



Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Signale, Systeme und Sensoren

Digitalisierung

M. Rudolph, B. Jasper, T. Schoch

Konstanz, 24. Juni 2019

Zusammenfassung (Abstract)

Thema:	Digitalisierung	
Autoren:	M. Rudolph	marvin.rudolph@htwg-konstanz.de
	B. Jasper	benjamin.jasper@htwg-konstanz.de
	T. Schoch	tobias.schoch@htwg-konstanz.de
Betreuer:	Prof. Dr. Matthias O. Franz	mfranz@htwg-konstanz.de
	Jürgen Keppler	juergen.keppler@htwg-konstanz.de
	Christoph Kaiser	christoph.kaiser@htwg-konstanz.de

In diesem Versuch wird die Umwandlung von analogen in digitale Daten und umgekehrt untersucht. Dabei wollen wir vorallem auf folgende Themen eingehen:

Der Umgang mit Analog-zu-Digital und Digital-zu-Analog Wandlern, die Messungen der Genauigkeit der AD- und DA- Wandlung, die Untersuchung des Zeitverhaltens der DA-Wandlung und die praktische Erfahrung mit dem Abtasttheorem.

Diese Themen behandeln wir mit der Multifunktionsbox ME-RedLab USB-1208LS. Mit dieser wandeln wir Analoge- in die Digitale Signale und umgekehrt.

So werden wir Werte auslesen aus der Multifunktionsbox und Werte auf die Multifunktionsbox geben. Zudem werden wir Genauigkeit der AD-Wandlung testen, sowie die Genauigkeit der DA-Wandlung.

Im Anschluss werden wir das Zeitverhalten der DA-Wandlung überprüfen und das Abtasttheorem behandeln.

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Listingverzeichnis	VI
0 Einleitung	1
1 Versuch 1 - Programmierung der AD/DA-Wandlerkarte	2
1.1 Fragestellung, Messprinzip, Aufbau, Messmittel	2
1.2 Messwerte	3
1.3 Auswertung	3
1.4 Interpretation	4
2 Versuch 2 - Genauigkeit der AD-Wandlung	5
2.1 Fragestellung, Messprinzip, Aufbau, Messmittel	5
2.2 Messwerte	6
2.3 Auswertung	6
2.4 Interpretation	6
3 Versuch 3 - Genauigkeit der DA-Wandlung	7
3.1 Fragestellung, Messprinzip, Aufbau, Messmittel	7
3.2 Messwerte	8
3.3 Auswertung	8
3.4 Interpretation	8
4 Versuch 4 - Zeitverhalten der DA-Wandlung	9
4.1 Fragestellung, Messprinzip, Aufbau, Messmittel	9
4.2 Messwerte	10

4.3	Interpretation	10
5	Versuch 5 - Abtasttheorem	11
5.1	Fragestellung, Messprinzip, Aufbau, Messmittel	11
5.2	Messwerte	12
5.3	Auswertung	14
5.4	Interpretation	14
	Anhang	16
A.1	Quellcode Aufgabe 1	16
A.2	Quellcode Aufgabe 4	17
A.3	Quellcode Aufgabe 5	18
A.4	Messprotokolle	19

Abbildungsverzeichnis

1.1	Konsolenausgabe als .txt Datei	3
2.1	Versuchsaufbau	5
3.1	Versuchsaufbau	7
4.1	Ausgegebene Sinusspannung	10
5.1	Messfrequenz von 2000Hz	12
5.2	Messfrequenz von 3000Hz	12
5.3	Messfrequenz von 4010Hz	13
5.4	Messfrequenz von 5000Hz	13
5.5	Messfrequenz von 6000Hz	13
5.6	Messfrequenz von 7000Hz	13
5.7	Messfrequenz von 8020Hz	13
5.8	Schaubild zu Aliasing	14
6.9	Protokoll Aufgabe 3	19
6.10	Protokoll Aufgabe 2	20

Tabellenverzeichnis

1.1	Vergleich der berechneten Referenzspektren	3
2.1	Messwerte Feinmessgerät, Multimeter, AD-Wandler	6
3.1	Messwerte DA-Wandlung	8

Listingverzeichnis

6.1	Versuch1.py	16
6.2	.py	17
6.3	Versuch5.py	18

0

Einleitung

In diesem Versuch wird die Umwandlung von analogen in digitale Daten und umgekehrt untersucht. Die Lernziele in dem letzten Versuch der Versuchsreihe sind unter anderem:

- Umgang mit Analog-zu-Digital und Digital-zu-Analog Wandlern
- Messung der Genauigkeit der AD- und DA- Wandlung
- Untersuchung des Zeitverhaltens der DA-Wandlung
- Praktische Erfahrung mit dem Abtasttheorem

Diese Themen behandeln wir mit der Multifunktionsbox ME-RedLab USB-1208LS. Mit dieser wandeln wir Analoge- in die Digitale Signale und umgekehrt.

