11B0033 AuV: Meilenstein 03
Audiosignale – Digitalisierung

Julius Schöning
j.schoening@hs-osnabrueck.de

Abgabedatum: 03. Dezember 2020 23:59:59

In diesem Meilenstein werden Sie Audiosignale digitalisieren. Dabei werden Sie die Signalabtastung und -quantisierung programmieren. Alle Aufgaben dieses Aufgabenblattes können Sie wahlweise mit Matlab oder der Scriptsprache Python bearbeiten.

Matlab steht Ihnen als Total Headcount Lizenz der Hochschule Osnabrück zur Verfügung. Installationsinformationen finden Sie unter `https://wiki.hs-osnabrueck.de/pages/viewpage.action?pageId=8749400`.

Gruppen

Der Meilenstein *Audiosignale – Digitalisierung* wird in Gruppenarbeit mit den bereits bekannten Mitgliedern umgesetzt. Dieser Meilenstein ist so ausgelegt, dass Sie diesen an Ihrem eigenen Computer bearbeiten können.

Eine Veränderung der Gruppenzusammensetzung ist nicht mehr möglich.

Abgabe

Die Ergebnisse aller Teilaufgaben sind schriftlich in einem strukturierten Praktikumsbericht im PDF-Format zusammenzufassen. Ergänzen Sie Ihre Ausführungen durch aussagekräftige Screenshots, Abbildungen, Plots und Tabellen. Mögliche Word und LaTeX Vorlagen für Praktikumsberichte finden Sie im Downloadbericht dieses Praktikum.

Den strukturierten Praktikumsbericht und alle Quellcodedateien, die Sie in diesem Meilenstein erzeugen oder auf die Sie sich in Ihrem Praktikumsbericht beziehen müssen in Ihrem Abgabearchiv vorhanden sein.

Für die Abgabe komprimieren Sie alle erstellen Matlab bzw. Python Quellcodedateien sowie Ihren strukturierten Praktikumsberichts als PDF in ein zip-Archive mit dem Namen *03_UserName1_UserName2_UserName3.zip*. Laden Sie das Zip-File bis spätestens zum 03. Dezember 2020 23:59:59 im Abgabebereich AuV Praktikum im OSCA als hoch.

Testat

Bereiten Sie sich für ein Testat mit ggf. schriftlichen Kurztest von ca. 10 Minuten vor. Inhalt dieses Testats werden die Themen der Meilensteine 03 und 04 sein. Ihre Gruppe erhält einen persönlichen Termin für die KW50 zugeteilt.

1 Aufgabe: Komplexe Funktion als Audiosignal

Mathematische Funktion können als Audiosignale hörbar gemacht werden. Programmieren Sie ein Script, das die Funktion

$$signal_{(t)} = 2 \cdot \frac{sin(600 \cdot \pi \cdot t^t)}{t^t}$$

a) als Audiosignal von $2,5$ Sekunden Länge über die Lautsprecher Ihres Computes wiedergibt und

b) über die Zeit von $2,5$ Sekunden als Funktionsplot visualisiert.

Verwenden Sie für die Wiedergabe und für die Visualisierung der Funktion die Abtastrate von $44.100Hz$. Speichern Sie Ihr Script unter *auv_03_Aufgabe_1.m* wenn Sie Matlab oder unter *auv_03_Aufgabe_1.py* wenn Sie Pyhton verwenden.

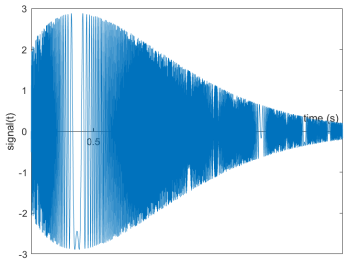
2 Aufgabe: Abtasten und Quantisieren

Funktion, wie die Funktion aus Aufgabe 1, sind zu jedem Zeitpunkt einen unendlich genaue Werte. Durch Zeitdiskretisierung (Abtastung; engl. Sampling) entsteht ein diskontinuierlich-analoges Signal. D.h. es gibt nur noch zu bestimmten Zeitpunkten einen unendliche genauen Wert. Das diskontinuierlich-analoges Signal wird durch die Quantisierung zu einem diskontinuierlich-diskreten Signal.

