



HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
UNIVERSITY OF APPLIED SCIENCES

Signale, Systeme und Sensoren

VERSUCH NAME

J. Altmeyer, M. Kieser

Konstanz, 13. Januar 2016

Zusammenfassung (Abstract)

| | | |
|-----------|-----------------------------|----------------------------------|
| Thema: | VERSUCH NAME | |
| 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

| | |
|--|-----------|
| Abbildungsverzeichnis | IV |
| Tabellenverzeichnis | V |
| Listingverzeichnis | VI |
| 1 Einleitung | 1 |
| 2 Versuch 1 - Genauigkeit der AD-Wandlung | 2 |
| 2.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 2 |
| 2.2 Messwerte | 2 |
| 2.3 Auswertung | 2 |
| 2.4 Interpretation | 2 |
| 3 Versuch 2 - Genauigkeit der DA-Wandlung | 4 |
| 3.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 4 |
| 3.2 Messwerte | 4 |
| 3.3 Auswertung | 4 |
| 3.4 Interpretation | 4 |
| 4 Versuch 3 - Zeitverhalten der DA-Wandlung | 5 |
| 4.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 6 |
| 4.2 Messwerte | 6 |
| 4.3 Auswertung | 7 |
| 4.4 Interpretation | 7 |
| 5 Versuch 4 - Abtasttheorem | 8 |
| 5.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 8 |
| 5.2 Messwerte | 8 |

| | | |
|---------------|--|-----------|
| 5.3 | Auswertung | 12 |
| 5.4 | Interpretation | 12 |
| Anhang | | 13 |
| A.1 | Quellcode für Versuche 1 - 4 | 13 |
| A.2 | Messergebnisse | 15 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 4.1 | generierter Sinus | 6 |
| 4.2 | Sinus Ausschnitt | 7 |
| 5.1 | 1000Hz FFT | 8 |
| 5.2 | 2000Hz FFT | 9 |
| 5.3 | 3000Hz FFT | 9 |
| 5.4 | 4000Hz FFT | 10 |
| 5.5 | 5000Hz FFT | 10 |
| 5.6 | 6000Hz FFT | 11 |
| 5.7 | 7000Hz FFT | 11 |
| 5.8 | 8000Hz FFT | 12 |
| 6.9 | Genauigkeitswerte der AD Wandlung | 15 |
| 6.10 | Genauigkeitswerte der DA Wandlung | 16 |

Tabellenverzeichnis

Listingverzeichnis

| | | |
|-----|------------------------------|----|
| 6.1 | QuellCodeV1 bis V4 | 13 |
|-----|------------------------------|----|

1

Einleitung

2

Versuch 1 - Genauigkeit der AD-Wandlung

2.1 Fragestellung, Messprinzip, Aufbau, Messmittel

2.2 Messwerte

2.3 Auswertung

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= 30mV Dies bedeutet, dass eine analoge Eingangsspannung in eine um $\pm 30mV$ abweichende Ausgangsspannung gewandelt wird. Der folgende AD Wandler kann dies wesentlich genauer.

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

Der theoretischer Quantisierungsfehler = 10mV beschreibt die Genauigkeit des AD-Wandlers. So kann der AD Wandler beispielsweise eine Eingangsspannung von 1mV nicht von 9mV unterscheiden.

Quantisierungsfehler ist schuld!!!!!!

3

Versuch 2 - Genauigkeit der DA-Wandlung

3.1 Fragestellung, Messprinzip, Aufbau, Messmittel

3.2 Messwerte

3.3 Auswertung

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

3.4 Interpretation

Der theoretischer 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 $s=28mV$ Dies bedeutet, dass eine digitale Eingabewert in eine um $\pm 28mV$ abweichende Ausgangsspannung gewandelt wird.

