



**HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG**  
UNIVERSITY OF APPLIED SCIENCES

**Signale, Systeme und Sensoren**

# **Digitalisierung**

**J. Altmeyer, M. Kieser**

**Konstanz, 18. Januar 2016**

## **Zusammenfassung (Abstract)**

Thema:	Digitalisierung	
Autoren:	J. Altmeyer	jualtmey@htwg-konstanz.de
	M. Kieser	makieser@htwg-konstanz.de
Betreuer:	Prof. Dr. Matthias O. Franz	mfranz@htwg-konstanz.de
	Jürgen Keppler	juergen.keppler@htwg-konstanz.de
	Martin Miller	martin.miller@htwg-konstanz.de

Zusammenfassung etwa 100 Worte.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>V</b>
<b>Listingverzeichnis</b>	<b>VI</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Versuch 1 - Genauigkeit der AD-Wandlung</b>	<b>2</b>
2.1 Fragestellung, Messprinzip, Aufbau, Messmittel . . . . .	2
2.2 Messwerte . . . . .	2
2.3 Auswertung . . . . .	2
2.4 Interpretation . . . . .	3
<b>3 Versuch 2 - Genauigkeit der DA-Wandlung</b>	<b>4</b>
3.1 Fragestellung, Messprinzip, Aufbau, Messmittel . . . . .	4
3.2 Messwerte . . . . .	4
3.3 Auswertung . . . . .	4
3.4 Interpretation . . . . .	5
<b>4 Versuch 3 - Zeitverhalten der DA-Wandlung</b>	<b>6</b>
4.1 Fragestellung, Messprinzip, Aufbau, Messmittel . . . . .	6
4.2 Messwerte . . . . .	6
4.3 Auswertung . . . . .	8
4.4 Interpretation . . . . .	8
<b>5 Versuch 4 - Abtasttheorem</b>	<b>9</b>
5.1 Fragestellung, Messprinzip, Aufbau, Messmittel . . . . .	9
5.2 Messwerte . . . . .	9

5.3	Auswertung . . . . .	14
5.4	Interpretation . . . . .	14
<b>Anhang</b>		<b>15</b>
A.1	Quellcode für Versuche 1 - 4 . . . . .	15
A.2	Messergebnisse . . . . .	20

# Abbildungsverzeichnis

4.1	Generierter Sinus . . . . .	7
4.2	Sinus Ausschnitt . . . . .	7
5.1	Spektrum Sinus mit 1000 Hz . . . . .	10
5.2	Spektrum Sinus mit 2000 Hz . . . . .	10
5.3	Spektrum Sinus mit 3000 Hz . . . . .	11
5.4	Spektrum Sinus mit 4000 Hz . . . . .	11
5.5	Spektrum Sinus mit 5000 Hz . . . . .	12
5.6	Spektrum Sinus mit 6000 Hz . . . . .	12
5.7	Spektrum Sinus mit 7000 Hz . . . . .	13
5.8	Spektrum Sinus mit 8000 Hz . . . . .	13
5.9	Zeitverlauf Sinus 8000 Hz . . . . .	14
6.10	Genauigkeitswerte der AD Wandlung . . . . .	20
6.11	Genauigkeitswerte der DA Wandlung . . . . .	21

# **Tabellenverzeichnis**

# Listingverzeichnis

6.1	QuellCodeV1 bis V4 . . . . .	15
-----	------------------------------	----

# 1

## Einleitung

In dieser Versuchsreihe, werden Analog-Digital bzw. Digital-Analog Umwandlungen betrachtet. Zunächst wird die Genauigkeit eines AD bzw. DA Wandlers betrachtet. Daraufhin wird das Zeitverhalten der DA Wandlung anhand eines Sinus Signals betrachtet. Abschließend wird unter Beachtung/ nicht Beachtung des Abtasttheorems das Verhalten der DA-Wandlung im Grenzbereich analysiert.



## 2

# Versuch 1 - Genauigkeit der AD-Wandlung

Nachfolgend wird eine Messung der Genauigkeit einer Analog-Digital Wandlung durchgeführt.

## 2.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Wie genau ist, der AD-Wandler des ME-RedLab USB-1208LS? Um dies heraus zu finden, wird zum einen der theoretische Quantisierungsfehler betrachtet, welcher die Genauigkeit in Volt angibt, zum anderen wird der Messfehler im Vergleich zu einem Feinmessgerät(Keithley TRMS 179)ermittelt. Um Vergleichswerte zu haben, wird der Messfehler des Multimeter Philips PM 2503 ebenfalls im Vergleich zu einem Feinmessgerät ermittelt.

