



Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Signale, Systeme und Sensoren

Versuch 1 - Abstandsmesser

Christoph Kaiser, Marcel Kanne

Konstanz, 21. April 2018

Zusammenfassung (Abstract)

| | | |
|-----------|-----------------------------|-----------------------------------|
| Thema: | Versuch 1 - Abstandsmesser | |
| Autoren: | Christoph Kaiser | christoph.kaiser@htwg-konstanz.de |
| | Marcel Kanne | marcel.kanne@htwg-konstanz.de |
| Betreuer: | Prof. Dr. Matthias O. Franz | mfranz@htwg-konstanz.de |
| | Jürgen Keppler | juergen.keppler@htwg-konstanz.de |
| | Mert Zeybek | me431zey@htwg-konstanz.de |

In Versuch 1 des Praktikums im Fach Signale, Systeme und Sensoren wird ein Infrarot-Entfernungsmesser verwendet, der nach dem Triangulationsprinzip funktioniert. Bei dem Versuch selbst werden mit Hilfe des Entfernungsmessers 20 Abstandsmessungen im Bereich von 10cm bis 67cm durchgeführt, wobei jede Messung einen Abstand von 3 cm ausweist. Um die Messungen möglichst genau zu halten, wird ein Metermaß parallel zum Entfernungsmesser gelegt. An den jeweiligen Abständen wird nun ein im 90° abgewinkeltes Holzbrett gestellt, an dem das Infrarotlicht des Entfernungsmessers reflektieren kann. Schließlich wird mittels eines Oszilloskops die Messdaten aufgezeichnet und in ein entsprechendes Format abgespeichert. Somit können Kennlinie, sowie Mittelwert der Messungen berechnet werden. Zu guter Letzt wird ein DIN A4 Blatt zur Hilfe genommen, um die Längen und Breiten aufzuzeichnen. Mit Hilfe dieser Aufzeichnungen wird nun der Flächeninhalt des DIN A4 Blattes berechnet.

Inhaltsverzeichnis

| | |
|--|-----------|
| Abbildungsverzeichnis | IV |
| Tabellenverzeichnis | V |
| Listingverzeichnis | VI |
| 1 Einleitung | 1 |
| 2 Versuch 1: Ermittlung der Kennlinie des Abstandssensors | 2 |
| 2.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 2 |
| 2.1.1 Fragestellung | 2 |
| 2.1.2 Messprinzip | 2 |
| 2.1.3 Aufbau | 3 |
| 2.1.4 Messwerte | 5 |
| 2.1.5 Auswertung | 6 |
| 2.1.6 Interpretation | 6 |
| 3 Versuch 2 | 8 |
| 3.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 8 |
| 3.1.1 Fragestellung | 8 |
| 3.1.2 Messwerte | 9 |
| 3.1.3 Auswertung | 9 |
| 3.1.4 Interpretation | 10 |
| 4 Versuch 3 | 11 |
| 4.1 Fragestellung, Messprinzip, Aufbau, Messmittel | 11 |
| 4.1.1 Fragestellung | 11 |
| 4.1.2 Messwerte | 12 |

| | | |
|-----------------------------|-------------------------------|-----------|
| 4.1.3 | Auswertung | 12 |
| 4.1.4 | Interpretation | 12 |
| Anhang | | 13 |
| A.1 | Quellcode | 13 |
| A.1.1 | Quellcode Versuch 1 | 13 |
| A.1.2 | Quellcode Versuch 2 | 14 |
| A.1.3 | Quellcode Versuch 3 | 15 |
| A.2 | Messergebnisse | 18 |
| Literaturverzeichnis | | 19 |

Abbildungsverzeichnis

| | | |
|-----|---------------------------------|----|
| 2.1 | Triangulationsprinzip | 3 |
| 2.2 | Versuchsaufbau | 4 |
| 2.3 | Verkabelung | 4 |
| 2.4 | Messwerte | 5 |
| 2.5 | Kennlinie | 6 |
| 3.1 | Lineare Regression | 9 |
| 5.1 | Messerprotokoll | 18 |

