

Bachelor-Projektarbeit

im

Bachelor-Studiengang Elektro- und Informationstechnik (PO 12)

ENTWURF UND KONSTRUKTION EINER RAUCHKAMMER FÜR THZ MESSUNGEN

angefertigt von

Nils Brinkmann - 3049005

Onno Eberhard - 3048563

Jonatan Stock - 3048714

bei

Prof. Dr.-Ing. Jan C. Balzer

Zweitprüfer

Prof. Dr.-Ing. Andreas Czulwik

Fachgebiet

Nachrichtentechnische Systeme

an der

Universität Duisburg-Essen

Duisburg, 15. November 2019

Danksagungen

An dieser Stelle möchten wir uns gerne bei all denjenigen bedanken, die uns während der Ausfertigung dieser Bachelor-Projektarbeit unterstützend und motivierend zur Seite standen.

Zuallererst gebührt unser Dank Herrn Dr.-Ing. Thorsten Schultze, der es uns nicht nur ermöglichte, dieses Projekt zu beginnen, sondern uns auch die volle Zeit mit Rat und Tat unterstützt hat.

Ebenfalls gilt unser Dank Barbara Brox, die uns bei der Einrichtung unseres Arbeitsplatzes behilflich war und Benedikt Friederich, der uns mit der Einführung und Erklärungen zum Arbeiten mit dem 3D-Drucker entscheidende Fertigkeiten zur Realisierung unseres Projekts an die Hand gab.

Abschließend bedanken wir uns bei dem Fachbereich Nachrichtentechnische Systeme, insbesondere bei Prof. Dr.-Ing. Jan C. Balzer und Prof. Dr.-Ing. Andreas Czulwik, für die Bereitstellung der notwendigen Ressourcen und Räumlichkeiten zur Bearbeitung der Bachelor-Projektarbeit.

Three handwritten signatures in blue ink, arranged horizontally. The first signature is 'Mit Blum', the second is 'Ovo Eberhard', and the third is a stylized signature.

Duisburg, 26.04.2020

Inhalt

Danksagungen

Abbildungsverzeichnis

1	Motivation	1
2	Theoretischer Hintergrund	2
2.1	Aufbau des optischen Rauchmelders	2
2.2	Eigenschaften des Nephelometers	4
2.2.1	Integration über die Zeit.....	4
2.3	Terahertzstrahlung.....	7
3	Organisation der Projektarbeit.....	10
3.1	Ziele und Projektmanagement.....	10
3.2	Qualitätssicherung.....	12
4	Durchführung	13
4.1	Durchführung Gestell.....	13
4.1.1	Verwendete Materialien	14
4.1.2	Aufbau des Gestells	15
4.1.3	Nebelmaschine	17
4.1.4	Rotationseinheit.....	17
4.1.5	Filtersystem	18
4.2	Durchführung Nephelometer-Gehäuse.....	20
4.2.1	Fertigung mit dem 3D-Drucker.....	21
4.2.2	Entwürfe im Zuge der Gehäuseentwicklung.....	22
4.2.3	Lichtfang.....	25
4.2.4	Testdrucke.....	27
4.2.5	Deckel und Arduino-Gehäuse	29
4.3	Elektronische Umsetzung des Nephelometers	31

4.3.1	Arbeitspunktanalyse in der LED-Schaltung	33
4.3.2	Auswahl einer geeigneten Integrationszeit.....	40
4.3.3	Referenzmessung zur Umrechnung der Messwerte in dB	42
4.3.4	Bedienungsprogramm und Endprodukt	52
5	Fazit und Ausblick.....	56
A	Technische Zeichnungen.....	57
A.1	Grundgestell	57
A.1.1	Dreitafelprojektion: Grundgestell	57
A.1.2	Dreitafelprojektion: Deckel & Zusammenbau	59
A.2	Nephelometer	61
A.2.1	Dreitafelprojektion: Dunkelkammer.....	61
A.2.2	Dreitafelprojektion: Deckel	63
A.2.3	Dreitafelprojektion: Arduino Gehäuse	65
B	Projektmanagement.....	67
B.1	GANTT-Diagramm.....	67
B.2	ToDo-Liste	71
C	Schaltplan.....	75
D	Quellcodes	77
D.1	Spice-Modell: LED	77
D.2	Spice-Modell: Transistor	77
D.3	Messung / Plot (Arduino)	78
D.4	Auswertung und Grafiken (Python).....	79
D.5	Nephelometer-Steuerung (Arduino)	86
D.6	Bedienungsprogramm (MATLAB).....	87
	Literaturverzeichnis	90
	Versicherung an Eides statt	92

Abbildungsverzeichnis

Abbildung 2.1: Prinzip eines Streulichtrauchmelders	3
Abbildung 2.2: Einordnung des Terahertz-Spektrums	8
Abbildung 3.1: Grobe Vorplanung.....	11
Abbildung 4.1: 40x40 Profil wie verwendet.....	14
Abbildung 4.2: Prinzipieller Aufbau eines Nephelometers	20
Abbildung 4.3: Dunkelkammer, erster Entwurf	22
Abbildung 4.4: Dunkelkammer, zweiter Entwurf	23
Abbildung 4.5: Dunkelkammer, dritter Entwurf	24
Abbildung 4.6: Dunkelkammer, vierter Entwurf/finaler Entwurf	25
Abbildung 4.7: Abstrahlcharakteristik der verwendeten IRED [11]	26
Abbildung 4.8: CAD-Modell der Wandstruktur aus dem finalen Entwurf.....	28
Abbildung 4.9: links: CAD-Modell Lichtfang; rechts: CAD-Modell Ring	29
Abbildung 4.10: Entwurf des Gehäuses für den Mikrocontroller	30
Abbildung 4.11: Entwurf des Deckels der Dunkelkammer	30
Abbildung 4.12: Die LED-Schaltung.....	33
Abbildung 4.13: SPICE-Simulationsschaltungen zur Bestimmung der genauen Kennlinien von LED und FET	34
Abbildung 4.14: U-I-Kennlinie der LED.....	35
Abbildung 4.15: Ausgangskennlinie des Transistors ($U_{GS} = 5\text{ V}$)	36
Abbildung 4.16: Iteration aus Tabelle 4.2 in graphischer Darstellung	38
Abbildung 4.17: Simulationsschaltung mit der der Graph in Abbildung 4.18 erstellt wurde.....	39
Abbildung 4.18: Schaltverhalten mit $R = 33\ \Omega$ bei $U_0 = 5\text{ V}$	39
Abbildung 4.19: Testmessung zur Bestimmung der Integrationszeit T	41
Abbildung 4.20: Ergebnis der Referenzmessung	43
Abbildung 4.21: Die Messdaten $x(t)$ und die jeweils verschobenen Daten $x(t)$, bei denen der Hintergrundstörwert abgezogen wurde.....	45
Abbildung 4.22: Umrechnungsfaktoren zwischen den Messwerten verschiedener Integrationszeiten	46

Abbildung 4.23: Links: Die Fensterfunktion $g(t)$ mit verschiedenen Werten des Parameters α . Rechts: Die verschobenen Messdaten $x(t)$ ($T = 200$ ms)	48
Abbildung 4.24: Oben: Die Messung $m(t)$ Mittig: Die Messung $x(t)$ Unten: Die zwei Kurven $m(t)$ und $y(t)$	49
Abbildung 4.25: Die resultierenden Transformationen	51
Abbildung 4.26: Die grafische Oberfläche des Bedienungsprogramms.	54
Abbildung 4.27 Links: Messung bei niedriger Temperatur Rechts: Messung bei hoher Temperatur	54

1 Motivation

In der Welt der Wissenschaft ist es von großer Relevanz, dass Forschungsergebnisse anschaulich dargestellt werden. Für Dritte ist es so leichter, einen komplexen Sachverhalt zu verstehen. Erfolgreich kommunizierte Forschungsergebnisse erreichen ein großes Publikum auch abseits der Wissenschaft und können so weiterführende Ideen anregen.

Terahertzstrahlung ist aktuell das Objekt intensiver Forschung. Sie liegt im Spektrum der elektromagnetischen Wellen zwischen der Infrarotstrahlung und der Mikrowellenstrahlung. Auch einige Fachbereiche der Universität Duisburg-Essen engagieren sich in diesem Forschungsgebiet und versuchen diesen Strahlungsbereich aus allen Richtungen zu erschließen. Da viele Materialien für Terahertzstrahlung unterschiedlich durchlässig sind, eignet sich Terahertzstrahlung, ähnlich wie Röntgenstrahlung, für bildgebende Verfahren. Diesen Eigenschaften der Terahertzstrahlung entspringt die Forschung hinsichtlich einer Anwendung im Bereich des Brandschutzes bzw. der Brandbekämpfung.

Um die Möglichkeiten dieser Technologie anschaulich darzustellen, soll ein Demonstrator entworfen werden, der die optischen Umgebungsbedingungen (Eintrübung der Luft durch Rauch) eines Brandes simuliert und somit einen Beleg für die Anwendbarkeit von Terahertzstrahlung in einem entsprechenden Einsatzgebiet liefert.