## 2.2 Messwerte

Siehe Abbildung 6.10 im Anhang A.2.

## 2.3 Auswertung

**theoretischer Quantisierungsfehler:** Bei einem Eingangsspannungsbereichs von -10 V bis 10V des 11-Bit-AD-Wandlers ergibt sich ein theoretischer Quantisierungsfehler von  $\Delta U = 0,0098V$

### **Standardabweichung:**

Multimeter Philips Std  $s=0.02862533842594704$

AD Wandler Std  $s=0.0029325756597230355$

## **2.4 Interpretation**

Multimeter Philips: Standardabweichung  $s= 30\text{mV}$  Dies bedeutet, dass eine analoge Eingangsspannung in eine um  $\pm 30\text{mV}$  abweichende Ausgangsspannung gewandelt wird. Der folgende AD Wandler kann dies wesentlich genauer.

AD Wandler: Standardabweichung  $s= 3\text{mV}$  Dies bedeutet, dass eine analoge Eingangsspannung in eine um  $\pm 3\text{mV}$  abweichende Ausgangsspannung gewandelt wird.

Der theoretische Quantisierungsfehler  $= 10\text{mV}$  beschreibt die Genauigkeit des AD-Wandlers. So kann der AD Wandler beispielsweise eine Eingangsspannung von  $1\text{mV}$  nicht von  $9\text{mV}$  unterscheiden.

Zusammenfassend kann man sagen, dass im wesentlichen der Quantisierungsfehler an der Ungenauigkeit des AD-Wandlers schuld ist. Eine Erhöhung der Bit Zahl würde hier also auch zu einer Erhöhung der Genauigkeit der AD-Wandlung führen.

# 3

## Versuch 2 - Genauigkeit der DA-Wandlung

Nachfolgend wird eine Messung der Genauigkeit einer Digital-Analog Wandlung durchgeführt.

### 3.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Wie genau ist, der DA-Wandler des ME-RedLab USB-1208LS? Um dies heraus zu finden, wird zum einen der theoretische Quantisierungsfehler betrachtet, welcher die Genauigkeit in Volt angibt, zum anderen wird der Messfehler im Vergleich zu einem Feinmessgerät(Keithley TRMS 179)ermittelt.

### 3.2 Messwerte

Siehe Abbildung 6.11 im Anhang A.2.

### 3.3 Auswertung

Bei einem Ausgangsspannungsbereichs von 0V bis 5V des 10-Bit-DA-Wandlers ergibt sich ein theoretischer Quantisierungsfehler von  $\Delta U = 0,0049V$

### 3.4 Interpretation

Der theoretische Quantisierungsfehler = 5mV beschreibt die Genauigkeit des DA-Wandlers. So kann der DA Wandler beispielsweise eine Eingangsspannung von 1mV nicht von 5mV unterscheiden.

DA Wandler: Standardabweichung  $s=28\text{mV}$  Dies bedeutet, dass eine digitale Eingabewert in eine um  $\pm 28\text{mV}$  abweichende Ausgangsspannung gewandelt wird.

Zusammenfassend ist zu sagen, dass die Ungenauigkeit des DA Wandlers überwiegend von der großen Standardabweichung abhängt. Der Fehler der durch Quantisierung entsteht ist vergleichsweise gering. Eine Erhöhung der Bit zahl würde demnach keine signifikante Auswirkung auf die Genauigkeit haben.

# 4

## Versuch 3 - Zeitverhalten der DA-Wandlung

In diesen Versuch soll das Zeitverhalten des DA-Wandlers genauer untersucht werden. Hierfür wird mit einem Programm eine Sinusspannung ausgegeben und analysiert.

### 4.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Welche Genauigkeit hat die DA-Wandlung in Bezug auf die Sample Rate? Um dies zu überprüfen, wird ein einfacher Sinus generiert, siehe dazu Listing 6.1 die Methode *get\_sin()*. Dieser besteht aus 100 Werten und ist um 1 auf der Y-Achse nach oben verschoben, da der DA-Wandler nur positive Spannungen im Bereich von 0 bis 5 Volt ausgeben kann. Die einzelnen Werte des Sinus werden nun hintereinander mit einer Pause von 10 ms auf die Karte des DA-Wandlers geschrieben (Listing 6.1, Methode *versuch4()*). Mit dem Oszilloskop wird dieses Signal aufgezeichnet.

