

01 - Basics

---

# EEG-Datenverarbeitung

---

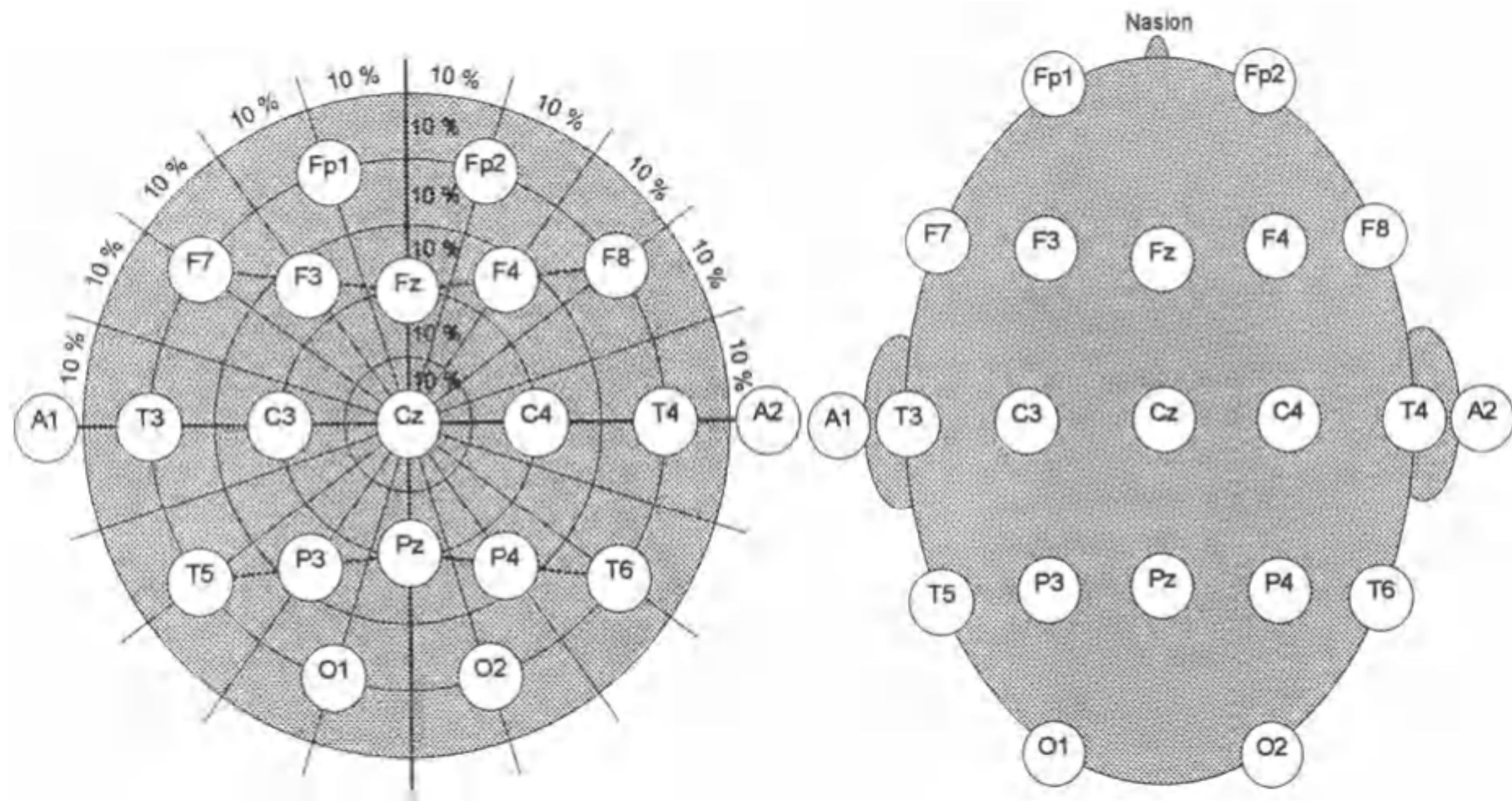