2 Theoretischer Hintergrund

Brände können Gründe für erhebliche Personen- und Sachschäden sein. So verursachten Brände im Jahr 2008 allein in Deutschland Sachschäden in einer Höhe von 2,85 Mrd. € [1]. Für das Jahr 2015 verzeichnet der deutsche Feuerwehrverband 192.078 Einsätze aufgrund von Bränden, wobei insgesamt 367 Menschen ihr Leben verloren [2]. Um die Höhe der Schäden möglichst gering zu halten, ist eine frühe Branderkennung von großer Relevanz. Je früher ein Brand entdeckt wird, desto eher können entsprechende Maßnahmen zur Brandbekämpfung eingeleitet werden. Eine effektive Branddetektion ist somit überall vonnöten, wo Menschen oder Sachen durch einen Brand zu Schaden kommen könnten. Da Brand keine direkt messbare Größe ist, muss auf andere (meist physikalische) verbrennungscharakteristische Größen zurückgegriffen werden wie z. B. Temperatur, Streuung und Dämpfung von Licht, Strahlungsintensität u. ä. Die energetischen und stofflichen Umsetzungen eines Brandes sind in ihren Ausprägungen (je nach Brandort und -ursache) sehr unterschiedlich, weshalb die verwendete Sensorik auf das Betriebsumfeld und somit auf die zu erwartenden Messgrößen und Messwertbereiche abgestimmt sein muss. Ein Schwelbrand¹, ausgelöst durch fehlerhafte Elektrik, hat beispielsweise eine starke Rauchentwicklung zur Folge, ohne dabei eine hohe Intensität an Wärme abzustrahlen. Ein wärmesensitiver Branddetektor würde in solch einem Fall erst einen Brand melden, wenn das Feuer des Schwelbrandes bereits auf heißer verbrennendes Material übergesprungen wäre. Auch aus diesem Grund werden in privaten Haushalten vor allem Rauchmelder zur Branddetektion eingesetzt.

2.1 Aufbau des optischen Rauchmelders

Es ist insbesondere der Rauch, der eine Gefahr für die Gesundheit des Menschen im Falle eines Brandes darstellt. Die zwei gängigsten Ausführungen eines Brandmelders sind der Ionisationsrauchmelder und der optische Rauchmelder. Dem Ionisationsrauchmelder soll an dieser Stelle jedoch keine weitere Beachtung geschenkt

¹ Verbrennung bei unzureichender Sauerstoffzufuhr und somit niedriger Verbrennungstemperatur.

werden, denn: „Ionisationsrauchmelder waren bis vor wenigen Jahren die am weitesten verbreiteten Melder für Brandrauch. Sie sind jedoch in zunehmenden Maße durch optische Rauchmelder ersetzt worden, weil der praktische Umgang des I-Melders durch die Strahlenschutzbestimmungen behindert wird.“ [1] Das grundlegende Prinzip eines Rauchmelders ist die Detektion von Partikeln in einem Aerosol². Der sogenannte Streulichtrauchmelder detektiert entsprechende Partikel über Lichtstreuung. Der exemplarische Aufbau eines solchen Rauchmelders besteht aus einer Strahlungsquelle, z. B. einer LED³ oder einer LD⁴ und einem Photosensor (i. d. R. eine Photodiode). LED und PD⁵ sind hier in einem rechten Winkel zueinander angeordnet (Abb. 2.1). Über die Variation dieses Winkels können zusätzliche Informationen über das Aerosol erlangt werden. Das in den Lichtkegel der LED gelangende Aerosol streut das Licht, je nach Zusammensetzung unterschiedlich stark. Die Photodiode detektiert das (in ihre Richtung) gestreute Licht und wandelt die Information der Strahlungsintensität in ein elektrisch nutzbares Signal um (z. B. Photostrom).

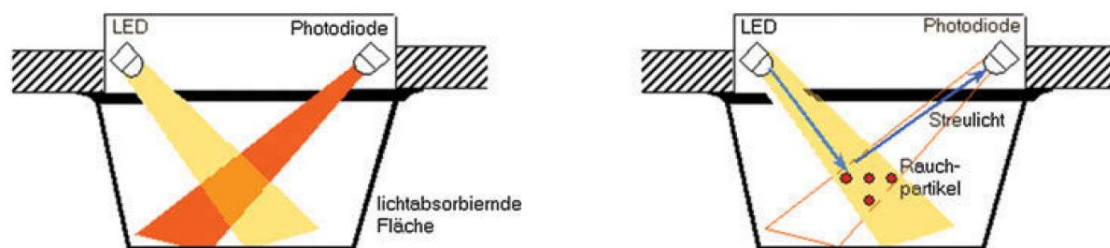


Abbildung 2.1: Prinzip eines Streulichtrauchmelders [1]

Damit Störsignale wie reflektiertes Licht oder Umgebungslicht einen möglichst geringen Einfluss auf das zu detektierende Signal nehmen, ist die Messanordnung in einer abgedunkelten Kammer, die im Bereich der Wellenlänge der LED einen geringen Reflexionskoeffizienten aufweist, zu installieren. Neben fremd eingestrahlt und ungewollt reflektiertem Licht können zudem auch andere Störgrößen auftreten. Es gibt eine Vielzahl an Aerosolen, die nicht von einem Brand ausgehen, für die ein

² Heterogenes Gemisch von Partikeln in einem Gas.

³ engl. light emitting diode (dt. Leuchtdiode, Lumineszenzdiode).

⁴ engl. laser diode (dt. Laserdiode).

⁵ Photodiode

Streulichtrauchmelder ebenfalls sensitiv ist. So können beispielsweise durch Wasserdampf oder Staubeinlagerungen Falschmeldungen ausgelöst werden.

Von außen betrachtet arbeitet ein Rauchmelder in seiner einfachsten Form nach dem Prinzip eines Grenzwertschalters, d. h. er sendet lediglich zwei Signalzustände: Grenzwert überschritten oder Grenzwert nicht überschritten. Interne Kalkulationen, Filter oder Ruhewertnachführungen werden bei dieser Betrachtungsweise nicht berücksichtigt.

2.2 Eigenschaften des Nephelometers

Um ein Aerosol näher zu untersuchen, werden die erzeugten Analogwerte der oben dargestellten Messapparatur ausgeleitet. Folglich lassen sich Messwerte in einem kontinuierlichen Wertebereich darstellen, die somit genauere Informationen über die untersuchte Suspension liefern. Ein solches Messinstrument wird als Nephelometer bezeichnet. Es ermöglicht die Anzahl der Partikel n , sowie deren Durchmesser d zu bestimmen. Des Weiteren kann indirekt, über die Anzahl der Partikel, die Dämpfung von Licht der verwendeten Wellenlänge λ bestimmt werden. Exemplarisch ergeben sich die entsprechenden Verhältnismäßigkeiten aus Gleichung 2.1 (Rayleigh-Gleichung) für ein kleines d gegenüber λ :

$$I \sim I_0 \cdot \frac{n \cdot d^6}{r^2 \cdot \lambda^4} \quad (2.1)$$

Mit der gestreuten Lichtintensität I , der einfallenden Strahlung I_0 und r dem Abstand zum beleuchteten Volumenelement. Bei der Charakterisierung eines Aerosols mittels Nephelometer wird davon ausgegangen, dass sich die Intensität des gestreuten Lichts nur äußerst langsam ändert.

2.2.1 Integration über die Zeit

Signalspitzen, die über einen kurzen Zeitraum auftreten, sind demnach als Störgrößen zu werten. Sie können beispielsweise durch einen Fremdkörper, der in den

Probenraum eindringt oder durch eine inhomogene Verteilung der Partikel im Aerosol hervorgerufen werden. Um diesen Einfluss der Störgrößen zu kompensieren kann das Messsignal numerisch aufbereitet werden. Das heißt, dass etwaige Signalspitzen durch den Einsatz entsprechender Algorithmik „herausgerechnet“ werden. Durch diese Methode lässt sich der Einfluss der Störgrößen nahezu vollständig kompensieren. Allerdings wird durch entsprechende Berechnungen zusätzlich eine leistungsstarke Berechnungseinheit nötig, die den Aufbau des Nephelometers komplexer macht. Darüber hinaus besteht die Möglichkeit, dass Teile des Messsignals fälschlicherweise als Störsignale erkannt werden und somit das Ausgangssignal verfälscht wird. Eine Alternative hierzu stellt die Integration des Messsignals über eine vorgegebene Zeit dar. Somit wird ein kontinuierlicher Mittelwert über die Integrationszeit ermittelt. Folglich hat eine Signalspitze umso weniger Einfluss auf den Messwert, je größer die Integrationszeit ist. Der Umkehrschluss ist, dass mit steigender Integrationszeit die zeitliche Auslösung des Messinstruments sinkt. Vorteil dieser Messmethode ist, dass jeder Teil des Signals (auch Störgrößen) einen Beitrag zum Messsignal liefern. Sogenannte false negatives⁶ sind dadurch ausgeschlossen. Prinzipiell kann eine Signalintegration numerisch oder elektrisch erfolgen. Bei der elektrischen Integration wird ein Kondensator über die Dauer der Integrationszeit geladen. Die nach Ablauf der Integrationszeit anliegende Spannung wird gemessen und ausgewertet.

Die gemessene Intensität des Streulichts, durch die die Dichte des Aerosols in der Dunkelkammer $x(t)$ beschrieben wird und welche durch diese LED-PD-Anordnung aufgenommen werden kann, ist eine Zufallsvariable mit starker Varianz. Für die Dämpfung, die mit dem Nephelometer bestimmt werden soll, ist dieses Signal zu stark verrauscht. Hier ist deutlich interessanter, was der momentane Mittelwert $\bar{x}(t)$ der gemessenen Dichte im Zeitraum $t - T$ bis t ist, wobei ein geeignetes T gefunden werden muss. Dieser Mittelwert kann gebildet werden, indem das Ausgangssignal der PD kontinuierlich in Intervallen der Periodendauer Δt abgefragt und numerisch ein Mittelwert über $N = \frac{T}{\Delta t}$ Werte berechnet wird. Dann würde gelten:

⁶ (dt. falsch Negativ), fälschliche Deklaration eines Signals als Störsignal

$$\bar{x}(t) = \frac{1}{N} \sum_{i=0}^{N-1} x(t - i\Delta t). \quad (2.2)$$

Da die genaue physikalische Abhängigkeit zwischen tatsächlicher Partikeldichte und gemessener Ausgangsspannung nicht bekannt ist, spielen genaue Spannungswerte hierbei keine wichtige Rolle. Diese Abhängigkeit muss im Folgenden durch eine Kalibrierung bestimmt werden. Es kommt deshalb in der obigen Rechnung nicht auf den konstanten Faktor $\frac{1}{N}$ an, sodass es ausreicht, nur die Summe der N gemessenen Spannungen zu berechnen, die dieselbe Information wie der Mittelwert enthält.