### 4.2 Messwerte

Das erzeugte Sinus Signal des DA-Wandlers wurde mit dem Oszilloskop aufgenommen und abgespeichert. Dieser Sinus wird in Abbildung 4.1 dargestellt. In Abbildung 4.2 ist ein Teilausschnitt des Signals zu sehen.

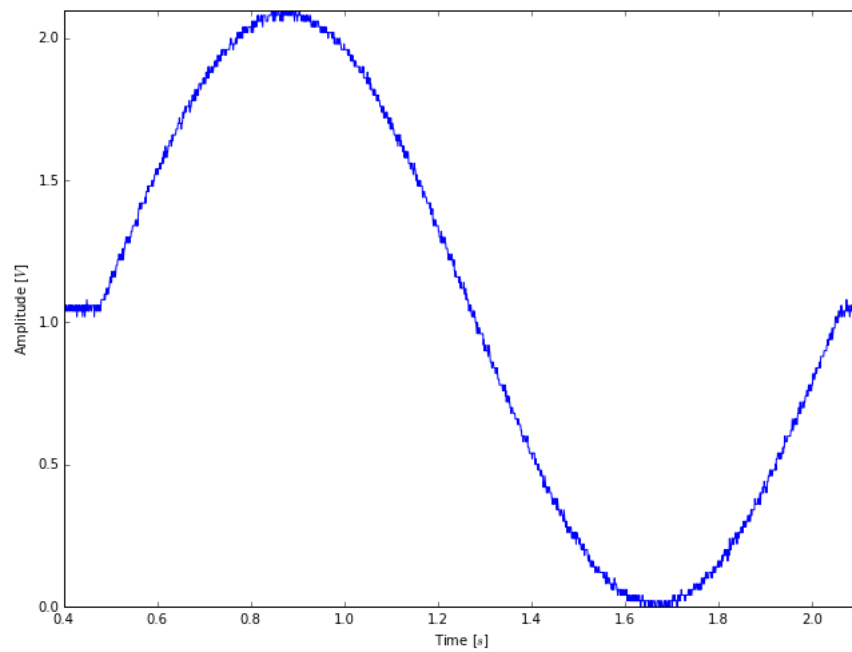


Abbildung 4.1: Generierter Sinus

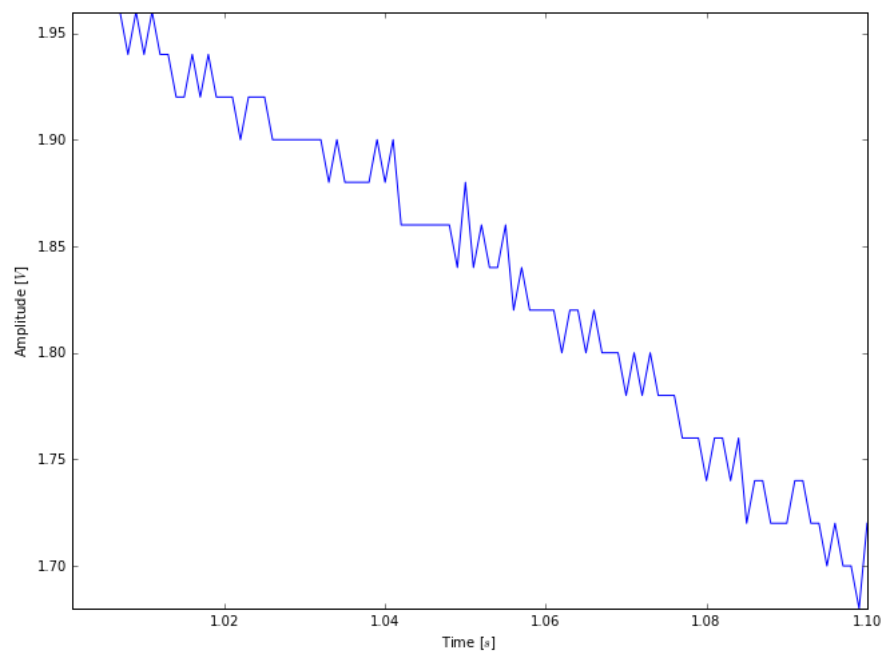


Abbildung 4.2: Sinus Ausschnitt

### 4.3 Auswertung

In dem Ausschnitt des Sinus (Abbildung 4.2) sind nun die einzelnen Stufen zu erkennen. Da das Signal Spannungsschwankungen aufweist, ist es schwer, einen Stufenübergang eindeutig zu bestimmen. Durch ablesen bzw. Vermessen der Dauer der Stufen ergibt sich somit ein mittleres  $\Delta t$  von ungefähr  $10ms$  und damit eine Frequenz von ca. 100 Hz. Die Zeit  $\Delta t$  variiert ein wenig von Stufe zu Stufe.