MARIO BABILON & NICLAS HÖRBER

---

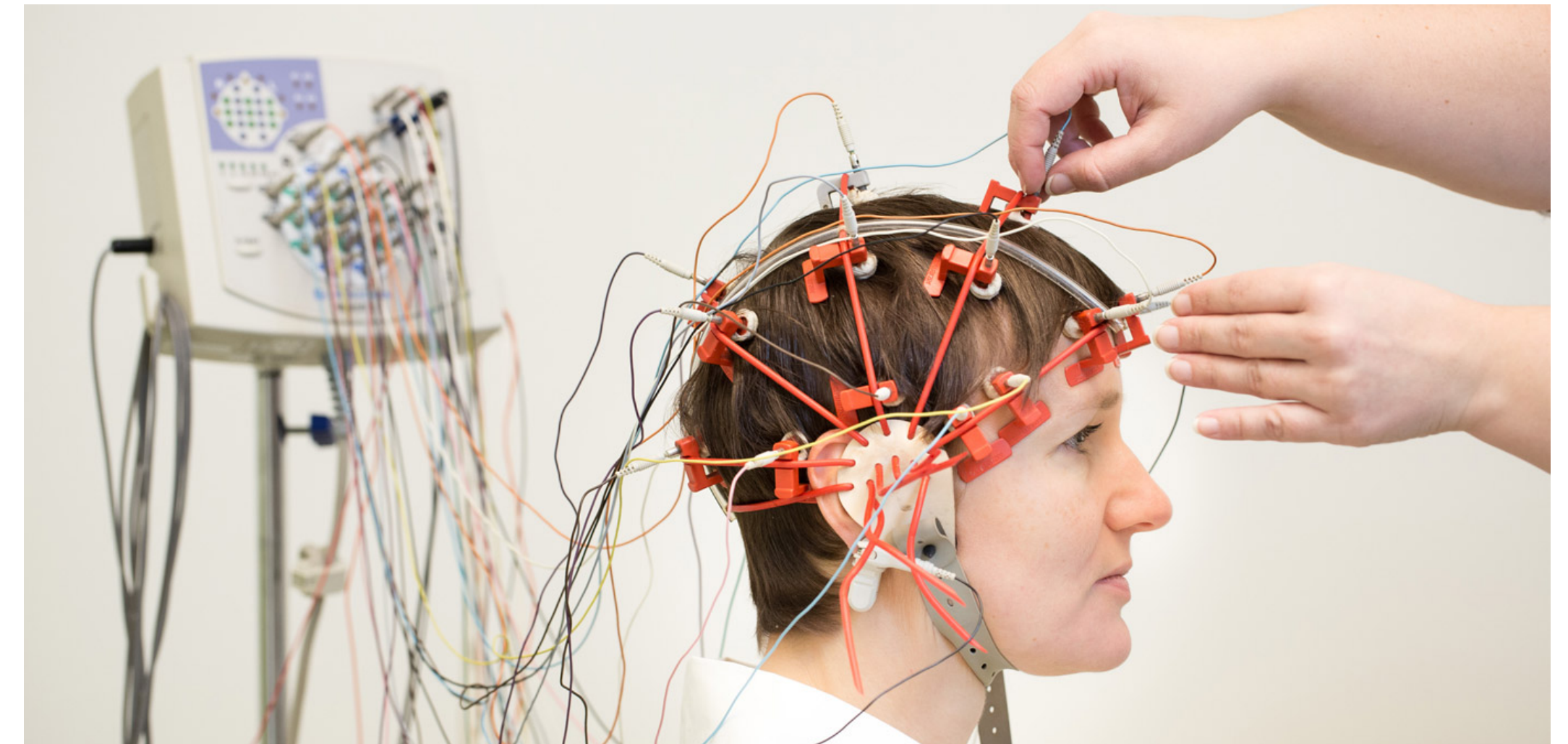
# Einführung: Elektroenzephalogramm (EEG)

