



Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Signale, Systeme und Sensoren

Aufbau, Kalibrierung und Einsatz eines einfachen Entfernungsmessers

T. Schoch, L. Stratmann

Konstanz, 11. April 2019

Zusammenfassung (Abstract)

Thema:	Aufbau, Kalibrierung und Einsatz eines einfachen Entfernungsmeßers	
Autoren:	T. Schoch	tobias.schoch@htwg-konstanz.de
	L. Stratmann	luca.stratmann@htwg-konstanz.de
Betreuer:	Prof. Dr. Matthias O. Franz Jürgen Keppler	mfranz@htwg-konstanz.de juergen.keppler@htwg-konstanz.de
	Mert Zeybek	me431zey@htwg-konstanz.de

In dem Versuch haben wir einen Entfernungsmeßer dazu verwendet, um die bereits in der Vorlesung behandelten Vorgehensweisen zum Thema Kalibrierung, Fehlerbehandlung und Fehlerrechnung anzuwenden. Der Distanzsensor der Marke "Sharp" benutzt für das Triangulationsprinzip Infrarot-LEDS mit einer Linse. Diese geben Lichtstrahlen von sich, um dann wiederum reflektiert zu werden und durch die zweite Linse zu gelangen. Je nachdem wo der Lichtstrahl auftrifft, wandelt der Signalprozessor die Leitfähigkeit in eine Spannung um. Da das Ausgangssignal anti-proportional ist, wird mit der zunehmenden Entfernung das Ausgangssignal kleiner. Die Entfernung liegen zwischen 10cm und 70cm. Durch ein Oszilloskop können wir den Spannungsverlauf des Abstandssensors überprüfen.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Listingverzeichnis	V
1 Einleitung	1
2 Versuch 1: Ermittlung der Kennlinie des Abstandssensors	2
2.1 Fragestellung, Messprinzip, Aufbau, Messmittel	2
2.2 Messwerte	4
2.3 Auswertung	5
2.4 Interpretation	7
3 Versuch 2: Modellierung der Kennlinie durch lineare Regression	8
3.1 Fragestellung, Messprinzip, Aufbau, Messmittel	8
3.2 Messwerte	9
3.3 Auswertung	10
3.4 Interpretation	12
4 Versuch 3: Flächenmessung mit Fehlerrechnung	13
4.1 Fragestellung, Messprinzip, Aufbau, Messmittel	13
4.2 Messwerte	13
4.3 Auswertung	13
4.4 Interpretation	13
Anhang	14
A.1 Quellcode	14
A.1.1 Quellcode Versuch 1: Messung Kalibrierung	14

A.1.2	Quellcode Versuch 2	16
A.1.3	Quellcode Versuch 3	18
A.2	Messergebnisse	19

Abbildungsverzeichnis

2.1	Aufbau im Labor	3
2.2	Durchschnittliche Spannung	5
2.3	Standartabweichung der Spannung	6
2.4	Unterschiede durch Störfaktoren	7
3.1	Logarithmus der Ein- und Ausgangswerte	10
3.2	Kennlinie	10
3.3	Lineare Regression	11
5.1	Messungen aus dem Labor	19

Tabellenverzeichnis

2.1	Messwerte Kalibrierung	4
3.1	Logarithmus von der Distanz und der Durchschnittsspannung	9

Listingverzeichnis

5.1	Ermittlung der Kennlinie des Abstandssensors	14
5.2	Lineare Regression	16
5.3	Flächenmessung mit Fehlerrechnung	18

1

Einleitung

[?] [?]

2

Versuch 1: Ermittlung der Kennlinie des Abstandssensors

2.1 Fragestellung, Messprinzip, Aufbau, Messmittel

Im ersten Versuch werden wir die Kennlinie des Abstandssensors ermitteln. Für den Aufbau des Projektes verbinden wir den Abstandssensor an 'Output 3' des Labornetzgerätes 'EA-PS 2342-06 B' durch Ground (-) und dem 5V Anschluss (+). Der Abstandssensor lautet 'GP2Y0A21YK0F' und wurde von der Firma 'Sharp' entworfen. Das Netzgerät wird auf 5V Gleichspannung eingestellt. Das Oszilloskop von 'Tektronix' mit dem Namen 'TDS 2022B' wird an den Abstandssensor mit Ground(-), sowie an den Signalausgang angeschlossen. Dies wird im Oszilloskop mit einem Adapter an Channel1 angeschlossen. Nachdem das Oszilloskop richtig eingestellt wurde, haben wir es mit dem PC verbunden. Über ein Programm konnten wir so das Oszilloskop mit dem Computer verbinden. Ein Programm hat uns geholfen den aktuellen Bildschirm des Oszilloskopes auf dem Bildschirm zu empfangen. Zudem kann das Programm die empfangenen Daten über die Ausgangsspannung in eine '.csv' Datei ausgeben. So konnten wir für jede Messung einen Screenshot und eine '.csv' Datei erstellen. Ein hochkant stehendes Holzbrett definiert den Abstand. Die 21 zu messenden Werte liegen zwischen 10cm und 70cm in jeweils 3cm Abständen. Mit einem Meterstab haben wir einen Richtwert für den Abstand zwischen Abstandssensor und Holzbrett. Nachdem wir durch die erschwerten Lichtverhältnisse die richtige Lage des Abstandssensors gefunden haben, haben wir über das Programm sowohl Screenshots vom Bildschirm des Oszilloskopes gemacht, als auch die Daten in einer '.csv' Datei gespeichert. Zudem haben wir die gemessenen Längen mit deren dazugehörigen Ausgangsspannung handschriftlich in einer Tabelle aufgeschrieben.