### 4.4 Interpretation

Da ein Sinus Signal mit einer Sample Rate von 100 Hz auf die Karte des DA-Wandlers geschrieben wurde, war zu erwarten, dass die einzelnen Stufen bzw. Samples auch in diesem Zeit-Bereich liegen. Es wurden aber auch leichte Abweichungen (Jitter) festgestellt, was zeigt, dass die DA-Wandlung nicht optimal ist.

Auffallend ist auch, dass das komplette Sinus Signal eine Dauer von 1.6 Sekunden aufweist. Eigentlich müsste das Signal genau eine Sekunde lang sein (Sinus mit 1 Hz), weil wir 100 Sinus-Werte mit einer Sample Rate von 100 Hz auf die Karte schreiben. Das könnte daran liegen, dass der DA-Wandler maximal 100 Samples pro Sekunde ausgeben kann und wir somit genau an der Grenzfrequenz liegen. Dadurch kommt der DA-Wandler eventuell nicht "hinterher" und ist ab und zu langsamer.

# 5

## Versuch 4 - Abtasttheorem

Dieser Versuch zeigt, was passiert, wenn man das Abtasttheorem nicht einhält und ignoriert. Anhand von diversen Sinus Signalen wird dies untersucht.

### 5.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Es soll geklärt werden, welche Auswirkungen und welche Bedeutung das Abtasttheorem für die AD-Wandlung hat. Dafür werden mit einem externen Sinusgenerator 8 Signale erzeugt. Diese unterscheiden sich in ihrer Frequenz. Beginnend mit 1000 Hz wird die Frequenz in 1000 Hz Schritten erhöht, bis eine Frequenz von 8000 Hz erreicht ist. Der AD-Wandler nimmt diese Signale auf und mit dem Programm (Listing 6.1) werden sie abgespeichert und verarbeitet.

### 5.2 Messwerte

Folgend sind die Spektren aller erzeugten Signale abgebildet. Abbildung 5.9 zeigt zusätzlich den zeitlichen Verlauf des erzeugten Signals mit 8000 Hz.