So werden wir Werte auslesen aus der Multifunktionsbox und Werte auf die Multifunktionsbox geben. Zudem werden wir Genauigkeit der AD-Wandlung testen, sowie die Genauigkeit der DA-Wandlung.

Im Anschluss werden wir das Zeitverhalten der DA-Wandlung überprüfen und das Abtasttheorem behandeln.

1

Versuch 1 - Programmierung der AD/DA-Wandlerkarte

1.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Im Labor verwenden wir die Multifunktionsbox ME-RedLab USB-1208LS. Diese Multifunktionsbox kann Analoge Signale in Digitale umwandeln oder umgekehrt.

Bei der Analogeingabe liegt der Eingangsspannungsbereich bei $\pm 10\text{V}$ im single-ended Modus. Der Ausgangsspannungsbereich liegt bei $0\text{V} - 5\text{V}$.

Als erstes müssen wir den AD/DA Wandler via USB mit dem Computer verbinden. Im Anschluss sollen wir mit dem Programm InstaCal in der Programmiergruppe 'RedLab' prüfen, ob die ME-RedLab-Box richtig erkannt wurde.

Die Kartenkonfiguration soll für die Box USB-1208LS auf 8 Single Ended Channels eingestellt sein.

Im Anschluss sollen wir ein Pythonskript schreiben zum Auslesen einer Reihe von Werten aus der Karte und ihrer Darstellung.

Im zweiten Teil des ersten Versuches sollen wir eine Funktion schreiben, die eine Konsolenausgabe auf die Karte ausgibt. Diese Funktion können wir im Anschluss mit dem Messgerät überprüfen.

1.2 Messwerte

Nach der Ausführung des Programms werden folgende Werte auf der Konsole ausgegeben:

16BitValue	Samplerate	VoltageValue
2454	8021	1.97266

Tabelle 1.1: Vergleich der berechneten Referenzspektren

[illegible]

Abbildung 1.1: Konsolenausgabe als .txt Datei

1.3 Auswertung

In dem zweiten Teil des ersten Versuches haben wir eine Konsoleneingabe auf die Karte ausgegeben. Dabei kann man Werte zwischen 0 und 5 Volt angeben.

Nachdem wir nun einen Wert von 3V in die Konsole durch das Pythonskript eingegeben haben, wurde von uns nun das Messgerät verwendet.

Im Anschluss haben wir eine Spannung von 3V mit dem Messgerät in dem Analogeingang gemessen.

1.4 Interpretation

Unsere Karte hat die Konsoleneingabe akzeptiert, da der Wert am Ausgang der AD/DA Karte auch 3V betragen hat. Somit ist die Karte und unser Skript funktionstüchtig.

Der Voltwert beträgt 1.97266V und die Samplerate beläuft sich auf 8021. Diese Samplerate ist auch die doppelte Nyquistfrequenz. Der 16 Bitwert beträgt 2454.

2

Versuch 2 - Genauigkeit der AD-Wandlung

2.1 Fragestellung, Messprinzip, Aufbau, Messmittel

In diesem Versuch wird an der Spannungsquelle eine Spannung eingestellt, welche dann mit dem Multimeter Philips PM 2503, dem AD-Wandler und dem Feinmessgerät Keithley TRMS 179 (als Referenzwert) gemessen wird.