ben, welche am Ende des Versuches vom Tutor unterschrieben wurde. Anschließend haben wir in Python programmiert, um die Dateien aus den '.csv' einzulesen mit genfromtxt(). Um den Einschwingvorgang nicht mit zu berechnen haben wir die ersten 1000 Zeilen übersprungen. Nachdem werden wir den Durchschnitt sowie die Standardabweichung berechnet haben, visualisieren wir in einer Kennlinie den Durchschnitt sowie die Standardabweichung mittels matplotlib.

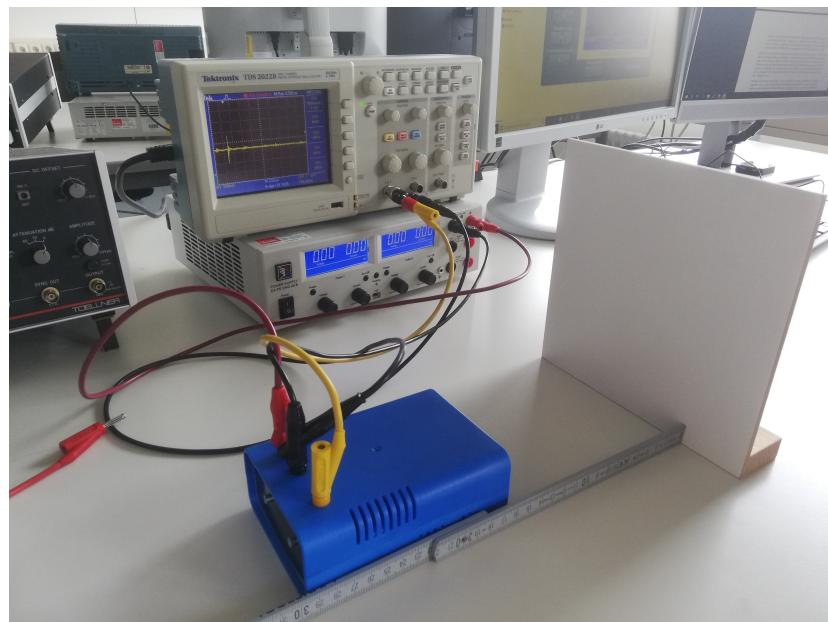


Abbildung 2.1: Aufbau im Labor

2.2 Messwerte

Tabelle [2.2] zeigt die von Hand notierten, sowie die in Python programmierten Werte.

Distanz	Spannung	Durchschnitt	Standartabweichung
10cm	1,34V	1.3318680589410588	0.020263173048016166
13cm	1,15V	1.1497102771228773	0.020678191630490284
16cm	1,05V	1.0469130452117372	0.02123462061404347
19cm	0,935V	0.9307492279888693	0.02269412487931415
22cm	0,838V	0.8345054697083348	0.020575634388180667
25cm	0,775V	0.8345054697083348	0.020575634388180667
28cm	0,696V	0.6915484407804515	0.02668684323820322
31cm	0,657V	0.6540259565267772	0.01868491936225132
34cm	0,617V	0.6141258593084946	0.02008623727146757
37cm	0,580V	0.5766833013111707	0.01849989543490117
40cm	0,560V	0.5602197652453677	0.02037490284636567
43cm	0,519V	0.5173026806826474	0.01873941748579766
46cm	0,499V	0.49640358628691506	0.020962190865871817
49cm	0,479V	0.4752847033362587	0.019953591718188307
52cm	0,457V	0.4526673207927892	0.020468492232078573
55cm	0,434V	0.4228371513728602	0.11289993464369412
58cm	0,412V	0.41786212422798	0.019085619538662855
61cm	0,395V	0.39262735831746653	0.018698669721518596
64cm	0,374V	0.3728471420971699	0.01960614919901995
67cm	0,395V	0.3910489372319112	0.02277015569326621
70cm	0,374V	0.36915083915708796	0.020094136680969384

Tabelle 2.1: Messwerte Kalibrierung

2.3 Auswertung

In der folgenden Abbildung sind die Messergebnisse der durchschnittlichen Spannung nochmals visuell dargestellt. Die Messergebnisse wurden mit matplotlib in Python visualisiert.

