nun die Berechnung der Geschwindigkeit inkl. Unsicherheit für beide Methoden. Die Fehlerrechnung wird nur noch für die analytische Methode ausgeführt, da die Werte für bei Methoden die gleichen sind:

```
s_v = 1/m**2 * s_m
print('Die Unsicherheit von v ist \t s_v = %5.4f m/s' %(s_v))
```

Die Unsicherheit von v ist $s_v = 0.1452 \text{ m/s}$

Das Messergebnis kann also wie folgt angegeben werden:

$$v = (1,6484 \pm 0,1452) \text{ m/s}$$

Ist diese Angabe sinnvoll? Wenn wir das so konkret fragen, dann vermutlich nicht... Wenn der Fehler bereits in der ersten Stelle nach dem Komma signifikant bemerkbar den Schätzwert beeinflusst, warum sollte man sich dann die Mühe machen noch weitere Nachkommastellen hinzuschreiben? Also **sinnvoll runden**:

$$v = (1,6 \pm 0,2) \text{ m/s}$$

Warum 0,2 und nicht 0,1? **Fehler werden immer aufgerundet!**

Nun könnte noch der relative Fehler $\Delta v/v$ berechnet werden.

```
print('Die relative Unsicherheit von v ist \t s_v = %5.4f %%' %(s_v/v*100))
```

Die relative Unsicherheit von v ist $s_v = 8.8057 \%$

10.3 Vergleich verschiedener Fit-Funktionen

10.3.1 Messdaten: Karussel

Wir stellen uns im Folgenden ein Kinderkarussel vor und wir wollen die Geschwindigkeit der Feuerwehrautos ermitteln. Das Auto selbst verfügt über kein Tachometer - es bleibt also eigentlich nur die Möglichkeit, die Zeit eines Umlaufs zu stoppen und aus der Geometrie des Karussells den zurückgelegten Weg zu bestimmen. Daraus lässt sich die Geschwindigkeit leicht berechnen.

Beispielsweise könnte man die Zeit nach jedem Umlauf messen, die näherungsweise konstant sein sollte (bis auf die 1. Runde, wo noch beschleunigt wird). Es könnte sich also ein linearer Zusammenhang zwischen zurückgelegtem Weg und der benötigten Zeit vermuten:

$$y = m \cdot x + b \Rightarrow y - b - m \cdot x = 0$$

Folgende Messwerte wurden ermittelt:

- Weg x (m): 12, 24, 36, 42, 60, 72, 84, 96, 108, 120
- Zeit t (s): 12.2, 17, 22.1, 33.2, 34.4, 59.1, 60.2, 65.7, 69.9, 70.1

```
import numpy as np
import matplotlib.pyplot as plt
# Matplotlib Settings:
plt.style.use('default') # Matplotlib Style wählen
plt.figure(figsize=(7,5)) # Plot-Größe
plt.rcParams['font.size'] = 10; # Schriftgröße

x = [12, 24, 36, 42, 60, 72, 84, 96, 108, 120] # Messwerte der Strecke x in m
```

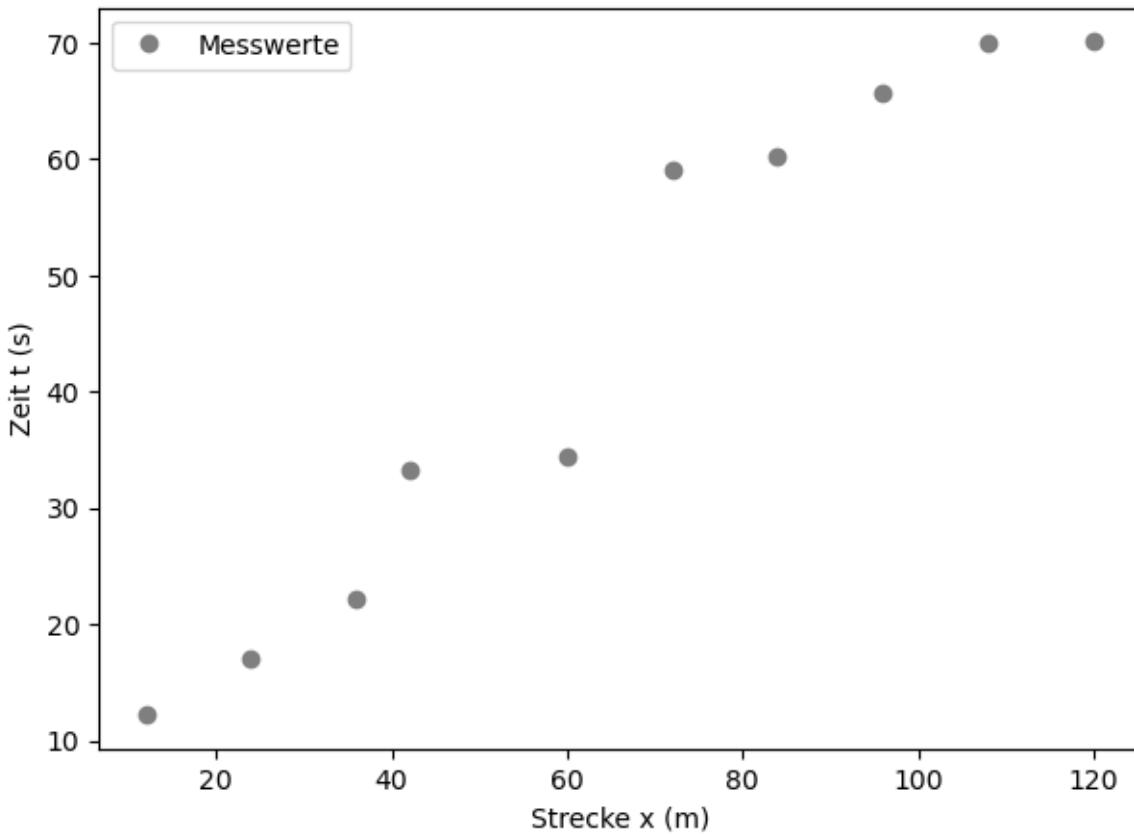
(continues on next page)

(continued from previous page)

```
t = [12.2, 17, 22.1, 33.2, 34.4, 59.1, 60.2, 65.7, 69.9, 70.1] # Messwerte der Zeit t
# in sek.

x = np.array(x) # konvertiere die Messwerte in ein Numpy-Array
t = np.array(t) # konvertiere die Messwerte in ein Numpy-Array

plt.plot(x,t, 'o', label = 'Messwerte', ms=6, color="tab:gray")
plt.xlabel('Strecke x (m)')
plt.ylabel('Zeit t (s)')
plt.legend()
plt.show()
```



10.3.2 Analytische Lineare Regression

Weitere Information findest du [hier](#) im Vorlesungsskript.

```
m = (np.mean(x*t) - np.mean(x)*np.mean(t)) / (np.mean(x**2) - np.mean(x)**2)
b = np.mean(t) - m * np.mean(x)
print('Die Steigung ist \t\t m = %5.4f s/m' %(m))
print('Der Ordinatenabschnitt ist \t b = %5.4f s' %(b))

plt.plot(x,t, 'o', label = 'Messwerte', ms=6, color="tab:gray")
plt.plot(x,m*x+b, lw=3, color="tab:red", label = 'analytische Lin. Reg.: y = %5.3fx+' +
        '%5.3f' %(m,b)) # plot Ausgleichsgerade mit m und b
plt.xlabel('Strecke x (m)')
```

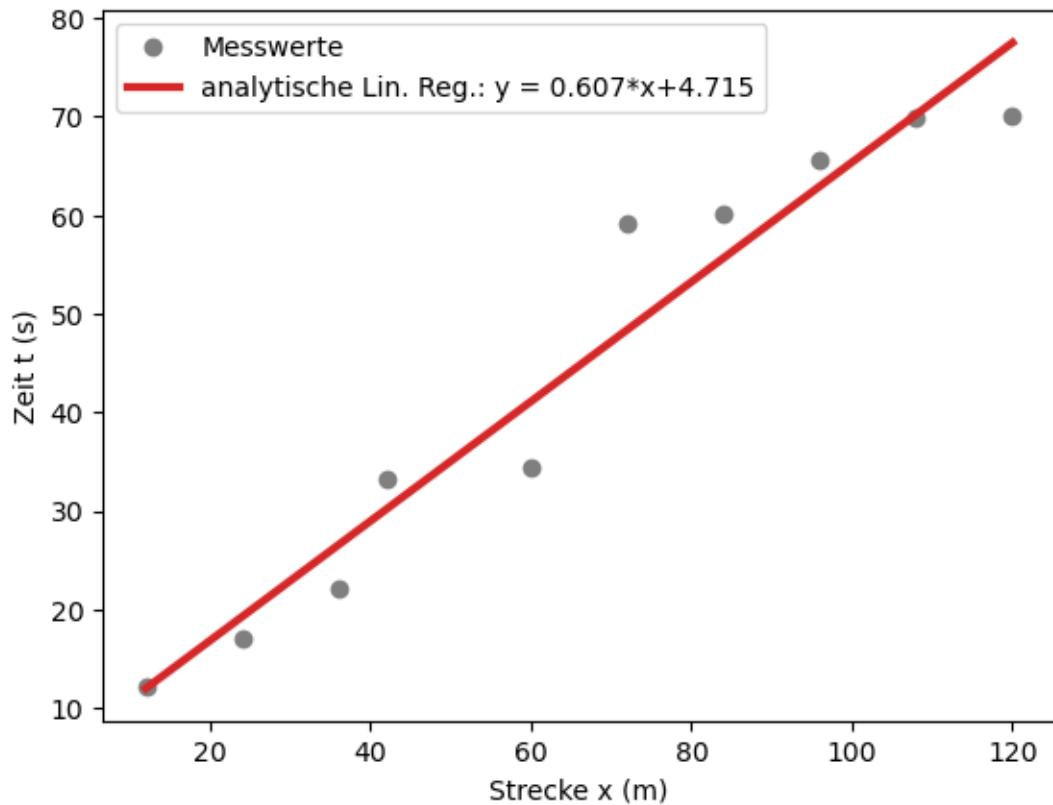
(continues on next page)

(continued from previous page)

```
plt.ylabel('Zeit t (s)')
plt.legend()
plt.show()
```

Die Steigung ist
Der Ordinatenabschnitt ist

$m = 0.6067 \text{ s/m}$
 $b = 4.7148 \text{ s}$



10.3.3 Lineare Regression mit Python: scipy

Mit einem Datenanalyseprogramm, sei es hier in Python oder in Matlab oder Maple, können Hilfsfunktionen und Pakete geladen werden, um Fit-Funktionen automatisch auf Messwerte anzupassen. Dies ist in der Regel schneller und insbesondere auch für andere Zusammenhänge, neben linearen Zusammenhängen, viel einfacher in der Umsetzung. Man sollte jedoch stets das Ergebnis kontrollieren, z.B. anhand von grafischen Darstellungen, und sich die Unsicherheiten genau ansehen. Des Weiteren können in der Fitroutine Startparameter angegeben werden, welche den Schätzwerten schon recht nah sein sollten. Das Fitergebnis kann nämlich unter Umständen sehr stark von der Wahl der Startparameter abhängen, gerade wenn es um komplexere Fitfunktionen geht. Im Folgenden wollen wir uns aber noch einmal das Beispiels des Karussells ansehen, um die beiden Methoden miteinander zu vergleichen.

