

# Sportinformatik

## WS 2019/2020 Übung 5

Abgabetermin: Mittwoch, 27. November 2019, 8 Uhr

### Aufgabe 1: Frequenzgang

**14 Punkte**

Der Frequenzgang eines kausalen FIR-Filters wird im Allgemeinen mit folgender Formel berechnet ( $j$  ist die imaginäre Einheit):

$$H(\omega) = \sum_{k=0}^{N-1} b_k e^{-j\omega k}$$

Mit  $C(\omega) = |H(\omega)|$  erhält man den Amplituden-Frequenzgang.

Es kann auch die in der Vorlesung erarbeitete Formel

$$C(\omega) = \sqrt{(A(\omega))^2 + (B(\omega))^2}$$

mit

$$A(\omega) = \sum_{k=0}^{N-1} b_k \cos(k\omega)$$

und

$$B(\omega) = \sum_{k=0}^{N-1} b_k \sin(k\omega)$$

verwendet werden, um den Amplituden-Frequenzgang zu bestimmen.

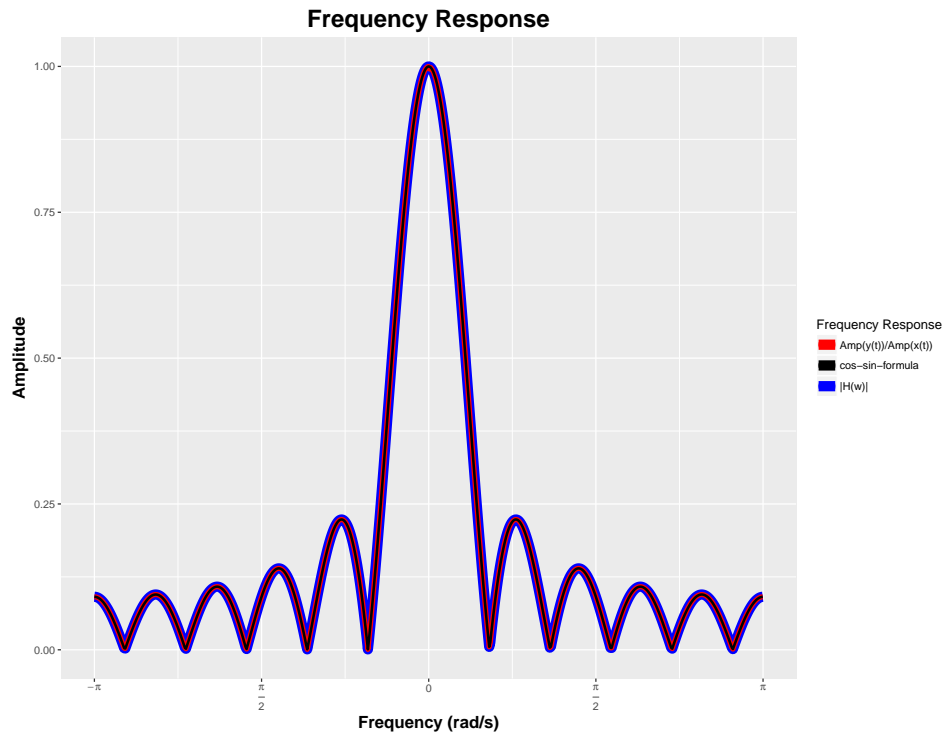
1. Verwenden Sie eine der Formeln um den Amplituden-Frequenzgang eines gleitenden Mittelwertfilters (z.B.  $n = 11$ ) für Frequenzen zwischen  $-\pi$  und  $\pi$  zu berechnen. 4
2. Eine andere Möglichkeit den Amplituden-Frequenzgang eines Filters zu bestimmen ist, ein Kosinus-Signal  $x_\omega(t) = \cos(\omega t)$  der Frequenz  $\omega$  zu filtern. Man erhält das Ausgangssignal  $y_\omega(t)$ .