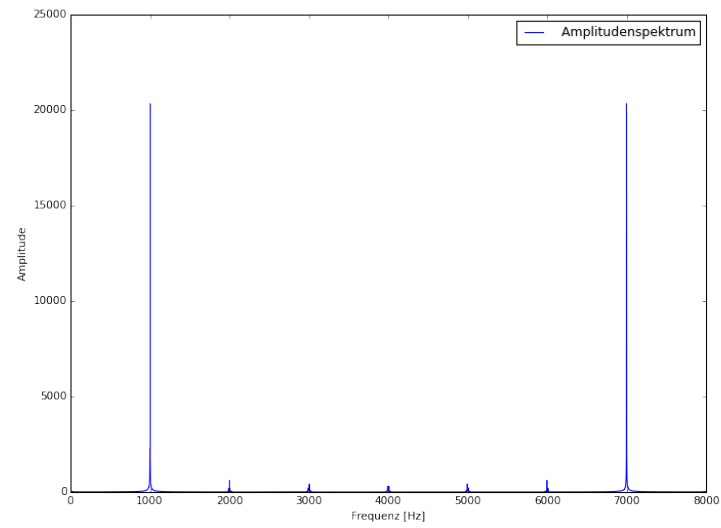


Abbildung 5.1: Spektrum Sinus mit 1000 Hz

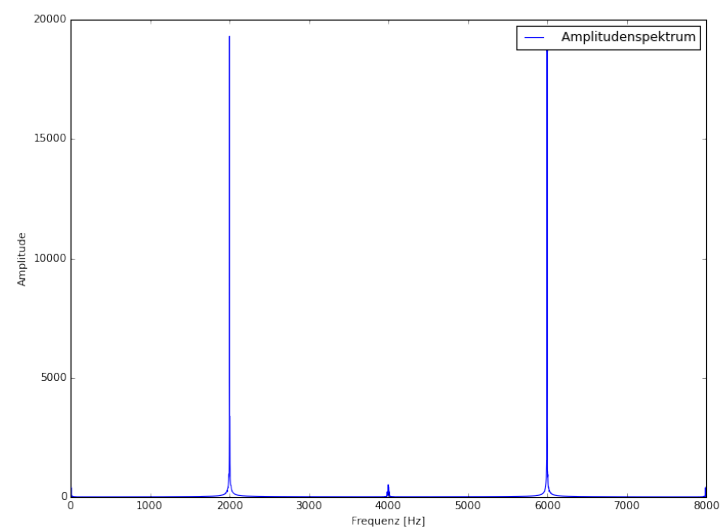


Abbildung 5.2: Spektrum Sinus mit 2000 Hz

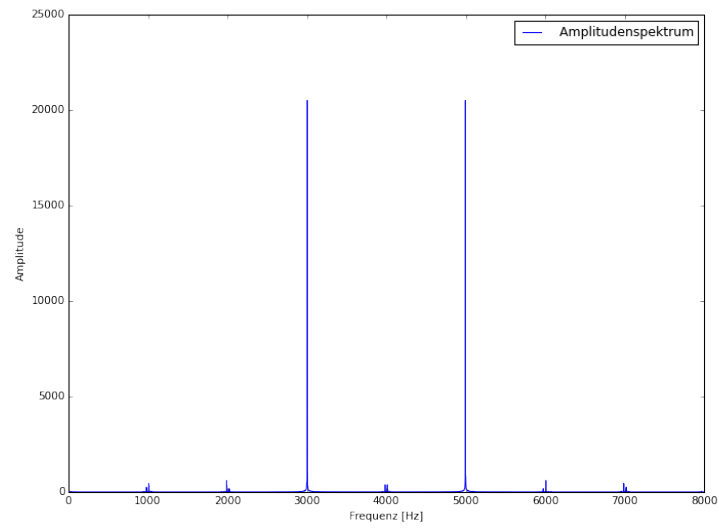


Abbildung 5.3: Spektrum Sinus mit 3000 Hz

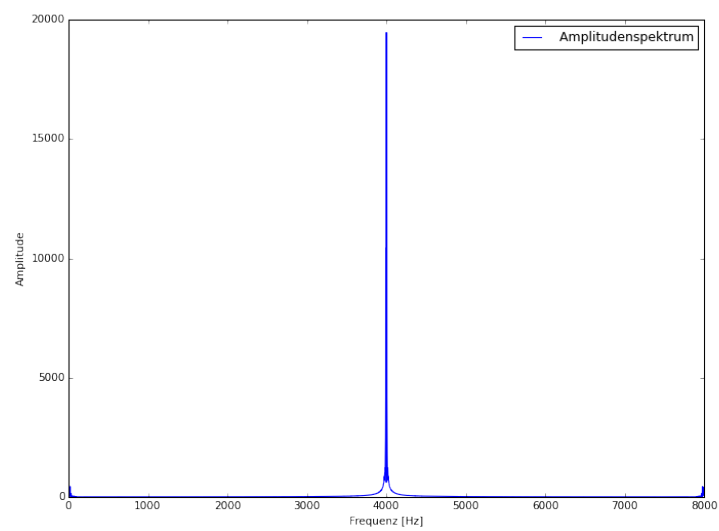


Abbildung 5.4: Spektrum Sinus mit 4000 Hz

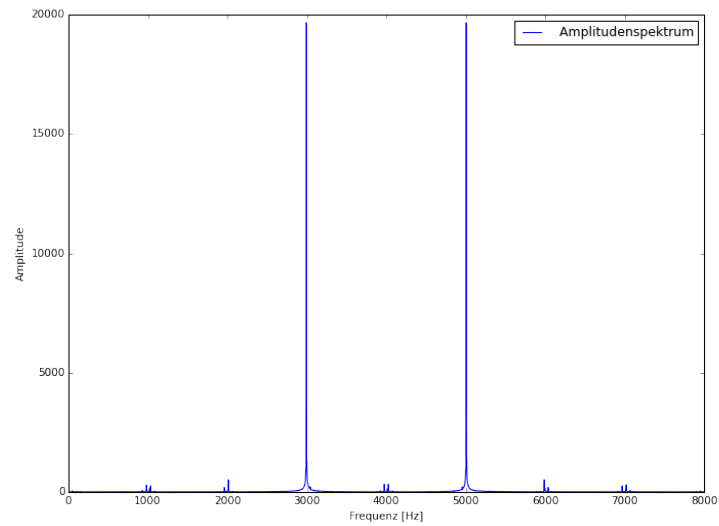


Abbildung 5.5: Spektrum Sinus mit 5000 Hz

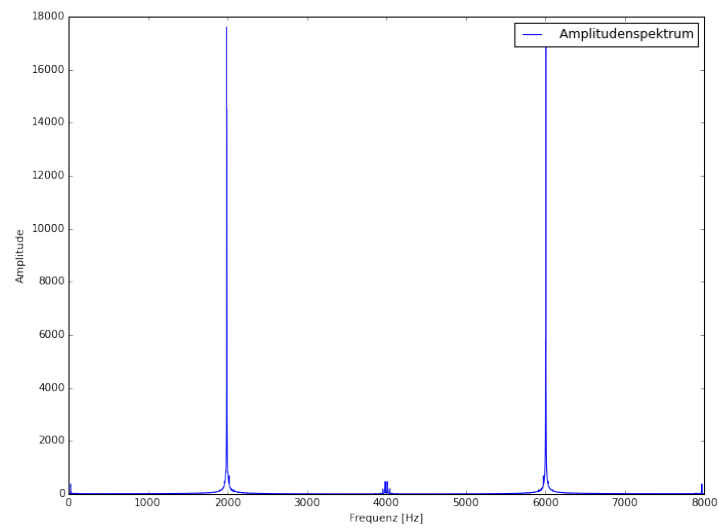


Abbildung 5.6: Spektrum Sinus mit 6000 Hz

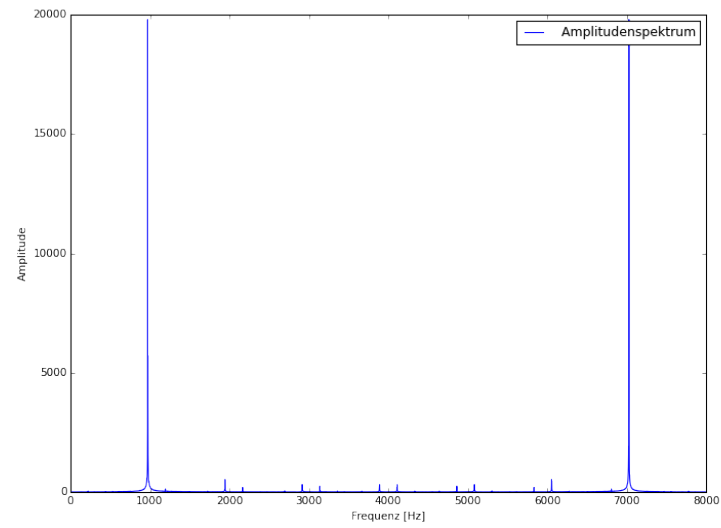


Abbildung 5.7: Spektrum Sinus mit 7000 Hz

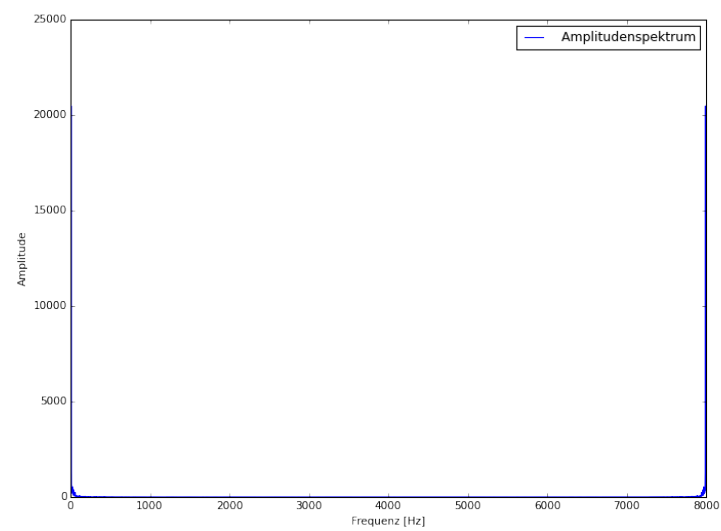


Abbildung 5.8: Spektrum Sinus mit 8000 Hz

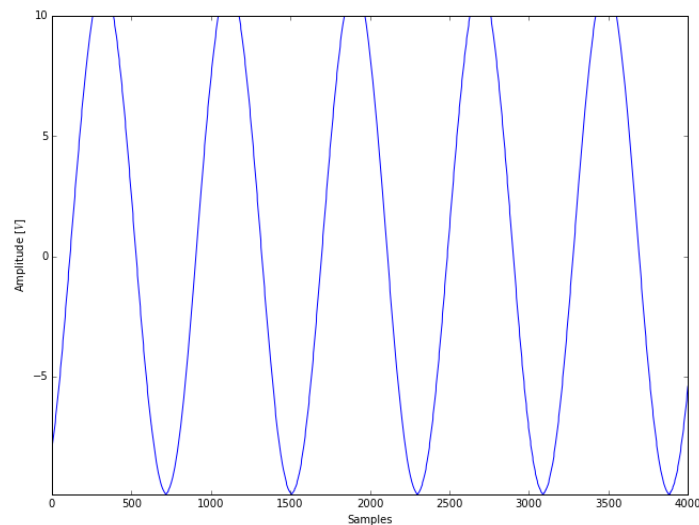


Abbildung 5.9: Zeitverlauf Sinus 8000 Hz

### 5.3 Auswertung

Da der AD-Wandler eine Abtastfrequenz von 8000 Hz besitzt, liegt die Nyquist-Frequenz bei der Hälfte, also 4000 Hz. Das Abtasttheorem besagt nun, dass die maximal vorkommende Frequenz im abgetasteten Signal nicht größer als die Nyquist-Frequenz sein darf, um das Signal verlustfrei rekonstruieren zu können. D.h. es sollten in diesem Fall nur Frequenzen mit höchstens 4000 Hz aufgenommen werden.

### 5.4 Interpretation

Die aufgenommenen Spektren zeigen nun sehr schön, dass sich ab der Nyquist-Frequenz die beiden Peaks "überholen". Man würde dadurch beispielsweise denken, dass die Frequenz des Signals mit 6000 Hz (Abbildung 5.6) 5000 Hz