Als erstes definieren wir uns eine allgemein lineare Fitfunktion und anschließend nutzen wir das `scipy`-Paket und rufen die Optimierungsfunktion auf, um unsere Messwerte zu modellieren. Die Ausgangsparameter werden in einem weiteren Array abgespeichert, welches einerseits die gesuchten Variablen m und b beinhaltet, andererseits die Kovarianzmatrix enthält. Aus der Wurzel die Diagonalelemente dieser Kovarianzmatrix erhält man die Standardabweichung der bestimmten Variablen.

```

from scipy.optimize import curve_fit

def fit_lin(x, b, a): # Fit Funktion für eine Gerade mit Steigung b
    return b*x + a

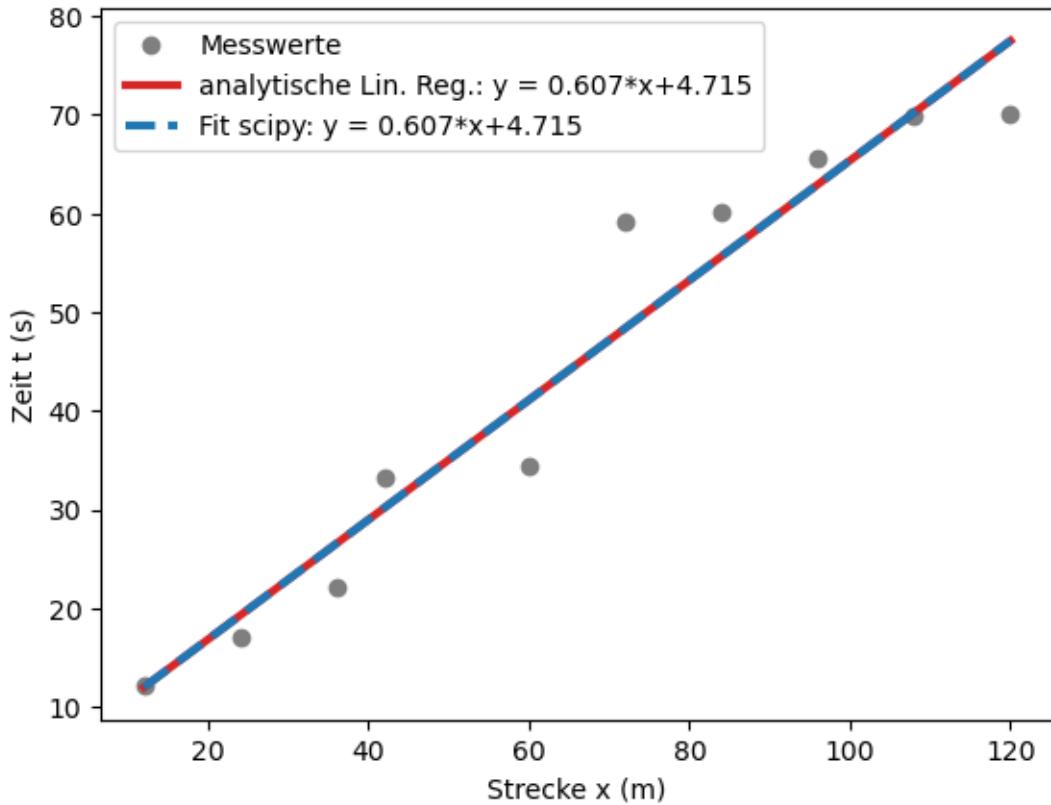
# Fit der Daten:
fit_out = curve_fit(fit_lin, x, t)

print('Die scipy-Fitparameter lauten:')
print('Die Steigung ist \t\t m = (%.4f +- %.4f) s/m' %(fit_out[0][0], np.sqrt(fit_
    _out[1][0][0])))
print('Der Ordinatenabschnitt ist \t b = (%.4f +- %.4f) s' %(fit_out[0][1], np.
    sqrt(fit_out[1][1][1])))

plt.plot(x,t, 'o', label = 'Messwerte', ms=6, color="tab:gray")
plt.plot(x,m*x+b, lw=3, color="tab:red", label = 'analytische Lin. Reg.: y = %.3f*x+_
    %.3f' %(m,b)) # plot Ausgleichsgerade mit m und b
plt.plot(x,fit_out[0][0]*x+fit_out[0][1], '--', lw=3, color="tab:blue", label = 'Fit_
    -scipy: y = %.3f*x+%.3f' %(fit_out[0][0],fit_out[0][1])) # plot Fitfunktion mit_
    -b und a
plt.xlabel('Strecke x (m)')
plt.ylabel('Zeit t (s)')
plt.legend()
plt.show()

```

Die scipy-Fitparameter lauten:
 Die Steigung ist $m = (0.6067 \pm 0.0534) \text{ s/m}$
 Der Ordinatenabschnitt ist $b = (4.7148 \pm 3.9581) \text{ s}$



Man erkennt, dass die Regressionskoeffizienten der beiden Methoden sehr gut übereinstimmen. Die Unterschiede betragen lediglich:

```
print('Unterschied in den Steigungen: \t\t %5.3e' % (m-fit_out[0][0]))
print('Unterschied in den Ordinatenabschnitten: %5.3e' % (b-fit_out[0][1]))
```

```
Unterschied in den Steigungen: 8.582e-13
Unterschied in den Ordinatenabschnitten: -8.125e-12
```

10.3.4 Lineare Regression mit Python: lmfit

Prinzipiell ist es wundervoll für Python für die Optimierung von Problemen zu nutzen und das scipy-Paket bietet sehr robuste und einfach Nutzung. Andererseits unterscheidet sich das Verfahren nicht besonders von denjenigen, die wir in C oder Fortran bekommen würden. D.h. es gibt einige Herausforderungen in der praktischen Anwendung, warum scipy nicht die optimale Wahl ist:

- Der Benutzer muss sich die Reihenfolge der Variablen und deren Bedeutung merken. (Was war noch mal `fit_out[0][0]`, die Steigung oder der Ordinatenabschnitt?)
- Wenn ich eine Variable nicht fitten möchte, also ich möchte zum Beispiel die Steigung auf fest Null setzen, so muss ich mir eine neue Funktion ohne diese Variable definieren. Insbesondere für komplexere Funktionen und Zusammenhänge wird dieses Verfahren schnell unübersichtlich. Einfacher wäre es, der Optimierungsfunktion zu sagen, welche Variablen ich in diesem Durchlauf gefittet haben möchte, und welche konstant bleiben sollen.
- Zwischen den verschiedenen Variablen können keine Beziehungen hergestellt werden. Es können lediglich Grenzen angegeben werden, in denen sich die Variablen befinden.

Der Grund für die eingeschränkte Nutzung hat historische Gründe und basiert hauptsächlich auf Fit-Routinen, die beispielsweise für C oder Fortran entwickelt wurden. Python bietet aber mehr, da es unterschiedlich Objekte und Datenstrukturen bietet, statt nur mit Arrays zu rechnen. Daher möchten wir uns noch einem etwas besser geeignetem Paket widmen, und das eben vorgestellt Beispiel noch einmal durchrechnen.

Das Paket lmfit definiert und benutzt Parameterobjekte anstelle von einfachen Zahlen als Variablen. Dies hat folgende Vorteile:

- Wir müssen uns endlich keine Reihenfolgen von Variablen mehr merken, wir können ihn ab sofort bedeutungsvolle Namen geben.
- Es müssen keine Arrays mehr für Grenzen eingegeben werden und die Reihenfolge der Grenzen, im Array, die vorher durch die Reihenfolge Variablen bestimmt war, spielt nun auch keine Rolle mehr.
- Wir können einfach Parameter fixieren ohne Zielfunktionen neu definieren zu müssen.
- Algebraische Beschränkungen der Parameter sind ab sofort möglich.

Eine ausführliche Dokumentation findet man hier: <https://lmfit.github.io/lmfit-py/index.html>

```
from lmfit import minimize, Parameters
from lmfit import Model

def f_lin(x, steigung, abschnitt):
    return steigung*x + abschnitt

model = Model(f_lin)
params = model.make_params(steigung=0.7, abschnitt=10.0)
result = model.fit(t, params, x=x)

print('lmfit hat eine eigene print-Ausgabe:')
result.params.pretty_print()

print()
print('Fit Report:')
print(result.fit_report())

plt.plot(x,t, 'o', label = 'Messwerte', ms=6, color="tab:gray")
plt.plot(x,m*x+b, lw=3, color="tab:red", label = 'analytische Lin. Reg.: y = %5.3f*x+ %5.3f' %(m,b)) # plot Ausgleichsgerade mit m und b
plt.plot(x,fit_out[0][0]*x+fit_out[0][1], '--', lw=3, color="tab:blue", label = 'Fit-scipy: y = %5.3f*x+%5.3f' %(fit_out[0][0],fit_out[0][1])) # plot Fitfunktion mit b und a
plt.plot(x,result.best_fit, ':', lw=3, color="tab:green", label = 'Fit lmfit: y = %5.3f*x+%5.3f' %(result.params['steigung'].value,result.params['abschnitt'].value))
plt.xlabel('Strecke x (m)')
plt.ylabel('Zeit t (s)')
plt.legend()
plt.show()
```

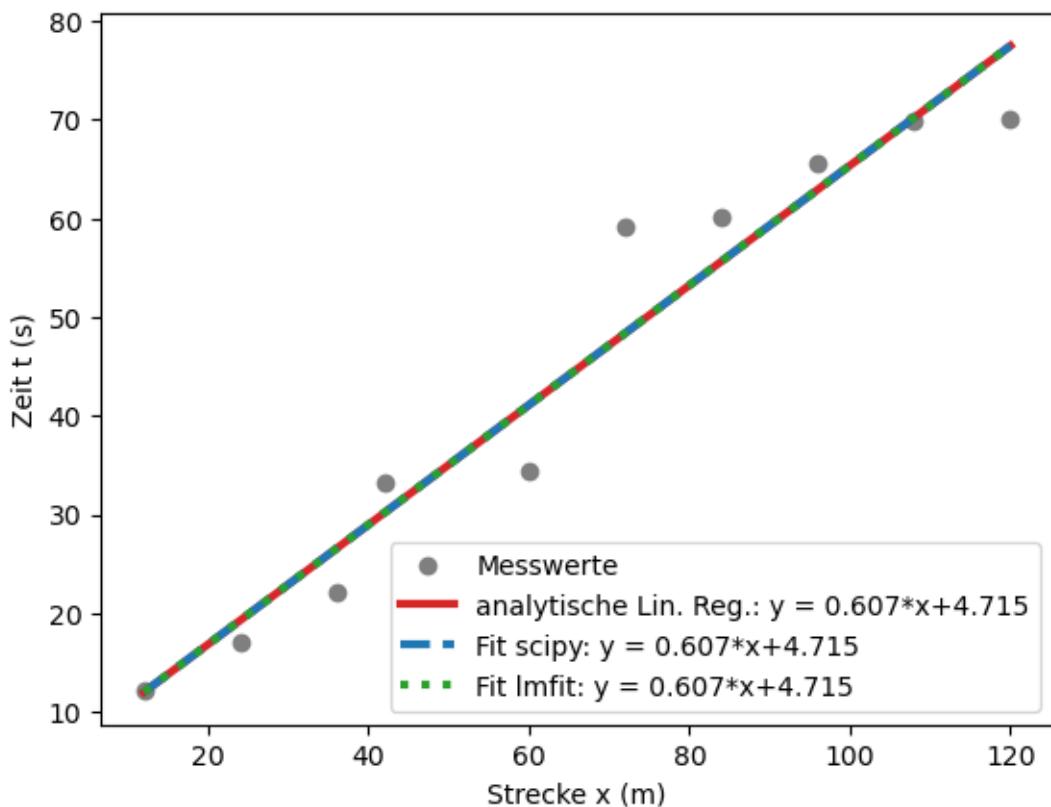
```
lmfit hat eine eigene print-Ausgabe:
Name      Value      Min      Max      Stderr      Vary      Expr  Brute_Step
abschnitt  4.715     -inf      inf     3.958     True      None   None
steigung    0.6067    -inf      inf    0.05342    True      None   None

Fit Report:
[[Model]]
    Model(f_lin)
```

