

Aufgabenblatt 6

In diesem Aufgabenblatt werden Filter untersucht. Die folgenden Schlüsselwörter und Funktionen sind dabei von besonderer Bedeutung:

audioread, playblocking, audioplayer
fft, ifft, log10
fir1, filter, rectwin, gausswin, hamming, hann, freqz
semilogx, title, xlabel, ylabel, hold, legend, grid

6.1 FIR-Filter

Für den Entwurf von FIR-Filtern (Finite-Impulse-Response-Filter) gibt es in Matlab unterschiedliche Möglichkeiten. Im folgenden werden wir zunächst die Funktion *fir1* nutzen.

a) Schreiben Sie in Matlab folgende Funktion:

```
plotfir (order, fc, window);
```

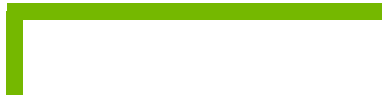
In dem Parameter *order* wird die Filterordnung, in *fc* die Grenzfrequenz (Cut-Off-Frequency) und in *window* das zu verwendende Fenster für den FIR-Filter übergeben. Letzteres muss ein Vektor der Länge *order* + 1 sein. Gehen Sie innerhalb der Funktion von der festen Abtastfrequenz 40 kHz aus.

Die Funktion soll die Taps eines FIR Tiefpassfilters berechnen, indem *fir1* verwendet wird. Anschließend ermitteln Sie den Frequenzgang des berechneten Filters, indem Sie die folgende Matlab-Funktion verwenden (wobei Sie die berechneten Taps als Parameter übergeben):

```
[h, w] = freqz (tap, 1, 10000);
```

Das Ergebnis *w* (Omega) enthält einen Vektor mit den normierten Winkelfrequenzen. Dabei entspricht *w* = 1.0 der halben Abtastfrequenz 20 kHz (auch als Nyquistfrequenz bezeichnet). Passend zu diesem Vektor wird in *h* die Verstärkung zu den unterschiedlichen Frequenzen ebenfalls als Vektor zurückgegeben.

Im letzten Schritt soll in der Funktion *plotfir* der berechnete Frequenzgang als Bodediagramm angezeigt werden. Hierzu müssen Sie die normierten Winkelfrequenzen in *w* in echte Frequenzen und den Vektor *h* in dB-Werte umrechnen. Stellen Sie anschließend die Amplituden als Bodediagramm dar. Verwenden Sie hierzu die Matlab-Funktion *semilogx*. Aktivieren Sie das Grid und achten Sie auf eine korrekte Achsenbeschriftung.



- b) Verwenden Sie Ihre Funktion *plotfir*, um in einem einzelnen Diagramm die Frequenzgänge von vier FIR-Tiefpassfiltern anzuzeigen, die sich jeweils nur im verwendeten Fenster unterscheiden. Ihre Funktion soll hierzu mit folgenden Parametern aufgerufen werden:

- order = 10, fc = 10 kHz, window = rectwin
- order = 10, fc = 10 kHz, window = gausswin
- order = 10, fc = 10 kHz, window = hamming
- order = 10, fc = 10 kHz, window = hann

Beschriften Sie das Diagramm mit einer aussagekräftigen Legende.

- c) Was für ungefähre Dämpfungen haben die Filter bei der Grenzfrequenz $f_c = 10$ kHz? Vergleichen Sie die ermittelten Werte mit dem, was Sie als Hilfe zur Funktion *fir1* in Matlab nachlesen können. Welches Filterdesign hat die größte Steilheit in der Grenzfrequenz? Welcher Filter erscheint Ihnen die beste Charakteristik zu haben? Geben Sie Ihre Antworten als Kommentare in dem Matlab-Skript ab.

Hinweis: Um in Matlab Hilfe zu erhalten, Selektieren Sie im Editor die Zeichenfolge fir1 und drücken Sie F1.