- Nichtinvasive Messung elektrischer Aktivitäten des Gehirns
  - Signale sind äußerst schwach: 5-100 Mikrovolt ( $\mu\text{V}$ )
  - Elektrischen Signale entstehen durch zwei Hauptmechanismen:
    1. **Erregende Signale (exzitatorisch):** Erhöhen die Wahrscheinlichkeit, dass eine Nervenzelle feuert
    2. **Hemmende Signale (inhibitorisch):** Verringern die Wahrscheinlichkeit des Feuerns
- => minimale Spannungsänderungen, die an der Schädeloberfläche messbar sind
-

# Einführung: Elektroenzephalogramm (EEG)



RONALD G. SCHMID, 1995, S.8





---

# Einführung: Elektroenzephalogramm (EEG)

EEGO8 von ANTNeuro

-> 2000Hz



---

# Aufgaben

- **Aufgabe 1: Statistische Kennwerte berechnen**

Lesen Sie die EEG-Daten (*LinkeGehirnHaelfte.csv*) mit Hilfe von `readtable` in MATLAB ein und berechnen sie anschließend folgende Kennwerte: Mittelwert, Median, Standardabweichung, Varianz, Minimum, Maximum, Spannweite

-> Geben Sie die berechneten Werte tabellarisch mit sinnvollen Beschriftungen aus.

-> Interpretieren Sie die Kennwerte. Was sagen sie über die Aktivität in diesem Kanal aus.

- **Aufgabe 2: Boxplot erstellen und interpretieren**

Erstellen Sie einen Boxplot der EEG-Daten mit sinnvoller Achsenbeschriftung und Legende.

-> Markieren Sie Mittelwert, Median, Minimum und Maximum zusätzlich als Textbox im Plot.

-> Beschreiben Sie die Ergebnisse des Boxplots.

-> Was könnte eine starke Asymmetrie im Kontext der EEG-Messung bedeuten?

---

---

# Aufgaben

- **Aufgabe 3: Histogramm mit Zusatzinformationen**

Erstellen Sie ein Histogramm der EEG-Werte

-> Fügen Sie vertikale Linien für Mittelwert (rot), Median (blau) und die Nulllinie (gestrichelt schwarz) ein.

-> Passen Sie die Achsen an, um die Symmetrie bzw. Asymmetrie gut zu erkennen.

-> Interpretieren Sie das Histogramm: Wie ist die Verteilung geformt? Gibt es Auffälligkeiten?

---

---

# Aufgaben

- **Aufgabe 4: Daten plotten (15min)**

Lesen Sie erneut die Daten mit readtable ein und verschaffen Sie sich einen Überblick über die Daten

-> Erstellen Sie die X-Achse so, dass sie die Zeit in Sekunden für jeden Datenpunkt enthält

-> Plotten Sie das Signal y gegen die Zeitachse x

-> Beschriften Sie den Plot sinnvoll (Achse, Titel)

-> Schauen Sie sich die Daten von einem kurzen Zeitraum (nur eine Sekunde) an. Gibt es hier ein regelmäßiges Muster? Könnte das Signal Artefakte enthalten?

---

# Aufgaben

- **Aufgabe 5: Fast Fourier-Transformation (20min)**

Führen Sie eine FFT der Daten durch.

-> Erstellen Sie die zugehörige Frequenzachse und berechnen Sie die normierte Amplitude

-> Stellen Sie das Frequenzspektrum im Bereich von **0 bis 200Hz** grafisch dar

-> Beschriften Sie den Plot aussagekräftig

-> In welchem Bereich befinden sich die dominanten Frequenzen in Ihrem Signal? Woran könnte das liegen?