Eine Alternative zu dieser digitalen Berechnung ist die Verwendung eines analogen Integrators. In der Berechnung des Mittelwerts hat dies den Effekt, dass die Zeit Δt zwischen zwei Messungen gegen 0 geht:

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \bar{x}(t) &= \lim_{\Delta t \rightarrow 0} \frac{1}{T} \sum_{i=0}^{\frac{T}{\Delta t}} x(t - i\Delta t) \Delta t \\ &= \frac{1}{T} \int_{t-T}^t x(\tau) d\tau \end{aligned} \quad (2.3)$$

In der oberen Grenze der Summe konnte die -1 vernachlässigt werden, da $\frac{T}{\Delta t} \xrightarrow{\Delta t \rightarrow 0} \infty$. Auch hier ist aus dem oben diskutierten Grund der konstante Faktor $\frac{1}{T}$ uninteressant. Tatsächlich wird von der Integratorschaltung ein konstanter Faktor c dazu gefügt, der ebenso ignoriert werden kann. Von dem Integrator wird also ein Signal $x(t)$ nach der folgenden Formel in ein Signal $I_T[x](t)$ transformiert:

$$I_T[x](t) = c \int_{t-T}^t x(\tau) d\tau. \quad (2.4)$$

T wird hier als Integrationszeit beschrieben. Damit Gleichung 2.4 stimmt, muss zum Zeitpunkt $t - T$ der Kondensator komplett entladen sein. Wie oben beschrieben,

enthält dieser Ausdruck dieselbe Information wie der Mittelwert des Signals (tatsächlich mehr Informationen als ein mit endlich großem Δt berechneter Mittelwert).

2.3 Terahertzstrahlung

Optische Strahlung liegt in einem Bereich von ca. 100 nm bis zu ca. 1 mm und umfasst Infrarotstrahlung, sichtbares Licht und UV-Strahlung [3]. Mit optischen Verfahren, wie sie zum Beispiel in Handykameras eingesetzt werden, ist es möglich, auf verschiedene Entfernungen Oberflächen hochauflösend zu erfassen. Sie finden Anwendung in vielen Bereichen, zivil, militärisch und industriell. Mit optischen Verfahren ist es nicht möglich, in Körper hinein- oder durch Körper hindurchzuschauen; nur die Oberfläche ist erkennbar.

Radar- oder Mikrowellenstrahlung durchdringt Körper. Mit Mikrowellenstrahlung ist es möglich, über große Distanzen und durch Wolken und Nebel Fremdkörper zu orten, wie es unter u.a. im Flug- oder Schiffsverkehr als Radar oder Sonar Anwendung findet.

Detektion von Fremdkörpern kann auch mit Röntgenstrahlung erfolgen. Dies erhöht für Firmen häufig den Aufwand erheblich. Es müssen hohe Standards eingehalten werden, um Strahlenschutzbedingungen zu erfüllen [4].

Terahertzstrahlung liegt in dem Zwischenbereich zwischen optischer und Mikrowellenstrahlung und somit auf dem Übergang zwischen Elektronik und Optik [5]. Die Strahlung ist dazu in der Lage, viele dielektrische Stoffe, wie z.B. trockenes Holz, zu durchdringen. Sie wird durch polare Stoffe wie Wasser absorbiert und durch Metall reflektiert. Auch Sprengstoffe sind häufig organische Verbindungen und aufgrund ihres spezifischen Terahertzspektrums erkennbar. Diese Eigenschaft der Terahertzstrahlung, Stoffe zerstörungsfrei durch einen Umschlag oder Textilien hindurch analysieren zu können, wird im Rahmen von z.B. Körper- oder Briefscannern zur Untersuchung auf Sprengstoffe oder unerlaubte Objekte bereits genutzt [6]. Auch für die Lebensmittelindustrie ist die Strahlung sehr interessant. Während Metallteile von empfindlichen Metalldetektoren erkannt werden, werden zum Beispiel

Glassplitter in Schokolade oder bislang mit Röntgenstrahlung erfasst. Das hat hohe Kosten im Unterhalt und in der Wartung zur Folge, da Röntgenstrahlung krebserregend ist und deswegen nur unter Auflagen genutzt werden darf.

Terahertzstrahlung hat eine Photonenenergie im Bereich von wenigen Millielektro-nenvolt. Sie ist nichtionisierend und ungefährlicher als Röntgenstrahlung.

Terahertzstrahlung liegt in dem Zwischenbereich des optischen und des elektroni-schen Strahlungsbereiches. Die Grafik zeigt eine Einordnung der Terahertzstrah-lung zwischen diese beiden Bereiche. Früher war aufgrund der Nähe zur Infrarot-strahlung der Begriff „fernes Infrarot“ geläufig. [5]

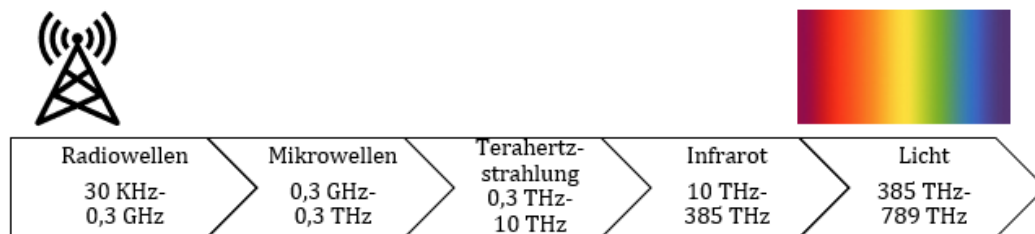


Abbildung 2.2: Einordnung des Terahertz-Spektrums

Laut verschiedenen Quellen liegt Terahertzstrahlung in einem Bereich von ca. 0,3 THz bis zu 6, 10 oder 30 THz. Recht einig sind sich die Quellen beim Anfang der Strahlung, dieser befindet sich im Submillimeterbereich. Damit beginnt sie bei 1mm, was ungefähr 0,3 THz entspricht. 3 THz entspricht im Vakuum einer Wellenlänge von ungefähr 100 μm . [7]

Es wird in diesem Bereich von der Terahertz-Lücke gesprochen, da diese Wellenlängen lange nicht nutzbar waren. Optische Sensoren konnten die Wellenlängen noch nicht erfassen, während für Hochfrequenzelektronik die Frequenzen zu hoch waren. Diese Lücke wird momentan aus beiden Richtungen erschlossen, erste günstige Systeme zur Erfassung und Erzeugung von Terahertzstrahlung sind verfügbar.

Für kostengünstige Systeme waren lange Zeit gerade die elektronischen Quellen interessant. Diese arbeiten häufig im unteren Terahertz-Bereich [7]. Sie sind günstig zu produzieren und schaffen durch intensive Entwicklung immer höhere

Frequenzen. Doch die Produktion von Lasern wird gleichzeitig immer günstiger, so-
dass nicht klar ist, welches Verfahren sich letztendlich für welche Anwendung
durchsetzt.

3 Organisation der Projektarbeit

3.1 Ziele und Projektmanagement

Aus der Aufgabenstellung gehen folgende Ziele hervor:

- Erstellen eines Zeit- und Arbeitsplans
- Entwurf und Konstruktion eines mobilen Gerüsts (bevorzugt aus Aluminiumprofilen) mit Messkammer, welches auch als Plattform für einen Aerosolgenerator, Nephelometer, Entlüftung, Rotationstisch und ein THz-System dienen soll.
- Entwurf und Realisierung eines einfachen Nephelometers zur Messung der Aerosolkonzentration in der Messkammer.
- Test des Nephelometers im Rauchkanal des Heinz-Luck Brandentdeckungslabors.
- Zusammenbau und Test des Demonstrators bestehend u.a. aus dem mobilen Gerüst mit Messkammer, Aerosolgenerator, Nephelometer, Entlüftung, Rotationstisch und THz-System.
- Dokumentation der Arbeit und Vorstellung des Demonstrators im Rahmen eines Vortrags.

Dazu wurden eigene Ziele definiert, die erreicht werden sollten:

- Einfache Bedienbarkeit des Gestells
- Kostengünstige Lösungsansätze
- Stabilität und Dichtheit sollen gegeben sein

Gleichzeitig soll der angefertigte Demonstrator qualitativ den Ansprüchen einer Bachelor-Projektarbeit genügen.

Für die Visualisierung der Ziele gibt es in dieser Arbeit zwei Werkzeuge, die zu großen Teilen Anwendung finden:

- Gantt-Diagramm oder Balkenplan zur groben, übersichtlichen Zeitplanung der zu erledigenden Aufgaben mit geplanter und tatsächlich genutzter Zeit. Dabei ist auch eine Terminliste mit stattgefundenen Besprechungs- und Messungsterminen.
- Liste mit Unterteilungen der groben Aufgaben in kleinere Aufgaben zur Kontrolle und zum Abhaken.