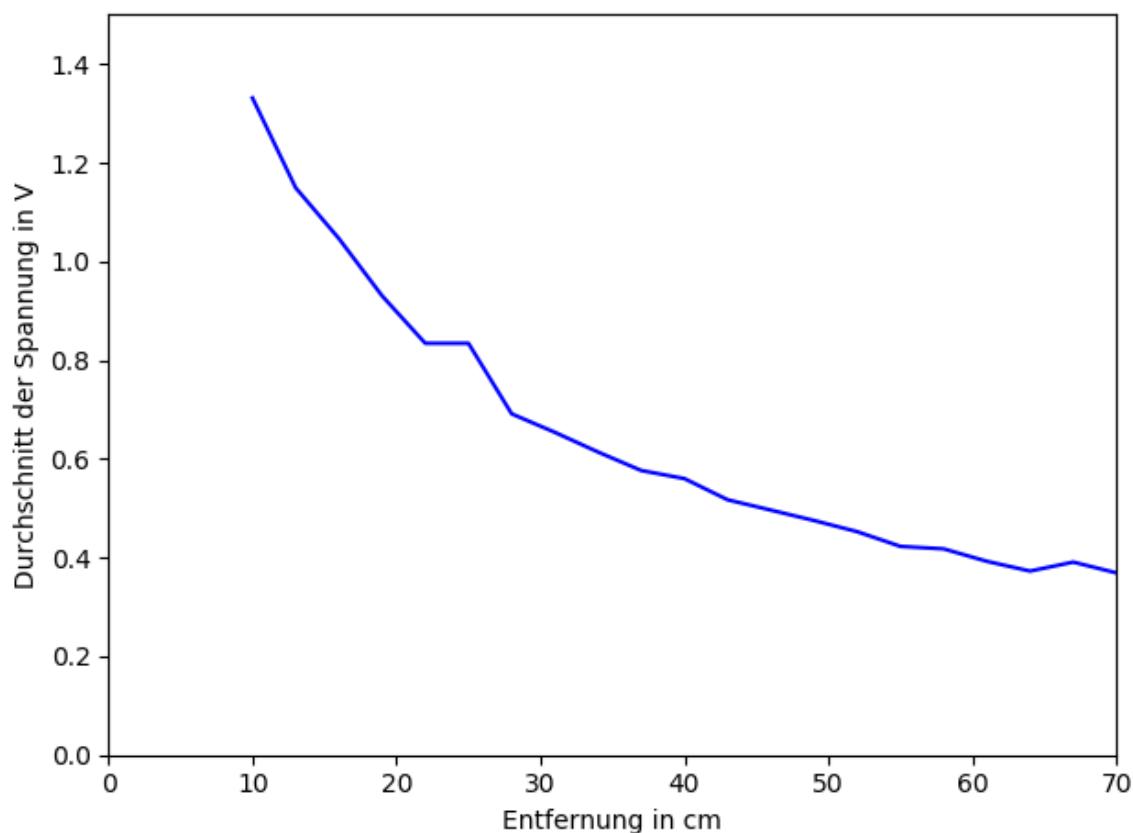


Abbildung 2.2: Durchschnittliche Spannung

In der folgenden Abbildung sind die Messergebnisse der Standartabweichung visuell dargestellt. Die Messergebnisse wurden mit matplotlib in Python visualisiert.