beträgt, in Wahrheit liegt sie jedoch bei 6000 Hz. Dieser Fehler gilt für alle Frequenzen über der Nyquist-Frequenz. Im zeitlichen Verlauf in Abbildung 5.9 kann man erkennen, dass das aufgenommene Signal keineswegs 8000 Hz beträgt, wie es eigentlich sein sollte. Stattdessen liegt die Frequenz des Sinus bei ca. 10 Hz. Anmerkung: Eigentlich müssten es 0 Hz sein, aber aufgrund von Ungenauigkeiten entsteht dieser Wert.

Offensichtlich führt das Überschreiten der Nyquist-Frequenz zu einer Fehlinterpretation des Signals (Spektrums). Dadurch werden Frequenzen ermittelt bzw. hinzugefügt, welche in dem originalen Signal gar nicht vorgekommen sind.

# Anhang

## A.1 Quellcode für Versuche 1 - 4

```
1
2 # -*- coding: utf-8 -*-
3 """
4 Created on Mon Jan 11 14:15:07 2016
5
6 @author: edc07
7 """
8
9
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 import redlab as rl
14 import time
15 from TekTDS2000 import *
16
17
18 def versuch1():
19     out(1)
20     #print(str(get_input(4000, 8000)))
21     print('fertig')
22
23
24 def versuch2():
25     #print(str(np.mean(get_input(4000, 8000))))
26
27     mult_array = np.array([0.103, 0.196, 0.2, 0.2, 0.194, 0.198, 0.199, 0.199, 0.2, 0.198])
28     ad_array = np.array([0.015, 0.018, 0.013, 0.015, 0.016, 0.013, 0.012, 0.012, 0.014, 0.022])
29     print("Multimeter Philips Std s={}".format(getStd(mult_array)))
30     print("AD Wandler Std s={}".format(getStd(ad_array)))
```

```

31