Tabellenverzeichnis

Listingverzeichnis

1

Einleitung

Optische Abstandssensoren finden in unserer heutigen Welt immer mehr Einzug. Das System selbst funktioniert auf vielerlei Wege. Der Weg der Triangulation ist dabei bereits seit 800 n.Ch bekannt und wird seither verwendet. In unserer heutigen Zeit ist dieses System nicht mehr wegzudenken. Da das System schnell, kostengünstig und zuverlässig arbeitet, sind Abstandssensoren in der Automobilindustrie unabdingbar. Sie sorgen für ein komfortables und zeitgleich sicheres Fahren, stressfreies Einparken und vieles mehr. Im folgenden wird festgehalten, wie das Berechnen der Kennlinie, sowie das Kalibrieren des Infrarot-Entfernungsmesser funktioniert wird, um so eine Entfernungsmessung durchzuführen. [1]
[2]

2

Versuch 1: Ermittlung der Kennlinie des Abstandssensors

2.1 Fragestellung, Messprinzip, Aufbau, Messmittel

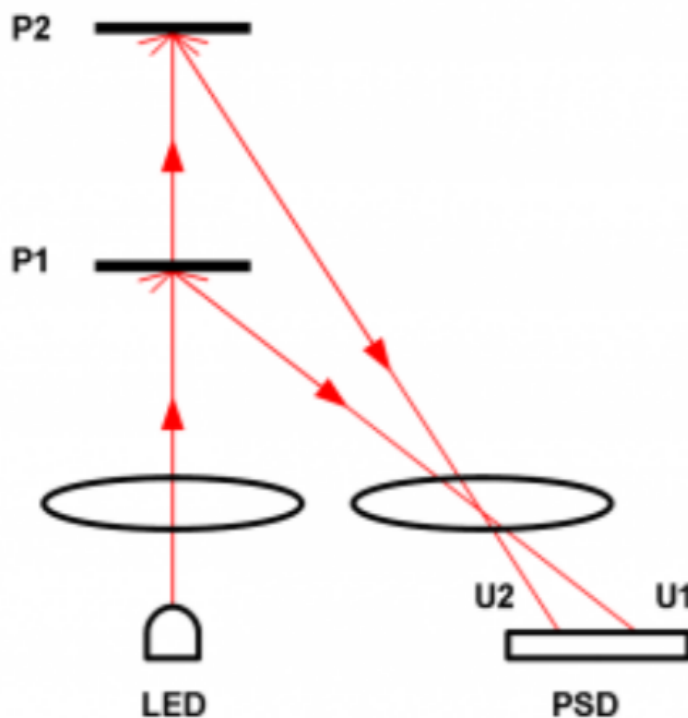
2.1.1 Fragestellung

Der erste Teil der Aufgabe besteht darin, die Ausgangsspannung des Infrarot-Entfernungsmessers mittels eines Oszilloskops zu bestimmen. Hierfür werden 20 Entfernungswerte im Bereich von 10cm bis 67cm gemessen und die entstehenden Resultate mit Hilfe der Toolbox 'TekTDS2000', sowie einen Rechner ausgelesen. Eine Bildschirmaufnahme der Software soll dies verdeutlichen. Im Anschluss wird mit der Programmiersprache Python, welche sich besonders gut für große Datenmengen eignet, ein Skript entwickelt, welches aus den gemessenen Daten den Mittelwert, sowie die Standardabweichen berechnet und dieser in Form eines Koordinatensystems ausgibt. Anhand des Koordinatensystems können die Daten somit mit den des Oszilloskops verglichen werden.

2.1.2 Messprinzip

Zum Messen der Entfernungen werden unter anderen ein Computer mit der Programmiersprache Python, ein Oszilloskop, sowie ein Abstandssensor vom Typ GP2Y0A21YK0F der Firma Sharp und ein entsprechendes Netzteil, welches eine kontinuierliche Versorgungsspannung von 5V bereitstellt, benötigt. Der Abstandssensor funktioniert nach dem Triangulationsprinzip. Im inneren des Sensors befindet sich eine Infrarot-Diode, die mit Hilfe einer Linse einen gebündelten Lichtstrahl aussendet. Befindet sich ein Gegenstand in der Laufbahn

des Lichtstrahls, wird dieser reflektiert und durch eine zweite Linse gebündelt. Anschließend wird der Strahl von einem optischen Positionssensor(OPS, engl. position sensitive detector PSD, siehe Abbildung) eingefangen. Der Ein- bzw. Austrittswinkel des Lichtstrahles spielt dabei eine entscheidende Rolle. Sollte der Winkel nicht stimmen, können fehlerhafte Messwerte die Folge sein. Stimmen beide Winkel, können die Lichtstrahlen mit Hilfe eines Signalprozessors in eine Spannung umgerechnet werden, die wiederum als Ausgangsspannung am Sensor anliegt.