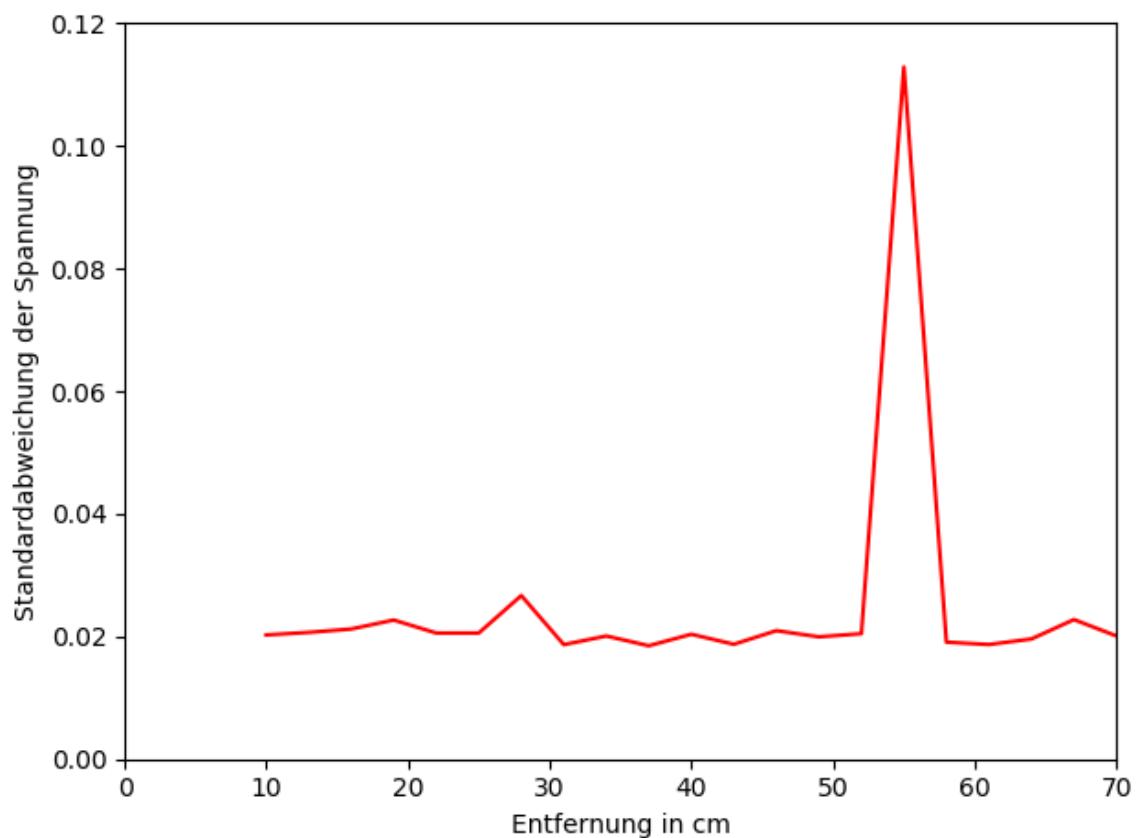


Abbildung 2.3: Standartabweichung der Spannung

2.4 Interpretation

Wie man gut in der Tabelle und der Abbildung 2.2 ablesen kann, wird die Spannung stets niedriger. Dies liegt an der Anti-proportionalität, dass mit der zunehmenden Entfernung zwischen Holzbrett und Abstandssensor die vom Signalprozessor übertragene Spannung geringer wird.

Leider haben wir bei der Generierung der Dateien einen Fehler gemacht und bei 22cm und 25cm zu spät die Single Sequenz aktualisiert, weshalb eine Gerade in dem Plot zwischen 22cm und 25cm genau gleich ist. Zudem geht bei Messung zwischen 64cm und 67cm Abstand die Spannung nach oben. An dem Tag der Messung war das Wetter sehr wechselhaft, was zu einer Erhöhung der Werte führen könnte.

Bei der Messung 55cm ist eine sehr hohe Standardabweichung im Vergleich zu den anderen Werten. Dies liegt daran, dass das Oszilloskop durch andere Störfaktoren gestört wurde und so eine ungleiche Single Sequenz ergeben hat. In dem rechten Bild der Abbildung 2.4 ist im Vergleich nochmals gut zu sehen, wie die hohe Standartabweichung zu stande kommt.

