

3. Bewertetes Aufgabenblatt

Gesamtpunktzahl: 20 Punkte

In diesem Aufgabenblatt geht es um die Themenbereiche *Diskrete Fourier-Transformation*, *Fensterung*, *Abtastung*, *Aliasing* und *DFT-Systeme*. Als Soundmaterial werden mehrere Instrument-Aufnahmen zur Verfügung gestellt, aus denen Sie zwei auswählen können. Die Sounddateien haben eine Abtastrate von $F_s = 44100$ und sind als Stereo .wav-Files abgespeichert, die sie auf einen Kanal reduzieren sollen.

Für die folgenden Aufgabenteile erstellen Sie bitte MATLAB Skripte, bzw. Funktionen.

1

Programmieren Sie eine Implementierung der diskreten Fourier-Transformation (DFT) in einer Matlab-Funktion. Als Eingabe soll diese Funktion das zu analysierende Zeitsignal als einen Vektor beliebiger Länge N , sowie die Abtastrate F_s erhalten. Als Ausgabe soll die Funktion die komplexe DFT, das Betrags-, sowie das Phasenspektrum ausgeben. Zusätzlich soll sich ein Fenster öffnen, in dem das Betrags- und Phasenspektrum grafisch dargestellt werden im Bereich von $[0, \frac{F_s}{2}]$. Die Spezifikation im Funktionskopf kann also so aussehen:

```
function [ X, X_abs, X_angle ] = dft_imp( x, Fs )
% DFT_IMP calculates the discrete fourier transformation of the input time
% series x and outputs its complex DFT coefficients as X, its absolute
% spectrum as X_abs and its phase spectrum as X_angle. Additionally it opens
% a figure that displays the absolute and the phase spectrum in the region
% from 0 to Fs/2 in two subplots.
```

Sie können für die Berechnung eine `for`-Schleife benutzen, um die Summenbildung der DFT zu realisieren. Achten Sie dabei auf die Symmetrie der diskreten Fouriertransformation für reelle Zeitsignale sowie auf die korrekte Achsenskalierung für die grafischen Darstellungen. Vergessen Sie bei den grafischen Darstellungen auch nicht durch die Anzahl der Stützstellen zu teilen, um eine korrekte Skalierung zu erhalten. Erstellen Sie ausserdem eine neue Funktion, die durch geeignete Modifizierung der ersten Funktion eine IDFT Implementiert. **(5 Punkte)**

2

Benutzen Sie die von Ihnen in **1.** erstellte Funktion um die DFT-Koeffizienten der zwei Sounddateien zu berechnen und vergleichen Sie die zwei Betragsspektren. Gehen Sie dabei auf die evtl. Unterschiede in den Grundfrequenzen und Obertönen ein. Überprüfen Sie ihre selbstgeschriebene DFT indem sie die gleichen Sounddateien auch mit der Matlab-internen `fft.m` Funktion transformieren und die Differenzen zwischen ihrer DFT und der Matlab-internen FFT plotten. **(2 Punkte)**

3

Erstellen Sie ein Sinussignal mit einer Frequenz von 2 kHz , einer Abtastrate von 16 kHz und einer Länge von 2 Sekunden. Stellen Sie die Grundperiode des Signals in Samples fest. Benutzen Sie die Matlab-Fensterfunktionen `rectwin()`, `hann()` & `bartlett()` um jeweils zwei Fenstervarianten zu erzeugen:

1. eines, welches ein ganzzahliges Vielfaches (z.B. 20) der Grundperiode lang ist,
2. eines, was die gleiche Länge wie das erste Fenster plus die Länge einer Viertelperiode des 2 kHz -Sinus als Länge hat.

Fenstern Sie nun mit jedem der 6 resultierenden Fenster das Sinussignal und berechnen Sie die jeweiligen Betragsspektren. Vergleichen Sie anschließend die Betragsspektren und erläutern Sie wie es zu den Unterschieden kommt. **(1 Punkt)**

4

Wählen Sie eine der beiden Audiodateien aus und erstellen Sie mit `fir1()` einen Lowpassfilter, dessen Impulsantwort so viele Samples hat wie das Audiosignal und eine Grenzfrequenz bei 5000 Hz hat. Wir wollen das Ausgangssignal $y[n] = x[n] * h[n]$ (mit $x[n]$: Eingangssignal, $h[n]$: Impulsantwort des Lowpassfilters) berechnen. Um Zeit bei der Berechnung des Ausgangssignals zu sparen, transformieren Sie beide Folgen (Eingangssignal $x[n]$ und Impulsantwort $h[n]$) ohne weitere Bearbeitung mithilfe der DFT in den Frequenzbereich um die Faltung im Zeitbereich durch eine Multiplikation im Frequenzbereich zu ersetzen. Transformieren Sie das Ergebnis der Multiplikation mit einer inversen DFT zurück in den Zeitbereich und hören Sie sich das so resultierende Ausgangssignal an. Welcher offensichtliche Fehler tritt hier auf? Wie kann man diesen Fehler vermeiden? Erzeugen Sie ein Ausgangssignal ohne diesen Fehler. **(2 Punkte)**

5

Erzeugen Sie ein Betrags- und Phasenspektrum für eine Abtastrate von 16 kHz , sodass mithilfe der DFT-Rücktransformation (`ifft()`-Funktion in Matlab) ein reelles Sinussignal der Frequenz 400 Hz und einer Länge von exakt 1000 Samples entstehen kann.

Benutzen Sie hierzu die Symmetrie der DFT für reelle Zeitsignale und achten Sie darauf, dass aus Betrags- und Phasenspektrum ein kombiniertes Spektrum erzeugt werden muss, dass der IDFT übergeben wird. Überprüfen Sie ihr Ergebnis, indem Sie sich das Signal anhören und plotten. Zum anhören können sie das Signal mit `repmat()` verlängern.

Bonus: Experimentieren Sie mit dieser Methode, indem Sie ein komplexeres Spektrum (z.B. das eines Sinus-Sweeps) erzeugen und das zugehörige Zeitsignal daraus herstellen. **(5 Punkte (+ 3 Punkte))**

6 Form der Abgabe

Abgabetermin ist **Freitag, der 14. Februar 2013**. Die Abgabe umfasst:

- den geschriebenen Matlab-Code (.m-Dateien),
- alle generierten Grafiken,

- eine PDF-Datei mit einer Diskussion der Ergebnisse, sowie den Namen der Autoren mit Matrikelnummer,

verpackt in einer ZIP-Datei mit den Namen der Gruppenmitglieder (z.B.: Name1_Name2.zip). Die Abgabe erfolgt per email an die Adresse `vitali.rotteker AT campus.tu-berlin.de`. Hilfreiche Funktionen zur Durchführung dieses Aufgabenblatts sind: `audioread()` bzw. `wavread()`, `plot()`, `stem()` `subplot()`, `axis()`, `sound()`, `soundsc()`, `sin()`, `fft()`, `ifft()`, `size()`, `zeros`, `ones`, `abs()`, `angle()`, `freqz()`, `fvtool()`.

Achten Sie bei allen Plots bitte auf eine sinnvolle Darstellung (Achseneinteilung und -beschriftung, Plotbereich).