Quelle: roboticlab.eu

Abbildung 2.1: Triangulationsprinzip

2.1.3 Aufbau

Der Infrarot-Abstandssensor wird mit 5V Versorgungsspannung des Netzteils versorgt. Der Minuspol des Abstandssensors, sowie der Signalempfänger des Oszilloskops werden mit dem Minuspol des Netzteils verbunden. Die Austrittsspannung des Sensors wird an den Signalempfänger des Oszilloskops angeschlossen. Der offene Seite des Sensors wird in einer Geraden zum Objekt gestellt. Beim Objekt handelt es sich in unserem Versuch um ein im 90°

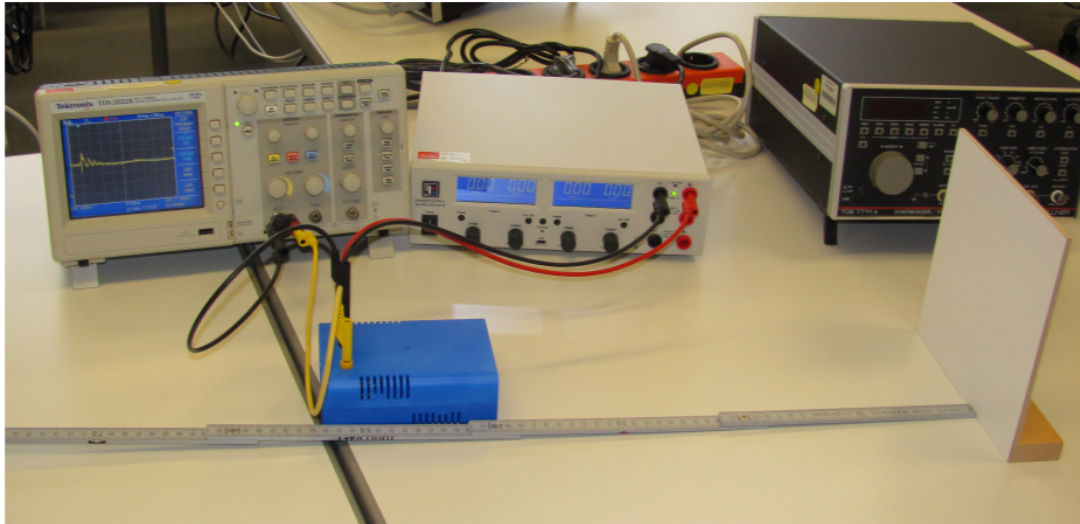


Abbildung 2.2: Versuchsaufbau

angewinkelt Holz Brett, welches das Licht reflektieren soll. Um genaue Messergebnisse zu erhalten, ist darauf zu achten, dass der gesamte Aufbau in einer Geraden liegt, da das Licht sonst nicht korrekt reflektiert werden kann. Die Abstände können mit einem Metermaß oder ähnliches zurecht gerückt werden. Um die Längen- und Breitenangaben des DIN A4 Blattes zu erhalten, wird dieses direkt vor dem Abstandssensor gelegt. Das Holzbrett wird nun an die zu messenden Kanten des Blattes positioniert, um dort den Lichtstrahl zu reflektieren.