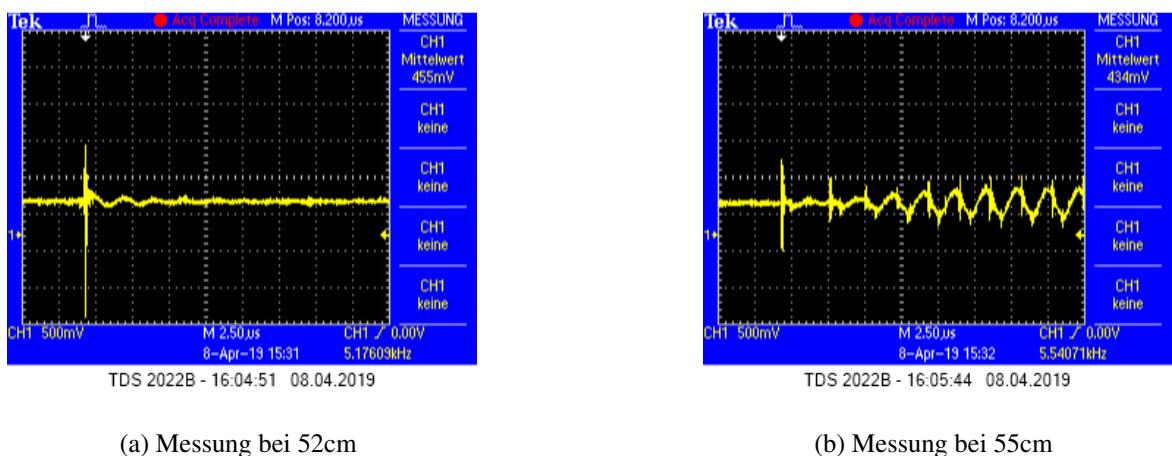


Abbildung 2.4: Unterschiede durch Störfaktoren

3

Versuch 2: Modellierung der Kennlinie durch lineare Regression

3.1 Fragestellung, Messprinzip, Aufbau, Messmittel

3.2 Messwerte

Tabelle [3.2] zeigt die in Python logarithmierten Werte.

Distanz	log cm	log Durchschnitt
10cm	2.302585092994046	0.28658251236288684
13cm	2.5649493574615367	0.13950997769809692
16cm	2.772588722239781	0.04584587706574458
19cm	2.9444389791664403	-0.0717653956802384
22cm	3.091042453358316	-0.18091598142798918
25cm	3.2188758248682006	-0.18091598142798918
28cm	3.332204510175204	-0.3688220785875914
31cm	3.4339872044851463	-0.42460823943365683
34cm	3.5263605246161616	-0.48755538925392977
37cm	3.6109179126442243	-0.5504620343701235
40cm	3.6888794541139363	-0.579426134298413
43cm	3.7612001156935624	-0.659127119920277
46cm	3.828641396489095	-0.7003660010868976
49cm	3.8918202981106265	-0.7438412790564594
52cm	3.9512437185814275	-0.792597814482305
55cm	4.007333185232471	-0.8607681589957209
58cm	4.060443010546419	-0.8726037472131969
61cm	4.110873864173311	-0.9348943146021321
64cm	4.1588830833596715	-0.9865867500420398
67cm	4.204692619390966	-0.9389225676676274
70cm	4.248495242049359	-0.9965499402651674

Tabelle 3.1: Logarithmus von der Distanz und der Durchschnittsspannung

3.3 Auswertung

In der folgenden Abbildung wurden die Messergebnisse logarithmiert und daraufhin mit matplotlib in Python visualisiert.

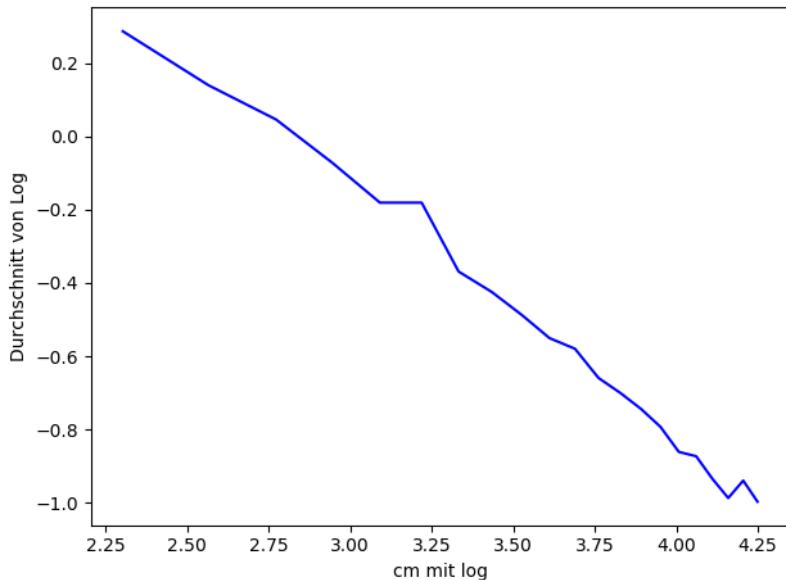


Abbildung 3.1: Logarithmus der Ein- und Ausgangswerte

In der folgenden Abbildung wurde die Kennlinie visualisiert mit matplotlib in Python.