- d) Verändern Sie die Grenzfrequenz f_c auf 100 Hz. Beantworten Sie die in der vorangehenden Teilaufgabe gestellten Fragen erneut. Was für eine Grenzfrequenz haben die Filter nun tatsächlich (berücksichtigen Sie hierzu die in der Hilfe nachzulesende Dämpfung in der Grenzfrequenz). Haben Sie eine Erklärung für den sich zeigenden Effekt?
- e) Vergrößern Sie die Filterordnung für alle Filter solange, bis die Frequenzgänge wieder einigermaßen den Erwartungen entsprechen. Geben Sie zu jedem Filter die minimale Ordnung an, an der der Frequenzgang den Erwartungen etwa wieder entspricht. Wie erklären Sie sich das Ergebnis?

6.2 IIR-Filter

Mit dieser Aufgabe soll das zuvor generierte Ergebnis mit dem eines IIR-Filters verglichen werden.

- a) Übernehmen Sie aus der vorangehenden Aufgabe zunächst alles, was erforderlich ist, um den Frequenzgang eines FIR-Filters mit folgenden Parametern anzuzeigen:
- order = 1000, fc = 100 Hz, window = hamming

Als Abtastrate verwenden Sie weiterhin 40 kHz.

- b) Rufen Sie den Filter-Designer von Matlab auf (unter Apps zu finden). Erzeugen Sie einen elliptischen IIR-Tiefpassfilter, der die Frequenzen bis 100 Hz mit maximal 1 dB dämpft und der bei der Frequenz 200 Hz



das Signal wenigstens genauso gut filtert, wie der zuvor in Teilaufgabe a erzeugte FIR-Filter. Achten Sie darauf, dass die Abtastfrequenz weiterhin 40 kHz sein soll. Schauen Sie sich das Ergebnis zunächst im Filter-Designer an. Schauen Sie sich auch den Pole-Zero-Plot an. Wieviele Nullstellen und Polstellen gibt es? Welche Ordnung hat der generierte Filter?

c) Stellen Sie die Frequenzgänge der in den Teilaufgaben a und b berechneten Filter in einem einzelnen Graphen dar. Gehen Sie folgendermaßen vor:

- Exportieren Sie den im Filter-Designer erzeugten Filter als matlab-Funktion. Rufen Sie hierzu den Menüpunkt „File->Generate MATLAB Code->Filter Design Function“ auf. Speichern Sie das Design unter dem Namen *EllipticFilter*.
- Erweitern Sie das unter Teilaufgabe a geschriebene Skript, indem Sie die erzeugte Funktion *EllipticFilter ohne Parameter* aufrufen. Die Funktion erzeugt ein Ergebnis *Hd*, dass Sie mit Hilfe der oben bereits verwendeten Funktion *freqz* in einen Frequenzgang umrechnen können:

$$[h, w] = \text{freqz}(Hd, 10000);$$

- Geben Sie den Frequenzgang schließlich mit *semilogx* in dem selben Graphen aus, der auch für den Frequenzgang des FIR-Filters genutzt wurde (orientieren Sie sich bei der Lösung dieser Aufgabe an dem, was Sie in Teilaufgabe 6.1a programmiert haben).

Bewerten Sie das Ergebnis und schreiben Sie Ihre Einschätzung als Kommentar in das Matlab-Skript.

6.3 Brumm-Filter

Speziell in der Audiotechnik gibt es das Problem, dass das allgegenwärtige 50 Hz Netzsignal auf die zu verarbeitenden Audiosignale überspricht und dann als störendes Brummen wahrgenommen werden kann. Diese Aufgabe adressiert genau dieses Problem.

- a) Laden Sie sich über Moodle zunächst die Audiodatei *sample4.wav* auf Ihren Rechner. Lesen Sie die Datei über ein Matlab-Skript ein und geben Sie sie über die Soundkarte aus, um einen ersten Eindruck zu dem nachfolgend zu verarbeitenden Audiosignal zu erhalten.
- b) Nutzen Sie den Filter-Designer von Matlab, um einen FIR-Hochpassfilter so zu designen, dass sich damit das Brummen entfernen lässt. Aus Kostengründen darf der Filter maximal eine Ordnung von 300 haben. Verwenden Sie als Design-Methode *Window* wobei ein Hamming-Fenster zum Einsatz kommen soll. Testen Sie den erzeugten Filter an dem

zuvor eingelesenen Signal. Geben Sie das Ergebnis über die Soundkarte aus.