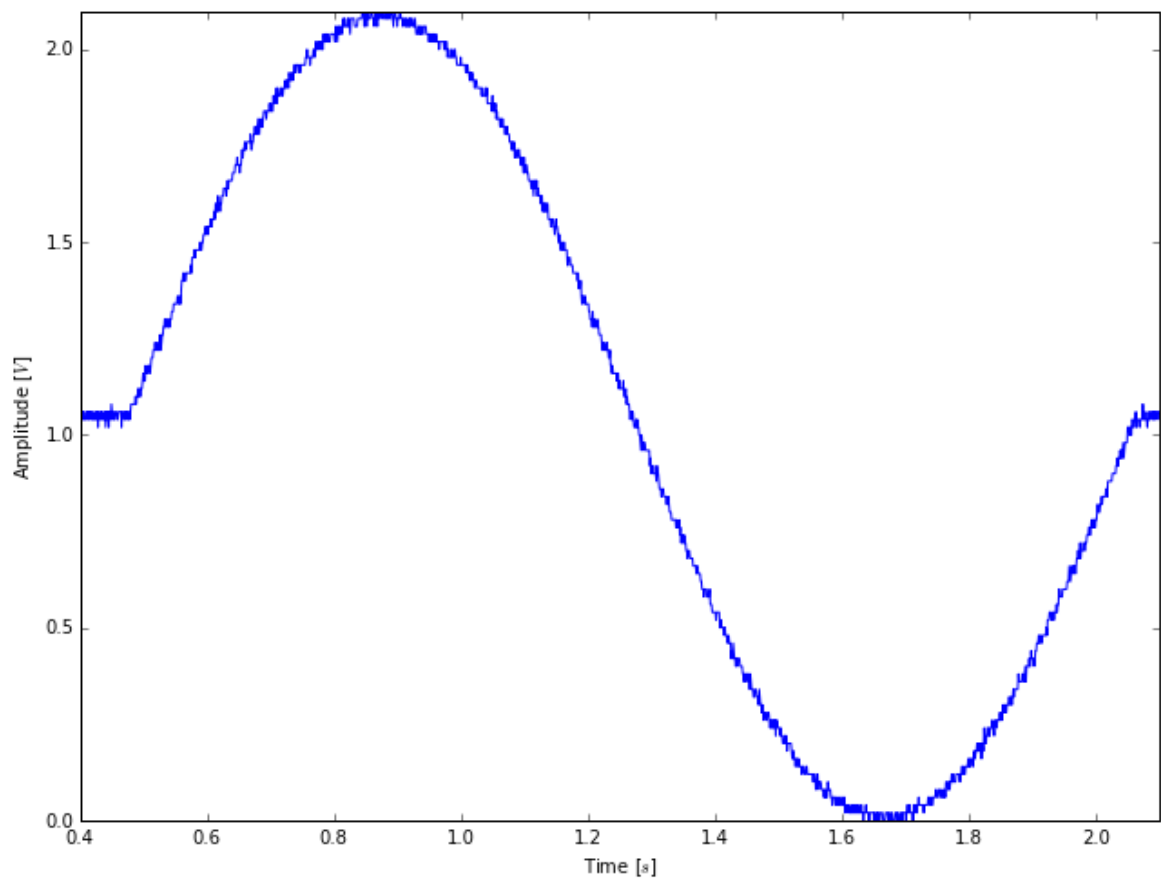
DA Wandler ist schuld!!!!!!!!!!!!!!!!!!!!