Dann kann man den Amplituden-Frequenzgang  $C(\omega)$  durch

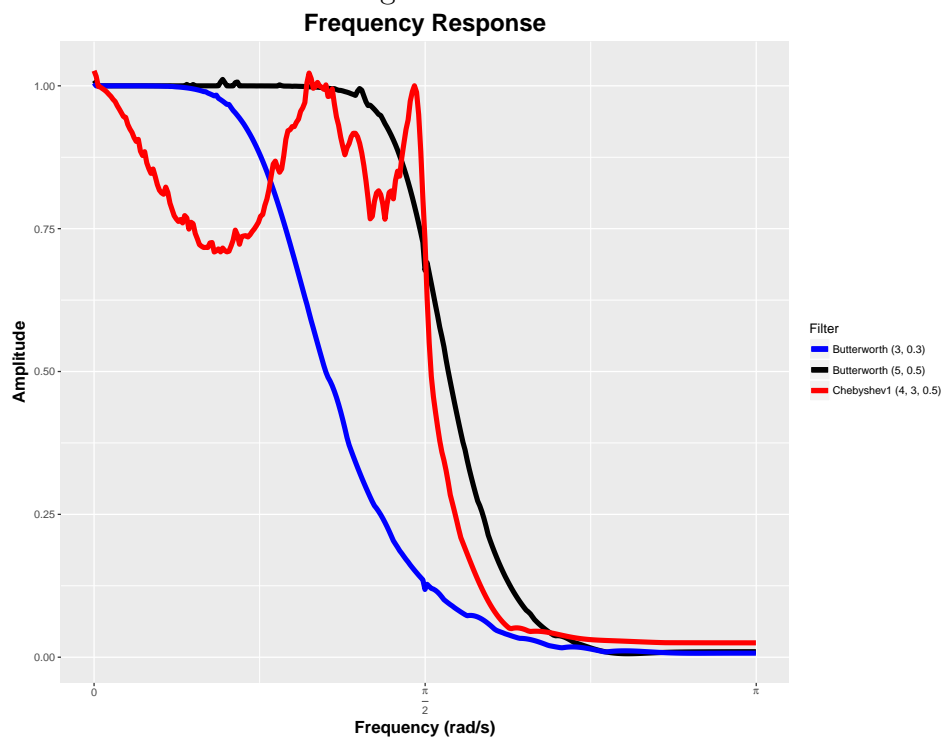
$$C(\omega) = \frac{\text{Amplitude}(y_\omega(t))}{\text{Amplitude}(x_\omega(t))}$$

bestimmen. Implementieren Sie eine Funktion, die  $C(\omega) = \frac{\text{Amplitude}(y_\omega(t))}{\text{Amplitude}(x_\omega(t))}$  für den gleitenden Mittelwertfilter aus 1. bestimmt. 4

3. Plotten Sie die Ergebnisse aus 1. und 2. ähnlich wie in der folgenden Grafik: 2



4. Mit Hilfe der Funktionen `butter` und `cheby1` aus dem Paket `signal` können Koeffizienten für weitere Filtertypen erzeugt werden. Modifizieren Sie ihre Funktion aus 2. so, dass als Parameter die Koeffizienten-Vektoren  $b$  und  $a$  übergeben werden können und die Kosinus-Signale mit dem durch  $b$  und  $a$  definierten Filter modifiziert werden. Zum filtern können Sie die Funktion `filter` (ebenfalls aus dem Paket `signal`) verwenden. Plotten sie den Amplituden-Frequenzgang für unterschiedliche Filter ähnlich wie in der folgenden Grafik: 4



## Aufgabe 2: Metriken

**4x1 Punkte**

Metriken erlauben es, den Unterschied zwischen zwei Signalen  $x$  und  $y$  der Länge  $N$  zu quantifizieren. Bekannte (Fehler-)Metriken sind z.B.:

$$\text{Mean-Absolute-Error: } \text{MAE}(x, y) = \frac{1}{N} \sum_{i=1}^N |x_i - y_i|$$

$$\text{Mean-Squared-Error: } \text{MSE}(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

$$\text{Root-Mean-Squared-Error: } \text{RMSE}(x, y) = \sqrt{\text{MSE}(x, y)} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2}$$

$$\text{Mean-Absolute-Percentage-Error: } \text{MAPE}(x, y) = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{x_i - y_i}{x_i} \right|$$

Implementieren Sie die Funktionen `mae(x,y)`, `mse(x,y)`, `rmse(x,y)` und `mape(x,y)`.