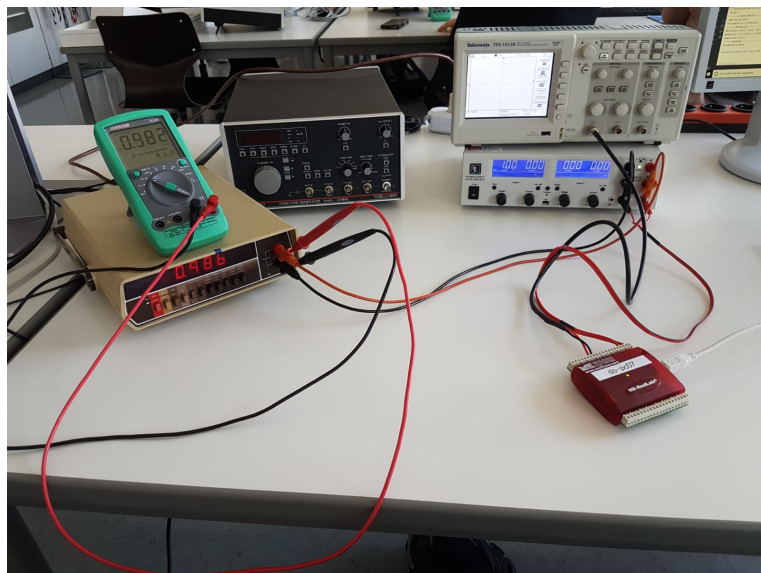


Abbildung 2.1: Versuchsaufbau

2.2 Messwerte

Folgende Messwerte wurden von uns während den Messungen zwischen 0 und 10 Volt notiert:

Spannung(V)	Feinmessgerät(V)	Multimeter	Messfehler	AD-Wandler	Messfehler
1	0.993	0.988	0.005	0.986	0.007
2	2.049	2.04	0.009	2.041	0.008
3	3.055	3.044	0.011	3.046	0.009
4	4.065	4.05	0.015	4.052	0.013
5	5.074	5.06	0.014	5.058	0.016
6	5.987	5.96	0.027	5.976	0.011
7	7.044	7.02	0.024	8.046	0.008
8	8.054	8.03	0.024	8.046	0.008
9	9.063	9.040	0.023	9.052	0.011
10	10.073	10.04	0.033	9.961	0.012

Tabelle 2.1: Messwerte Feinmessgerät, Multimeter, AD-Wandler

2.3 Auswertung

Der theoretische Quantisierungsfehler für den AD-Wandler beträgt

$$\Delta U_{AD} = \frac{10V - 1V}{2^{11}} = 4.4mV \quad (2.1)$$

Für das Multimeter ergibt sich aus den Messwerten die folgende Standardabweichung

$$S_M = 9mV \quad (2.2)$$

und für den AD-Wandler

$$S_{AD} = 2.8mV \quad (2.3)$$

2.4 Interpretation

Anhand der Standardabweichungen lässt sich feststellen, dass das Multimeter ungenauer ist, als der AD-Wandler. Die gemessene Standardabweichung von 2.8mV ist niedriger als der theoretische Quantisierungsfehler von 4.4mV. Daraus lässt sich schließen, dass die Messwerte im Fehlertoleranzbereichs des theoretischen Quantisierungsfehler liegen. Deshalb ist dieser theoretisch berechnete Wert für die AD-Wandlung plausibel.

3

Versuch 3 - Genauigkeit der DA-Wandlung

3.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Hier sollen nun Analogwerte vom DA-Wandler gemessen werden. Die DA-Wandler wird programmiert Spannungen auszugeben, die dann vom Oszilloskop gemessen werden. Der Versuchsaufbau sieht dabei wie folgt aus:

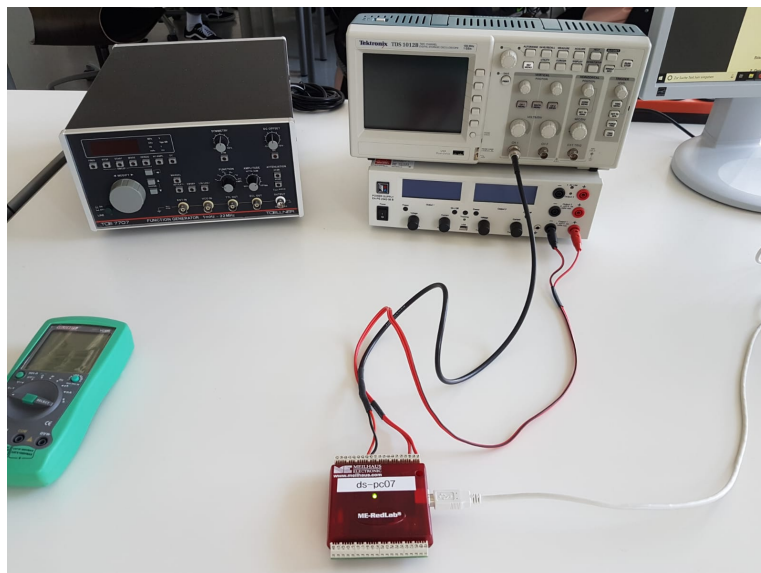


Abbildung 3.1: Versuchsaufbau

3.2 Messwerte

Folgende Messwerte wurden von uns bei den Messungen zwischen 0 und 5 Volt notiert:

Spannung(V)	Oszilloskop(V)	Messfehler(V)
0.5	0.557	0.057
1	1.05	0.05
1.5	1.59	0.09
2	2.08	0.08
2.5	2.58	0.08
3	3.11	0.11
3.5	3.62	0.12
4	4.15	0.15
4.5	4.66	0.16
5	5.14	0.14

Tabelle 3.1: Messwerte DA-Wandlung

3.3 Auswertung

Der theoretische Quantisierungsfehler für den DA-Wandler beträgt

$$\Delta U_{\text{DA}} = \frac{5V - 0.5V}{2^{10}} = 4.4mV \quad (3.1)$$

Aus den Messwerten ergibt sich die folgende Standardabweichung

$$S_{\text{DA}} = 38.5mV \quad (3.2)$$

3.4 Interpretation

Die Standardabweichung(38.5mV) liegt deutlich über dem theoretischen Quantisierungsfehler(4.4mV), was darauf schließen lässt, dass die Wandlung des digitalen Signals in ein analoges sehr fehlerbehaftet ist. Dazu tragen folgende Faktoren bei: thermisches Rauschen, Rundungsfehler, Überlastungsfehler.

4

Versuch 4 - Zeitverhalten der DA-Wandlung

4.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Ziel des Versuchs ist zu sehen, wie genau sich mit einer Wandlerkarte eine Sinusschwingung erzeugen lässt. Der Versuchsaufbau lautet folgendermaßen:

Der AD/DA Wandler wird via USB an den Computer angeschlossen. Der AD/DA Wandler wird zudem an den Funktionsgenerator angeschlossen.

Der AD/DA Wandler wird zudem an das Oszilloskop an Channel1 angeschlossen.

Die Verbindung zum Oszilloskop wird verwendet um die Sinusschwingung auszugeben.

Das Python Programm (s. task4.py) errechnet ein Array von Sinuswerten mit 30 Werten pro Schwingung.

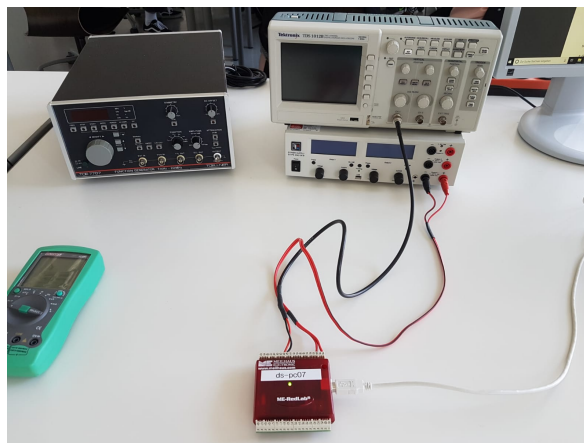


Abbildung 4.1: Versuchsaufbau

4.2 Messwerte

Nachdem das Sinussignal von dem AD/DA Wandler zum Oszilloskopen geleitet wird, machen wir mit dem Programm auf dem Computer im Labor ein Screenshot des Oszilloskopen. Das Ergebnis ist in dem folgenden Schaubild zu erkennen.