4

Versuch 3 - Zeitverhalten der DA-Wandlung

4.1 Fragestellung, Messprinzip, Aufbau, Messmittel

4.2 Messwerte



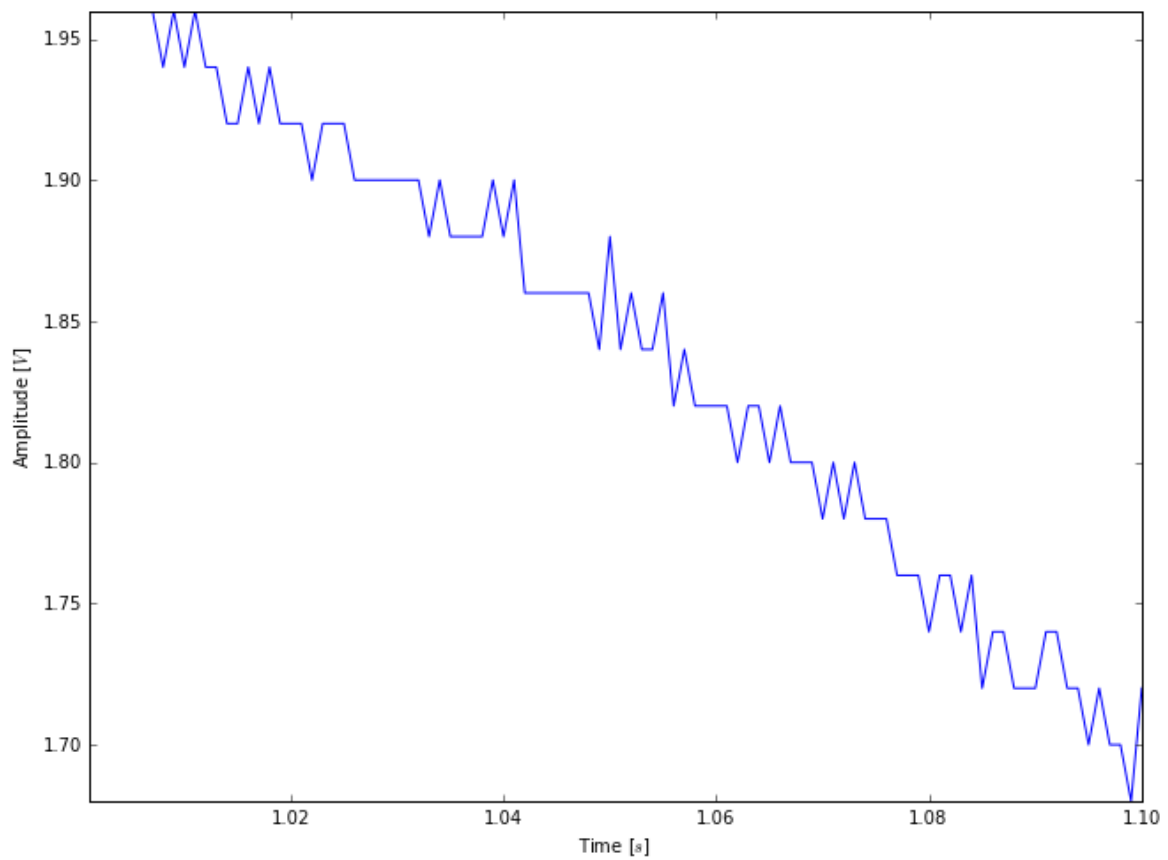


Abbildung 4.2: Sinus Ausschnitt

4.3 Auswertung

Durch ablesen aus der Abbildung 4.2 ergibt sich ein $\Delta t = 10ms$

4.4 Interpretation

Somit ergibt sich ca. eine maximale Ausgabefrequenz von 100Hz

5

Versuch 4 - Abtasttheorem

5.1 Fragestellung, Messprinzip, Aufbau, Messmittel

5.2 Messwerte

Abtastfrequenz= 8000Hz