(continues on next page)

(continued from previous page)

```
[[Fit Statistics]]
# fitting method = leastsq
# function evals = 6
# data points = 10
# variables = 2
chi-square = 276.888038
reduced chi-square = 34.6110047
Akaike info crit = 37.2102814
Bayesian info crit = 37.8154516
[[Variables]]
steigung: 0.60665381 +/- 0.05342018 (8.81%) (init = 0.7)
abschnitt: 4.71484116 +/- 3.95814322 (83.95%) (init = 10)
[[Correlations]] (unreported correlations are < 0.100)
C(steigung, abschnitt) = -0.883
```



Anhand der ausführlichen Ausgabe erkennt man sofort, wie viele Informationen man aus der Benutzung dieses Paketes ziehen kann, ohne aufwendige Fehlerfortpflanzung betreiben zu müssen. Außerdem kann man Beispielsweise die Fit-Methode ändern (anstelle von least-squares), man kann sich die Anfangs-Fitfunktion ansehen (und parallel zur 'besten' Fit-Funktion anzeigen lassen), man kann Parameter konstant setzen oder sich Standardfehler, χ^2 und Korrelationen (zwischen den Fit-Parametern, nicht den Messwerten!) anzeigen lassen.

10.4 Lineare Regression mit Berücksichtigung von Fehlerbalken

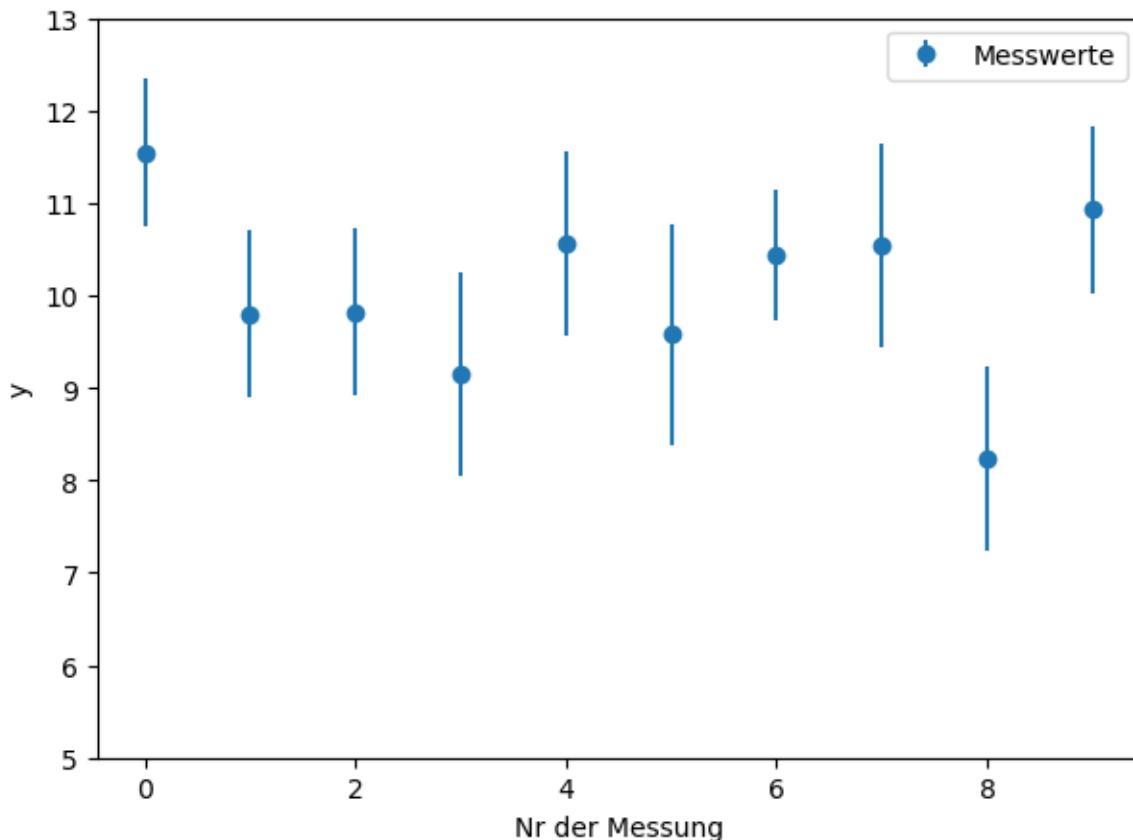
Gegeben sei eine Messreihe von 10 Messwerten mit annähernd konstanten Verhalten. In diesem sollen nun Kurvenanpassungen vorgenommen werden, unter Berücksichtigung von Fehlerbalken. Folgende Messdaten inkl. Unsicherheiten wurden aufgenommen. Die Unsicherheit der x -Achse vernachlässigen wir hier der Einfachheit halber.

```
import numpy as np
import matplotlib.pyplot as plt
# Matplotlib Settings:
plt.style.use('default') # Matplotlib Style wählen
plt.figure(figsize=(7,5)) # Plot-Größe
plt.rcParams['font.size'] = 10; # Schriftgröße

y = [11.55, 9.8, 9.82, 9.15, 10.57, 9.58, 10.44, 10.55, 8.23, 10.93] #Messwerte y_i
s_y = [0.8, 0.9, 0.9, 1.1, 1.0, 1.2, 0.7, 1.1, 1.0, 0.9] #Unsicherheiten Fall 1
x = [0., 1., 2., 3., 4., 5., 6., 7., 8., 9.] #Messwerte y_i

y = np.array(y) #konvertiere die Messwerte in ein Numpy-Array
x = np.array(x) #konvertiere die Messwerte in ein Numpy-Array
s_y = np.array(s_y) #konvertiere die Unsicherheiten in ein Numpy-Array

plt.errorbar(x,y, fmt='o', xerr = None, yerr = s_y, label = 'Messwerte')
plt.xlabel('Nr der Messung')
plt.ylabel('y')
plt.ylim([5,13])
plt.legend()
plt.show()
```



10.4.1 Der Einfluss von Fehlerbalken auf den Fit

Im Allgemeinen gilt, je kleiner die Unsicherheit eines Messwertes ist, desto wichtiger ist dieser Messwert für die Mittelwertbildung. Die analytische Lösung würde wiefolgt aussehen:

$$\begin{aligned} \bullet \quad S(c) &= \sum_{i=1}^N \frac{(y_i - c)^2}{s_i^2} \\ \bullet \quad 0 = \frac{dS}{dc} &= \sum_{i=1}^N \frac{2(y_i - c)}{s_i^2} \\ \Rightarrow \hat{c} &= \frac{1}{\sum 1/s_i^2} \sum_{i=1}^N \frac{1}{s_i^2} y_i \end{aligned}$$

Mit der Definition $w_i \equiv \frac{1}{s_i^2}$ erhalten wir als wichtiges Ergebnis die Formel für den *gewichteten Mittelwert*:

$$\hat{c} = \frac{1}{\sum w_i} \sum_{i=1}^N w_i y_i$$

Der Mittelwert ist die mit $1/s_i^2$ gewichtete Summe der Einzelmessungen. Die Gewichte entsprechen der obigen Erwartung: Messwerte mit den kleinsten s_i bekommen bei der Mittelwertbildung das größte Gewicht. Am Vorfaktor $1/\sum w_i$ kann man ablesen, dass die Zahl der Messwerte N in diesem Fall durch die Summe der Gewichte $\sum w_i$ ersetzt werden muss.

Ist der Fehler für alle Messwerte konstant, so kann s_i einfach auf die andere Seite multipliziert werden und verschwindet aus der Gleichung. Ist der Fehler Null, nutzen wir die Gütfunktion, welche minimiert werden muss:

$$\bullet \quad Q(c) := \sum_{i=1}^n (y_i - f(x))^2 = \sum_{i=1}^n (y_i - c)^2 = \min?$$

```
def S(y,c): # Minimierungsfunktion mit Fehler
    return ((y-c)**2 / s_y).sum()

def Q(y,c): # Minimierungsfehler ohne Fehler
    return ((y-c)**2).sum()
```

Die Minimierung kann einfach ausgeführt werden, indem die Gütfunktion für verschiedene Funktionsparameter (c) ausprobiert wird, im Folgenden werden für c 100 Werte zwischen 8 und 12 ausprobiert:

```
c_val = np.linspace(8,12,100)
S_c = []
Q_c = []

for c in c_val:
    S_c.append(S(y,c))
    Q_c.append(Q(y,c))

id_min_S = np.argmin(S_c)
id_min_Q = np.argmin(Q_c)

print("Minimum unter Einbezug von Unsicherheiten ist bei c =", c_val[id_min_S])
print("Minimum ohne Einbezug von Unsicherheiten ist bei c =", c_val[id_min_Q])

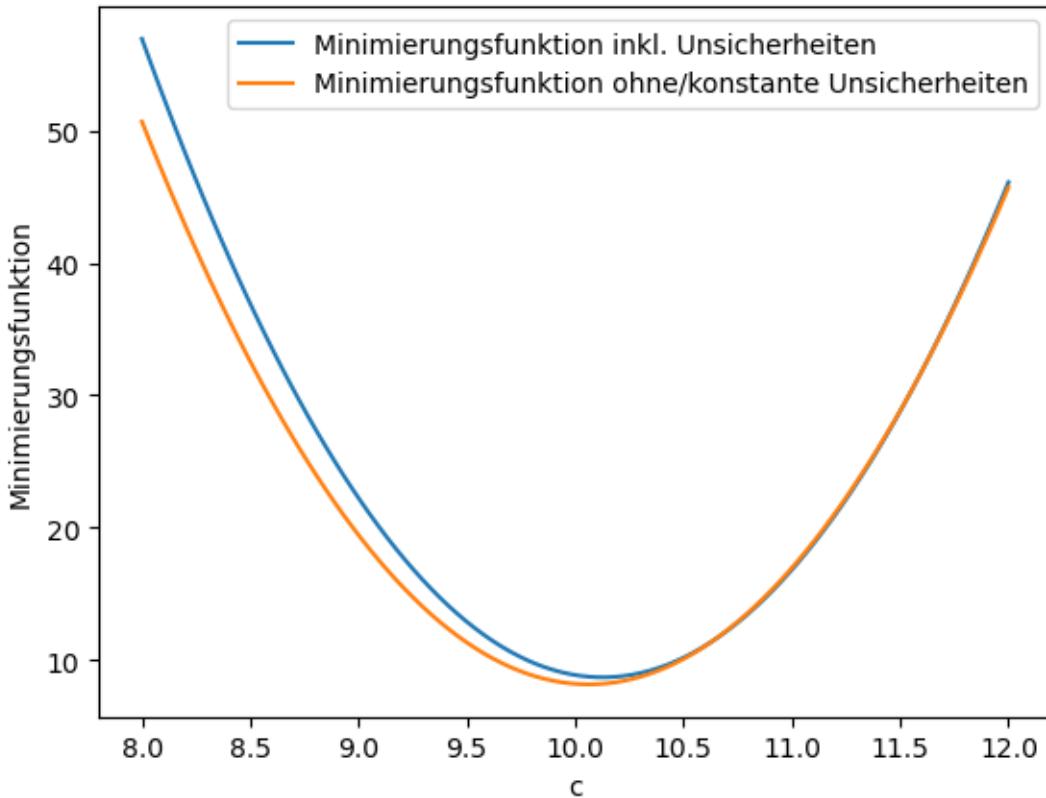
plt.figure()
plt.plot(c_val, S_c, label='Minimierungsfunktion inkl. Unsicherheiten')
plt.plot(c_val, Q_c, label = 'Minimierungsfunktion ohne/konstante Unsicherheiten')
plt.xlabel('c')
```

(continues on next page)

(continued from previous page)

```
plt.ylabel('Minimierungsfunktion')
plt.legend()
plt.show()
```

Minimum unter Einbezug von Unsicherheiten ist bei $c = 10.141414141414142$
 Minimum ohne Einbezug von Unsicherheiten ist bei $c = 10.06060606060606$



Schon bei dieser schnellen Analyse sehen wir, dass es einen Unterschied gibt, ob wir die Fehlerbalken miteinbeziehen, oder ob keine Unsicherheiten vorhanden sind. Sind die absoluten Unsicherheiten für alle Messwerte die gleichen, so trifft der Fall *ohne* Unsicherheiten zu.