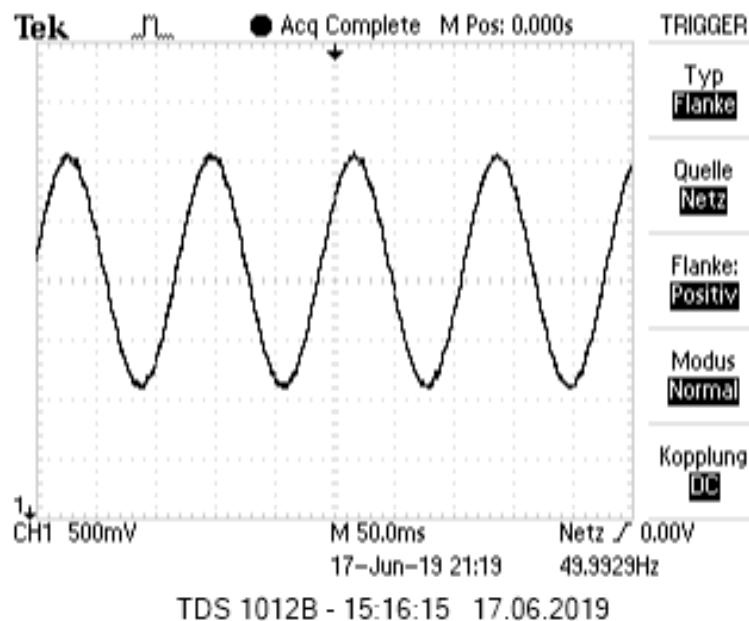


Abbildung 4.2: Ausgegebene Sinusspannung

4.3 Interpretation

Die Sinusspannung entspricht sehr gut einer realen Sinusschwingung, jedoch entsteht man aufgrund der kleinen Anzahl der Amplitudenstufen(30) ein Treppeneffekt, wodurch das Signal bei näherer Betrachtung leicht rauschig wirkt.

5

Versuch 5 - Abtasttheorem

5.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Im fünften Versuch dürfen wir eine Abtastfrequenz zwischen 6000 und 8000 auswählen. Wir haben uns für eine Frequenz von 8000 entschieden. Mittels Python und dem RedLab-Befehl `rl.cbInScanRate` werden wir die tatsächliche Abtastfrequenz des AD-Wandlers für die Abtastfrequenz auslesen.

Zudem haben wir die Nyquistfrequenz ermittelt. Im Anschluss sollen wir von der halben Nyquistfrequenz bis zur doppelten Nyquistfrequenz, die Frequenz des Sinusgenerators variieren. Nach der Programmierung des Sinusgenerators, haben wir diese auf 7 verschiedene Frequenzen angewendet.

Im Anschluss daraufhin sollen die Kurven für das Protokoll plotten. So bekommen wir im Anschluss für 7 verschiedene Frequenzen 7 verschiedene Plots.

In Abbildung ist es gut zu sehen, wie unser Aufbau erfolgt ist:

Der AD/DA Wandler wurde per USB an den Computer angeschlossen. Der AD/DA Wandler wurde ebenfalls auch noch an den Funktionsgenerator angeschlossen, welcher davor auf Sinusschwingungen eingestellt wurde.

5.2 Messwerte

Die folgenden Messwerte sind entstanden, indem wir nun einen Sinuswert eingegeben haben in unseren Sinusgenerator, denn wir mittels Python programmiert haben. Die Frequenz haben wir dabei auf 7 verschiedene Nyquistfrequenz eingestellt:

- 2000Hz
- 3000Hz
- 4010Hz
- 5000Hz
- 6000Hz
- 7000Hz
- 8020Hz

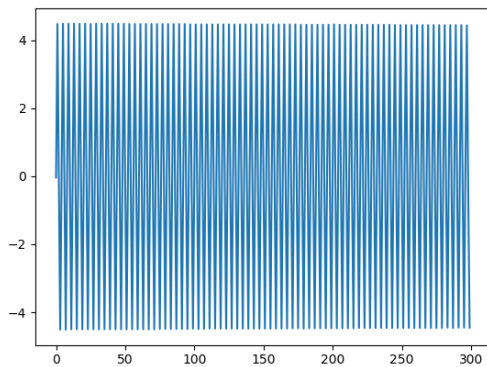


Abbildung 5.1: Messfrequenz von 2000Hz

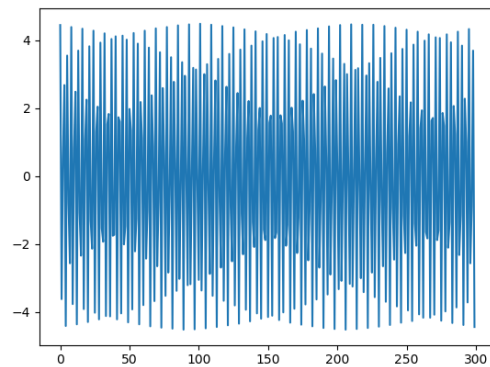


Abbildung 5.2: Messfrequenz von 3000Hz

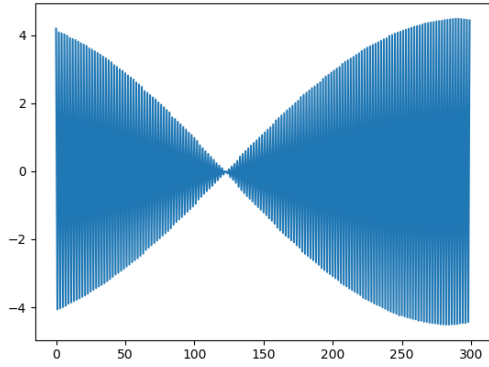


Abbildung 5.3: Messfrequenz von 4010Hz

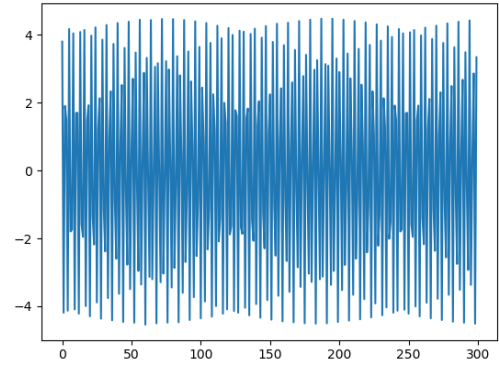


Abbildung 5.4: Messfrequenz von 5000Hz

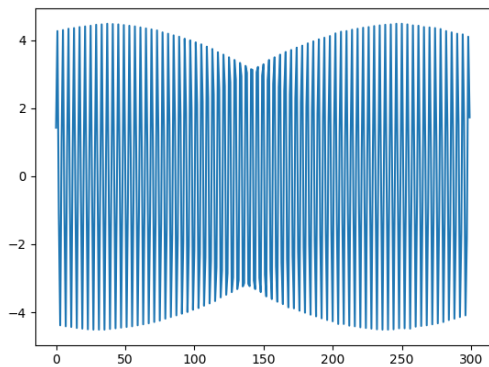


Abbildung 5.5: Messfrequenz von 6000Hz

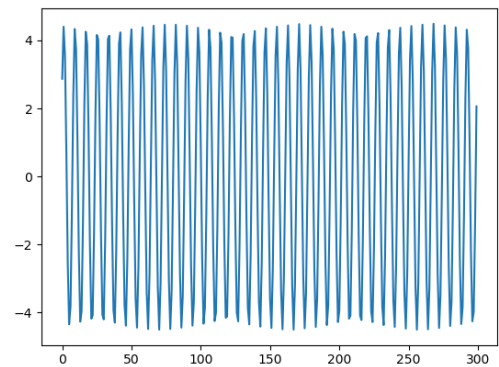


Abbildung 5.6: Messfrequenz von 7000Hz

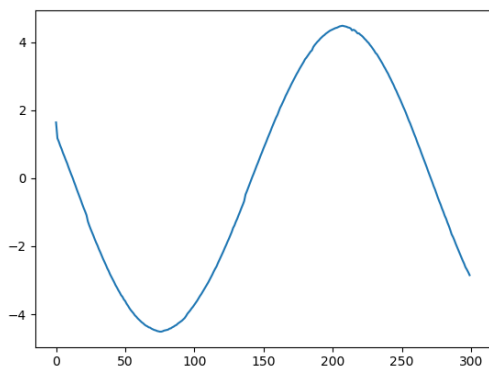


Abbildung 5.7: Messfrequenz von 8020Hz

5.3 Auswertung

Da wir von Anfang an eine Abtastfrequenz von 8000Hz verwendet haben, waren wir sehr nahe an der tatsächlichen Abtastfrequenz von 8021. Diese haben wir durch den Befehl `cbInScanRate(0, 0, 0, 8000)` erfahren.

Da dies jedoch die doppelte Nyquistfrequenz ist, wollen wir noch die normale Nyquistfrequenz erfahren.

Dies wird logischerweise ermöglicht durch die Division der doppelten Nyquistfrequenz durch 2:

$$8021 / 2 = 4010.5$$

5.4 Interpretation

In den Bildern 6.2, 6.3, 6.4 und 6.5 geschieht das sogenannte Aliasing.

Da die Abtastfrequenz bei diesen Plots kleiner ist als die doppelte Grundfrequenz, überlappen sich die Kopien des Spektrums.

Diese Überlappung wird Aliasing genannt.

