



# HOCHSCHULE OSNABRÜCK

UNIVERSITY OF APPLIED SCIENCES

Fakultät Ingenieurwissenschaften und Informatik  
Studiengang Medieninformatik  
Modul Audio- und Videotechnik

## **Praktikumsbericht**

### **Meilenstein 3** **Audiosignale - Digitalisierung**

Wintersemester 2020/2021

Dozent:

Prof. Dr. Julius Schöning

Verfasserin:

Patrick Felschen, Julian Voß

Matrikelnummer:

932056, 934505

Datum der Abgabe:

03.12.2020

## **I. Inhaltsverzeichnis**

1 Aufgabe: Komplexe Funktion als Audio Signal .....	1
2 Aufgabe: Abtasten und Quantisieren .....	2
2.1 Abtastung .....	2
2.1.1 Einfluss der Abtastrate.....	3
2.1.2 Beobachtung bei niedriger Abtastrate .....	3
2.2 Quantisierung .....	3
2.2.1 Auditive Wahrnehmung der Quantisierung .....	4
2.3 Abtastung und Quantisierung vs. Speicherplatz und Klangqualität .....	4
3 Vom Zeit- zu Frequenzbereich .....	5
4 Frequenzbereich einer Audiodatei.....	5
4.1 3D Frequenzplot .....	5
4.2 3D Frequenzanalyse und Klangqualität .....	5
IV. Literaturverzeichnis .....	6
VI. Anhang.....	7

## II. Abbildungsverzeichnis

Abbildung 1: Aufgabe 1 Plot.....	1
Abbildung 2: Plot 1000Hz .....	2
Abbildung 3: Plot 5 Hz .....	2
Abbildung 4: Plot > 44,1kHz hier 50kHz.....	2
Abbildung 7: Plot Faktor 10.....	3
Abbildung 6: Plot Faktor 5.....	3
Abbildung 5: Plot Faktor 0,5.....	3

## 1 Aufgabe: Komplexe Funktion als Audio Signal

In dieser Aufgabe wird eine mathematische Funktion mittels MATLAB als Audiosignal hörbar gemacht. Für die Wiedergabe und Visualisierung der Funktion wird eine Abtastrate von 44100 Hz verwendet. Die Länge des Signals beträgt 2,5 Sekunden. In Abbildung 1 wird der Plot der Funktion dargestellt.

Die Variable „t“ (s. Anhang 1) enthält eine 1xn Matrix, wobei n die Anzahl an abgetasteten Werten wiedergibt. Über den Befehl „sound“ mit den Parametern „signal“, berechnet aus der Formel:

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

und der „sampleRate“<sup>1</sup> wird der Ton zuletzt über die Lautsprecher ausgegeben.

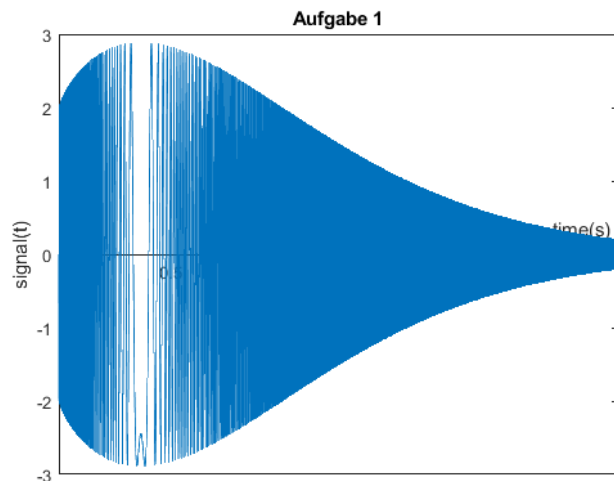


Abbildung 1: Aufgabe 1 Plot

---

<sup>1</sup> Abtastrate

## 2 Aufgabe: Abtasten und Quantisieren

In diesem Abschnitt wird zunächst der Einfluss der Abtastrate und abschließend der Quantisierung eines Signals dargestellt. Als Funktion wird hier das gleiche Signal wie in der Aufgabe zuvor verwendet.