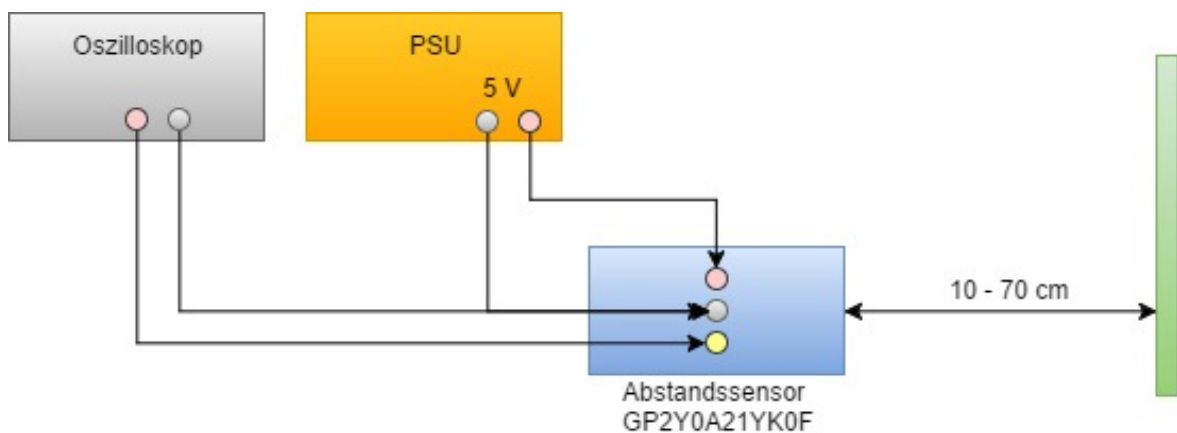


Abbildung 2.3: Verkabelung

2.1.4 Messwerte

03.04.18

Versuch 1

Beteiligte: Christoph Kaiser 234182
Marcel Kanne 234074

| (in cm) Distanz | Mittelwert der Spannung |
|-----------------|-------------------------|
| 10 | 2,02 V |
| 13 | 1,76 V |
| 16 | 1,51 V |
| 18 | 1,35 V |
| 22 | 1,20 V |
| 25 | 1,08 V |
| 28 | 1,00 V |
| 31 | 931 mV |
| 34 | 874 mV |
| 37 | 803 mV |
| 40 | 773 mV |
| 43 | 757 mV |
| 46 | 717 mV |
| 48 | 666 mV |
| 52 | 597 mV |
| 55 | 577 mV |
| 58 | 545 mV |
| 61 | 522 mV |
| 64 | 508 mV |
| 67 | 486 mV |

kurze Seite = 1,14 V
Lange Seite = 836 mV

Gunnar Kettner

Abbildung 2.4: Messwerte

2.1.5 Auswertung

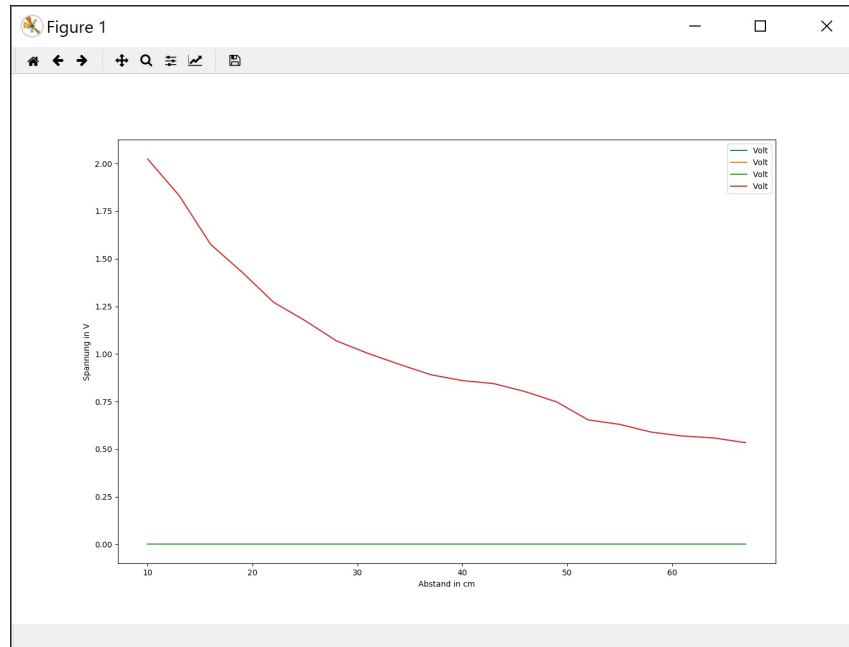


Abbildung 2.5: Kennlinie

Beim Ausmessen der Ergebnisse wurde festgestellt, dass die Ausgangsspannung des Infrarot-Abstandssensors, bei einem Abstand von 10 cm zum reflektierenden Objekt, 2,02V beträgt. Bei einem Abstand von 67 cm beträgt die Ausgangsspannung lediglich 496mV. Die Ergebnisse des DIN A4 Blattes betragen 1,14V an der kurzen Seite und 896mV an der langen Seite.