32 def versuch3():
33     da_array = np.array([0.011, 0.019, 0.027, 0.041, 0.049, 0.058, 0.072, 0.080, 0.090, 0.096])
34     print("DA Wandler Std s={ }".format(getStd(da_array)))
35
36 def versuch4():
37     for x in get_sin():
38         out(x)
39         time.sleep(0.01)
40
41     save_input_oszi()
42     print(getInputData('sinus.csv')[0])
43
44     plotRecord(getInputData('sinus.csv'))
45
46
47 def versuch5():
48     np.savetxt('8000.csv', rl.cbVInScan(0, 0, 0, 4000, 8000, 1))
49
50     plotFFT(getInputData('1000.csv'), 8000, '1000fft.png')
51     plotFFT(getInputData('2000.csv'), 8000, '2000fft.png')
52     plotFFT(getInputData('3000.csv'), 8000, '3000fft.png')
53     plotFFT(getInputData('4000.csv'), 8000, '4000fft.png')
54     plotFFT(getInputData('5000.csv'), 8000, '5000fft.png')
55     plotFFT(getInputData('6000.csv'), 8000, '6000fft.png')
56     plotFFT(getInputData('7000.csv'), 8000, '7000fft.png')
57     plotFFT(getInputData('8000.csv'), 8000, '8000fft.png')
58
59
60 def plotFFT(rec, sampleRate, filename=""):
61
62     #fft
63     # n = Anzahl der Schwingungen innerhalb der gesamten Signaldauer
64     c = np.fft.fft(rec)
65     n = np.abs(c)
66     sampleTime = 1 / sampleRate
67
68     count = np.arange(0, len(n)) * (1 / (len(n) * sampleTime))
69
70     # Anzahl der Schwingungen innerhalb der gesamten Signaldauer dargestellt
71     dpi=75
72     fig, axN = plt.subplots(figsize=(800/dpi,600/dpi), dpi=dpi)