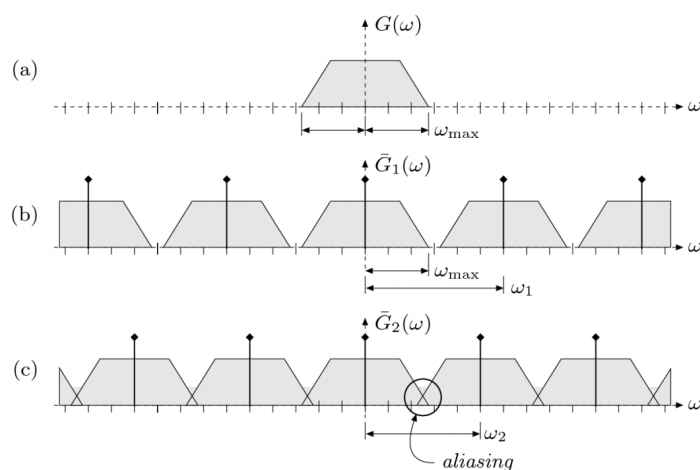


Abbildung 5.8: Schaubild zu Aliasing

Bei den anderen Schaubildern 6.1, 6.6 und 6.7 geschieht kein Aliasing. Dafür muss die Abtastfrequenz größer sein, als die doppelte Grundfrequenz des Signals. Bei der tatsächlichen Nyquistfrequenz von 4010.5Hz ist die Überlappung beziehungsweise das Aliasing am größten. Umso weiter sich die Abtastfrequenz von der Nyquistfrequenz entfernt, desto geringer wird das Aliasing. Erst bei der Aufnahme der Nyquistfrequenz von 7000Hz, ist das Aliasing verschwunden. Während der ersten Aufnahme noch kein Aliasing zu sehen, beginnt dies dafür dann bei der zweiten aufgenommenen Frequenz von 3000Hz. Beim letzten Messwert von 8020Hz sieht man genau eine Schwingung, da sich die Frequenz nur um 1Hz von der doppelten Nyquistfrequenz(8021Hz) unterscheidet.

Anhang

A.1 Quellcode Aufgabe 1

```
45 import redlab as rl
46
47 print("-----einzelneWerte-----")
48 print("16BitValue:" + str(rl.cbAIn(0, 0, 1)))
49 print("VoltageValue:" + str(rl.cbVIn(0, 0, 1)))
50 print("-----Messreihe-----")
51 print("Messreihe:" + str(rl.cbAInScan(0, 0, 0, 300, 8000, 1)))
52 print("Messreihe:" + str(rl.cbVInScan(0, 0, 0, 300, 8000, 1)))
53 print("Samplerate:" + str(rl.cbInScanRate(0, 0, 0, 8000)))
54 print("Nyquist:" + str(rl.cbInScanRate(0, 0, 0, 8000) / 2))
55 print("-----Ausgabe-----")
```

Listing 6.1: Versuch1.py

A.2 Quellcode Aufgabe 4

```
45 import redlab as rl
46 from time import sleep
47
48 import numpy as np
49 import matplotlib.pyplot as plt
50
51 print("-----einzelneWerte-----")
52 print("16BitValue:" + str(rl.cbAIn(0, 0, 1)))
53 print("VoltageValue:" + str(rl.cbVIn(0, 0, 1)))
54 print("-----Messreihe-----")
55 print("Messreihe:" + str(rl.cbAInScan(0, 0, 0, 300, 8000, 1)))
56 print("Messreihe:" + str(rl.cbVInScan(0, 0, 0, 300, 8000, 1)))
57 print("Samplerate:" + str(rl.cbInScanRate(0, 0, 0, 8000)))
58 print("Nyquist:" + str(rl.cbInScanRate(0, 0, 0, 8000) / 2))
59 print("-----Ausgabe-----")
60
61 sin = [np.sin(2 * np.pi * 2 * (i / 30)) for i in range(0, 30)]
62
63 while True:
64     for x in range(0, 30):
65         rl.cbVOut(0, 0, 101, sin[x] + 2)
66         sleep(1/30*8021/4)
67
68 vec = np.zeros((8000))
69 data = rl.cbVInScan(0, 0, 0, 300, 8000, 1)
70 plt.plot(data)
71 plt.show()
```