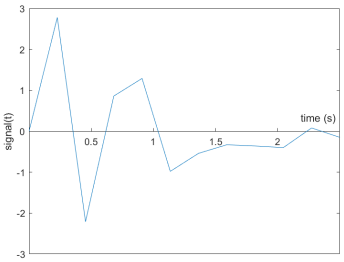
Für diese Aufgabe legen Sie eine Kopie des Scripts *auv_03_Aufgabe_2.m* an und benennen dieses *auv_03_Aufgabe_2.m*. Entfernen Sie nun die Funktion als Audiosignal wiedergibt, durch auskommentieren.

2.1 Abtastung

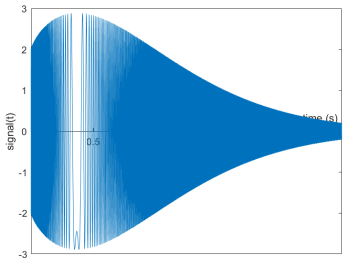
a) bestimmen Sie die Abtastraten der Abbildungen a, b und c indem Sie Ihr Script entsprechend verändern



a



b



c

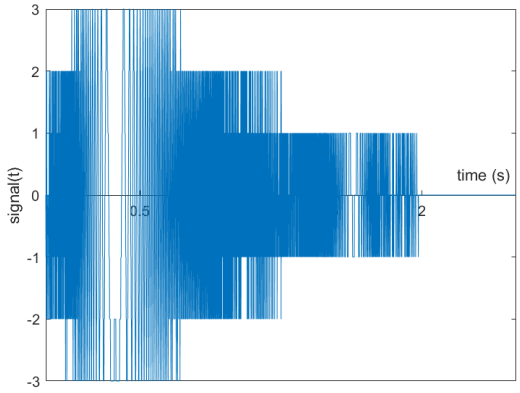
b) beschreiben Sie den Einfluss der Abtastrate auf den Funktionsplot; berücksichtigen Sie in Ihrer Ausführungen auch die Abweichung zur ursprünglichen Funktion

c) fügen Sie Ihrem Programmcode die Variable $audioData$ zu, in der Sie zu jedem abgetasteten Zeitpunkt, den Funktionswert und den Abtastzeitpunkt speichern. Bei einer Abtastrate von $44.100Hz$ bei $2,5$ Sekunden Länge sollte diese Variable die Größe von $2x110.250$ haben

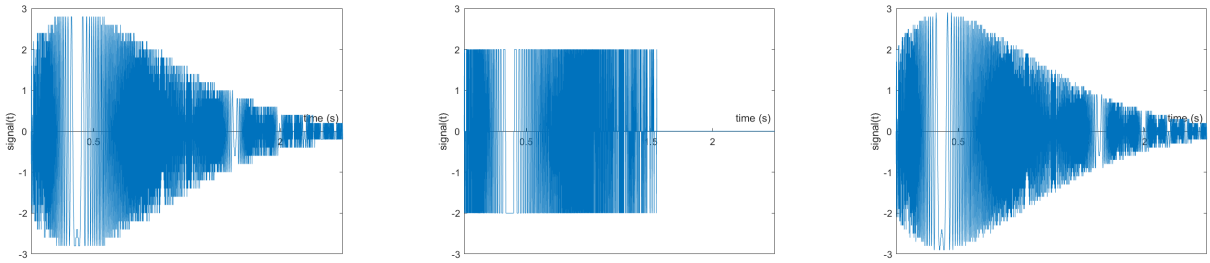
d) kommentieren Sie, nur für diese Teilaufgabe, die Wiedergabe als Audiosignal ein; verändern Sie die Abtastrate, beschreiben Sie Ihre Wahrnehmung und begründen Sie warum Abtastraten kleiner $1.000Hz$ zu einem Programmfehler führen

2.2 Quantisierung

a) erweitern Sie das Programm, sodass die Funktionswerte mathematisch Richtig auf die nächste Ganzzahl $\mathbb{Z}$ gerundet werden
Hinweis: Der Funktionsplot bei einer Abtastrate von $1.000Hz$ und einer Quantisierung auf $\mathbb{Z}$ ist untenstehend abgebildet.



b) bestimmen Sie die Quantisierungsschritte der Abbildungen a, b und c indem Sie Ihr Script entsprechend anpassen
Hinweis: Die Abtastrate beträgt immer $1.000Hz$.



a b c

c) kommentieren Sie, nur für diese Teilaufgabe, die Wiedergabe als Audiosignal ein; setzen die die Abtastrate auf $44.000Hz$ und verändern Sie die Quantisieren auf I) Zehnteln, II) Viertel und III) Hunderstel, beschreiben und begründen Sie Ihre Wahrnehmung