### 2.1 Abtastung

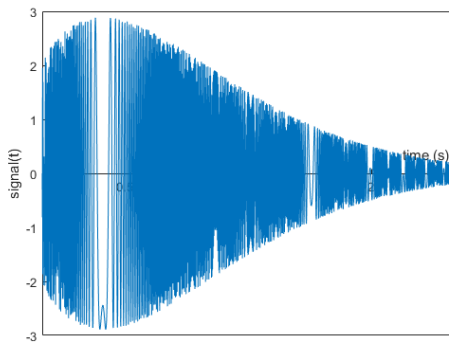


Abbildung 2: Plot 1000Hz

Die Einbrüche des Signals lassen darauf hindeuten, dass die Abtastfrequenz niedriger als 44,1kHz sein muss.

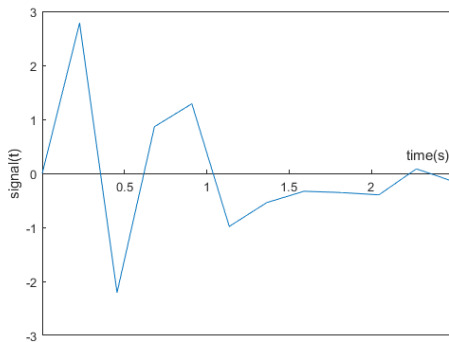


Abbildung 3: Plot 5 Hz

Durch Ablesen der Anzahl der Funktionswerte innerhalb einer Sekunde kann eine Frequenz von 5Hz festgestellt werden.

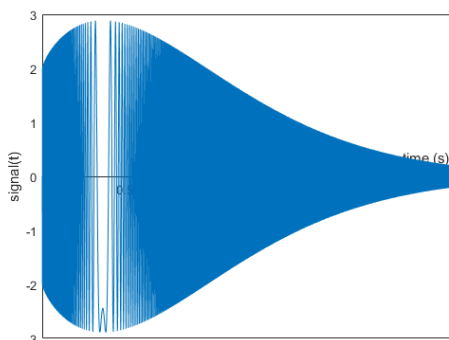


Abbildung 4: Plot > 44,1kHz hier 50kHz

Das Signal ist sehr genau und einzelne Funktionswerte sind nicht ablesbar, die Abtastrate ist hier größer als 44,1kHz. Im Gegensatz zu Abbildung 2 sind wenig Einbrüche erkennbar.

### 2.1.1 Einfluss der Abtastrate

Je höher die Abtastrate ist, desto genauer ist der Funktionsplot. Bei hohen Abtastraten werden Unterschiede im Funktionsplot nicht mehr sichtbar.

### 2.1.2 Beobachtung bei niedriger Abtastrate

Die Ausgabe des Audiosignals bei einer Abtastrate von 1kHz unterscheidet sich stark vom Ausgangssignal (Abtastrate 44,1kHz). Bei einer Abtastrate von unter 1kHz ist keine Audioausgabe möglich, da der Treiber der Soundkarte dies nicht zulässt (Device Error: Invalid sample rate). Um das Ursprungssignal ohne Informationsverlust wiederherstellen zu können, wird eine Abtastrate, welche größer als die doppelte Maximalfrequenz des Signals ist, benötigt. [1] [2]

## 2.2 Quantisierung

Die folgenden Abbildungen zeigen jeweils mit welcher Genauigkeit das Signal bei einer Abtastrate von 1kHz verarbeitet wurde. Es wurde in MATLAB die „round“-Funktion verwendet, welche auf volle Integer Werte rundet. Mittels des Faktors wird der Funktionswert jeweils multipliziert, gerundet und danach wieder durch den Faktor dividiert (s. Anhang 2).