10.4.2 Kurvenanpassung mit Python: `scipy`

Im Folgenden Code-Block wollen wir die Analyse dieser einfachen Messreihe noch einmal mittel `scipy`-Paket wiederholen und eine lineare Regression und konstante Regression auf die Messdaten anwenden. Wir definieren also zwei Fit-Funktionen:

```
def fit_lin(x, b, a): # Funktion für lineare Regression
    return b*x + a

def fit_const(x, a): # Funktion für konstante Regression
    return a + x - x

def f(x, c_val): # Funktion um Minimierungsfunktion in Diagramm zu zeichnen
    return c_val + x-x
```

Diese beiden Funktionen werden nun benutzt, um die Daten zu modellieren. Wir testen jeweils beide Fälle, nämlich mit und ohne Fehlerbalken:

```
from scipy.optimize import curve_fit
# ----- Mit Fehlerbalken: ----- #

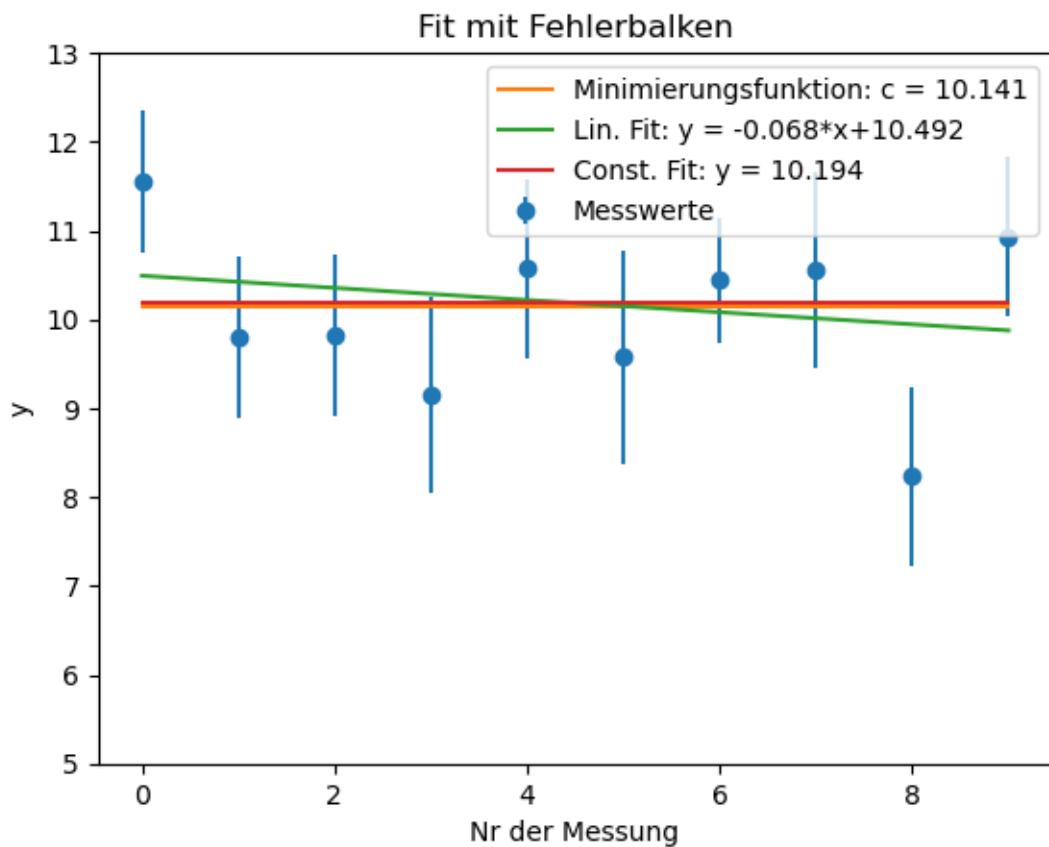
fit_lin_out_err = curve_fit(fit_lin, x, y, sigma=s_y)
fit_const_out_err = curve_fit(fit_const, x, y, sigma=s_y)

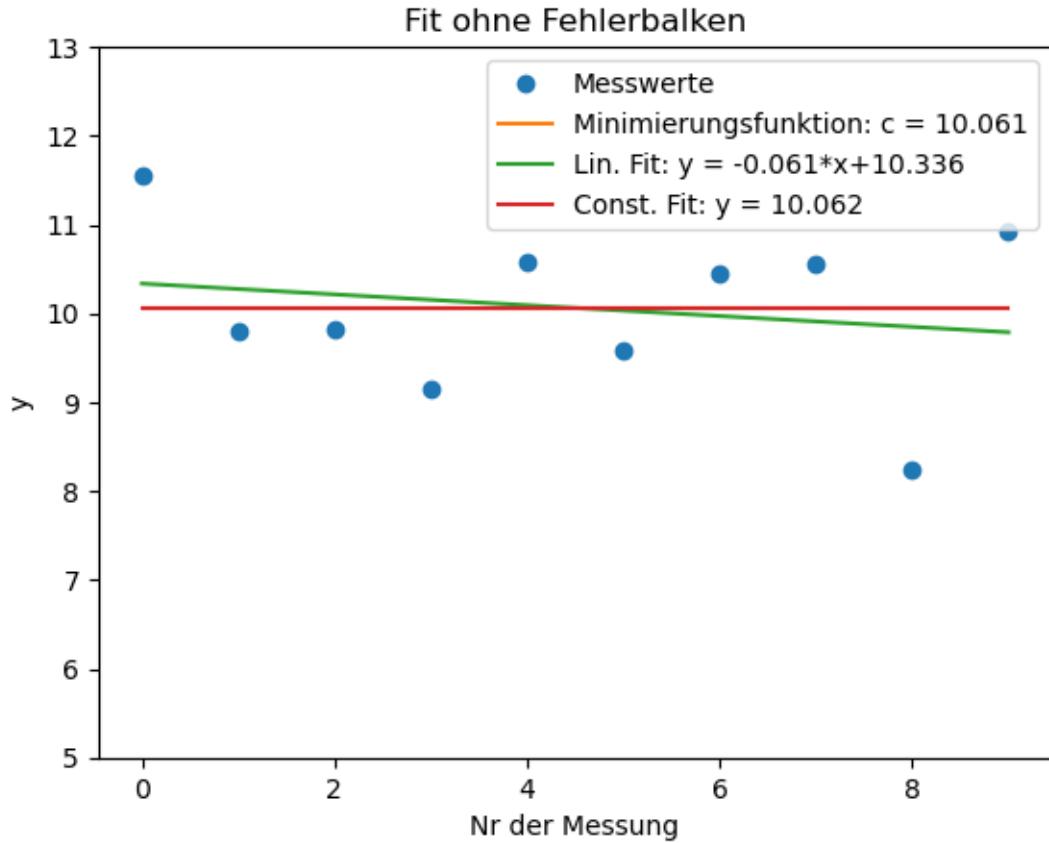
plt.errorbar(x,y, fmt='o', xerr = None, yerr = s_y, label = 'Messwerte')
plt.plot(x,f(x, c_val[id_min_S]), label = 'Minimierungsfunktion: c = %5.3f'%(c_
    →val[id_min_S]))
plt.plot(x,fit_lin(x, fit_lin_out_err[0][0], fit_lin_out_err[0][1]), label = 'Lin. u_
    →Fit: y = %5.3f*x+%5.3f'%(fit_lin_out_err[0][0], fit_lin_out_err[0][1]))
plt.plot(x,fit_const(x, fit_const_out_err[0][0]), label = 'Const. Fit: y = %5.3f'
    →%(fit_const_out_err[0][0]))
plt.xlabel('Nr der Messung')
plt.ylabel('y')
plt.ylim([5,13])
plt.legend()
plt.title('Fit mit Fehlerbalken')
plt.show()

# ----- Ohne Fehlerbalken: ----- #

fit_lin_out = curve_fit(fit_lin, x, y)
fit_const_out = curve_fit(fit_const, x, y)

plt.plot(x,y, 'o', label = 'Messwerte')
plt.plot(x,f(x, c_val[id_min_Q]), label = 'Minimierungsfunktion: c = %5.3f'%(c_
    →val[id_min_Q]))
plt.plot(x,fit_lin(x, fit_lin_out[0][0], fit_lin_out[0][1]), label = 'Lin. Fit: y =
    →%5.3f*x+%5.3f'%(fit_lin_out[0][0], fit_lin_out[0][1]))
plt.plot(x,fit_const(x, fit_const_out[0][0]), label = 'Const. Fit: y = %5.3f'%(fit__
    →const_out[0][0]))
plt.xlabel('Nr der Messung')
plt.ylabel('y')
plt.ylim([5,13])
plt.legend()
plt.title('Fit ohne Fehlerbalken')
plt.show()
```





10.4.3 Kurvenanpassung mit Python: lmfit

Das vorangegangene Beispiel mit scipy zeigt auf, wie aufwändig es ist eine Fitparameter zu fixieren. Es muss eine neue Funktion mit weniger Freiheitsgraden definiert werden. Wie oben schon angekündigt, lässt sich dies mit dem lmfit-Paket etwas einfacher lösen und soll hier anhand des Beispiels noch einmal visualisiert werden.

```
from lmfit import minimize, Parameters
from lmfit import Model

# ----- Mit Fehlerbalken: ---- #

def f_lin(x, steigung, abschnitt):
    return steigung*x + abschnitt

model = Model(f_lin)
params = model.make_params(steigung=0.0, abschnitt=10.0)
result_lin = model.fit(y, params, x=x, weights = 1/s_y) # Fehlerbalken werden über_
# die 'weights = 1/sigma' Option berücksichtigt

params['steigung'].set(0.0, vary=False)
result_const = model.fit(y, params, x=x, weights = 1/s_y)

#print('Fit Parameter der Linearen Regression:')
result_lin.params.pretty_print()
#print('Fit Report:')
```

(continues on next page)