-> Falls nötig entfernen Sie sehr niederfrequente Anteile unter 0.3Hz

---



---

# Aufgaben

- **Aufgabe 6: Artefakte entfernen (50Hz) (25min)**

Führen Sie eine Filterung der Daten durch:

-> Implementieren Sie einen Butterworth Bandstop-Filter

-> Implementieren Sie einen IIR-Notch Filter

-> Stellen Sie das rohe & die beiden gefilterten Signale nebeneinander in der FFT-Darstellung dar

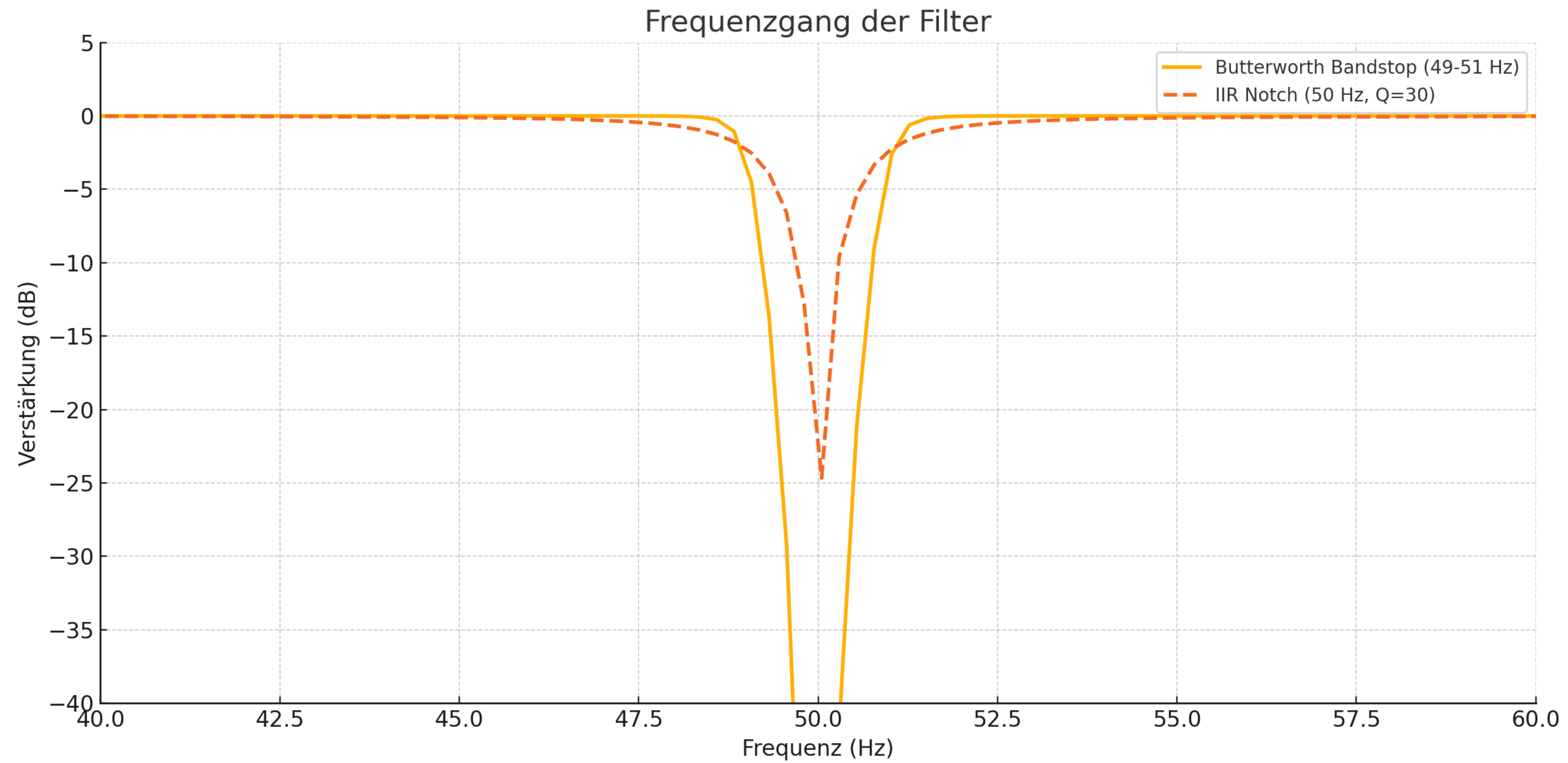
-> Stellen Sie das rohe & die beiden gefilterten Signale nebeneinander in der Zeit-Amplituden-Darstellung dar