Die Aufgabenliste wird während des Projektes genutzt, um den Überblick zu behalten. Die Aufgabenliste und Teile des Zeitplanes werden während des Projektes ständig erweitert und angepasst. Die Zeitrechnung erfolgte im Vorhinein nach einem recht einfachen Schema:

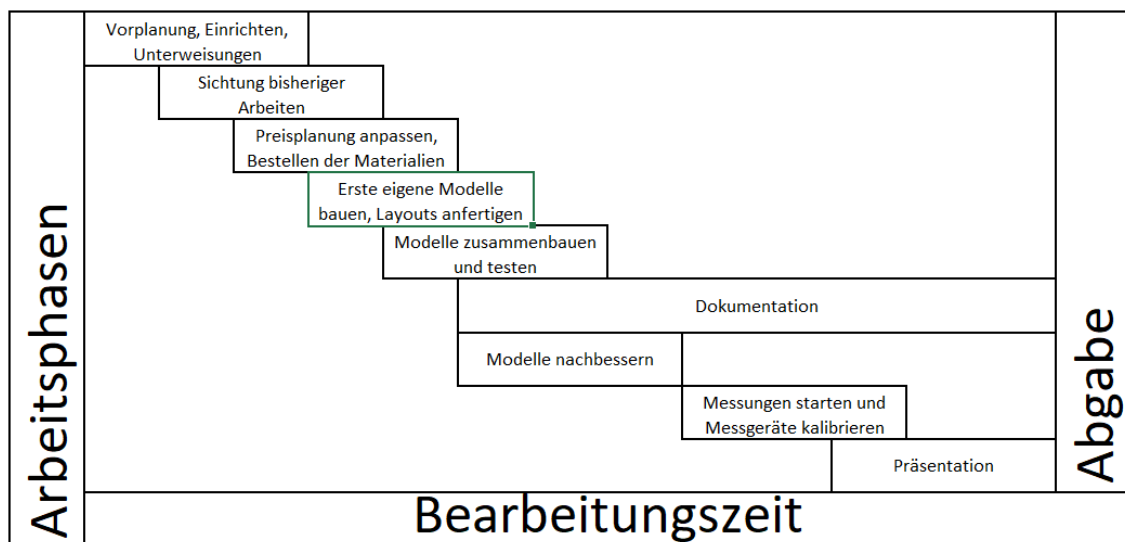


Abbildung 3.1: Grobe Vorplanung

In Abb. 3.1 ist ein Ablaufplan zu sehen, der das Projekt grob in die Arbeitsphasen einteilt, wie sie im Projekt vorgesehen sind.

Pro Woche ist ein Arbeitstag vorgesehen, der nur für die Bachelor-Projektarbeit genutzt wird. Es wird der Aufwand verschiedener Aufgaben geschätzt. Für die Projekt- und Aufgabenplanung sind auch Lieferzeiten und Verzögerungen in Bestellungen zu beachten. Die Dokumentation findet vollständig außerhalb dieser Zeiten statt. Der Gantt-Plan und die Aufgabenliste mit den geplanten Einteilungen und zeitlichen Zielsetzungen befinden sich im Anhang B.

3.2 Qualitätssicherung

Die Qualitätsansprüche für das Projekt sind so anzusetzen, dass das Endprodukt auf Messen und Kongressen präsentiert werden kann. Wichtige zu erreichende Punkte der Qualitätskontrolle sind:

- Das Gestell ist klein und stabil genug, um Transporte und die Benutzung unbeschadet zu überstehen.
- Das Gestell ist weitestgehend hermetisch geschlossen; Aerosol entweicht nicht oder nur wenig.
- Das Gestell ist sauber verarbeitet.
- Aerosol kann ohne großen Aufwand in das Gestell eingeleitet werden.
- Das Nephelometer liefert Messdaten mit ausreichender Genauigkeit; Messungen sind reproduzierbar und werden von Außenbedingungen nicht beeinflusst.
- Die Filteranlage reinigt in kurzer Zeit die Luft, sodass es geöffnet werden kann.

Der erste Punkt ist wichtig, damit das Gestell dauerhaft als Demonstrator genutzt werden kann und sich nicht nur auf ein Projekt beschränkt.

Zusätzlich soll der Demonstrator nicht während des Betriebes oder bei Öffnung die Rauchmeldeanlage eines Gebäudes auslösen. Deswegen sind Dichtigkeit und die Fähigkeit, das Aerosol schnell aus der Luft zu filtern, ein wichtiger Punkt. Die notwendigen Bedingungen der Qualitätskontrolle werden in den einzelnen Unterkapiteln behandelt, in diesen wird dann auch weiter darauf eingegangen, falls nötig.

4 Durchführung

Zielsetzung und Projektplanung wurden bereits in dem zugehörigen Unterkapitel beschrieben. In diesem Unterkapitel soll darauf eingegangen werden, welche Vorgehensweisen während des Projektes genutzt werden.

Die Inhalte des Projektes können grundsätzlich in zwei Arten von Aufgaben eingeteilt werden. Für die einen Aufgaben des Projektes bietet es sich an, nach den bekannten Strukturen zu gehen und die gelernte Methodik anzuwenden. Für den zweiten Bereich ist es erforderlich, eine oder vielleicht mehrere Versuchsreihen durchzuführen, um die Anforderungen zu erfüllen.

Viele Teile der Aufgabenstellung fallen hierbei in den zweiten Bereich. So sind unter anderem die Kenntnisse, die zur technischen Konstruktion des Gestells erforderlich sind, kein Studieninhalt. Es fallen gerade die Wandkonstruktion des Nephelometers oder dessen Programmierung in den Bereich des Erkenntnisgewinns. Um diese anzufertigen, müssen verschiedene Modelle konzipiert und geprüft werden. Es erfolgt eine Modellkonstruktion, meist in Proben, die getestet wird. Das Modell wird dann entweder fallen gelassen oder in das Bauteil integriert.

4.1 Durchführung Gestell

Das Gestell hat recht hohe Anforderungen zu erfüllen. Es soll transportierbar sein, was die Dimensionen des Gestells auf Kofferraumgröße beschränkt. Gleichzeitig muss es stabil sein und möglichst viel Testraum für die Antenne bieten. Das Gestell muss ohne großen Aufwand zu öffnen sein und Umbauten erlauben. Die Anforderungen an die Wände, Boden und Decken sind vielfältig: der Deckel und drei der Wände sollen durchsichtig sein. Der Deckel kann geöffnet werden. Eine Wand soll undurchsichtig sein, allerdings soll diese mit der Antenne durchleuchtbar sein. Der Boden benötigt eine Führungsschiene für den Motor und soll stabil genug sein, um die restlichen Aufbauten wie Nebelmaschine, Filtersystem und Sensorik tragen zu können. Das gesamte Gestell wird mit Nebel gefüllt, daher muss es möglichst luftdicht sein. Ausführungen für Leitungen sind vorgesehen, die ebenfalls luftdicht sein

4.1.1 Verwendete Materialien

Technical drawing of a square plate with a central gear-like hole and four circular holes. The overall dimensions are 40x40. The central hole has a diameter of $\varnothing 13.7$ (M16). The four corner holes have a diameter of $\varnothing 5$ (M6). The dimensions are as follows:

- Overall width: 40
- Overall height: 40
- Distance from top edge to top of corner hole: 4.5
- Distance from left edge to left of corner hole: 4.5
- Distance from center of corner hole to center of central hole: 22
- Distance from center of central hole to bottom of corner hole: 8
- Distance from center of central hole to right of corner hole: 4
- Distance from bottom edge to bottom of corner hole: 10

Drei der vier Wände und die Decke bestehen aus Acrylglas mit 5 mm Dicke. Dieses Acrylglas ist auf die passenden Maße zugeschnitten. Acrylglas ist gemeinsam mit Polycarbonat eine stabile Alternative zu Glas, die durchsichtig ist. Gleichzeitig ist das spezifische Gewicht von Acrylglas deutlich geringer als das von Glas ($1,18 \text{ g/cm}^3$ und $2,5 \text{ g/cm}^3$) [8]. Während Polycarbonat schlagfester ist als Acrylglas, ist Acrylglas kratzfester als Polycarbonat und vergilbt nicht bei UV-Bestrahlung. [9]

Der Boden des Gestells muss stabil genug sein, um alle darauf platzierten Objekte zu halten. Gleichzeitig muss Stoßfestigkeit gegeben sein. Es wurde geschäumtes PVC vorgesehen, doch aufgrund von Stabilitätsbedenken wurde stattdessen eine

Holzplatte eingesetzt. Diese ist eine mit Fichtenholz beschichtete Furnierholzplatte mit 6 mm Dicke.

Im Gestell werden zwei Arten von Dichtungen verwendet. Der Übergang des Deckels zum Gestell wird mit einem Halbrund-Dichtprofil abgedichtet. Dieses ist nachgebend. Es drückt sich in die gegenüberliegende Nut und dichtet somit die Fläche ab. Die Flächenelemente sind in ein U-Dichtprofil eingelassen. Dieses hält die Flächenelemente in den Nuten, ohne Druck auf die Scheiben auszuüben, und dichtet gleichzeitig ab. Für weitere Leckagen im Gestell ist Silikon eingesetzt.

4.1.2 Aufbau des Gestells

In die 1100 mm und die 700 mm Grundprofile sind jeweils Bohrungen für Standardverbinder eingelassen. Dies erlaubt den Zusammenbau eines Quaders. Zur Stabilisierung und Montage der Schiene für den Motor ist im Boden ein weiteres Grundprofil eingelassen. Der Deckel ist ein Rechteck, das mit Schwerlastscharnieren auf dem Quader angebracht wird.

Transportfähigkeit bedeutet, dass das Gestell formstabil sein muss, um Stöße, Schläge oder Verdrehungen des Deckels bei Öffnung gewährleisten zu können. Dazu werden neben Standardverbindern ebenfalls Verdrehsicherungen eingesetzt. Die Schrauben werden mit dem empfohlenen Drehmoment vom 30-35 Nm angezogen.

Die Scheiben werden mit U-Dichtprofil in die Nut eingesetzt. Dieses hat auf beiden Seiten eine Breite von ca. 1 mm und beschränkt somit die Scheibenbreite auf max. 6 mm. Kontakt zum Boden wird durch ein geripptes Gummiprofil hergestellt, das als Fuß und als Kratzschutz für die Aluminium-Grundträger dient.

Leitungen, die in das Gestell hineingelegt werden, sind mit einer Verschraubung gesichert und laufen durch diese gebündelt hinaus.