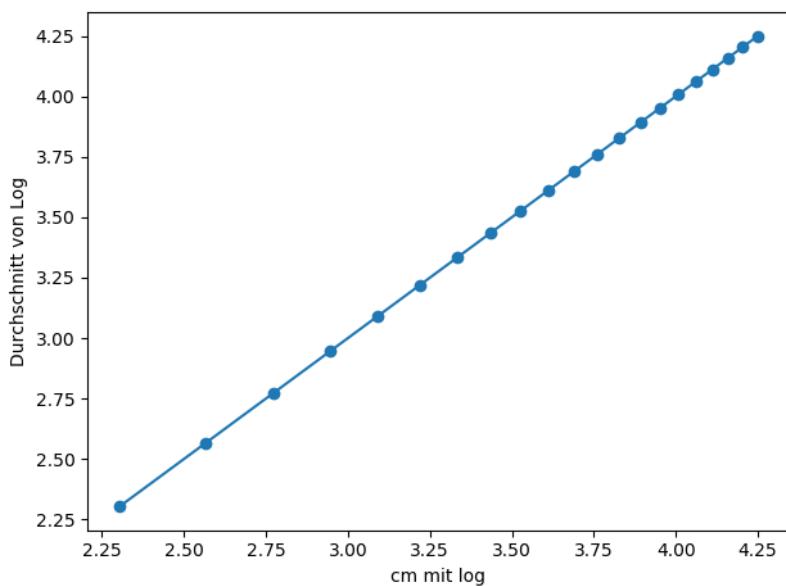


Abbildung 3.2: Kennlinie

In der folgenden Abbildung wurde die Lineare Regression realisiert mit matplotlib in Python.

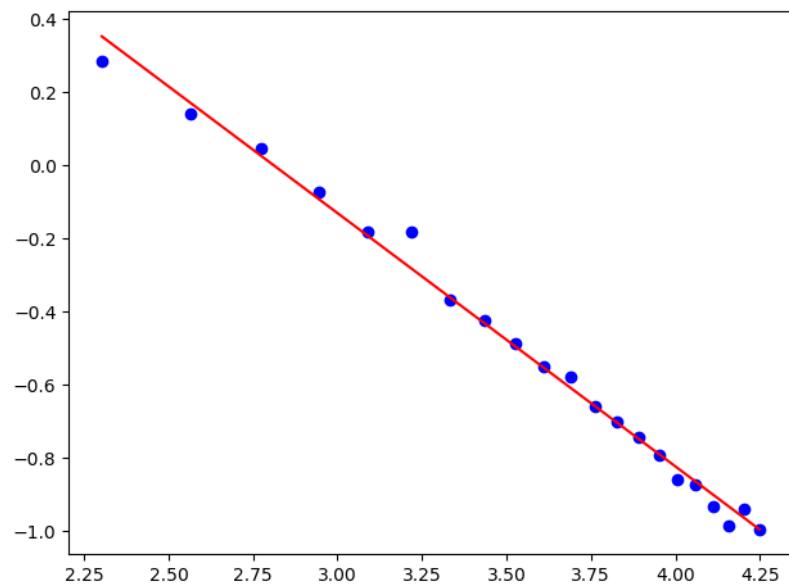


Abbildung 3.3: Lineare Regression

3.4 Interpretation

In Abbildung 3.1 sieht man nur wenige Unstimmigkeiten. Bei der x-Achse an der Stelle 3.25cm liegt der Fehler an der Messung, für die unsere Messung verantwortlich ist. Leider haben wir denselben Wert zweimal eingelesen, ohne die Single Sequenz zu aktualisieren, was zu diesem Ausschlag führt. Kurz vor dem Ende der Visualisierung geht der Wert nach oben, was an einer Erhöhung der Spannung liegt. Dies ist auch sichtbar in der Tabelle 2.1. Vermutlich ist dies passiert durch veränderte Wetterverhältnisse, die die Lichtverhältnisse im Raum veränderten.

In Abbildung 3.3 zieht sich die Ausgleichsgerade durch die einzelnen Punkte um den Abstand minimal werden zu lassen. Abgesehen von einzelnen Ausreißern wie zum Beispiel dem Wert bei 3.25cm, der durch einen Messfehler entstanden ist wie schon weiter oben berichtet, liegen die Punkte sehr nah an der Ausgleichsgeraden. Dies lässt darauf deuten, dass die meisten Punkte durch den Abstandssensor richtig gemessen wurden.