## Aufgabe 3: Weight estimation

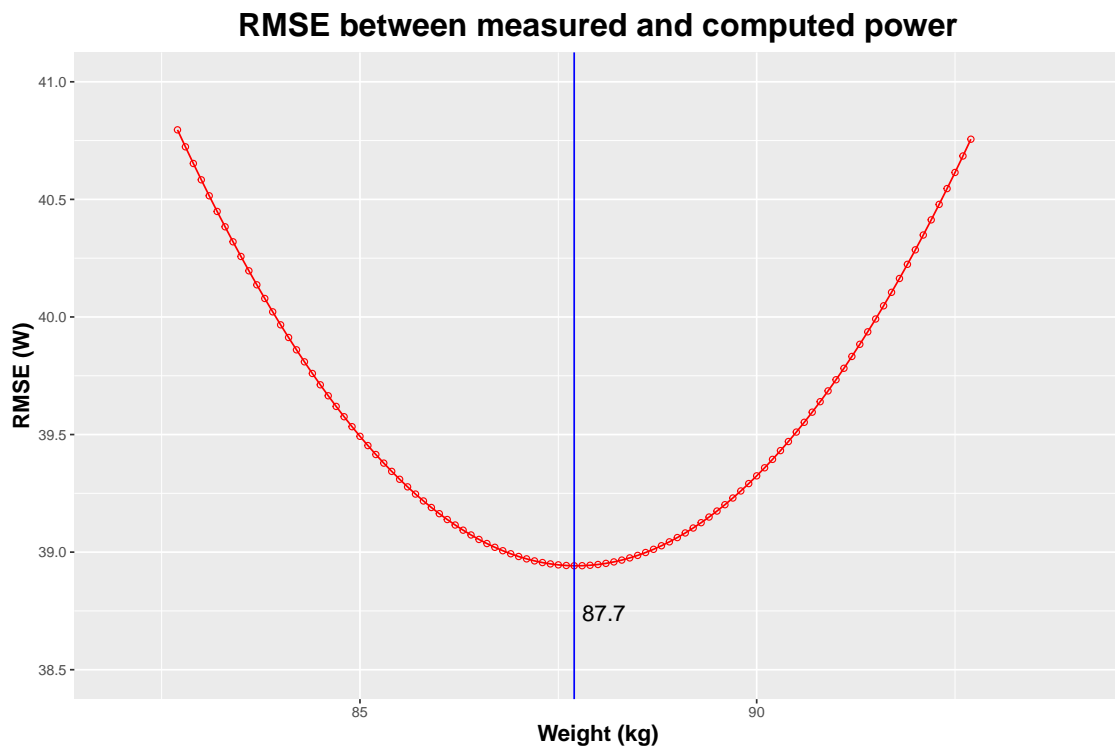
**9 Punkte**

Die Datei `strava_data.mat` enthält Geschwindigkeits- und Leistungsdaten, die auf dem Strava Segment Norton Summit aufgezeichnet wurden. Höhen- und Steigungsprofil des Segments sind bereits aus Aufgabe 3 des letzten Übungsblatts bekannt (`segment.mat`).

Datensätze, die keine aufgezeichnete Leistung enthalten wurden entfernt. Für diese Datensätze soll später die Leistung mit der Leistung-Geschwindigkeits-Formel von Übungsblatt 4 (*Korrektur beachten!*) berechnet werden.

Da keine Information zum Gewicht vorliegt, suchen wir das Gewicht, dass für die in `strava_data.mat` enthaltenen Datensätze den geringste Fehler produziert.

Implementieren Sie dazu eine Funktion `weightEstimation(strava_data, ERROR_FUN)`, die für Werte zwischen 50 kg und 100 kg für alle Datensätze aus der Geschwindigkeit die Leistung berechnet, diese mit den aufgezeichneten Leistungswerten vergleicht, den mittleren Fehler für jedes Gewicht berechnet und das Ergebnis wie folgt visualisiert:



Testen Sie die Funktion, in dem Sie für `ERROR_FUN` ihre Funktionen aus Aufgabe 2 übergeben. Unterscheiden sich die Ergebnisse?

---

**Gesamtpunktzahl:**

**27 Punkte**