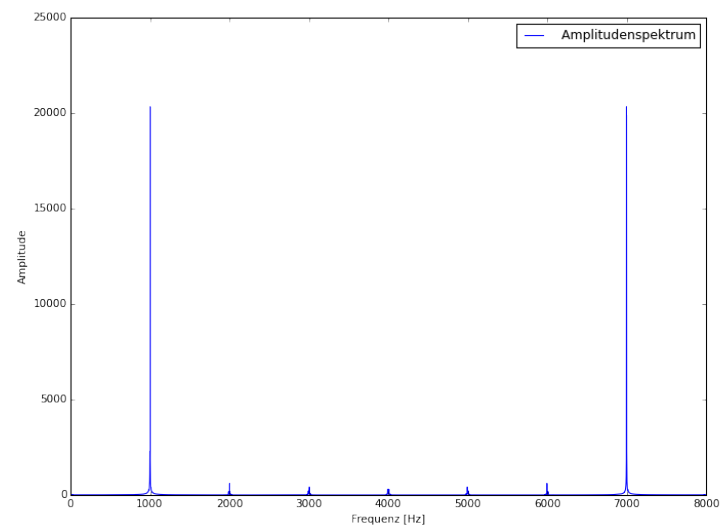


Abbildung 5.1: 1000Hz FFT

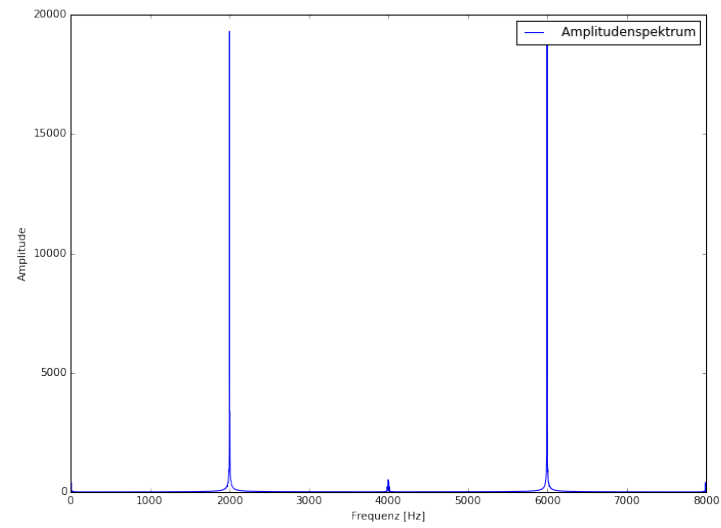


Abbildung 5.2: 2000Hz FFT

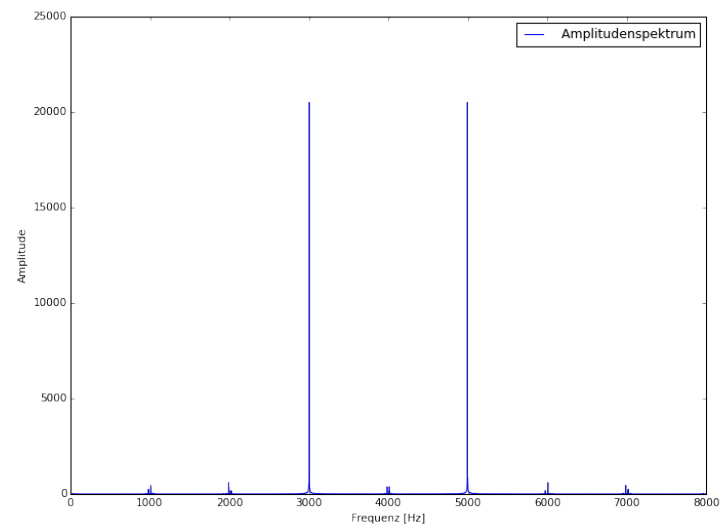


Abbildung 5.3: 3000Hz FFT

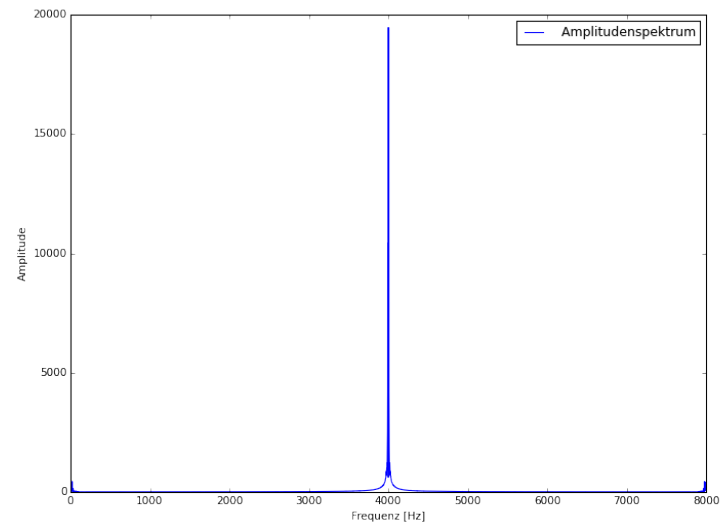


Abbildung 5.4: 4000Hz FFT

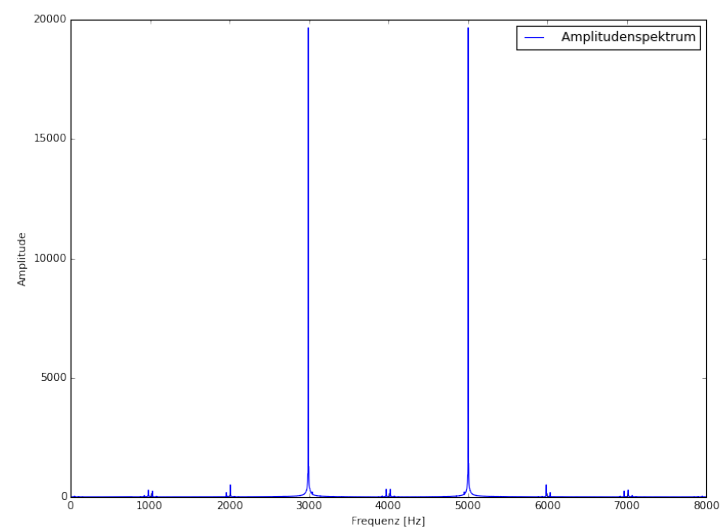


Abbildung 5.5: 5000Hz FFT

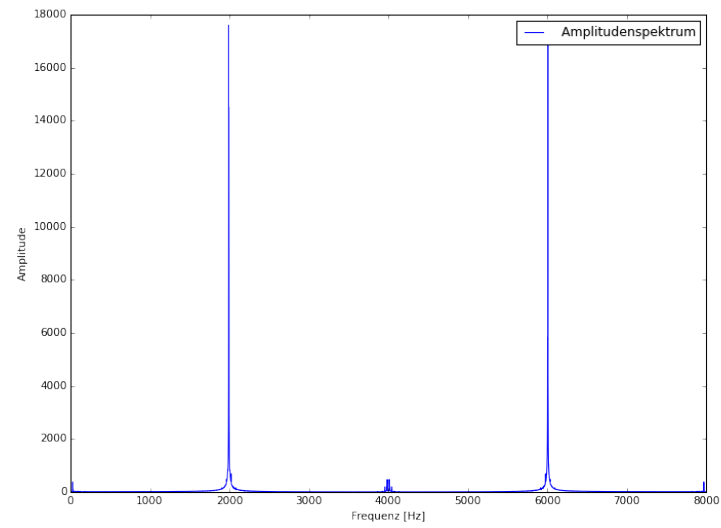


Abbildung 5.6: 6000Hz FFT

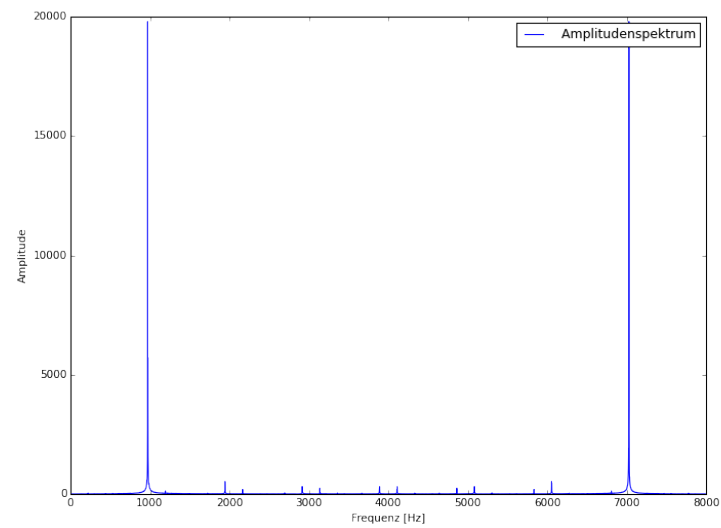


Abbildung 5.7: 7000Hz FFT