2.1.6 Interpretation

Während Drehspul-Messwerke durch den Parallaxen Fehler, aber auch durch ungewöhnliche Temperaturen ungenaue Messwerte liefern, erhält man mit Hilfe des Oszilloskops hoch genaue Ergebnisse. Da beim Aufbau des Versuchs, aber auch beim Ablauf selbst kleine Veränderungen Ungenauigkeiten hervorrufen können, müsste man das Ergebnis für jeden Abstand mehrfach bestimmen, um so die systematische Unterschätzung des Messfehlers zu reduzieren. Laut der Analyse unserer aufgezeichneten Messdaten, nimmt die Ausgangsspannung antiproportional und exponentiell zum Abstand ab. Dies führt bei zu großen Abständen zu einer ungenauen Messdatenaufzeichnung, da so zudem Störsignale aufgezeichnet werden. Da das Lichtsignal zudem, bedingt durch eine nicht genaue Positionierung der Holzplatte,

nicht korrekt reflektiert werden kann, werden die Messdaten verstärkt verfälscht. Somit muss bei jeder Messdurchführung auf eine genaue Positionierung des Aufbaus geachtet werden, damit die Resultate für den weiteren Ablauf brauchbar sind.

3

Versuch 2

3.1 Fragestellung, Messprinzip, Aufbau, Messmittel

3.1.1 Fragestellung

Mit Hilfe der aufgezeichneten Messwerte soll nun eine Umrechnungsvorschrift mittels Python angewandt werden, mit der wir den Zusammenhang in Form einer Grafik darstellen können. Da der Sharp-Entfernungssensor nicht auf der Methode der linearen Regression der Form $y = x^a$ anwendbar ist und dieser somit eine nicht lineare Kennlinie ausweist, müssen sowohl die Eingangs- als auch die Ausgangsspannungen logarithmiert werden. Hierdurch erhalten wir neue Paare aus Eingangs- und Ausgangswerten $x = e^{x'}$. Setzt man beides in die Kennlinie ein, so ergibt sich $y' = \ln(y) == \ln(x^a) = a * \ln(x) = a * \ln(e^{x'}) = a * x'$. Nach der doppelten Logarithmierung ist die Kennlinie eine Gerade mit Steigung a. Die Parameter dieser Gerade können wir wiederum mit der linearen Regression schätzen. Somit erhalten wir aus der Regression die Parameter a und b für den Zusammenhang $y' = a * x' + b$. Mit Hilfe der doppelten Logarithmierung geschieht die Rückrechnung auf den ursprünglichen Zusammenhang $y = \exp(a * \ln(x) + b) = e^b * x^a$.