(continued from previous page)

```

print(result_lin.fit_report())

#print('Fit Parameter der Konstanten Regression:')
#result_const.params.pretty_print()
#print('Fit Report:')
#print(result_const.fit_report())

plt.errorbar(x,y, fmt='o', xerr = None, yerr = s_y, label = 'Messwerte')
plt.plot(x,f(x, c_val[id_min_S]), label = 'Minimierungsfunktion: c = %5.3f'%(c_
    _val[id_min_S]))
plt.plot(x,fit_lin(x, fit_lin_out_err[0][0], fit_lin_out_err[0][1]), label = 'Lin.-
    →Fit: y = %5.3f*x+%5.3f'%(fit_lin_out_err[0][0], fit_lin_out_err[0][1]))
plt.plot(x,fit_const(x, fit_const_out_err[0][0]), label = 'Const. Fit: y = %5.3f'-
    →%(fit_const_out_err[0][0]))
plt.plot(x,result_lin.best_fit, '--',label = 'lmfit Lin-Fit: y = %5.3f*x+%5.3f'-
    →%(result_lin.params['steigung'].value,result_lin.params['abschnitt'].value)) #_
    →plot Fitfunktion mit b und a
plt.plot(x,result_const.best_fit, '--',label = 'lmfit Const-Fit: y = %5.3f*x+%5.3f'-
    →%(result_const.params['steigung'].value,result_const.params['abschnitt'].value)) #_
    →plot Fitfunktion mit b und a
plt.xlabel('Nr der Messung')
plt.ylabel('y')
plt.ylim([5,13])
plt.legend()
plt.title('Fit mit Fehlerbalken')
plt.show()

# ----- Ohne Fehlerbalken: ---- #

def f_lin(x, steigung, abschnitt):
    return steigung*x + abschnitt

model = Model(f_lin)
params = model.make_params(steigung=0.0, abschnitt=10.0)
result_lin = model.fit(y, params, x=x)

params['steigung'].set(0.0, vary=False)
result_const = model.fit(y, params, x=x)

#print('Fit Parameter der Linearen Regression:')
result_lin.params.pretty_print()
#print('Fit Report:')
print(result_lin.fit_report())

#print('Fit Parameter der Konstanten Regression:')
result_const.params.pretty_print()
#print('Fit Report:')
print(result_const.fit_report())

plt.plot(x,y, 'o', label = 'Messwerte')
plt.plot(x,f(x, c_val[id_min_Q]), label = 'Minimierungsfunktion: c = %5.3f'%(c_
    _val[id_min_Q]))
plt.plot(x,fit_lin(x, fit_lin_out[0][0], fit_lin_out[0][1]), label = 'scipy Lin. Fit:-
    →y = %5.3f*x+%5.3f'%(fit_lin_out[0][0], fit_lin_out[0][1]))
plt.plot(x,fit_const(x, fit_const_out[0][0]), label = 'scipy const. Fit: y = %5.3f'-
    →%(fit_const_out[0][0]))

```

(continues on next page)

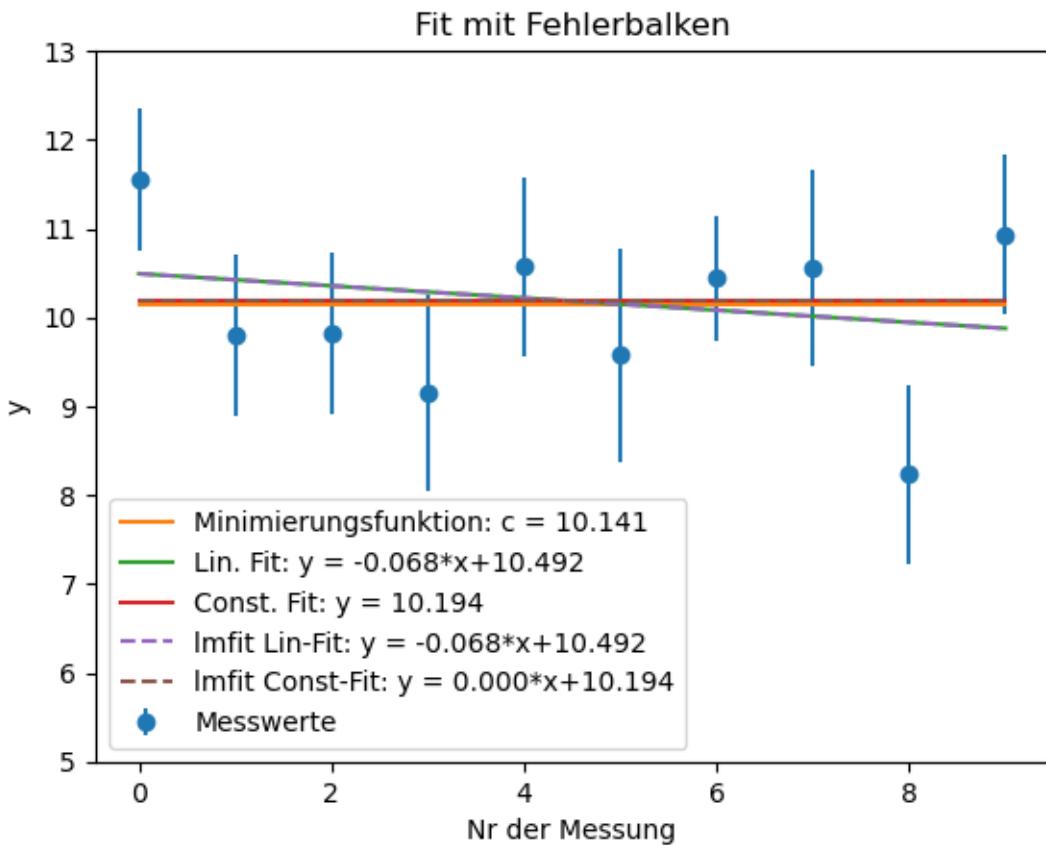
(continued from previous page)

```

plt.plot(x,result_lin.best_fit, '--',label = 'lmfit Lin-Fit: y = %5.3f*x+%5.3f'
          %(result_lin.params['steigung'].value,result_lin.params['abschnitt'].value)) #_
          ↪plot Fitfunktion mit b und a
plt.plot(x,result_const.best_fit, '--',label = 'lmfit Const-Fit: y = %5.3f*x+%5.3f'
          %(result_const.params['steigung'].value,result_const.params['abschnitt'].value)) #_
          ↪plot Fitfunktion mit b und a
plt.xlabel('Nr der Messung')
plt.ylabel('y')
plt.ylim([5,13])
plt.legend()
plt.title('Fit ohne Fehlerbalken')
plt.show()

```

Name	Value	Min	Max	Stderr	Vary	Expr	Brute_Step
abschnitt	10.49	-inf	inf	0.5446	True	None	None
steigung	-0.06839	-inf	inf	0.1031	True	None	None
[[Model]]							
Model(f_lin)							
[[Fit Statistics]]							
# fitting method = leastsq							
# function evals = 6							
# data points = 10							
# variables = 2							
chi-square = 8.81129304							
reduced chi-square = 1.10141163							
Akaike info crit = 2.73449106							
Bayesian info crit = 3.33966124							
[[Variables]]							
steigung: -0.06839389 +/- 0.10313985 (150.80%) (init = 0)							
abschnitt: 10.4919542 +/- 0.54460553 (5.19%) (init = 10)							
[[Correlations]] (unreported correlations are < 0.100)							
C(steigung, abschnitt) = -0.825							



```

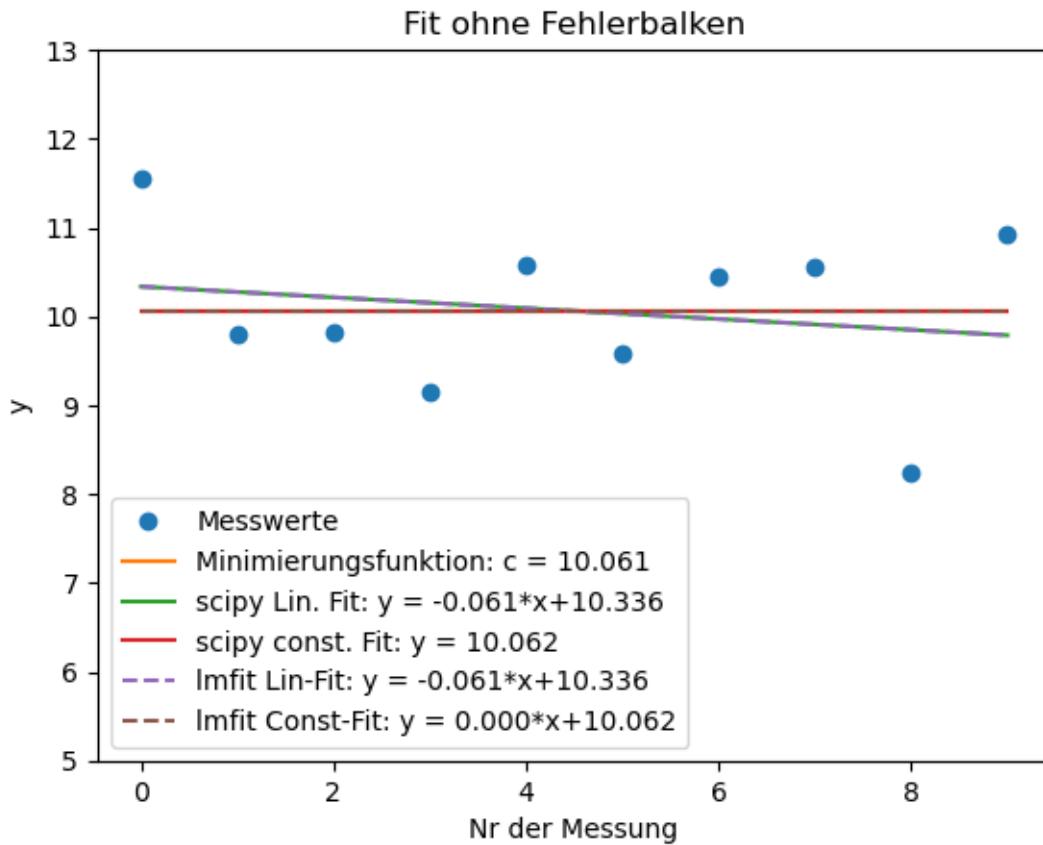
Name      Value      Min      Max      Stderr      Vary      Expr  Brute_Step
abschnitt 10.34     -inf     inf      0.5822    True      None   None
steigung  -0.06085  -inf     inf      0.1091    True      None   None
[[Model]]
  Model(f_lin)
[[Fit Statistics]]
  # fitting method = leastsq
  # function evals = 6
  # data points   = 10
  # variables     = 2
  chi-square      = 7.84870061
  reduced chi-square = 0.98108758
  Akaike info crit = 1.57762897
  Bayesian info crit = 2.18279916
[[Variables]]
  steigung: -0.06084848 +/- 0.10905032 (179.22%) (init = 0)
  abschnitt: 10.3358182 +/- 0.58216937 (5.63%) (init = 10)
[[Correlations]] (unreported correlations are < 0.100)
  C(steigung, abschnitt) = -0.843
Name      Value      Min      Max      Stderr      Vary      Expr  Brute_Step
abschnitt 10.06     -inf     inf      0.301     True      None   None
steigung      0       -inf     inf      0        False     None   None
[[Model]]
  Model(f_lin)
[[Fit Statistics]]
  # fitting method = leastsq
  # function evals = 4

```

(continues on next page)

(continued from previous page)

```
# data points      = 10
# variables       = 1
chi-square        = 8.15416000
reduced chi-square = 0.90601778
Akaike info crit   = -0.04056867
Bayesian info crit = 0.26201643
[[Variables]]
steigung: 0 (fixed)
abschnitt: 10.0620000 +/- 0.30100129 (2.99%) (init = 10)
```



10.5 Fourier-Transformation: Entfernung von Hintergrundrauschen in einem Audiosignal

Um das Verständnis für die Fourier-Transformation und ihre Möglichkeiten zu vertiefen, werden wir hier einige Audiosignale filtern. Zuerst wirst du ein Audiosignal mit einem hohen Brummton erzeugen und dann den Brummton mit Hilfe der Fourier-Transformation entfernen.

