

## 4 Filterentwicklung mit Matlab

### Für alle Versuche: Ergebnispräsentation in der Übung:

- Beschreibung des Versuchszwecks
- Code
- Plots
- ggf. Hörbeispiele
- Diskussion

### 4.1 Entwurf eines FIR-Filters nach der Fenstermethode

a) Entwerfen Sie mit Matlabs *filterDesigner* ein FIR-Filter mit diesen Kennwerten:

- Abtastfrequenz = 44,1 kHz.
- Tiefpass, Ende des Passbandes bei 1500 Hz, Beginn des Stopbandes ab 1600 Hz.
- Am Beginn des Stopbandes soll die Dämpfung mindestens 40 dB betragen.
- Probieren Sie verschiedene Window-basierte Entwurfsmethoden (rect, hamming, chebychev) aus und nehmen Sie den Entwurf, der die Spezifikation mit möglichst geringer Filterordnung erfüllt.

**Hinweis:** Das Eingabefeld „Filter Order“ des Filter Designers ist falsch beschriftet: Hier müssen Sie die Filterlänge eintragen. Die Filterordnung ist um 1 niedriger.

**Hinweis:** Um eine Idee von der benötigten Filterlänge zu bekommen, können Sie vorab erst mal ein *Equiripple*- FIR-Filter entwerfen, und bei der Filterlänge *Minimum Order* angeben.

- Exportieren Sie das Filter mit File > Export auf den Workspace.
- Dokumentieren Sie mit einem Screenshot die Einstellungen und den Frequenzgang.
- Geben Sie die Filterordnung an.
- Ist das Filter *linearphasig*?

b) Schreiben Sie ein Matlab-Skript, das folgende Anforderungen erfüllt:

- Über eine Variable `select` kann zwischen 4 Signalquellen ausgewählt werden:

**select == 1** wav-Datei `ACDClike.wav`

**select == 2** chirp-Signal (44,1 kHz Abtastrate, 0 ... 5 s, 50 ... 5000 Hz).

**select == 3** Rauschen mit Befehl `rand`, Abtastrate = 44,1 kHz, Dauer 5 s.

**select == 4** Rechteckimpulsfolge, Abtastrate = 44,1 kHz, Dauer = 1 s, Pulsbreite = 5 ms, Wiederholrate 0 20 Hz (Tipp: Funktion `pulsetran`).

Das ausgewählte Signal wird mit **filter** gefiltert, dann:

- Plot
- Normierung und Audio-Wiedergabe des ursprünglichen und des gefilterten Signals.

## 4. 2 Entwurf eines IIR-Filters mit Second-Order-Sections

a) Entwerfen Sie mit Matlabs *filterDesigner* ein IIR-Filter mit diesen Kennwerten:

- Abtastfrequenz = 44,1 kHz.
- Tiefpass mit Grenzfrequenz 1500 Hz, Stopband beginnt bei 1600 Hz.
- Am Beginn des Stopbandes soll die Dämpfung mindestens 80 dB betragen.
- Probieren Sie verschiedene Filterkonzepte (Butterworth, Chebychev, Elliptic).
- Nehmen Sie den Entwurf, der die Spezifikation mit möglichst geringer Filterordnung erfüllt.
- Exportieren Sie das Filter mit **File > Export** auf den Workspace. Das Filter wird als Matrix von *L Second-Order-Sections* (SOS) abgespeichert. Jede Zeile in der Filtermatrix beschreibt eine SOS:

$$sos = \begin{bmatrix} b_{10} & b_{11} & b_{12} & 1 & a_{11} & a_{12} \\ b_{20} & b_{21} & b_{22} & 1 & a_{21} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{L0} & b_{L1} & b_{L2} & 1 & a_{L1} & a_{L2} \end{bmatrix}$$

- Dokumentieren Sie mit einem Screenshot die Einstellungen und den Frequenzgang.
- Geben Sie die Filterordnung an.
- Ist das Filter *linearphasig*?

Schreiben Sie ein Matlab-Skript, das folgende Anforderungen erfüllt:

- Über eine Variable `select` kann zwischen 4 Signalquellen ausgewählt werden:

**select == 1** wav-Datei `ACDCLike.wav`

**select == 2** chirp-Signal (44,1 kHz Abtastrate, 0 ... 5 s, 50 ... 5000 Hz).

**select == 3** Rauschen mit Befehl **rand**, Abtastrate = 44,1 kHz, Dauer 5 s.

**select == 4** Rechteckimpulsfolge, Abtastrate = 44,1 kHz, Dauer = 1 s, Pulsbreite = 5 ms, Wiederholrate 0 20 Hz (Tipp: Funktion `pulsetran`).

Das ausgewählte Signal wird mit `sosfilter` gefiltert, dann:

- Plot
- Normierung und Audio-Wiedergabe des ursprünglichen und des gefilterten Signals.

### **4. 3 Entwurf eines Notch-Filters zur Unterdrückung sinusförmiger Störsignale**

- a) Entwerfen Sie mit Matlabs *filterDesigner* (oder nach dem Beispiel im Vorlesungsskript) ein IIR-Filter mit diesen Kennwerten:
- Abtastfrequenz = 44,1 kHz.
  - Störunterdrückung bei 880 Hz (nur dort).
  - Bandbreite 100 Hz.
  - Export des Filters.
- b) Schreiben Sie ein Matlab-Skript, das folgende Anforderungen erfüllt:
- Einlesen des Gitarrentons `GitRiff_880Hz.wav`.
  - Filterung mit dem Notch-Filter, dann
    - Plot
    - Normierung und Audio-Ausgabe des ursprünglichen und des gefilterten Signals.