---

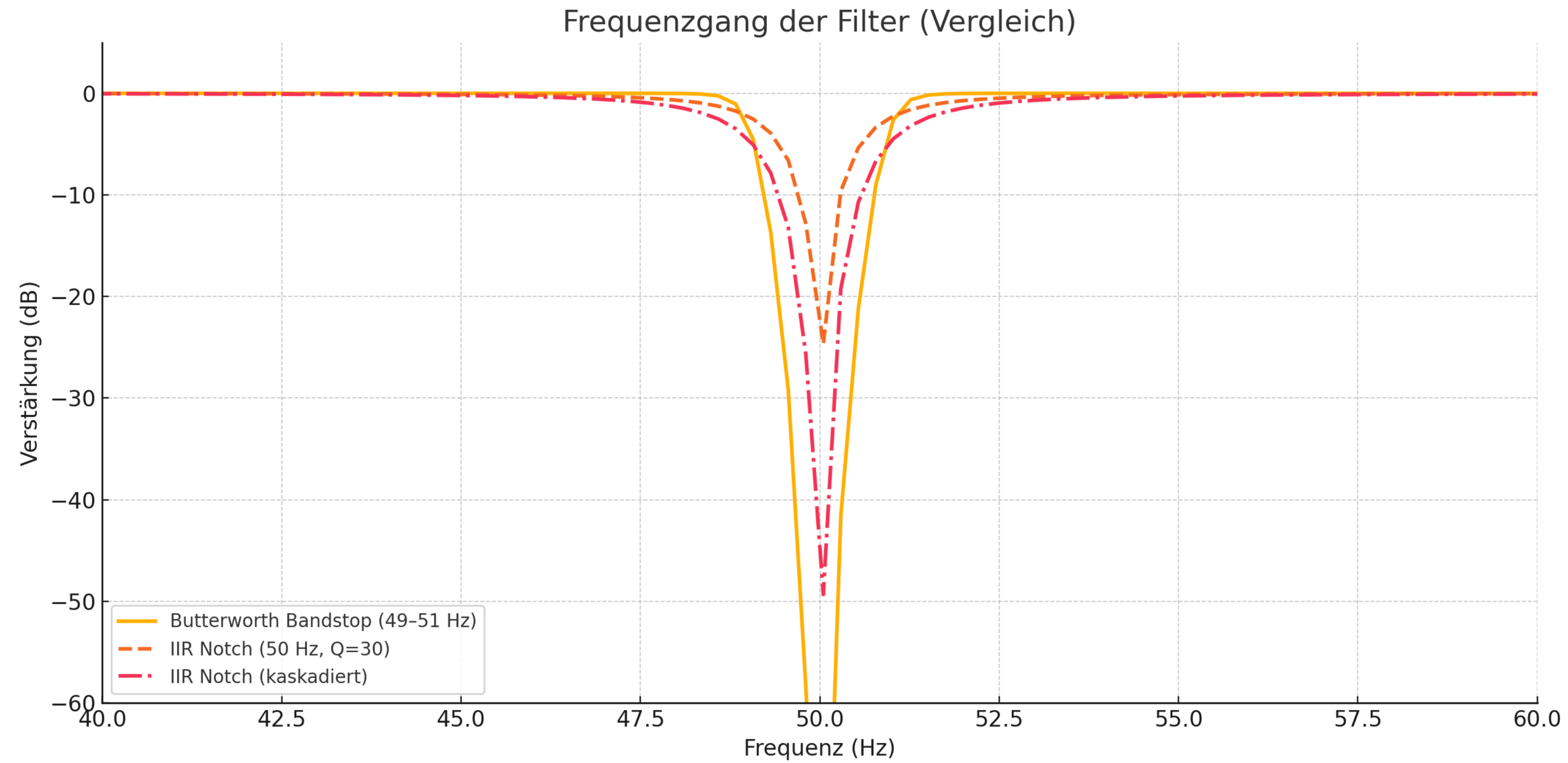
# Filter

Eigenschaft	Butterworth Bandstop (49–51 Hz)	IIR Notch (50 Hz, Q=30)
Filtertyp	Bandstop (breites Frequenzband)	Kerbfilter (sehr schmalbandig)
Frequenzbereich	49 bis 51 Hz	Ca. 50 Hz (sehr eng)
Filterbreite	2 Hz	Sehr schmal (weniger als 1 Hz je nach Q)
Q-Faktor	Nicht spezifiziert	Definiert Schmalbandigkeit (hier 30)
Einfluss auf Nachbarmfrequenzen	Relativ breit, beeinträchtigt 49-51 Hz	Minimal, nur exakt 50 Hz
Anwendung	Entfernen eines breiten Störbandes	Entfernen einer schmalen Störfrequenz