10.5.1 Signal erstellen

Sinuswellen werden manchmal als reine Töne bezeichnet, weil sie eine einzige Frequenz darstellen. Hier werden wir werden Sinuswellen verwenden, um das Audio zu erzeugen, da diese deutliche Spitzen im resultierenden Frequenzspektrum bilden. Ein weiterer Vorteil von Sinuswellen ist, dass sie mit 'NumPy' einfach zu erzeugen sind.

Als erstes laden wir einige Libraries und Einstellung:

```
# Noise Tone
#Benötigte Libraries:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import time
import warnings
warnings.filterwarnings('ignore')

# Matplotlib Settings:
plt.style.use('default') # Matplotlib Style wählen
plt.figure(figsize=(8,4)) # Plot-Größe
plt.rcParams['font.size'] = 10; # Schriftgröße
```

<Figure size 800x400 with 0 Axes>

Als Nächstes definieren wir eine Funktion zur Erzeugung einer Sinuswelle, da Sie diese später mehrfach verwenden werden. Die Funktion nimmt eine Frequenz, freq, und gibt dann die x- und y-Werte zurück, die wir für die Darstellung der Welle verwenden werden.

```
def generate_sine_wave(freq, sample_rate, duration):
    t = np.linspace(0, duration, sample_rate * duration, endpoint=False)
    frequencies = t * freq
    # 2pi because np.sin takes radians
    y = np.sin((2 * np.pi) * frequencies)
    return t, y
```

- SAMPLE_RATE bestimmt, wie viele Datenpunkte das Signal verwendet, um die Sinuswelle pro Sekunde darzustellen. Hätte das Signal also eine Abtastrate von 10 Hz und wäre eine Sinuswelle von fünf Sekunden, dann hätte es $10 * 5 = 50$ Datenpunkte.
- DURATION ist die Länge des erzeugten Samples.

Die x-Koordinaten der Sinuswelle liegen in gleichmäßigen Abständen zwischen 0 und DURATION, daher verwendet der Code NumPy's linspace(), um sie zu erzeugen. Er nimmt einen Startwert, einen Endwert und die Anzahl der zu erzeugenden Abtastwerte an. Die Einstellung endpoint=False ist wichtig, damit die Fourier-Transformation richtig funktioniert, da sie davon ausgeht, dass ein Signal periodisch ist.

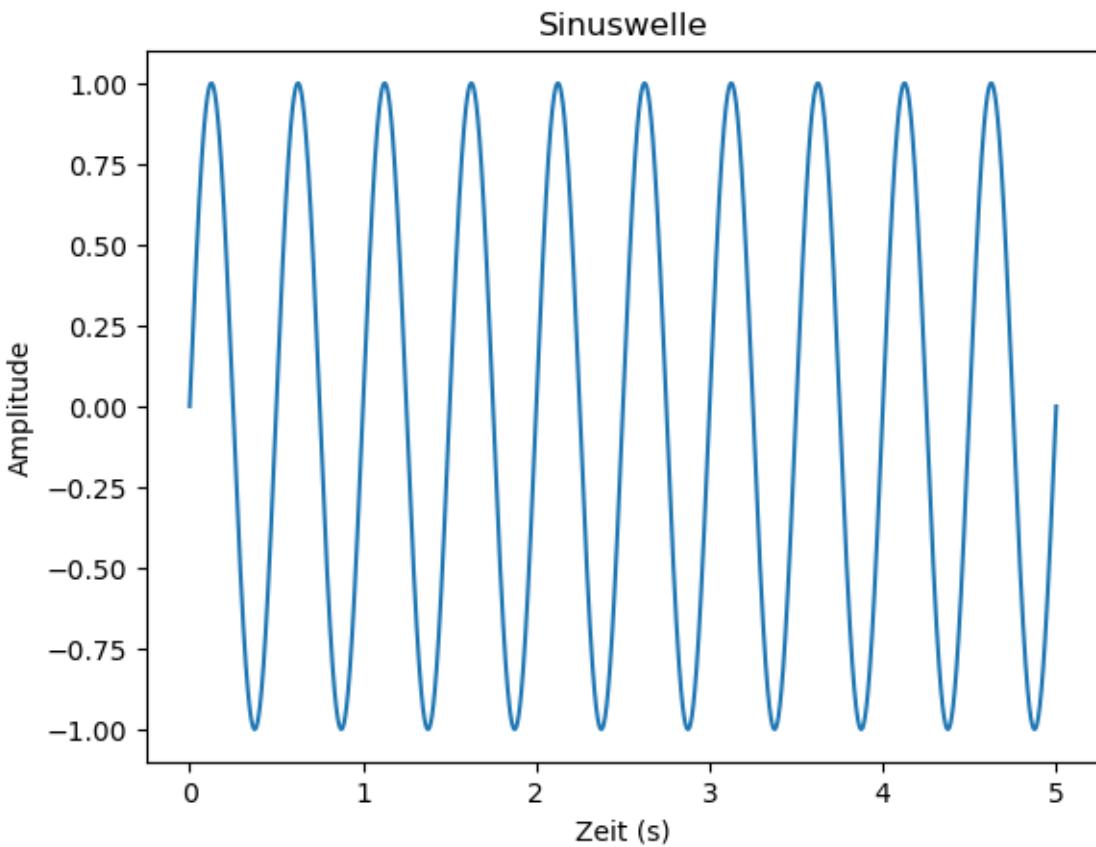
`np.sin()` berechnet die Werte der Sinusfunktion an jeder der x-Koordinaten. Das Ergebnis wird mit der Frequenz multipliziert, damit die Sinuswelle mit dieser Frequenz schwingt, und das Produkt wird mit 2π multipliziert, um die Eingabewerte in Bogenmaß umzurechnen.

Nachdem die Funktion definiert ist, verwenden wir sie, um eine Sinuswelle von zwei Hertz zu erzeugen, die fünf Sekunden lang dauert, und stellen sie mit Matplotlib dar. Deine Sinuskurve sollte etwa so aussehen:

```
SAMPLE_RATE = 44100 # Hertz
DURATION = 5 # Seconds

# Generate a 2 hertz sine wave that lasts for 5 seconds
x, y = generate_sine_wave(2, SAMPLE_RATE, DURATION)

plt.plot(x,y, 'tab:blue')
plt.xlabel('Zeit (s)')
plt.ylabel('Amplitude')
plt.title('Sinuswelle')
plt.show()
```



Die x-Achse stellt die Zeit in Sekunden dar, und da es zwei Spitzen für jede Sekunde gibt, kann man sehen, dass die Sinuswelle zweimal pro Sekunde schwingt. Diese Sinuswelle hat eine zu niedrige Frequenz, um hörbar zu sein. Im nächsten Abschnitt werden wir daher einige Sinuswellen mit höherer Frequenz erzeugen und sehen, wie Sie diese mischen können.

10.5.2 Audiosignale mischen

Die gute Nachricht ist, dass das Mischen von Audiosignalen aus nur zwei Schritten besteht:

- Addieren der Signale
- Normalisieren des Ergebnisses

Zunächst erzeugen wir die Signale:

```
_, nice_tone = generate_sine_wave(400, SAMPLE_RATE, DURATION)
_, noise_tone = generate_sine_wave(4000, SAMPLE_RATE, DURATION)
noise_tone = noise_tone * 0.3

mixed_tone = nice_tone + noise_tone
```

Dieses Codebeispiel ist nicht neu. Es erzeugt einen Ton mit mittlerer Tonhöhe und einen Ton mit hoher Tonhöhe, die den Variablen `nice_tone` bzw. `noise_tone` zugewiesen werden. Der hohe Ton wird als unerwünschtes Rauschen verwendet und daher mit 0,3 multipliziert, um seine Leistung zu verringern. Der Code addiert dann diese Töne zusammen. Beachte, dass wir den Unterstrich (`_`) verwenden, um die von `generate_sine_wave()` zurückgegebenen x-Werte zu verwerfen.

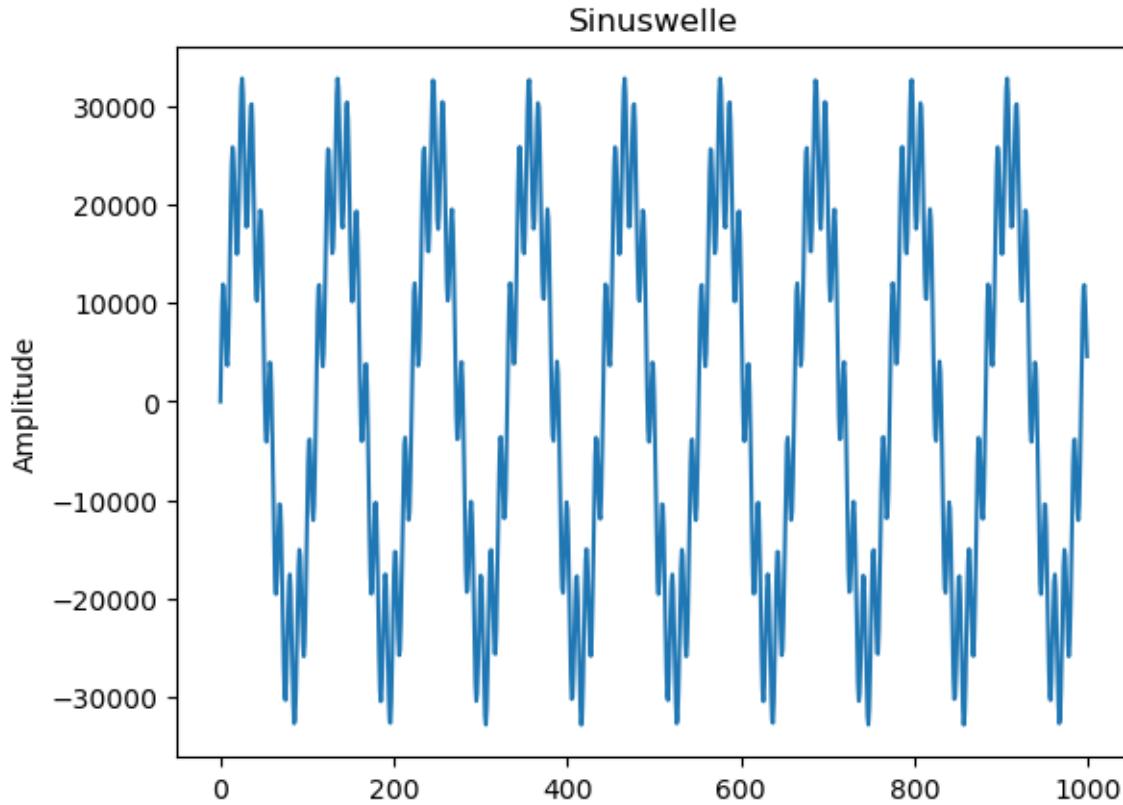
Der nächste Schritt ist die Normalisierung bzw. Skalierung des Signals, damit es in das Zielformat passt. Aufgrund der Art und Weise, wie das Audiosignal später gespeichert wird, ist das Zielformat eine 16-Bit-Ganzzahl, die einen Bereich von -32768 bis 32767 hat:

```
normalized_tone = np.int16((mixed_tone / mixed_tone.max()) * 32767)
```

Hier wird `mixed_tone` so skaliert, dass es genau in einen 16-Bit-Integer passt, und dann mit `np.int16` von NumPy in diesen Datentyp gecastet. Dividiert man `mixed_tone` durch seinen Maximalwert, wird es auf einen Wert zwischen -1 und 1 skaliert. Wird dieses Signal mit 32767 multipliziert, wird es zwischen -32767 und 32767 skaliert, was ungefähr dem Bereich von `np.int16` entspricht. Der Code zeichnet nur die ersten 1000 Abtastwerte auf, damit Sie die Struktur des Signals besser erkennen können.