3.1.2 Messwerte

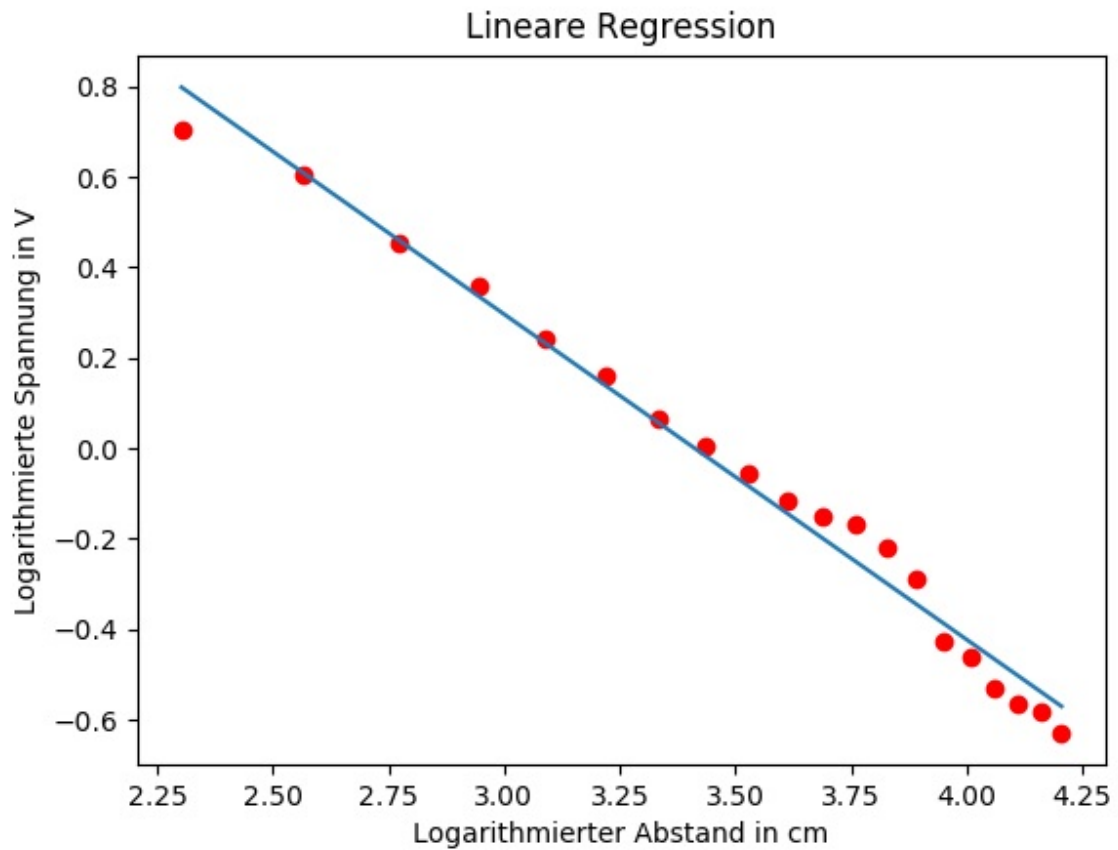


Abbildung 3.1: Lineare Regression

3.1.3 Auswertung

Aus den aufgezeigten Messwerten ist erkennbar, dass die lineare Regression die Ausgangskennlinie deutlich gleichmäßiger darstellt.

3.1.4 Interpretation

Um eine genaue Kennlinie bestimmen zu können, muss erneut mehrfache Messungen erfolgen, um Schwankungen zu reduzieren. Außerdem können die Schwankungen verstärkt reduziert werden, wenn beim Aufbau genaustens gearbeitet wird, damit Sensor und Zielobjekt auf der selben Gerade liegen. Beim Auslegen des Meterstabs können zudem kleine Schwankungen entstehen, da nur millimetergenau gearbeitet werden kann.

4

Versuch 3

4.1 Fragestellung, Messprinzip, Aufbau, Messmittel

4.1.1 Fragestellung

In Versuch 3 wollen wir den Messfehler des Abstandssensors bestimmen. Da der Abstand nicht direkt, sondern über eine Kombination aus Sharp-Sensor, Oszilloskop und berechneter Kennlinie, erzeugt wird, müssen wir zudem die Fehlerfortpflanzung durch die Kennlinie $e^b x^a$ berechnen. Der Messfehler soll mit bisher erlernten Methoden aus der Vorlesung geschätzt werden. Zur Berechnung selbst muss auf Grund der Anzahl der Messungen eine gewisse Korrektur eingebunden werden. Dabei soll der Vertrauensbereich für eine Sicherheit von 68% und 95% berechnet werden. An Hand eines DIN A4 Blattes soll nun der Flächeninhalt bestimmt werden. Die 20 zuvor gemessenen Messwerte sollen mit Hilfe einer Umrechnungsvorschrift die Spannungen in Entfernungswerte konvertiert werden. Um die Fläche des Blattes und das des Messergebnisses korrekt schätzen zu können, soll das Gaußsche Fehlerfortpflanzungsgesetz aus der Vorlesung angewandt werden. Die Berechnungen erfolgen mit Python und dem mathematischen Formelsatz von Numpy.