```

```

73 axN.plot(count[:,n:], color = "blue", label=" Amplitudenspektrum ")
74 # lässt X-Achse bei 0 beginnen
75 axN.autoscale(enable=True, axis='x', tight=True)
76 axN.legend(loc='upper right');
77 axN.set_xlabel("Frequenz [Hz]")
78 axN.set_ylabel("Amplitude")
79
80 # als png abspeichern
81 if filename is not '':
82     fig.savefig(filename, transparent=True, dpi=dpi)
83 return
84
85
86 def getStd(e_array):
87     return np.std(e_array)
88
89 def plotRecord(rec):
90     myDpi = 75
91     fig, ax = plt.subplots(figsize=(800/myDpi, 600/myDpi), dpi=myDpi)
92     ax.autoscale(enable=True, axis='x', tight=True)
93     ax.plot(rec[400:2100,0], rec[400:2100,1])
94     ax.set_xlabel('Time [$s$]')
95     ax.set_ylabel(' Amplitude [$V$]')
96
97
98 def save_input_oszi():
99     scope = TekTDS2000()
100
101     x,y = scope.getData(1,1,2500)
102     np.savetxt("sinus.csv" , np.transpose([x,y]), delimiter=",")
103
104
105 def getInputData(filename):
106     return np.genfromtxt(filename, delimiter=',')
107
108
109 def get_input(number, samplerate):
110     return rl.cbVInScan(0, 0, 0, number, samplerate, 1)
111
112
113 def out(voltage):
114     rl.cbVOut(0, 0, 101, voltage)

```



```
115
116
117 def get_sin(fs=100):
118     val = np.linspace(0, 2 * np.pi, fs)
119     return np.sin(val) + 1
120
121
122 def main():
123     versuch1()
124     versuch2()
125     versuch3()
126     versuch4()
127     versuch5()
128
129 if __name__ == '__main__':
130     main()
```

Listing 6.1: QuellCodeV1 bis V4



## A.2 Messergebnisse

2. Genauigkeit der AD Wandlung

2. Feinmessgerät analog AD Wandler

Keithley TR75179 Multimeter Philips HI 2503

V [Volt]	$u_{i,ref}$	$u_i$	$e_i$	$u_i$	$e_i$
1	0,997	1,1	-0,103	0,982	0,015
2	2,004	2,2	-0,196	1,986	0,018
3	3,000	3,2	-0,200	2,987	0,013
4	4,000	4,2	-0,200	3,985	0,015
5	5,006	5,2	-0,194	4,990	0,016
6	6,002	6,2	-0,198	5,989	0,013
7	7,001	7,2	-0,199	6,989	0,012
8	8,001	8,2	-0,199	7,989	0,012
9	9,000	9,2	-0,200	8,986	0,014
10	10,002	10,2	-0,198	9,980	0,022

$$e_i = u_{i,ref} - u_i$$

11.1.2016

*[Signature]*

Abbildung 6.10: Genauigkeitswerte der AD Wandlung

### 3. Genauigkeit der DA-Wandlung

Digitalwert von Konvertereingabe	Feinmessgerät	
$U_{i,ref}$	$U_i$	$e_i$
0,5	0,511	-0,011
1,0	1,019	-0,019
1,5	1,527	-0,027
2,0	2,041	-0,041
2,5	2,549	-0,049
3,0	3,058	-0,058
3,5	3,572	-0,072
4,0	4,080	-0,080
4,5	4,590	-0,090
5,0	5,096	-0,096

$$e_i = U_{i,ref} - U_i$$

11. 1. 2016

M. Mitz

Abbildung 6.11: Genauigkeitswerte der DA Wandlung