2.3 Abtastung und Quantisierung vs. Speicherplatz und Klangqualität

Die Abtastung und Quantisierung einer kontinuierlichen Funktion bzw. eines analogen Signals hat Einfluss auf die Klangqualität und den benötigen Speicherplatz.
Mit dem Befehl *save('sample_44.1k_q_0_01.mat','audioData')* können Sie den Inhalt einer Variable $audioData$ in eine Datei $sample\_44.1k\_q\_0\_01.mat$ schreiben.

a) digitalisieren Sie die Funktion aus Aufgabe 1 mit der Abtastrate $44.100Hz$, $22.000Hz$ bzw. $1.000Hz$ bei einer Quantisierung auf Hunderstel und speichern Sie jeweils die Variable $audioData$ als Datei $sample\_44.1k\_q\_0\_01.mat$, $sample\_22k\_q\_0\_01.mat$ bzw. $sample\_1k\_q\_0\_01.mat$

b) digitalisieren Sie die Funktion aus Aufgabe 1 mit der Abtastrate $44.000Hz$ bei einer Quantisierung auf Ganzzahlen, Viertel bzw. Zehnteln und speichern Sie jeweils die Variable $audioData$ als Datei $sample\_44k\_q\_1\_00.mat$, $sample\_44k\_q\_0\_25.mat$, bzw. $sample\_44k\_q\_0\_10.mat$

c) digitalisieren Sie die Funktion aus Aufgabe 1 mit der Abtastrate $44.000Hz$ ohne eine Quantisierung und speichern Sie die Variable $audioData$ als Datei $sample\_44k\_q\_0\_00.mat$

d) beschreiben Sie basierend auf den in a) bis c) erzeugten $*.mat$-Files den Zusammenhang zwischen Abtastung, Quantisierung und Speicherplatz

e) kommentieren Sie, nur für diese Teilaufgabe, die Wiedergabe als Audiosignal ein; beschreiben Sie basierend auf Ihren Erkenntnisse aus d) und Ihre Wahrnehmung der verschieden Abtastungen und Quantisierungen den Zusammenhang zwischen Abtastung, Quantisierung, Speicherplatz und Klangqualität

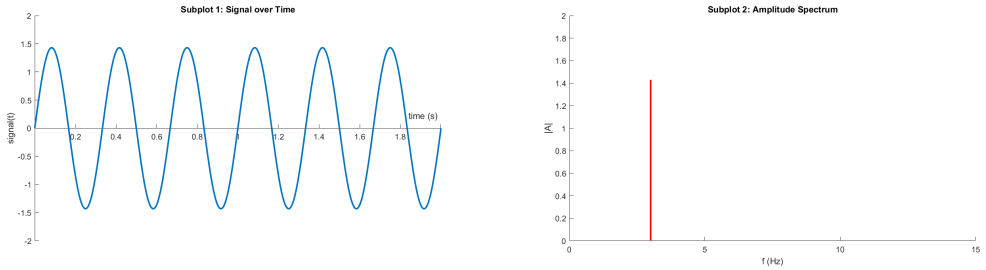
3 Vom Zeit- zu Frequenzbereich

Mit der Fourieranalyse können die Frequenzen und Amplituden aller sinusförmige Teilschwingungen eines Signals extrahiert werden.

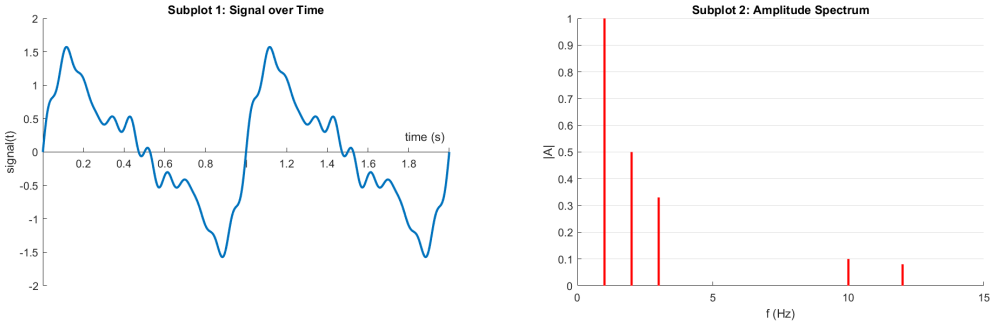
Erstellen Sie ein neues Script mit dem Namen *auv_03_Aufgabe_3.m*.