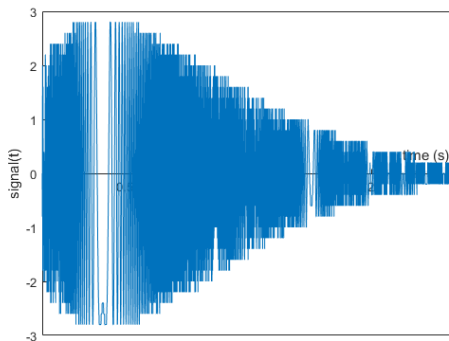


Abbildung 6: Plot Faktor 5

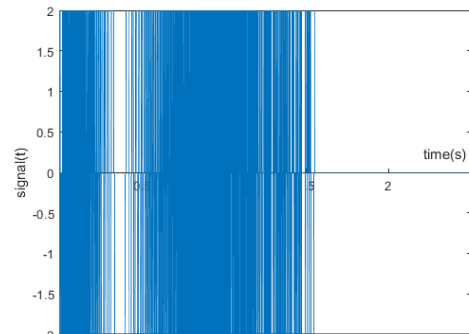


Abbildung 7: Plot Faktor 0,5

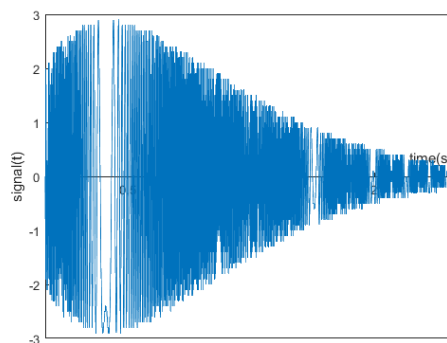


Abbildung 5: Plot Faktor 10

### **2.2.1 Auditive Wahrnehmung der Quantisierung**

Des Weiteren wird nun die Abtastrate auf 44kHz erhöht und es werden weitere Quantisierungen analysiert.

Zehntel: Es ist ein leichtes Kratzen im Audiosignal hörbar.

Viertel: Es ist ein deutliches Kratzen im Audiosignal hörbar.

Hundertstel: Das quantisierte Audiosignal unterscheidet sich kaum vom Ausgangssignal.

Je kleiner der Faktor der Quantisierung, desto mehr Quantisierungsfehler treten im Audiosignal auf.

### **2.3 Abtastung und Quantisierung vs. Speicherplatz und Klangqualität**

Zunächst werden die \*.mat-Files der Größe nach geordnet, um einen Vergleich zu erhalten. Die Auflistungsreihenfolge geht vom größten bis zur kleinsten Datei.

1. Ohne Quantisierung mit 44kHz (1520KB)
2. Quantisierung auf Hundertstel mit 44,1kHz (958KB)
3. Quantisierung auf Zehntel mit 44kHz (803KB)
4. Quantisierung auf Viertel mit 44kHz (740KB)
5. Quantisierung auf Ganzzahlen mit 44kHz (703KB)
6. Quantisierung auf Hundertstel mit 22kHz (484KB)
7. Quantisierung auf Hundertstel mit 1kHz (22KB)

Es wird deutlich, dass die Abtastrate ein größeres Kriterium für die Dateigröße, als die Quantisierung ist. Durch Verringerung der Abtastrate lässt sich somit die Dateigröße stärker verkleinern als durch eine grobe Quantisierung.

Die Dateien 3, 4, 5 mit einer Viertel, Zehntel und Ganzzahligen Quantisierung, weisen ein hörbares Rauschen auf. Bei einer zu niedrigen Abtastrate, Datei 7, wird der Ton des quantisierten Signals verschoben und ähnelt somit dem Ausgangssignal nicht mehr. Somit haben die Dateien 1, 2 und 6 die beste Klangqualität.