4

Versuch 3: Flächenmessung mit Fehlerrechnung

4.1 Fragestellung, Messprinzip, Aufbau, Messmittel

4.2 Messwerte

4.3 Auswertung

4.4 Interpretation

Anhang

A.1 Quellcode

A.1.1 Quellcode Versuch 1: Messung Kalibrierung

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 ten = -1
5 data = 0
6 vec = np.zeros((21, 3))
7
8 rang = ["10", "13", "16", "19", "22", "25", "28", "31", "34", "37", "40", "43", "46", "49", "52", "55", "58", "61",
9     "64", "67", "70"]
10
11 volt = ["1,34V", "1,15V", "1,05V", "0,935V", "0,838V", "0,775V", "0,696V", "0,657V", "0,617V", "0,580V",
12     "0,560V", "0,519V", "0,499V", "0,479V", "0,457V", "0,434V", "0,412V", "0,395V", "0,374V", "0,395V",
13     "0,374V"]
14
15 for x in rang:
16     ten += 1
17     data = np.genfromtxt('data/' + str(rang[ten]) + '.csv', delimiter=',', skip_header=1000, skip_footer=499,
18                         usecols=(4))
19
20     print("\hline")
21     print(str(rang[ten]) + "cm & " + str(volt[ten]) + " & " + str(np.mean(data)) + " & " + str(np.std(data)) + " \\\\"")
22
23     vec[ten, 0] = rang[ten]
24     vec[ten, 1] = np.mean(data)
25     vec[ten, 2] = np.std(data)
26
27 plt.plot([10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 61, 64, 67, 70],
28     [vec[0, 1], vec[1, 1], vec[2, 1], vec[3, 1], vec[4, 1], vec[5, 1], vec[6, 1], vec[7, 1], vec[8, 1], vec[9, 1],
```

```

29     vec[10, 1], vec[11, 1], vec[12, 1], vec[13, 1], vec[14, 1], vec[15, 1], vec[16, 1], vec[17, 1], vec[18, 1],
30     vec[19, 1], vec[20, 1]], 'b')
31 plt.ylabel('Durchschnitt der Spannung in V')
32 plt.xlabel('Entfernung in cm')
33 plt.axis([0, 70, 0, 1.5])
34 plt.show()
35
36 plt.plot([10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 61, 64, 67, 70],
37     [vec[0, 2], vec[1, 2], vec[2, 2], vec[3, 2], vec[4, 2], vec[5, 2], vec[6, 2], vec[7, 2], vec[8, 2], vec[9, 2],
38     vec[10, 2], vec[11, 2], vec[12, 2], vec[13, 2], vec[14, 2], vec[15, 2], vec[16, 2], vec[17, 2], vec[18, 2],
39     vec[19, 2], vec[20, 2]], 'r')
40 plt.ylabel('Standardabweichung der Spannung')
41 plt.xlabel('Entfernung in cm')
42 plt.axis([0, 70, 0, 0.12])
43 plt.show()

```

Listing 5.1: Ermittlung der Kennlinie des Abstandssensors

A.1.2 Quellcode Versuch 2

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import stats
4
5 ten = -1
6 data = 0
7 vec2 = np.zeros((21, 2))
8 vec3 = np.zeros((21, 1))
9
10 rang = ["10", "13", "16", "19", "22", "25", "28", "31", "34", "37", "40", "43", "46", "49", "52", "55", "58", "61",
11     "64", "67", "70"]
12
13 for x in rang:
14     ten += 1
15     data = np.genfromtxt('data/' + str(rang[ten]) + '.csv', delimiter=',', skip_header=1000, skip_footer=499,
16                         usecols=(4))
17
18     vec2[ten, 0] = np.log(int(rang[ten]))
19     vec2[ten, 1] = np.log(np.mean(data))
20     vec3[ten, 0] = (vec2[ten, 0] / vec2[ten, 1]) * vec2[ten, 1]
21
22     print("\hline")
23     print(str(rang[ten]) + "cm & " + str(vec2[ten, 0]) + " & " + str(vec2[ten, 1]) + " & " + str(vec3[ten, 0]) + " \\\\"")
24
25 x = np.array(
26     [vec2[0, 0], vec2[1, 0], vec2[2, 0], vec2[3, 0], vec2[4, 0], vec2[5, 0], vec2[6, 0], vec2[7, 0], vec2[8, 0],
27     vec2[9, 0], vec2[10, 0], vec2[11, 0], vec2[12, 0], vec2[13, 0], vec2[14, 0], vec2[15, 0], vec2[16, 0], vec2[17, 0],
28     vec2[18, 0], vec2[19, 0], vec2[20, 0]])
29 y = np.array(
30     [vec2[0, 1], vec2[1, 1], vec2[2, 1], vec2[3, 1], vec2[4, 1], vec2[5, 1], vec2[6, 1], vec2[7, 1], vec2[8, 1],
31     vec2[9, 1], vec2[10, 1], vec2[11, 1], vec2[12, 1], vec2[13, 1], vec2[14, 1], vec2[15, 1], vec2[16, 1],
32     vec2[17, 1], vec2[18, 1], vec2[19, 1], vec2[20, 1]])
33
34 gradient, intercept, r_value, p_value, std_err = stats.linregress(x, y)
35 mn = np.min(x)
36 mx = np.max(x)
37 x1 = np.linspace(mn, mx, 500)
38 y1 = gradient * x1 + intercept
39
40 plt.plot(x, y, 'ob')
```

```

41 plt.plot(x1, y1, 'r')
42 plt.show()
43
44 plt.plot([vec2[0, 0], vec2[1, 0], vec2[2, 0], vec2[3, 0], vec2[4, 0], vec2[5, 0], vec2[6, 0], vec2[7, 0], vec2[8, 0],
45           vec2[9, 0], vec2[10, 0], vec2[11, 0], vec2[12, 0], vec2[13, 0], vec2[14, 0], vec2[15, 0], vec2[16, 0],
46           vec2[17, 0], vec2[18, 0], vec2[19, 0], vec2[20, 0]],
47           [vec2[0, 1], vec2[1, 1], vec2[2, 1], vec2[3, 1], vec2[4, 1], vec2[5, 1], vec2[6, 1], vec2[7, 1], vec2[8, 1],
48           vec2[9, 1], vec2[10, 1], vec2[11, 1], vec2[12, 1], vec2[13, 1], vec2[14, 1], vec2[15, 1], vec2[16, 1],
49           vec2[17, 1], vec2[18, 1], vec2[19, 1], vec2[20, 1]], 'b')
50 plt.ylabel('Durchschnitt von Log')
51 plt.xlabel('cm mit log')
52 plt.show()
53
54 plt.plot([vec2[0, 0], vec2[1, 0], vec2[2, 0], vec2[3, 0], vec2[4, 0], vec2[5, 0], vec2[6, 0], vec2[7, 0], vec2[8, 0],
55           vec2[9, 0], vec2[10, 0], vec2[11, 0], vec2[12, 0], vec2[13, 0], vec2[14, 0], vec2[15, 0], vec2[16, 0],
56           vec2[17, 0], vec2[18, 0], vec2[19, 0], vec2[20, 0]],
57           [vec3[0, 0], vec3[1, 0], vec3[2, 0], vec3[3, 0], vec3[4, 0], vec3[5, 0], vec3[6, 0], vec3[7, 0], vec3[8, 0],
58           vec3[9, 0], vec3[10, 0], vec3[11, 0], vec3[12, 0], vec3[13, 0], vec3[14, 0], vec3[15, 0], vec3[16, 0],
59           vec3[17, 0], vec3[18, 0], vec3[19, 0], vec3[20, 0]], marker = "o", )
60 plt.ylabel('Durchschnitt von Log')
61 plt.xlabel('cm mit log')
62 plt.show()