Implementierung in MATLAB	<code>butter</code> + <code>filter</code>	<code>iirnotch</code> + <code>filtfilt</code>

# Filter



# Filter



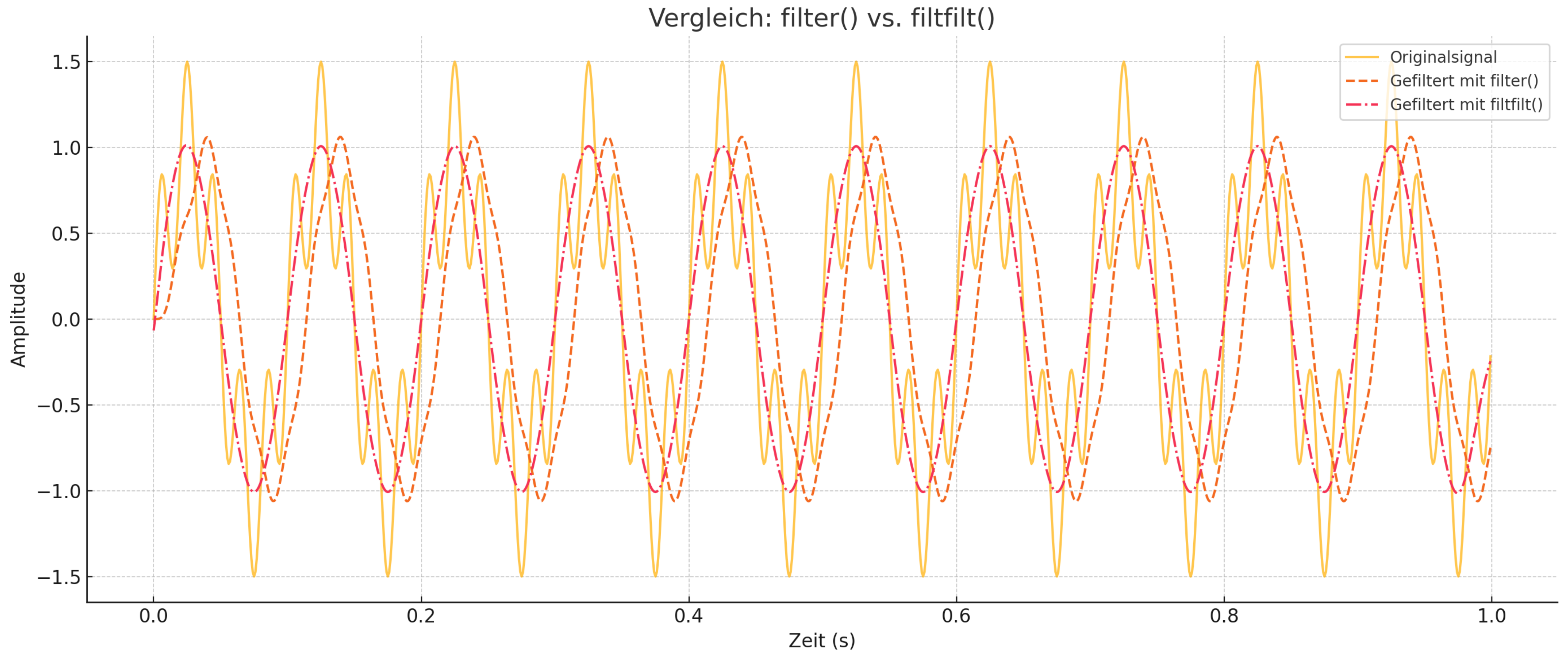


---

# Filter

- `filter()` ist ein gewöhnlicher Vorwärtsfilter, der Phasenverschiebung verursacht
  - Echtzeitverarbeitung
- `filtfilt()` wendet den Filter vorwärts und rückwärts an, sodass keine Phasenverschiebung entsteht (Zero-Phase-Filtering)
  - Offline Analyse
- Gerade beim Netzbrummen ist es sinnvoll, Phasenverzerrung zu vermeiden