Die Stabilität wird mit einer Berechnung der Statik überprüft. Zuerst werden Schwachstellen gesucht, dann wird deren Belastbarkeit berechnet. Zuletzt wird die

Belastbarkeit der Platten überprüft und Benutzungshinweise werden gegeben. Eine kurze Zusammenfassung der Berechnung zeigt, dass die größte Gefahr für das Metallgestell nicht die direkte Krafteinwirkung ist, sondern das Drehmoment, welches entsteht, wenn der Deckel geöffnet wird. Hierzu eine kurze Beispielrechnung zur direkten Krafteinwirkung, Formeln kommen aus dem Produktkatalog des Herstellers:

Randbedingungen sind, dass jede Stange des Gestells einem zweiseitig eingespannten Träger entspricht. Dies ist durch die Struktur des Quaders realistisch. Die stärkste Belastung, die ein Träger somit bei direkter Krafteinwirkung erfahren kann, ist in der Mitte zwischen den beiden eingespannten Trägern. Dabei sind aufgrund der Zugfestigkeit der Standardverbinder von 10 kN und der Schubfestigkeit der Standardverbinder von ca. 4 kN bei einem Anzugmoment von 30 Nm die Standardverbinder bei Zug- und Druckspannungen sehr hoch belastbar und somit nicht als Schwachstelle zu erkennen.

Die maximale Dehngrenze der Träger liegt bei 200 N/mm², aus dem Datenblatt des Trägers lässt sich eine Ersatzprofilfläche von 7,29 cm² ableiten. Das ergibt eine Belastung von 14,58 kN, bis sich eine dauerhafte Verformung einstellt.

Eine andere mögliche Schwachstelle des Gestells ist die Verformung der Flächen durch die Torsion, die entsteht, wenn der Deckel geöffnet wird.

Die verwendeten Verdrehsicherungen verhindern ein Verdrehen der Träger beim Öffnen des Deckels. Die Nutensteine, die zur Befestigung der Scharniere benutzt wurden, können bei hoher Belastung das Gestell verbiegen. [10].

$$M = \frac{m \cdot g \cdot l}{n} \quad (4.1)$$

Mit einem Gewicht des Deckels m von insgesamt 12,93 kg, einem Hebelarm von 0,4 m aufgrund von Schwerpunktverschiebung, einer Erdbeschleunigung von $9,8 \frac{\text{m}}{\text{s}^2}$

und 4 Nutensteinen, mit denen die Scharniere befestigt werden, ist die Belastung pro Nutenstein, wenn die Platte geöffnet steht, bei ca. 12 Nm. Somit ist angeraten, den Deckel der Box nicht aufzuschlagen. Es kann dabei ein Drehmoment >12 Nm entstehen. Dies kann das Gestell verbiegen. Von stärkeren Schlägen auf Flächenelemente ist ebenfalls abzuraten.

Das Gestell erfüllt somit grundlegend die Anforderungen an Stabilität und Transportfähigkeit.

4.1.3 Nebelmaschine

Der POWER-TINY ist eine kleine netzunabhängige Nebelmaschine mit Fernsteuerung. Die Spannungsversorgung wird herausgeführt, um einen Dauerbetrieb bei Präsentationen gewährleisten zu können. Der POWER-TINY verfügt über eine Verdampfungseinheit, die das Nebelfluid verdampft. Das erzeugt einen sehr hellen weißen Dampf, dieser ist hoch konzentriert und füllt innerhalb von Sekunden das gesamte Gestell aus. Der POWER-TINY hat nominelle Heizleistung von 400 W bei einer Aufheizzeit von 1 s. Es wird 10 ml Nebelfluid pro Minute verbraucht. Durch sein Eigengewicht von 5,6 kg steht die Nebelmaschine sicher in dem Gestell und bedarf keiner weiteren Sicherung.

4.1.4 Rotationseinheit

Bestandteil der Einheit ist ein Motor. Daran befestigt ist ein Hebelarm, an dessen Ende ein innerhalb der Lastgrenzen des Motors frei wählbarer Körper montiert wird. Der Motor ist auf einer Montageschiene angebracht, die ein Verschieben der Einheit zulässt.

Der Motor ist ein zweiphasiger Schrittmotor von ThorLabs. Dieser hat einen Bewegungswinkel von 360° und rotiert mit einer maximalen Frequenz von 10° pro Sekunde. Die Stromversorgung erfolgt gemeinsam mit der Ansteuerung per Mikro-USB 3.0. Zusätzlich ist der Motor über einen I/O-Port oder über

Vorwärts/Rückwärts-Tasten direkt am Gerät ansteuerbar. Maximal ist der Motor mit 22 N belastbar bei einem Drehmoment von 140 mNm.

4.1.5 Filtersystem

Der Luftfilter soll schnellstmöglich das Gestell von Aerosol befreien können. Dies ist erforderlich, um beim Öffnen des Gestells nicht versehentlich die Feuermeldeanlage im Präsentationsgebäude auszulösen. Die Tabelle erfasst eine Testreihe mit den verschiedenen Filtersystemen, die für eine Benutzung in Betracht kommen.

Eine langsame Methode der Luftreinigung wie mit dem Raumlufthereiniger oder dem HEPA-Filtermodul scheidet somit aus der Auswahl aus. Der Siemens-Staubsauger erzielt, wenn er außerhalb des Gestells steht, nur unzureichende Ergebnisse. Die Wirkung der Filtersysteme ist nicht gut genug, um in einem Durchgang die Luft vollständig zu reinigen. Das Filtersystem befindet sich im Gestell, da für die Luft mehrere Filtervorgänge erforderlich sind. Beste Ergebnisse in der Testreihe erzielte dabei ein beutelloser Zyklonstaubsauger mit Mehrfachfiltersystem. Der Staubsauger verfügt über ein Filtersystem aus Motorvorfilter und einem HEPA-12-Nachfilter. Mit der verwendeten Technologie schlägt sich das Aerosol in den Filtern nieder. Die Filter lassen sich ohne großen Aufwand herausnehmen und reinigen und sind auch mehrfach benutzbar.

Name	Filter	Methode	Dauer
JINPUS Luftreiniger 2,5 W	HEPA und Aktivkohle	Innen	>15 Minuten
Siemens Staubsauger VS06B112A 700 W	Staubbeutel, Hygienefilter, Motorvorfilter Papier	Innen	>3 Minuten
Siemens Staubsauger VS06B112A 700 W	Staubbeutel, Hygienefilter, Motorvorfilter Papier	Außen	>3 Minuten
HEPA-Filtermodul	Filtervlies	Innen	>10 Minuten
Beutelloser Staubsauger 700 W	Motorvorfilter Papier, Motorvorfilter Filtervlies, Zyklonzylinder, HEPA12-Filter	Innen	>3 Minuten

Tabelle 4.1: Vergleich verschiedener Filtersysteme

Der Kostenaufwand für spezialisierte, leistungsfähigere Luftfiltersysteme ist sehr viel höher. Der Zyklonstaubsauger ist somit für unser System die erste Wahl.

4.2 Durchführung Nephelometer-Gehäuse

Als ein funktionaler Teil des Nephelometers gelten besondere Anforderungen an das Gehäuse. So muss es der Elektronik nicht nur einen festen Sitz liefern, sondern die Sensorik ebenfalls von äußeren Einflüssen schützen. Auch die numerische Rechen- bzw. Kontrolleinheit muss einen ausgewiesenen Platz im Gehäuse finden. Zudem müssen am Gehäuse Montagevorrichtungen vorhanden sein, die es erlauben das Nephelometer an einem beliebigen Ort in der Rauchkammer zu positionieren. Die wesentliche Aufgabe des Gehäuses aber, liegt darin, den Probenraum von sichtbarem Licht⁷ und Licht im nahen Infrarot-Bereich⁸ abzuschirmen und dabei einen Luftaustausch mit der Umgebungsluft zu gewährleisten.

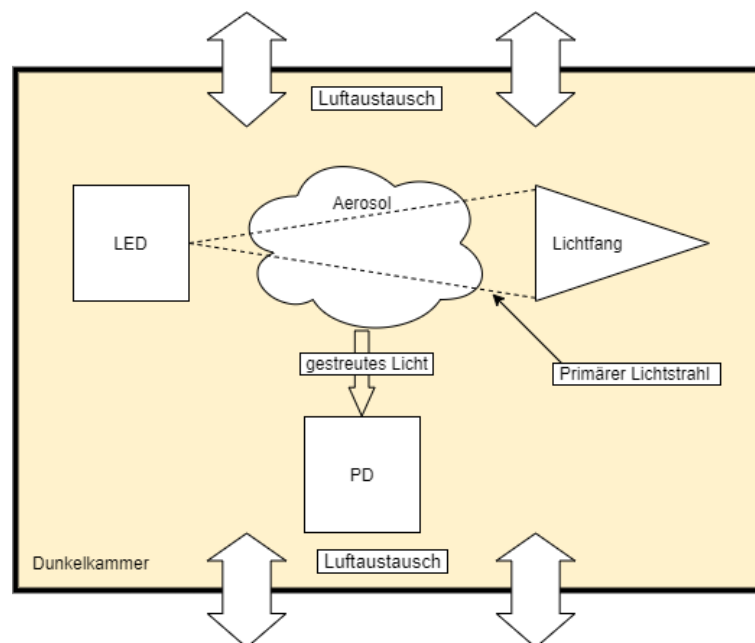


Abbildung 4.2: Prinzipieller Aufbau eines Nephelometers

Jener Luftaustausch muss dabei (im Idealfall) so effektiv ablaufen, dass die Zusammensetzung der Luft im Probenraum zu jedem Zeitpunkt der Zusammensetzung der Umgebungsluft entspricht. Aus diesen Vorgaben ergeben sich entsprechende Konstruktionsvorschriften für den Entwurf der Außenwände des Gehäuses. Die Wände müssen also eine geometrische Grundstruktur aufweisen, die kein Licht passieren lässt, aber für Fluide wie Luft leicht zu durchdringen ist. Der in Abbildung 4.2

⁷ Sichtbar sind elektromagnetische Wellen in einem Bereich von etwa 380 nm – 780 nm

⁸ Die Wellenlänge von nahem Infrarotlicht liegt in einem Bereich von 780 nm – 3 µm

dargestellte Lichtfang dient dazu eine direkte Reflexion des transmittierten primären Lichtstrahls zu verhindern.