4.1.2 Messwerte

| | Gemessene Werte | Python berechneter Mittelwert |
|--------|-----------------|-------------------------------|
| Länge | 896 mV | 1.1511 V |
| Breite | 1,14 V | 0.9176 V |

Lange Seite:

Standardabweichung: $s_{\bar{x}} = \sqrt{s^2} = \sqrt{\frac{1}{n*(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2} = \frac{s}{\sqrt{n}} = 0.004955$

Vertrauensbereich 68%: $x = \bar{x} \pm t * s_{\bar{x}} = 1.1511 \text{ V} \pm 1 * 0.004955 = 1.1511 \text{ V} \pm 0.004955$

Vertrauensbereich 95%: $x = \bar{x} \pm t * s_{\bar{x}} = 1.1511 \text{ V} \pm 1.96 * 0.004955 = 1.1511 \text{ V} \pm 0.009712$

Breite Seite:

Standardabweichung: $s_{\bar{x}} = \sqrt{s^2} = \sqrt{\frac{1}{n*(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2} = \frac{s}{\sqrt{n}} = 0.003821$

Vertrauensbereich 68%: $x = \bar{x} \pm t * s_{\bar{x}} = 0.9176 \text{ V} \pm 1 * 0.003821 = 0.9176 \text{ V} \pm 0.003821$

Vertrauensbereich 95%: $x = \bar{x} \pm t * s_{\bar{x}} = 0.9176 \text{ V} \pm 1.96 * 0.003821 = 0.9176 \text{ V} \pm 0.007489$

4.1.3 Auswertung

Nach den Resultaten konnte in Python eine Breite von 24.94 cm und eine Länge von 34.181 cm berechnet werden. Die Fläche des DIN A4 Blattes liegt bei 852.41 cm^2 . Diese Werte liegen nicht so fern von den realen Maßen.

4.1.4 Interpretation

Die Messung des DIN A4 Blattes liefert größere als die zu erwartenden Werte. Dies liegt wahrscheinlich an den Lichtverhältnissen, sowie an den Gegebenheiten der Messplatte (Unebenheiten), die die Reflektion des Infrarotstrahls verfälscht hat. Jedoch sind diese Werte noch recht nah an den Realen. Ein weiterer Grund für die Abweichung können natürlich auch Ungenauigkeiten beim Aufbau, wie nicht symmetrisches Aufstellen der Platte und natürlich die Abweichung der Messung vom wahren Wert.

DIN A4:

Breite: 21 cm

Länge: 29.7 cm

Fläche: 625 cm^2

Anhang

A.1 Quellcode

A.1.1 Quellcode Versuch 1

```
import numpy as np
import matplotlib.pyplot as plt
import cmath as math

#Versuch 1
s = []
logLenList = [] # Logarithmierte Laengen
wertList = [] # Logarithmierte Mittelwerte

for i in range(10, 68, 3):
    s.append(str(i) + '.cm.csv')
    logLenList.append(np.log(i))

datamittelarr = []
datastdarr = []
datamittelwert = 0
average = 0

for i in s:
    data = np.genfromtxt(i, delimiter=",", skip_header=1000)
    data = data[:,0:-1]

    for j in range(0, len(data)):
```

```

        datamittelwert = data[j] + datamittelwert

    datamittelwert = datamittelwert[1:]
    datamittelwert = datamittelwert/len(data)
    datamittelarr.append(datamittelwert)
    datamittelwert = 0

    datastd = np.std(data[:, -1])
    datastdarr.append(datastd)
    wertList.append(np.log(np.average(data[:, -1])))

distanz = np.zeros((len(datamittelarr)))
indx = 0
for i in range(0, len(distanz)*3, 3):
    distanz[indx] = i + 10.0
    indx += 1

# Ausgabe der berechneten Mittelwerte
for i in datamittelarr:
    print(i)

```

A.1.2 Quellcode Versuch 2

```

# Versuch 2
wertList2 = [] # Logarithmierte Mittelwerte
tmpSum1 = 0
tmpSum2 = 0

# Berechnung der Linearen Regression
for i in range(0, len(wertList)):
    tmpY = (wertList[i] - np.average(wertList))
    tmpX = (logLenList[i] - np.average(logLenList))
    tmpSum1 += tmpY*tmpX
for i in logLenList:
    tmpSum2 += (i - np.average(logLenList))**2
a = tmpSum1/tmpSum2

```

```

b = np.average(wertList) - (a*np.average(logLenList))
print("a = ", a , "; b = " , b)
linR = []
for x in logLenList:
    linR.append(a*x+b)

# Ausgabe Kennlinie
plt.plot(distanz , datamittelarr , label='Volt ')
plt.ylabel('Spannung in V')
plt.xlabel('Abstand in cm')
plt.legend()
plt.show()