# Filter



# Aufgaben

- **Aufgabe 7.1: Trend entfernen (20min)**

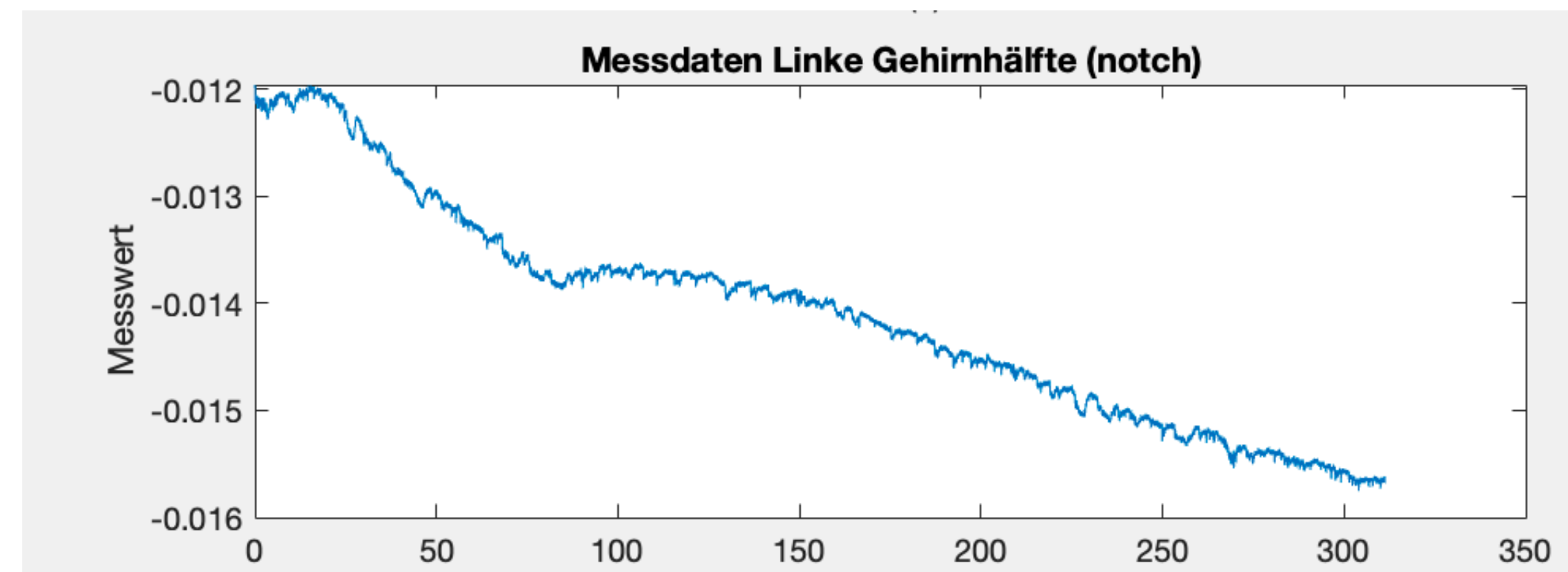
Führen Sie eine weitere Filterung der Daten durch (die 50Hz bereits weggefiltert):

-> Implementieren Sie zwei unterschiedliche Wege, um den negativen Trend zu eliminieren

-> Hierfür bietet Matlab bereits Funktionen an.