4.2.1 Fertigung mit dem 3D-Drucker

Zu Beginn des Projekts wurde festgelegt, dass das Gehäuse des Nephelometers in einem additiven Verfahren mittels 3D-Filamentdrucker⁹ hergestellt werden soll. Aus fertigungstechnischen und wirtschaftlichen Gründen wurde PLA¹⁰ als Druckmaterial gewählt. Während des Druckvorgangs wird das PLA-Filament erhitzt und durch eine 0,4 mm breite Düse extrudiert. Dabei wird das Werkstück in Schichten von 0,1 mm schrittweise gefertigt. Entsprechende Einschränkungen sind bei der Konstruktion zu berücksichtigen. So dürfen beispielsweise die Abmessungen des Werkstückes nicht die Abmessungen des Druckvolumens¹¹ übersteigen. Des Weiteren sollte eine Wandstärke von mindestens 1 mm eingehalten werden, sehr spitze Winkel sowie Überhänge, sollten, wenn möglich, vermieden werden und rechte Winkel sollten durch Abrundungen mit einem Radius von mindestens 1 mm ersetzt werden. Dies sind empirisch ermittelte Richtwerte, welche stark von der eingesetzten Drucktechnik und dem verwendeten Drucker abhängen. Um möglichst wenig mechanische Nachbearbeitung am Werkstück vornehmen zu müssen, wird (soweit möglich) auf Support verzichtet. Im Bereich des 3D-Filamentdrucks ist der sogenannte Support eine Stützstruktur, die überall dort zum Einsatz kommt, wo konstruktionsbedingt Überhänge entstehen. Entsprechende Regeln zur Erstellung von Support können im Slicing-Programm¹² festgelegt werden. Durch die Verwendung eines 3D-Druckers besteht die Möglichkeit einen Entwurf quasi direkt zu fertigen. Lieferzeiten, die unter Umständen einige Wochen betragen können, entfallen. Darüber hinaus bekommt der Konstrukteur einen dreidimensionalen und haptischen Eindruck des Werkstücks, welchen beispielsweise ein CAD-Programm nicht liefern kann. Etwaige Konstruktionsfehler können somit effizienter und vor allem kostengünstiger erkannt werden. Andererseits ist insbesondere durch die Verwendung

⁹ hier: ein Filamentdrucker der Marke Ultimaker: „Ultimaker S5“

¹⁰ kurz für Polylactide, umgangssprachlich auch Polymilchsäuren

¹¹ Bei Ultimaker S5: 330 mm * 240 mm * 300 mm

¹² Ein Programm, welches aus einer 3D-Datei die Steuerungsbefehle für den Drucker ableitet (hier: STL-Datei zu G-Code)

eines Filamentdruckers die Auswahl an Materialien beschränkt. Zudem treten, je nach Betriebskonfiguration, teilweise große Maßabweichungen auf.

4.2.2 Entwürfe im Zuge der Gehäuseentwicklung

Im Laufe des Projekts wurden einige Entwürfe eines Gehäuses erstellt, welche aus unterschiedlichen Gründen nicht für eine Endfertigung geeignet waren. Im Folgenden soll der Ablauf der Gehäuseentwicklung dargestellt werden.

Der erste Entwurf der Dunkelkammer (Abb. 4.3) basiert auf einer kreisförmigen Grundfläche mit tangential angeordneten, keilförmigen Strukturen, die auftreffendes Licht so reflektieren sollen, dass es nicht in das Innere der Dunkelkammer gelangt. Durch die Messapparatur eines Nephelometers ist eine rechtwinklige Anordnung der einzelnen Komponenten festgelegt.

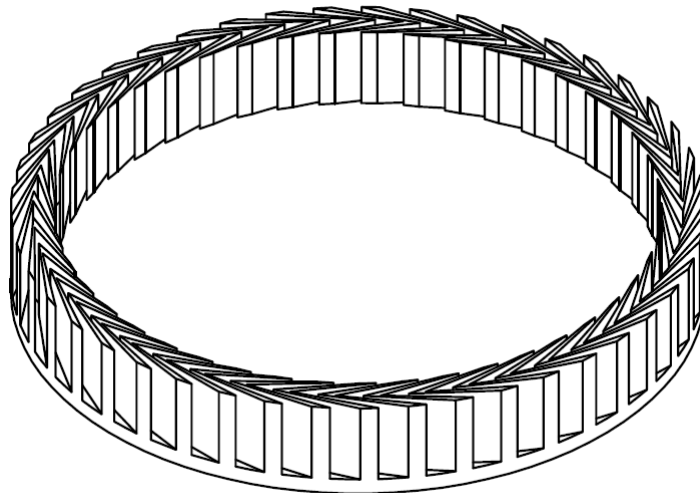


Abbildung 4.3: Dunkelkammer, erster Entwurf

Eine Positionierung einer solchen Anordnung (Abb. 4.2) innerhalb einer kreisförmigen Fläche ist allerdings nicht Platzeffizient. Geeigneter wäre eine Grundfläche mit ebenfalls rechtwinkligen Strukturen. Ausgehend von dieser Prämisse wurde ein zweiter Entwurf angefertigt, der dieses grundlegende geometrische Defizit nicht aufweist. Es wurde eine rechteckige Grundfläche gewählt und das Design der Wandstruktur des ersten Entwurfs übernommen. Darüber hinaus wurden

Montagevorrichtungen für die elektronischen Bauelemente sowie ein Lichtfang implementiert (Abb. 4.4).

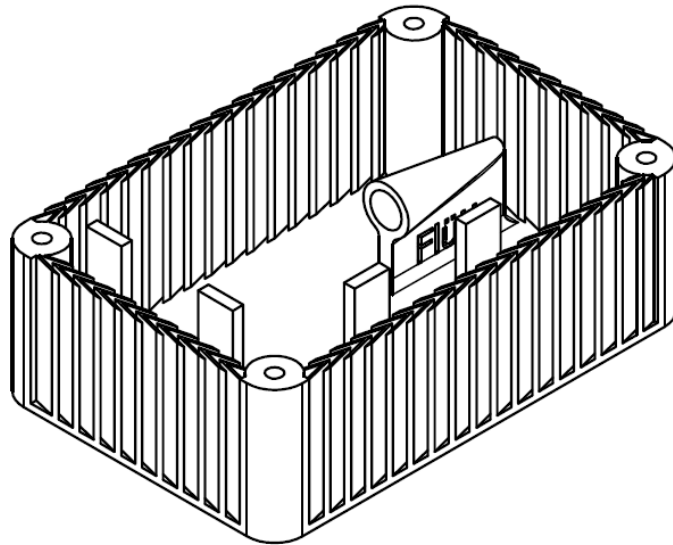


Abbildung 4.4: Dunkelkammer, zweiter Entwurf

In einer Simulation des Druckvorgangs mit Hilfe des Slicing-Programms stellte sich heraus, dass die keilförmigen Wandstrukturen (Abb. 4.4) zu filigran konstruiert wurden. Die Wandstärke dieser Strukturen unterschritt die Mindestdicke von 1 mm und somit auch das Auflösungsvermögen des Druckers. Darüber hinaus wurden die Montagevorrichtungen für die Trägerplatten der Elektronik fehlerhaft konstruiert. Zum einen wären die Halterungsstege zu dünn, als dass sie einen stabilen Sitz der Elektronik gewährleisten könnten. Zum anderen wären die Durchgangsbohrungen (in Abb. 4.4 nicht zu sehen) nicht von außen zugänglich, was eine Montage praktisch unmöglich machte.

In einem dritten Entwurf wurde das Ziel verfolgt, das Design der Wandstrukturen und des Lichtfangs zu verbessern. So wurden Öffnungsdurchmesser und Länge des Lichtfangs erhöht und die keilförmige Wandstruktur (Abb. 4.4 und 4.3) durch eine labyrinthartige Struktur ersetzt. Ferner wurden an den Stirnseiten Wände aus vollem Material eingebracht (Abb. 4.5).

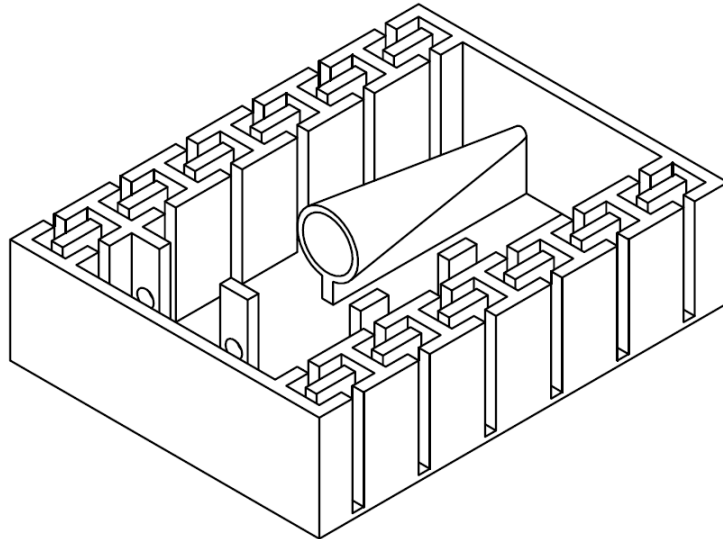


Abbildung 4.5: Dunkelkammer, dritter Entwurf

Einige der Mängel des zweiten Entwurfs wurden allerdings nicht behoben. Wie in Abbildung 4.5 zu sehen ist, wurden beispielsweise die Montagevorrichtungen für die Trägerplatten aus dem zweiten Entwurf übernommen. Des Weiteren entstand durch die Änderung des Designs der Wände eine neue Problematik: Die Stege in der Mitte der Labyrinthstruktur (jeweils zwischen zwei Doppel-T-Elementen) sind der Konstruktion aus Abbildung 4.5 zu Folge nur auf der Grundfläche befestigt. Es besteht somit mutmaßlich eine erhöhte Defektgefahr. Gleiches gilt für den Lichtfang, der einem nur dünnen Steg aufsitzt. Es besteht ebenfalls keine zweckmäßige Möglichkeit eine Abdeckung zu montieren.