```

Listing 5.2: Lineare Regression

A.1.3 Quellcode Versuch 3

```
1 import numpy as np
2
3
4 data = 0
5 data2 = 0
6 vec1 = np.zeros((2, 3))
7 korrektur_breit1 = 0
8 korrektur_lang1 = 0
9 korrektur_breit2 = 0
10 korrektur_lang2 = 0
11 faktor1 = 1
12 faktor2 = 1.96
13
14 data = np.genfromtxt('data/DinA4Breit.csv', delimiter=",", skip_header=1000, skip_footer=499,
15                     usecols=(4))
16 data2 = np.genfromtxt('data/DinA4Lang.csv', delimiter=",", skip_header=1000, skip_footer=499,
17                      usecols=(4))
18
19 vec1[0, 0] = 21
20 vec1[0, 1] = np.mean(data)
21 vec1[0, 2] = np.std(data)
22
23 vec1[1, 0] = 29.7
24 vec1[1, 1] = np.mean(data2)
25 vec1[1, 2] = np.std(data2)
26
27 korrektur_breit1 = 0.861 + faktor1 * vec1[0, 2]
28 korrektur_lang1 = 0.662 + faktor1 * vec1[1, 2]
29 korrektur_breit2 = 0.861 + faktor2 * vec1[0, 2]
30 korrektur_lang2 = 0.662 + faktor2 * vec1[1, 2]
31
32 print("68,26%: " + str(korrektur_breit1))
33 print("68,26%: " + str(korrektur_lang1))
34 print("95%: " + str(korrektur_breit2))
35 print("95%: " + str(korrektur_lang2))
```

Listing 5.3: Flächenmessung mit Fehlerrechnung

A.2 Messergebnisse

Distanz	Mittelwert	Einheit
10 cm	1,34 V	
13 cm	1,15 V	
16 cm	1,05 V	
19 cm	935 mV	
22 cm	838 mV	
25 cm	775 mV	
28 cm	696 mV	
31 cm	657 mV	
34 cm	617 mV	
37 cm	580 mV	
40 cm	560 mV	
43 cm	518 mV	
46 cm	499 mV	
49 cm	479 mV	
52 cm	457 mV	
55 cm	434 mV	
58 cm	413 mV	
61 cm	395 mV	
64 cm	394 mV	
67 cm	395 mV	
70 cm	374 mV	
Länge S.	29,7 cm	662 mV
Breite S.	21 cm	861 mV
Luca Strattmann, Tobias Schuch		8.4.19
<i>Luca Strattmann</i>		<i>Tobias Schuch</i>

Abbildung 5.1: Messungen aus dem Labor