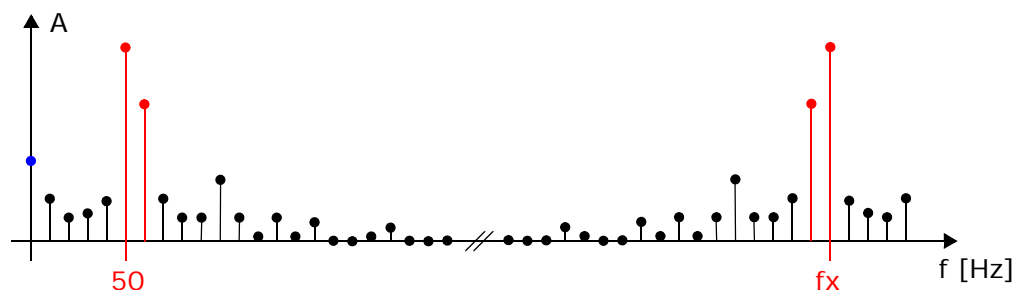
Hinweis: Um den Filter zu testen, können Sie die Matlab-Funktion `filter` verwenden. Die Funktion hat folgende Signatur:

$$y = \text{filter}(\text{Hd}, x);$$

- c) Wiederholen Sie den vorangehenden Versuch. Verwenden Sie diesmal jedoch einen IIR-Bandstopfilter, den Sie nach eigenen Vorstellungen designen dürfen. Beachten Sie dabei, dass das Netzbrummen nicht immer exakt bei 50 Hz liegt sondern um einen Mittelwert schwankt.
- d) Führen Sie einen dritten Versuch durch um das Netzbrummen zu entfernen. Diesmal führen Sie eine FFT durch, entfernen die störenden Frequenzanteile, indem Sie diese auf 0 setzen und wandeln Sie das Ergebnis mit einer `ifft` zurück in den Zeitbereich. Spielen Sie das generierte Ergebnis über die Soundkarte ab.

Hinweis: Beachten Sie, dass das Spektrum periodisch und symmetrisch ist. Dies ist in dem nachfolgenden Bild exemplarisch dargestellt (entspricht nicht dem, was Sie erhalten). Die linke Seite des dargestellten Spektrums entspricht dem, was auf der rechten Seite gespiegelt dargestellt ist.

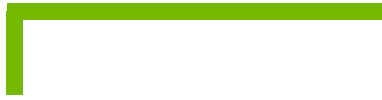
Wenn Sie die 50 Hz Spektrallinie entfernen wollen, dann müssen Sie auch die gespiegelte Spektrallinie (mit f_x gekennzeichnet) entfernen, d.h. auf 0 setzen. Beachten Sie weiter, dass es nicht Reicht genau 50 Hz zu entfernen, weil die Netzfrequenz immer leicht schwankt. Beachten Sie außerdem, dass der Gleichanteil für $f = 0$ Hz nicht gespiegelt auftaucht, sondern nur einmal vorhanden ist (blau dargestellt).



6.4 Kerb-Filter als Brummfilter

In der Realität streut das Netzsignal stark verzehrt in das Audiosignal ein. Dies führt dazu, dass das Brummen nicht nur als 50 Hz Signal wahrnehmbar ist, sondern auch über ganze Vielfache von 50 Hz. Wenn man also nur 50 Hz herausfiltert, hört man trotzdem noch ein 100 Hz Brummen. Filtert man auch 100 Hz heraus, hört man 150 Hz usw.

- a) Wiederholen Sie die vorangehende Aufgabe zunächst mit dem Audi-



odatei *sample5.wav* das Sie sich zuvor über Moodle auf Ihren Rechner laden müssen.

- b) Modifizieren Sie die zuletzt erarbeitete Lösung (unter Verwendung einer FFT und IFFT) so, dass alle Spektralanteile, die ein ganzes Vielfaches der Netzfrequenz 50 Hz aufweisen, zu 0 gesetzt werden. Spielen Sie das Ergebnis über die Soundkarte ab. Beachten Sie erneut, dass auch diesmal angrenzende Frequenzen berücksichtigt werden müssen, da das Netzsignal um die mittlere Frequenz 50 Hz variiert.