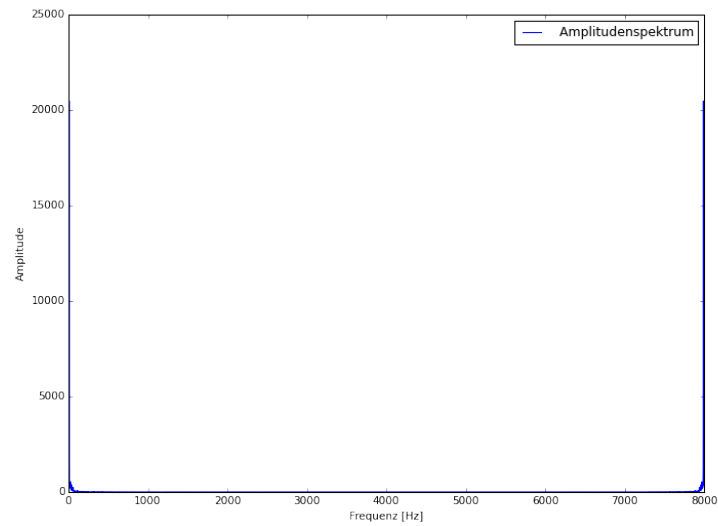


Abbildung 5.8: 8000Hz FFT

5.3 Auswertung

Bei obiger Abtastfrequenz ergibt sich eine Nyquist-Frequenz = 4000Hz.

5.4 Interpretation

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

$U_{i,ref}$ 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 6.9: Genauigkeitswerte der AD Wandlung

3. Genauigkeit der DA-Wandlung

| Digitalwert von Konvertereingebe | 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 6.10: Genauigkeitswerte der DA Wandlung