Dein Diagramm sollte in etwa so aussehen:

```
plt.plot(normalized_tone[:1000], 'tab:blue')
plt.ylabel('Amplitude')
plt.title('Sinuswelle')
plt.show()
```



Das Signal sieht aus wie eine verzerrte Sinuswelle. Die Sinuswelle ist der 400-Hz-Ton, den wir erzeugt haben, und die Verzerrung ist der 4000-Hz-Ton. Wenn man genau hinsieht, kann man erkennen, dass die Verzerrung die Form einer Sinuswelle hat.

Um den Ton anzuhören, können wir ihn in einem Format speichern, das ein Audioplayer lesen kann. Am einfachsten ist es, die Methode `wavfile.write` von SciPy zu verwenden, um sie in einer WAV-Datei zu speichern. 16-Bit-Ganzzahlen sind ein Standard-Datentyp für WAV-Dateien, also normalisieren Sie Ihr Signal auf 16-Bit-Ganzzahlen:

```
from scipy.io.wavfile import write
import IPython.display as ipd
# Remember SAMPLE_RATE = 44100 Hz is our playback rate
write("meinsinus.wav", SAMPLE_RATE, normalized_tone)
ipd.Audio('meinsinus.wav') # load a local WAV file
```

<IPython.lib.display.Audio object>

Dieser Code schreibt in eine Datei `meinsinus.wav` in dem Verzeichnis, in dem wir das Python-Skript ausführen. Danach rufen wir die Datei auf, sodass wir das Audio direkt anhören können. Du wirst einen tieferen und einen höheren Ton hören. Das sind die 400-Hz- und 4000-Hz-Sinuswellen, die du gemischst hat.

Wenn du diesen Schritt abgeschlossen hast, hast du dein Audiobeispiel fertig. Der nächste Schritt ist das Entfernen des hohen Tons mithilfe der Fourier-Transformation!

10.5.3 Fast Fourier-Transformation (FFT)

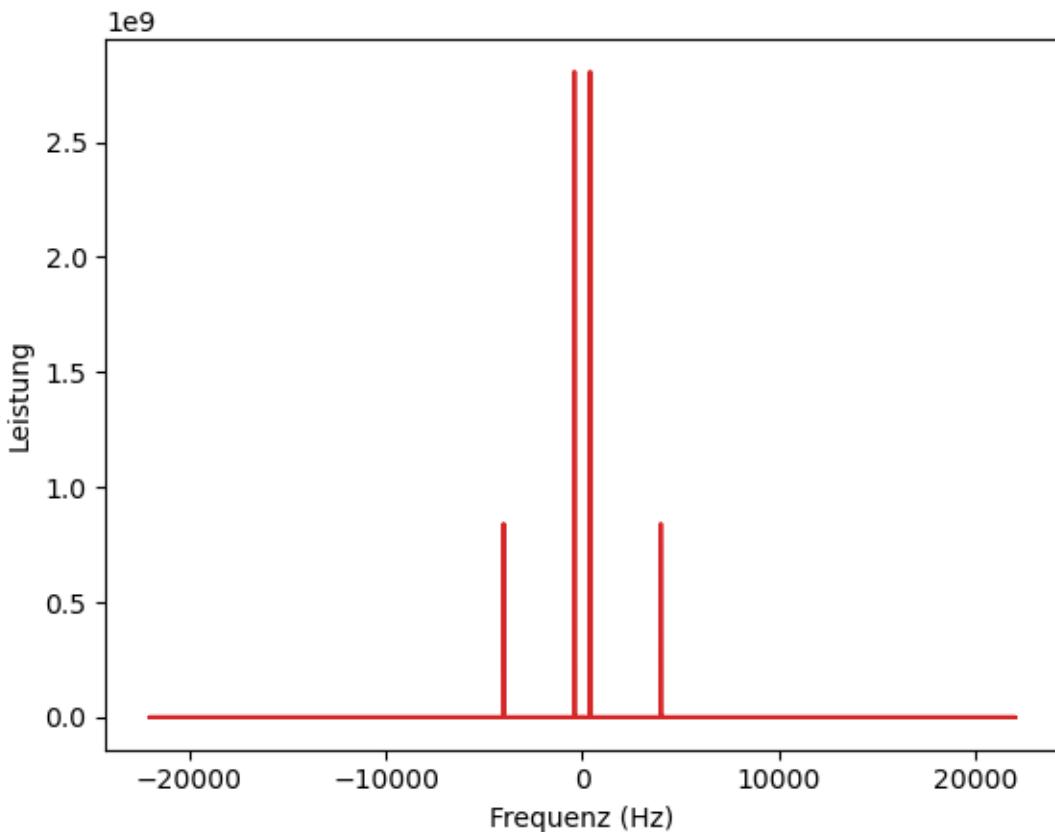
Die FFT ist ein Algorithmus, der die Fourier-Transformation implementiert und ein Frequenzspektrum für ein Signal im Zeitbereich, z. B. dein Audiosignal, berechnen kann:

```
from scipy.fft import fft, fftfreq

# Number of samples in normalized_tone
N = SAMPLE_RATE * DURATION

yf = fft(normalized_tone)
xf = fftfreq(N, 1 / SAMPLE_RATE)

plt.plot(xf, np.abs(yf), 'tab:red')
plt.xlabel('Frequenz (Hz)')
plt.ylabel('Leistung')
plt.show()
```



Man kann zwei Spitzen bei den positiven Frequenzen und Spiegelungen dieser Spitzen bei den negativen Frequenzen erkennen. Die Spitzen der positiven Frequenzen liegen bei 400 Hz und 4000 Hz, was den Frequenzen entspricht, die du in das Audiosignal eingegeben hast.

Die Fourier-Transformation hat Ihr kompliziertes, schwabbeliges Signal in die darin enthaltenen Frequenzen umgewandelt. Da du nur zwei Frequenzen hineingegeben hast, sind auch nur zwei Frequenzen herausgekommen. Die Negativ-Positiv-Symmetrie ist ein Nebeneffekt der Fourier-Transformation bei der Eingabe von reellen Werten, aber dazu später mehr.

In den ersten paar Zeilen importieren wir die Funktionen aus `scipy.fft`, die du später verwenden wirst, und definieren

eine Variable, N , die die Gesamtzahl der Abtastwerte im Signal speichert.

Der Code ruft zwei wichtige Funktionen auf:

- `fft()`: berechnet die Transformation
- `fftfreq()`: berechnet die Frequenzen in der Mitte jedes Bin in der Ausgabe von `fft()`. Ohne diese Funktion gäbe es keine Möglichkeit, die x-Achse des Frequenzspektrums darzustellen.

Ein Bin ist ein Bereich von Werten, die gruppiert wurden, wie in einem Histogramm. Für die Zwecke dieses Tutorials kannst du sie dir als einzelne Werte vorstellen.

Außerdem wird `np.abs()` für `yf` aufgerufen, weil die Werte der FFT komplex sind.

Note: Nebenbei bemerkt, hast du vielleicht bemerkt, dass `fft()` eine maximale Frequenz von knapp über 20 Tausend Hertz zurückgibt, 22050Hz, um genau zu sein. Dieser Wert ist genau die Hälfte unserer Abtastrate und wird als Nyquist-Frequenz bezeichnet. Dies ist ein grundlegendes Konzept in der Signalverarbeitung und bedeutet, dass die Abtastrate mindestens das Doppelte der höchsten Frequenz des Signals betragen muss.

10.5.4 Schnellere fast FT mittels `rfft()`

Das von `fft()` ausgegebene Frequenzspektrum wurde an der y-Achse gespiegelt, so dass die negative Hälfte ein Spiegelbild der positiven Hälfte war. Diese Symmetrie wurde durch die Eingabe von reellen Zahlen (nicht von komplexen Zahlen) in die Transformation verursacht.

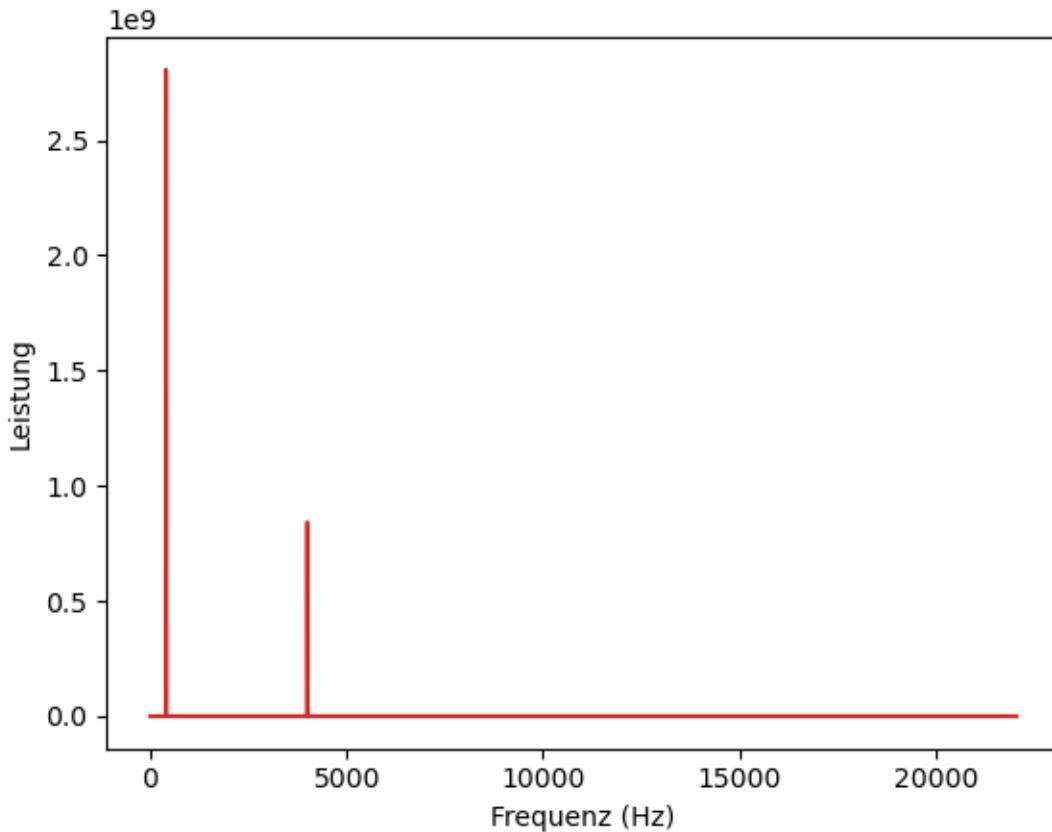
Wir können diese Symmetrie nutzen, um unsere Fourier-Transformation schneller zu machen, indem wir nur die Hälfte davon berechnen. `scipy.fft` implementiert diesen Speed-Hack in Form von `rfft()`.