```

A.1.3 Quellcode Versuch 3

```

# Versuch 3
# Einlesen der benoetigten Dateien
s2 = []
s2.append('kurz.csv')
s2.append('lang.csv')

std1 = []
std2 = []
data1 = []
data2 = []

for i in s2:
    if i == 'kurz.csv':
        data1 = np.genfromtxt(i, delimiter=",",
                               skip_header=1000)
        data1 = data1[:,0:-1]
        # Mittelwerte der gemessenen kurzen Seite
        wertList2.append((np.average(data1[:, -1])))

    if i == 'lang.csv':
        data2 = np.genfromtxt(i, delimiter=",",

```

```

skip_header=1000)
data2 = data2[:,0:-1]
# Mittelwerte der gemessenen langen Seite
wertList2.append((np.average(data2[:, -1])))

for i in data1[:, -1]:
    std1.append(math.sqrt(np.average(abs(i - np.average(i))**2)))
for i in data2[:, -1]:
    std2.append(math.sqrt(np.average(abs(i - np.average(i))**2)))

print("Standardabweichung kurze Seite: ", np.std(data1[:, -1]))
print("Standardabweichung lange Seite: ", np.std(data2[:, -1]))

mk = wertList2[0]
ml = wertList2[1]
kurz = (wertList2[0]/np.exp(b))**(1/a)
lang = (wertList2[1]/np.exp(b))**(1/a)
print("Mittelwert kurz: ", mk, "V")
print("Mittelwert lang: ", ml, "V")
print("Kurz:", kurz, "cm")
print("Lang:", lang, "cm")
print(kurz * lang, "cm^2")

# Berechnung der linearen Regression
logLenList2 = [] # Logarithmierte Laengen

logLenList2.append(np.log(21))
logLenList2.append(np.log(29.7))

for i in range(0, len(wertList2)):
    tmpY = (wertList2[i] - np.average(wertList2))
    tmpX = (logLenList2[i] - np.average(logLenList2))
    tmpSum1 += tmpY*tmpX
for i in logLenList2:

```

```

        tmpSum2 += (i - np.average(logLenList2))**2
a = tmpSum1/tmpSum2
b = np.average(wertList2) - (a*np.average(logLenList2))
print("a = ", a , "; b = " , b)
linR2 = []
for x in logLenList2:
    linR2.append(a*x+b)

# Ausgabe Lineare Regression
plt.plot(logLenList2 , wertList2 , "ro")
plt.plot(logLenList2 , linR2)
plt.ylabel('Logarithmierte Spannung in V')
plt.xlabel('Logarithmierter Abstand in cm')
plt.title('Lineare Regression')
plt.show()

```


A.2 Messergebnisse

08.04.18

Versuch 1

Beteiligte: Christoph Kaiser 2341132
Marcel Kanne 2341074

| (in cm) Distanz | Mittelwert der Spannung |
|-----------------|------------------------------------|
| 10 | 2,02 V |
| 13 | 1,76 V |
| 16 | 1,51 V |
| 18 | 1,35 V |
| 22 | 1,20 V |
| 25 | 1,08 V |
| 28 | 1,00 V |
| 31 | 881 mV |
| 34 | 874 mV |
| 37 | 803 mV |
| 40 | 773 mV |
| 43 | 757 mV |
| 46 | 717 mV |
| 49 | 666 mV |
| 52 | 597 mV |
| 55 | 577 mV |
| 58 | ⁴⁵ 548 mV |
| 61 | 522 mV |
| 64 | 508 mV |
| 67 | 496 mV |

kurze Seite = 1,14 V
Lange Seite = 836 mV

Gunnar Kanne

Abbildung 5.1: Messerprotokoll

Literaturverzeichnis

- [1] Franz, Prof. Dr. Matthias O.: *Vorlesung 10 - Sprache und Spracherkennung: Kurzzeit-Fouriertransformation, Erzeugung und Wahrnehmung von Sprache, Mustererkennung durch Korrelation*. In: *Vorlesung Technische Grundlagen der angewandten Informatik*, 2016.
- [2] Franz, Prof. Dr. Matthias O.: *Vorlesung 1 - Einführung*. In: *Vorlesung Technische Grundlagen der angewandten Informatik*, 2016.