-> Vergleichen und Bewerten Sie die Filtertechniken

-> Stellen Sie die nicht gefilterten und die gefilterten Daten nebeneinander da

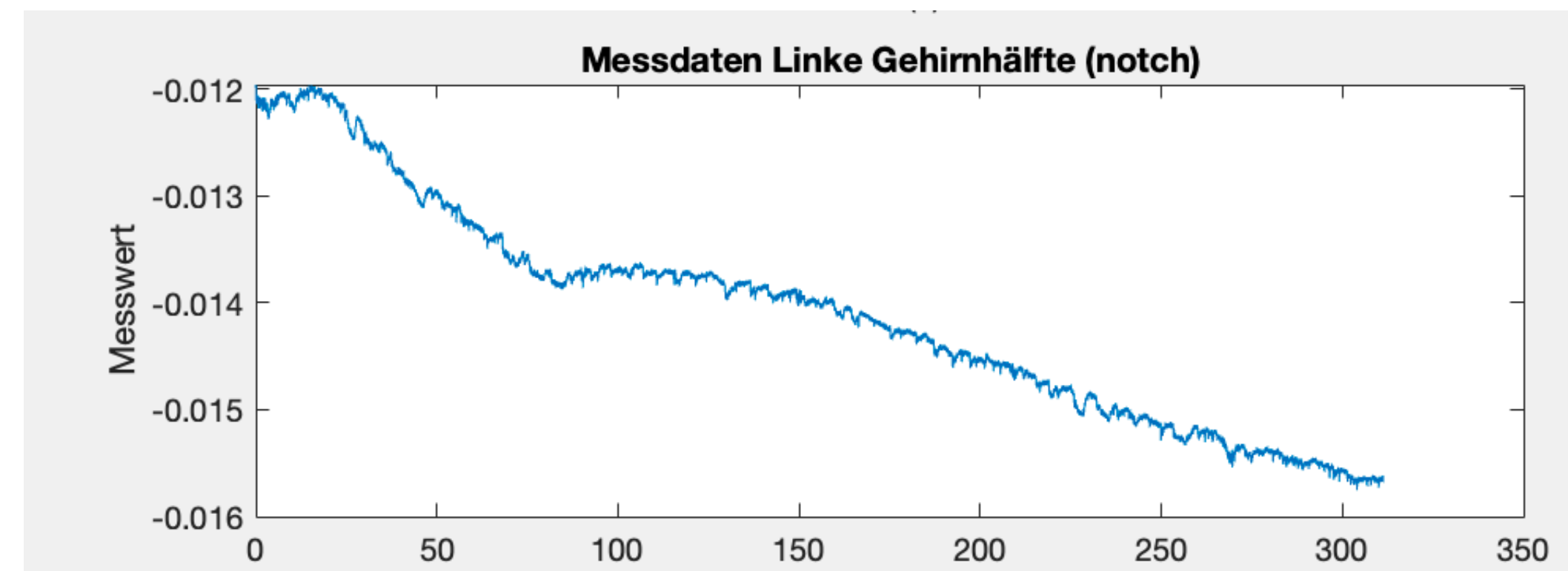


# Aufgaben

- **Aufgabe 7.2: Trend entfernen (20min)**

Führen Sie eine weitere Filterung der Daten durch (die 50Hz bereits weggefiltert):

- > Implementieren Sie eine eigene alternative zur Eliminierung der Trendlinie
- > Vergleichen und Bewerten Sie diese Alternative
- > Stellen Sie die nicht gefilterten und die gefilterten Daten (ihre Alternative und die beste Matlab Funktion) nebeneinander da
- > Stellen Sie den Besten der drei Wege für die ersten 3 Sekunden (6000 Datenpunkte) dar





---

# Aufgaben

- **Aufgabe 8: Eingrenzung auf die interessanten Frequenzen (0.3-32Hz) (35min)**

Führen Sie eine weitere Filterung der Daten durch:

- > Implementieren Sie einen Hoch- & Tiefpassfilter, sodass nur noch die Frequenzen von 0.3 - 32Hz vorhanden sind
  - > Stellen Sie das gesamte Signal dar
  - > Stellen Sie die ersten 3 Sekunden dar
  - > Stellen Sie die Daten anhand von der FFT dar
-