Das Tolle an `rfft()` ist, dass es ein sofortiger Ersatz für `fft()` ist. Durch das Einfügen von `rfft()` bleibt der Code größtenteils derselbe, nur mit ein paar wichtigen Änderungen:

```
from scipy.fft import rfft, rfftfreq

# Note the extra 'r' at the front
yf = rfft(normalized_tone)
xf = rfftfreq(N, 1 / SAMPLE_RATE)

plt.plot(xf, np.abs(yf), 'tab:red')
plt.xlabel('Frequenz (Hz)')
plt.ylabel('Leistung')
plt.show()
```



Da `rfft()` nur die Hälfte der Ausgabe von `fft()` liefert, wird eine andere Funktion für die Frequenzzuordnung verwendet, nämlich `rfftfreq()` anstelle von `fftfreq()`.

`rfft()` erzeugt immer noch eine komplexe Ausgabe, so dass der Code für die Darstellung des Ergebnisses derselbe bleibt.

Wie du siehst, ist das obige Bild nur die positive Seite des Frequenzspektrums, das `fft()` erzeugt. `rfft()` berechnet nie die negative Hälfte des Frequenzspektrums, was es schneller macht. Sie kann bis zu doppelt so schnell sein, aber einige Eingabelängen sind schneller als andere. Wenn du weißt, dass du nur mit reellen Zahlen arbeiten wirst, dann ist das ein Geschwindigkeits-Hack, den du kennen solltest.

Jetzt, da du das Frequenzspektrum des Signals hast, kannst du mit der Filterung fortfahren.

10.5.5 Filtern des Signals

Ein großer Vorteil der Fourier-Transformation ist, dass sie umkehrbar ist. Das bedeutet, dass alle Änderungen, die du am Signal im Frequenzbereich vornimmst, auch bei der Rücktransformation in den Zeitbereich gelten. Das kannst du dir zunutze machen, um dein Audiosignal zu filtern und die hohen Frequenzen loszuwerden.

Warning: Die in diesem Abschnitt gezeigte Filtertechnik ist für reale Signale nicht geeignet. Es wird ein weiteres Tutorial erstellt, wo digitale Signalverarbeitung und Filtertechniken eingeführt werden.

Die von `rfft()` zurückgegebenen Werte stellen die Leistung der einzelnen Frequenzbereiche dar. Wenn wir die Leistung eines bestimmten Bins auf Null setzen, werden die Frequenzen in diesem Bin im resultierenden Zeitsignal nicht mehr vorhanden sein.

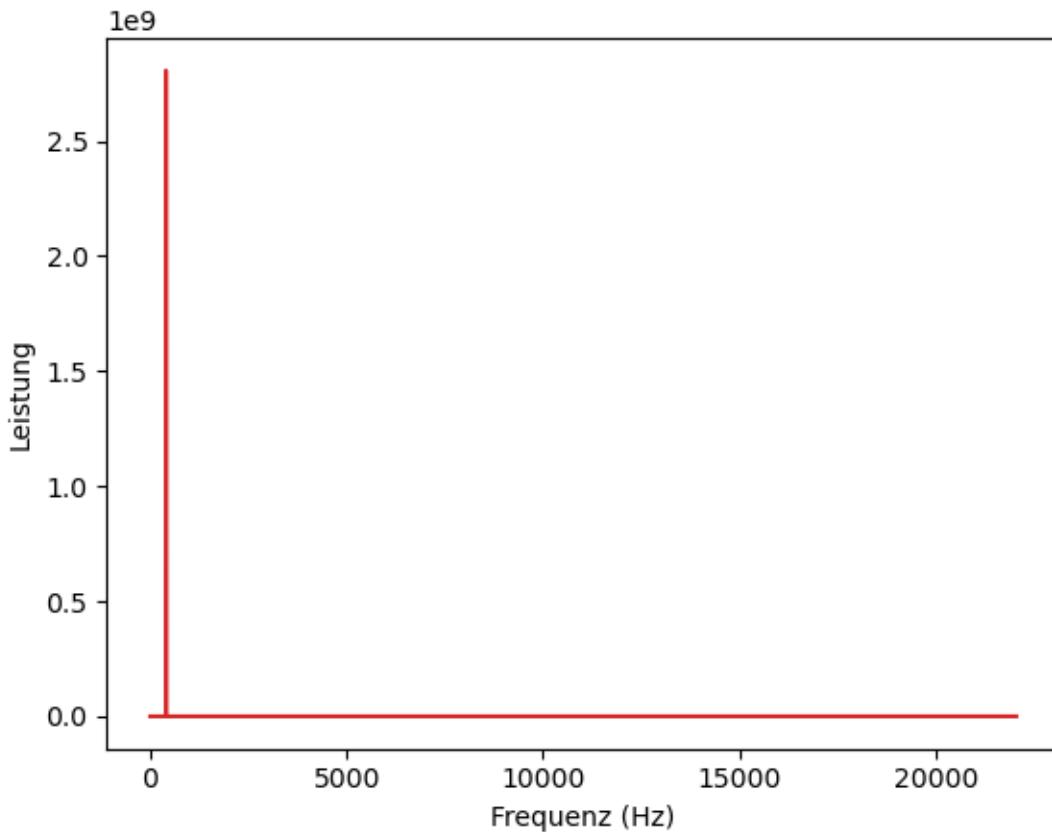
Anhand der Länge von `xf`, der maximalen Frequenz und der Tatsache, dass die Frequenzbins gleichmäßig verteilt sind, kann der Index der Zielfrequenz berechnet werden:

```
# The maximum frequency is half the sample rate
points_per_freq = len(xf) / (SAMPLE_RATE / 2)

# Our target frequency is 4000 Hz
target_idx = int(points_per_freq * 4000)
```

Du kannst im folgenden Code `yf` bei Indizes um die Zielfrequenz herum auf 0 setzen, um sie loszuwerden:

```
yf[target_idx - 1 : target_idx + 2] = 0
plt.plot(xf, np.abs(yf), 'tab:red')
plt.xlabel('Frequenz (Hz)')
plt.ylabel('Leistung')
plt.show()
```



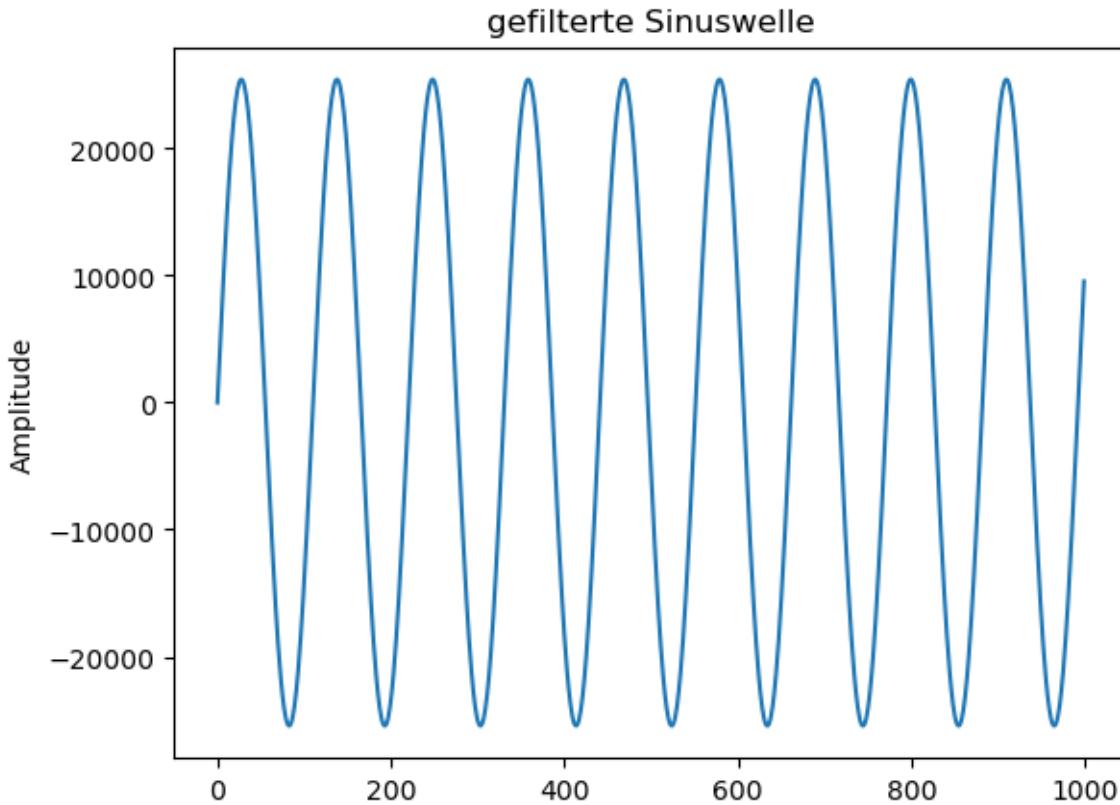
Da es nur eine Spitze gibt, sieht es so aus, als hätte es funktioniert!

10.5.6 Die inverse FFT anwenden

Als nächstes wenden wir die inverse Fourier-Transformation an, um in den Zeitbereich zurückzukehren. Da du `rfft()` verwendest, musst du `irfft()` verwenden, um die Umkehrung anzuwenden. Hätten wir `fft()` verwendet, dann wäre die inverse Funktion `ifft()` gewesen.

```
from scipy.fft import irfft

new_sig = irfft(yf)
plt.plot(new_sig[:1000], 'tab:blue')
plt.ylabel('Amplitude')
plt.title('gefilterte Sinuswelle')
plt.show()
```



Wie du siehst, hast du jetzt eine einzelne Sinuswelle, die mit 400 Hz schwingt, und du hast das 4000-Hz-Rauschen erfolgreich entfernt.

Noch einmal musst du das Signal normalisieren, bevor du es in eine Datei schreibst:

```
import IPython.display as ipd
norm_new_sig = np.int16(new_sig * (32767 / new_sig.max()))
write("clean.wav", SAMPLE_RATE, norm_new_sig)
ipd.Audio('clean.wav') # load a local WAV file
```

```
<IPython.lib.display.Audio object>
```


BIBLIOGRAPHY

- [Par20] Rainer Parthier. *Messtechnik: Vom SI-Einheitensystem über Bewertung von Messergebnissen zu Anwendungen der elektrischen Messtechnik*. Springer Fachmedien Wiesbaden, 2020. ISBN 978-3-658-27130-5 978-3-658-27131-2. URL: <http://link.springer.com/10.1007/978-3-658-27131-2> (visited on 2022-09-15), doi:10.1007/978-3-658-27131-2.
- [PL19] Fernando Puente León. *Messtechnik: Grundlagen, Methoden und Anwendungen*. Springer Berlin Heidelberg, 2019. ISBN 978-3-662-59766-8 978-3-662-59767-5. URL: <http://link.springer.com/10.1007/978-3-662-59767-5> (visited on 2022-09-15), doi:10.1007/978-3-662-59767-5.