Der finale Entwurf wurde, unter Berücksichtigung aller bisher genannten Defizite, mit besonderem Hinblick auf einfache Montage, Durchführbarkeit bzgl. des Druckprozesses und modulare Erweiterbarkeit konstruiert. Aus Abbildung 4.6 wird ersichtlich, dass nun die Möglichkeit besteht mehrere Gehäuse der gleichen Grundform übereinander zu positionieren und sie über Schraubverbindungen durch die Durchgangsbohrungen zu verbinden. Die Montagevorrichtungen für die Trägerplatten wurden vergrößert und mit den Außenwänden verbunden, wodurch die Schraubverbindung zur Befestigung der Trägerplatten von außen zugänglich wird.

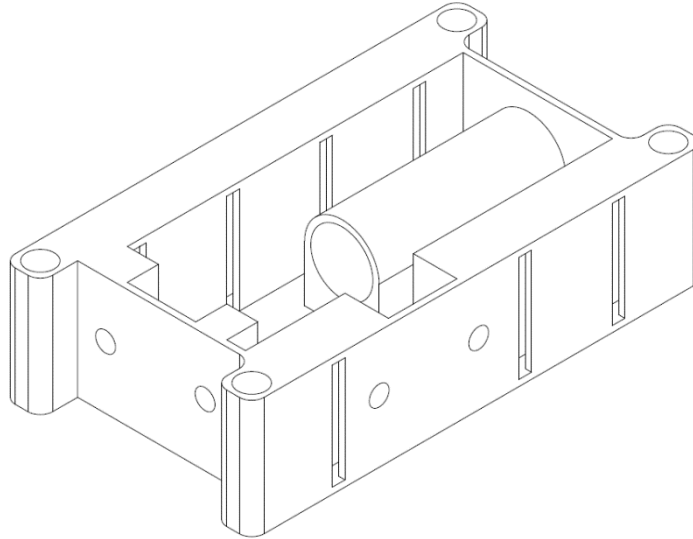


Abbildung 4.6: Dunkelkammer, vierter Entwurf/finaler Entwurf

Die Labyrinthstrukturen der Wände wurden oberhalb mit einer Deckschicht abgeschlossen, um die oben genannten Stege zu stabilisieren. Schließlich wurde aus fertigungstechnischen Gründen ein breiter Fuß unter den Lichtfang gesetzt. So wird eine problemlose Fertigung mittels 3D-Drucker ohne den Einsatz von Support möglich.

4.2.3 Lichtfang

Wie in Abbildung 4.6 zu sehen ist, nimmt der sogenannte Lichtfang einen Großteil des inneren Volumens der Dunkelkammer ein. Ein Lichtfang in dieser Größe ist notwendig, da das Fertigungsmaterial (schwarzes PLA) nicht ideal absorbierend wirkt und somit einen Teil des emittierten Lichtes reflektiert. Dies ist von besonderer Bedeutung für den durch das Aerosol transmittierten Teil des primären Lichtstrahls, welcher ohne Lichtfang von den Wänden reflektiert und somit einen Beitrag zum detektierten Signal liefern würde. Um diese Art von Messfehlern zu minimieren wird ein Lichtfang eingesetzt. Der Lichtfang soll den transmittierten, primären Lichtstrahl möglichst vollständig absorbieren. Dementsprechend ergeben sich die Dimensionen des Lichtfangs aus dem Abstrahlwinkel der verwendeten LED und der Strecke zwischen Lichtfang und LED. Für den Durchmesser d des Lichtfangs gilt somit:

$$d = 2r \geq 2 * \tan(\varphi_{LED}) * s \quad (4.2)$$

Mit Radius r , Abstrahlwinkel φ_{LED} und der Strecke s zwischen LED und Lichtfang.

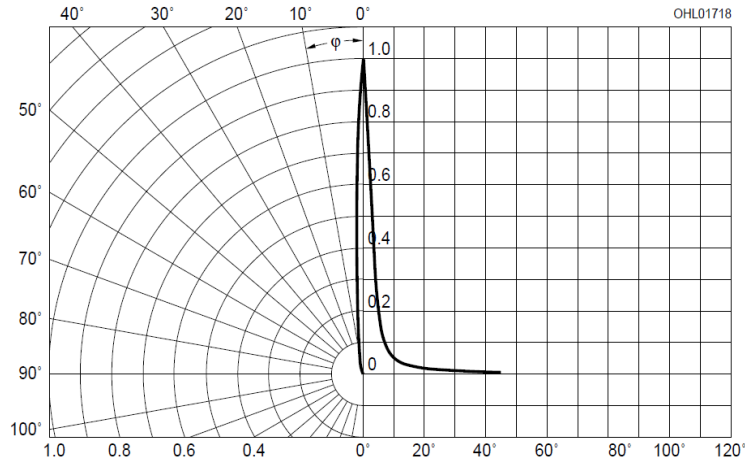


Abbildung 4.7: Abstrahlcharakteristik der verwendeten IRED¹³ [11]

Der Abstrahlwinkel einer LED ist der Winkel, bei welchem die optische Leistung auf die Hälfte des Maximalwerts gefallen ist. In diesem Fall: $\varphi_{SFH4550} = 3^\circ$ (Abb. 4.7). Mit einem Abstand von $s = 14$ mm zwischen LED und Lichtfang ergibt sich ein minimaler Durchmesser von $d_{min} = 1,47$ mm. Es wurde unter anderem aus fertigungstechnischen Gründen, aber auch um Bauteiltoleranzen auszugleichen ein Durchmesser von $d = 10$ mm $\approx 7 * d_{min}$ gewählt. Somit hat das Licht, was nicht vom Lichtfang absorbiert wird (geometrisch), eine Intensität von etwa 3 %. Der Öffnungswinkel des Kegels innerhalb des Lichtfangs wäre im Idealfall infinitesimal klein, was wiederum einen unendlich langen Lichtfang zur Folge hätte. Da dies nicht zu realisieren ist, wurde willkürlich festgelegt, dass ein Lichtstrahl, der im Abstrahlwinkel φ_{LED} auf den Lichtfang trifft, mindestens fünf Mal reflektiert werden muss, bevor der Einfallswinkel α_n auf die Innenwand des Lichtfangs 90° übersteigt und der Strahl somit die Richtung ändert. Für den Einfallswinkel α_n gilt:

$$\alpha_n = \alpha_{n-1} + 2 * \varphi_{Lichtfang} \quad (4.3)$$

unter der Bedingung, dass

¹³ engl. infrared emitting diode (dt. Infrarot emittierende Diode)

$$\alpha_5 < 90^\circ \quad (4.4)$$

und mit

$$\alpha_1 = \varphi_{\text{LED}} + \varphi_{\text{Lichtfang}} \quad (4.5)$$

ergibt sich für den Öffnungswinkel des Lichtfangs:

$$\varphi_{\text{Lichtfang}} < \frac{\alpha_5 - \varphi_{\text{LED}}}{9} = 9,67^\circ \quad (4.6)$$

Die Länge des Lichtfangs wurde demnach festgelegt auf 30 mm, wodurch sich ein Winkel von $\varphi_{\text{Lichtfang}} = 9,46^\circ$ ergibt, der die Bedingung aus Gleichung 4.5 erfüllt.

4.2.4 Testdrucke

Ein Lichtfang mit den oben genannten Maßen wurde folglich im finalen Entwurf der Dunkelkammer (Abb. 4.6) implementiert. Eine Simulation des Druckprozesses der Dunkelkammer mittels Slicing-Programm ergab eine Produktionszeit von 5 h und 39 min sowie einen Materialverbrauch von 20 g (entspricht etwa 2,59 m Filament). Aus diesem Grund wurden vorab einige Testdrucke durchgeführt, um zum einen die Funktionalität der Komponenten und zum anderen die Durchführbarkeit des Druckprozesses zu erproben. Zuerst wurde eine Probe der Wandstruktur (Abb. 4.8) gedruckt.

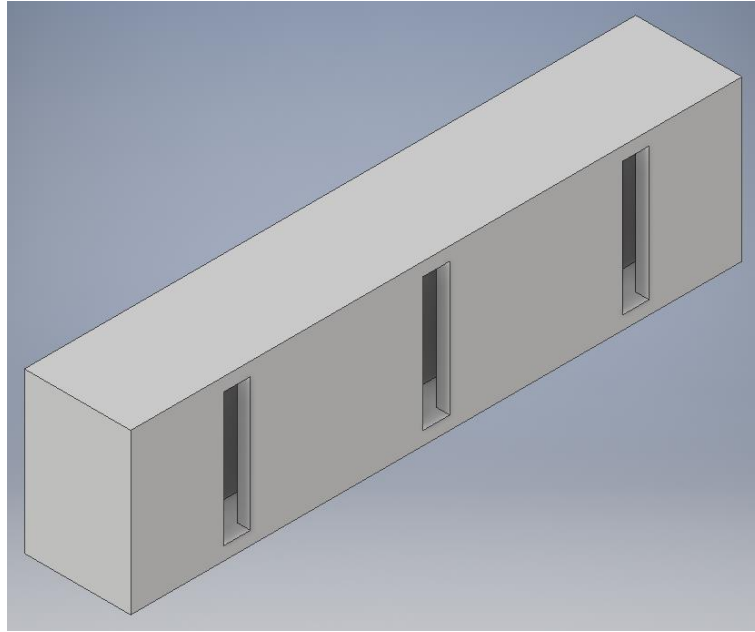


Abbildung 4.8: CAD-Modell einer Probe der Wandstruktur aus dem finalen Entwurf

Um einen Test unter realen Bedingungen zu gewährleisten, wurde die Probe originalgetreu aus der Zeichnung des finalen Entwurfs übernommen und auch in der gleichen Position auf dem Druckbett angeordnet, wie es für den finalen Entwurf vorgesehen war. Nach erfolgreichem Druck wurde die Probe einer Sichtprüfung hinsichtlich der Druckqualität und der Luftzirkulation unterzogen. Abschließend wurde die Probe in einem abgedunkelten Umfeld auf die Transmission von Licht überprüft. Dies wurde ebenfalls im Zuge einer Sichtprüfung, jedoch unter Einstrahlung von Licht aus verschiedenen Winkeln, durchgeführt.