### 3 Vom Zeit- zu Frequenzbereich

Um vom Amplitudenspektrum zur Signalgleichung zu kommen, werden jeweils die Grundfrequenzen der x-Achse und die Amplituden der y-Achse abgelesen und in die Ausgangsformel einer Sinusfunktion eingesetzt.

$$ausgangsform(t) = A \cdot \sin(f(\text{Hz}) \cdot F \cdot \pi \cdot t)$$

Somit ergibt sich folgende Signalgleichung mit  $F = 2$ :

$$\begin{aligned} signal(t) = & \sin(1 \cdot F \cdot \pi \cdot t) + 0,5 \cdot \sin(2 \cdot F \cdot \pi \cdot t) + 0,3 \cdot \sin(3 \cdot F \cdot \pi \cdot t) \\ & + 0,1 \cdot \sin(10 \cdot F \cdot \pi \cdot t) + 0,08 \cdot \sin(12 \cdot F \cdot \pi \cdot t) \end{aligned}$$

### 4 Frequenzbereich einer Audiodatei

In dieser Aufgabe wird ein neues Skript erstellt, welches eine Audiodatei einliest, die eingelesenen Werte entsprechend der Abtastrate in 0,5 Sekunden Stücke aufteilt und auf diese Wertebereiche eine FFT() Funktion anwendet. Anschließend werden die Amplituden über die Frequenz und Zeit geplottet.

#### 4.1 3D Frequenzplot

Nach dem Einlesen der Audiodatei wird ein 840432x1 Vektor erstellt. Dieser wird in 0,5 Sekunden Stücke aufgeteilt, indem jeweils 24000 Werte (Abtastrate \* 0,5) als eigene Spalte in einer Matrix gespeichert werden. Insgesamt werden so 36 Spalten erhalten, da der Letzte Vektor nur 432 Werte enthält, werden die fehlenden Werte mit 0 aufgefüllt. Die FFT() Funktion wendet Spaltenweise eine diskrete Fourier-Transformation auf die Matrix an. Die erhaltenen Amplitudenwerte lassen sich mit einer surf() Funktion über die Frequenz und Zeit plotten. Als Parameter werden der surf() Funktion die Frequenz (sampleRate), die Zeit (Spielzeit der Audiodatei) sowie die Amplitudenwerte übergeben. Mit view(24,33) wird ein geeigneter Kamerawinkel gewählt.

#### 4.2 3D Frequenzanalyse und Klangqualität

Ein Unterschied bei den Beiden Frequenzplots ist, dass sich die Amplitudenwerte unterscheiden. Dies erklärt den dumpfen Klang der Datei mit niedriger Daten- und Abtastrate.



#### IV. Literaturverzeichnis

- [1] „MathWorks,“ [Online]. Available: <https://de.mathworks.com/help/matlab/ref/sound.html>. [Zugriff am 23.11.2020].
- [2] „Wikipedia,“ [Online]. Available: <https://de.wikipedia.org/wiki/Nyquist-Shannon-Abtasttheorem>. [Zugriff am 23.11.2020].

## **VI. Anhang**

<b>Anhang 1: Quellcode Aufgabe 1 .....</b>	<b>8</b>
<b>Anhang 2: Quellcode Aufgabe 2.2 .....</b>	<b>8</b>

## Anhang 1: Quellcode Aufgabe 1

```
%created by Julius Schoening
%edited by Patrick Felschen, Julian Voss
clc;
close all;
clear;

%variables
A1=2; %amplitude signal 2
playbacktime = 2.5; % in seconds
sampleRate = 44100; % in Hz
F1=600; %frequency signal 1, 600 Hz
plotTime = 2/F1;

t = linspace(0, playbacktime, playbacktime*sampleRate);

%output graph preparation
tt = t.^t;
signal=A1*(sin(pi*F1*tt)./tt); % Signal equation

%plot signal(t) and its amplitude
subplot(1,1,1);
plot(t,signal);

xlim([0, playbacktime]);
xlabel('time(s)');
ylabel('signal(t)');
title('Aufgabe 1');
ax = gca;
ax.XAxisLocation = 'origin';

sound(signal,sampleRate);
```

## Anhang 2: Quellcode Aufgabe 2.2

```
factor = 0.5;
signal=round(A1*(sin(pi*F1*tt)./tt)*factor)/factor;
```