Listing 6.2: Versuch4.py

A.3 Quellcode Aufgabe 5

```
45 import redlab as rl
46 from time import sleep
47
48 import numpy as np
49 import matplotlib.pyplot as plt
50
51 print("-----einzelneWerte-----")
52 print("16BitValue:" + str(rl.cbAIn(0, 0, 1)))
53 print("VoltageValue:" + str(rl.cbVIn(0, 0, 1)))
54 print("-----Messreihe-----")
55 print("Messreihe:" + str(rl.cbAInScan(0, 0, 0, 300, 8000, 1)))
56 print("Messreihe:" + str(rl.cbVInScan(0, 0, 0, 300, 8000, 1)))
57 print("Samplerate:" + str(rl.cbInScanRate(0, 0, 0, 8000)))
58 print("Nyquist:" + str(rl.cbInScanRate(0, 0, 0, 8000) / 2))
59 print("-----Ausgabe-----")
60
61 sin = [np.sin(2 * np.pi * 2 * (i / 30)) for i in range(0, 30)]
62
63 vec = np.zeros((8000))
64 data = rl.cbVInScan(0, 0, 0, 300, 8000, 1)
65 plt.plot(data)
66 plt.show()
```

Listing 6.3: Versuch5.py

A.4 Messprotokolle

Aufgabe 3: Genauigkeit der DA-Wandlung

Quantisierungsfehler $\Delta U = \frac{5V - 0,5V}{2^{10}}$
 $= 0,0044V$

Spannung	Oszilloskop	AD-Wandler	Massfehler
0,5V	0,557V	0,5566V	0,057V
1V	1,05V		0,05V
1,5V	1,59V		0,09V
2V	2,08V		0,08V
2,5V	2,58V		0,08V
3V	3,11V		0,11V
3,5V	3,62V		0,12V
4V	4,15V		0,15V
4,5V	4,66V		0,16V
5V	5,14V		0,14V

$S = 0,0385V$

[Signature]

Abbildung 6.9: Protokoll Aufgabe 3

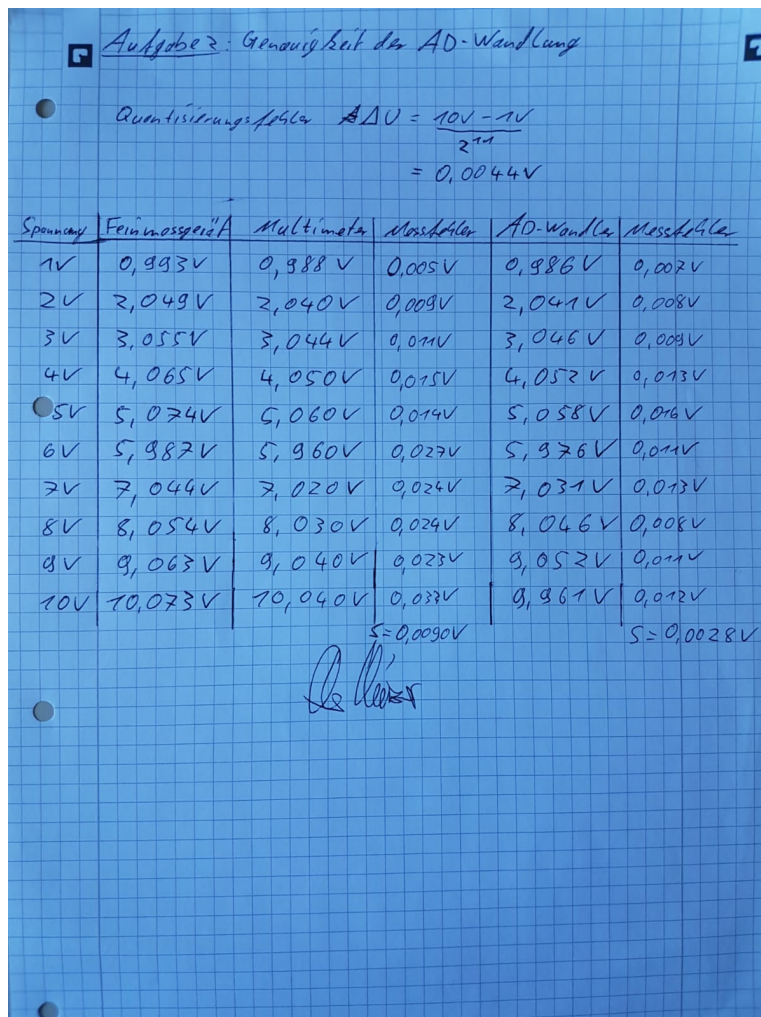


Abbildung 6.10: Protokoll Aufgabe 2