Als zweites wurde eine Probe des Lichtfangs (Abb. 4.9, links) gedruckt, welche ebenfalls originalgetreu aus dem finalen Entwurf übernommen wurde. Die hohle, kegelförmige Struktur des Lichtfangs hat in der gewählten Position auf dem Druckbett (Kreisquerschnitt senkrecht zum Druckbett) einen steilen Überhang zur Folge, welcher durch das Drucken mit Support (Generierung einer Stützstruktur) stabilisiert wurde. In einer anschließenden Sichtprüfung stellte sich heraus, dass durch die Entfernung der Support-strukturen Unebenheiten auf der Oberfläche des Konus zurückblieben. Zur Vermeidung dieser Unebenheiten wurde das Drucken des Lichtfangs ohne die Generierung von Support diskutiert. Um die Durchführbarkeit eines Werkstücks mit Überhängen dieser Größe zu validieren wurde ein Ring (Abb. 4.9, rechts) mit 12 mm Durchmesser ohne den Einsatz von Support gedruckt.

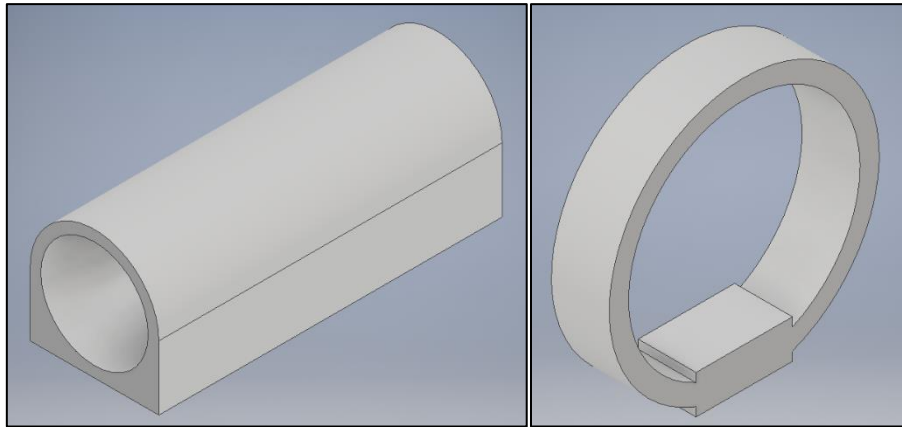


Abbildung 4.9: links: CAD-Modell einer Probe des Lichtfangs; rechts: CAD-Modell einer Probe des Rings zur Validierung der Druckbarkeit ohne Support

Aus den oben genannten Prüfungen geht hervor, dass die getestete Wandstruktur den Anforderungen genügt einen ungehinderten Luftaustausch bei gleichzeitiger Abschirmung vor sichtbarem Licht zu gewährleisten. Der Druck des Rings zeigte, dass der Lichtfang und folglich der gesamte finale Entwurf ohne Support gedruckt werden kann.

4.2.5 Deckel und Arduino-Gehäuse

Es wurde festgelegt, dass Sensorik und Kontrolleinheit räumlich getrennt installiert werden sollen. Der Mikrocontroller (hier: Arduino Nano) und periphere Elektronik sind auf einer Lochrasterplatine angebracht. Ein entsprechendes Gehäuse soll den Mikrocontroller inklusive Platine beherbergen und den gleichen Grundriss, wie das Gehäuse des Nephelometers aufweisen, damit sie übereinander angeordnet montiert werden können. Des Weiteren muss eine Montagevorrichtung vorhanden sein, die es ermöglicht das Nephelometer an beliebiger Stelle zu installieren. Aus diesen Vorgaben ergibt sich der in Abbildung 4.10 dargestellte Entwurf. Im Inneren des Gehäuses befinden sich vier Montagevorrichtungen, die entsprechend des Rastermaßes der Platine (ein Zehntel Zoll) positioniert sind. So ist es möglich die vorhandenen Löcher in der Platine zur Montage zu nutzen. An einer Stirnseite des Gehäuses ist eine Ausnehmung vorgesehen, durch welche der USB-Port des Mikrocontrollers zugänglich ist.

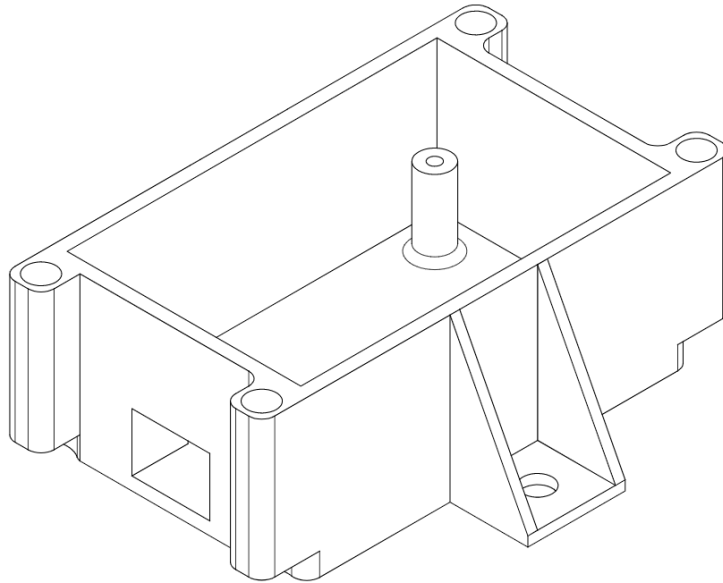


Abbildung 4.10: Entwurf des Gehäuses für den Mikrocontroller

Durch die Durchgangsbohrungen an den vier Ecken ist eine modulare Verbindung mit der Dunkelkammer möglich. Die Aussparungen an der Unterseite der Durchgangsbohrungen sind so konzipiert, dass die M4-Muttern, die zur Montage vorgesehen sind, bündig mit dem Boden des Gehäuses abschließen. An den Flanken des Gehäuses befindet sich je Seite eine Bohrung zur Befestigung des Nephelometers am Einsatzort. Die Oberseite des Mikrocontrollergehäuses wird somit von der Unterseite der Dunkelkammer verschlossen. Für den Verschluss der Dunkelkammer wurde ein Deckel entworfen (Abb. 4.11), der durch einen Steg am äußeren Rand die Seiten der Dunkelkammer überlappt. Somit ist gewährleistet, dass kein Licht durch einen etwaigen Montagespalt eindringen kann.

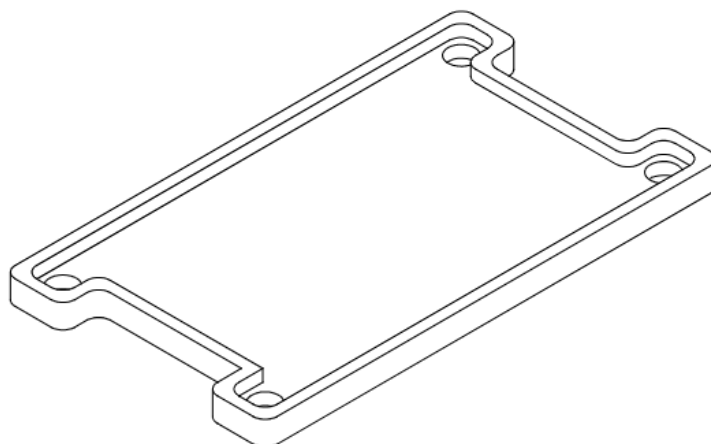


Abbildung 4.11: Entwurf des Deckels der Dunkelkammer

4.3 Elektronische Umsetzung des Nephelometers

Aus den oben beschriebenen Gründen haben wir uns in unserer Schaltung für einen analogen Integrator entschieden. Die benutzte Schaltung stammt aus [12], der Schaltplan des Nephelometers kann in Anhang C gefunden werden.

Die Stromversorgung für kommt von einem Arduino Nano (der über USB von einem Computer versorgt wird), an den auch die Anschlüsse „Output“ (Spannungswert am Integrator), „Reset“ (Transistorsteuerung zum Entladen des Integrationskondensators) und „Pulse“ (Transistorsteuerung zur Bedienung der LED) angeschlossen sind.

Im Schaltplan ist auf der linken Seite die LED-Schaltung zu sehen. Hier musste anhand der LED- und Transistorkennlinien ein geeigneter Wert für den Vorwiderstand gefunden werden. Die Überlegungen hierzu werden im nächsten Abschnitt diskutiert.

In der Mitte, im oberen Bereich des Schaltplans, ist ein Festspannungsregler zu sehen. Dieser wandelt die $0\text{ V} / +5\text{ V}$ Versorgungsspannung des Arduinos in eine $-5\text{ V} / 0\text{ V} / +5\text{ V}$ Spannung um, mit der der Operationsverstärker in der Integratorschaltung betrieben wird. Zwei 100 nF Kondensatoren werden zur Spannungsstabilisierung verwendet.

Rechts neben dieser Spannungsversorgung ist die PD, welche im Gehäuse mit einem Transimpedanzwandler verschaltet ist. Unter der PD ist der Rest der Integratorschaltung zu sehen. Der hier verwendete Kondensator legt die Zeitkonstante des Integrators fest. