

Aufgabenblatt 2

Mit dem zweiten Aufgabenblatt soll der Umgang mit Matlab vertieft werden. Desweiteren werden grundlegende Begriffe der Signalverarbeitung geklärt. Dabei sind insbesondere folgende Schlüsselworte und Funktionen von Bedeutung:

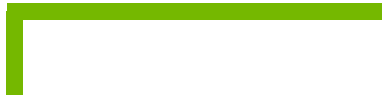
audioread, audiowrite, audioplayer, playblocking
audiorecorder, recordblocking, getaudiodata
awgn, sin
function, for

Wichtig: Geben Sie bitte nur eine Gesamtlösung für jede Aufgabe 2.1, 2.2 usw. ab. Lösungen für die Teilaufgaben a, b usw. sind nicht erwünscht. Am besten geben Sie ein einzelnes Matlab-Skript für das Aufgabenblatt 2 ab und verwenden Sektionen für jede einzelne Aufgabe 2.1, 2.2 usw. (dies ist nur in Matlab und nicht in Octave möglich).

Jede Gruppe (mit maximal zwei Studierenden) gibt die Lösungen bitte gemeinsam ab (also nur einmal). Vergessen Sie nicht, die Namen und Matrikel-Nummern aller an der Lösung beteiligten Studierenden als Kommentar in den ersten Zeilen jedes Matlab-Skripts einzufügen.

2.1 Audiodateien, Unterabtastung und Rauschen

- Schreiben Sie ein Skript, mit dem die Audiodatei „sample1.wav“ mit Hilfe der Funktion audioread in die Variable sample1 gelesen werden kann. Ermitteln Sie die Abtastrate und Länge der Audiodatei „sample1.wav“ und geben Sie beides in der Konsole aus.
Hinweis: Zur Lösung der Aufgabe laden Sie sich die Datei „sample1.wav“ über Moodle auf Ihren Rechner.
- Modifizieren Sie das Signal so, dass ein Monosignal entsteht. Dabei sollen die Abtastwerte des linken und rechten Kanals zu einem Ergebnis gemischt werden. Geben Sie die ersten vier Sekunden des Monosignals über die Soundkarte aus. Das Monosignal soll dabei genauso laut klingen wie das zuvor ausgegebene Stereosignal.
- Geben Sie die ersten vier Sekunden der Audiodatei ein drittes Mal aus. Diesmal soll das Monosignal einer Unterabtastung unterzogen werden. Erzeugen Sie hierzu ein Hilfssignal indem Sie jeden zehnten Abtastwert aus dem Monosignal verwenden. Geben Sie die ersten vier Sekunden des Ergebnisses über die Soundkarte korrekt aus (also ohne Micki-Maus-Effekt). Erläutern Sie als Kommentar, wie sich das Signal verändert hat und warum.



- d) Mischen Sie zum vorangehend erzeugten unterabgetasteten Monosignal weißes Rauschen mit einem Signal-Rausch-Abstand von 40 dB und geben Sie die ersten vier Sekunden des Ergebnisses über die Soundkarte aus. Verwenden Sie zur Erzeugung des weißen Rauschens die Matlab-Funktion `awgn`.

2.2 Linearität

- a) Lesen Sie die Audiodatei „sample1.wav“ erneut ein. Modifizieren Sie anschließend das Signal so, dass jeder Abtastwert quadriert wird.
- b) Spielen Sie die ersten vier Sekunden der berechneten Audiodaten über die Soundkarte ab. Bewerten Sie das Ergebnis der nichtlinearen Operation.

2.3 Erzeugen einer Audiodatei

- a) Zeichnen Sie über das Mikrophon ein 3s langes Stereosignal auf und speichern Sie dieses in einer Variablen. Verwenden Sie für die Aufzeichnung die Matlab-Funktionen `audiorecorder`, `recordblocking` und `getaudiodata`. Das Einlesen soll mit einer Abtastrate von 44100 Hz, 16 Bit und 2 Kanälen (stereo) erfolgen.
- b) Normieren Sie das Ausgangssignal auf die Maximalamplitude 1.0 und geben Sie das Ergebnis über die Soundkarte aus.
- c) Speichern Sie das aufgenommene und normierte Audiosignal in der Datei „mysample.wav“. Zur Kontrolle geben Sie die gespeicherte Sounddatei mit Hilfe eines Media-Players wieder (z.B. VLC usw.).

2.4 Abtasttheorem

- a) Schreiben Sie eine Funktion `mysine`, die ein Sinussignal über eine halbe Sekunde erzeugt und über die Soundkarte ausgibt. Außerdem soll die Sinusfunktion als Plot (über `fplot`) und die Abtastwerte mit Hilfe der Matlab-Funktion `stem` in einem Fenster ausgegeben werden. Der Funktionskopf ist wie folgt vorgegeben:

function mysine (f, fs, T)

Die Frequenz des Sinus wird über den Parameter `f` und die Abtastfrequenz über `fs` vorgegeben. Der Parameter `T` gibt an, welches Zeitintervall bei der Ausgabe mit `fplot` bzw. `stem` angezeigt werden soll (Beispiel: `T=0.1` bedeutet, dass `fplot` und `stem` den Graphen über das Zeitintervall von 0 bis 0.1 s anzeigen sollen; Der Berechnete Sinus soll aber trotzdem 0.5 s lang sein). Bei jedem Aufruf der Funktion `mysine` soll ein neues Fenster für die Ausgabe erzeugt werden. Sie erreichen dies, indem Sie die Matlab-Funktion `figure` vor der Ausgabe aufrufen.



- b) Schreiben Sie ein Programm, in dem Sie `mysine` in einer Schleife aufrufen und die Frequenzen 1700 Hz, 1800 Hz, ... 2300 Hz ausgeben. Als Abtastrate verwenden Sie bitte $f_s = 2000$ Hz. Setzen Sie außerdem $T = 0.001$ s.
- c) Interpretieren Sie das Ergebnis. Warum wird der Ton mit der Frequenz 2000 Hz nicht abgespielt? Warum steigt die Tonhöhe zunächst und sinkt dann wieder? Schreiben Sie Ihre Antworten als Kommentar in das Skript.