a) plotten Sie das Signal $signalFFT_{(t)} = 1,43 \cdot sin(2 \cdot F \cdot \pi \cdot t)$ mit $F = 3$ über die Zeit von $2,0$ Sekunden

b) plotten Sie den dazugehörigen Frequenzbereich dieses Signals $signalFFT_{(t)}$.
Hinweis: Verwenden die dafür die Matlab-Funktion *fft()*.
Ihr Ergebnis sollten wie unten abgebildet aussehen.



c) formulieren Sie das Signal zu den unten abgebildeten Zeit- zu Frequenzbereichen, begründen Sie wie Sie zur Signalgleichung gekommen sind, überprüfen Sie Ihr Ergebnis mit Ihrem Script



4 Frequenzbereich einer Audiodatei

Das Frequenzbereich einer Audiodatei kann über die Zeit geplottet werden. Erstellen und erörtern Sie ein neues Script mit dem Namen *auv_03_Aufgabe_4.m*, dass das unten beschriebenen Programm umsetzt.

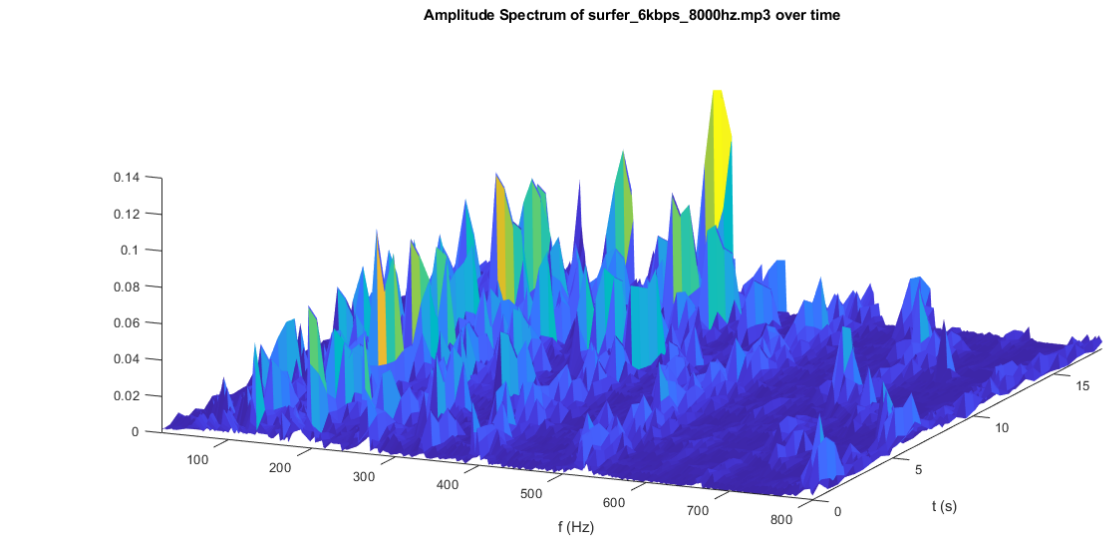
4.1 3D Frequenzplot

a) mit der Funktion *[mono,sampleRate] = audioread('surfer_320kbps_48000hz.mp3')* können Sie die Audiodatei $surfer\_320kbps\_48000hz.mp3$ einlesen

b) zerteilen Sie jetzt den Vector $mono$ in $0,5$ Sekunden Stücke; Hinweis: die Abtastrate ist pro Sekunde, d.h. $Abtastrate \cdot 0,5$ ist die Anzahl der Datenpunkte in einer halben Sekunde

c) wenden Sie jetzt die FFT auf jedes $0,5$ Sekunden Stück an und speichern Sie die Amplitudenwerte

d) mit der Funktion *surf(frequency,time,amplitude, 'EdgeColor', 'none','LineStyle', 'none')* gefolgt von der Funktion *view(24,33)* können jetzt die Amplituden über die Frequenz und Zeit geplottet werden.
Mit den Anzeigegrenzen $20 - 500Hz$ sieht das Ergebnis unten abgebildet aus.



4.2 3D Frequenzanalyse und Klangqualität

a) untersuchen Sie die Unterschiede zwischen den 3D Frequenzplot der Audiodatei $surfer\_320kbps\_48000hz.mp3$ und $surfer\_6kbps\_8000hz.mp3$

b) stellen Sie die wahrgenommene Klangqualität in Bezug zu dem jeweiligen 3D Frequenzplot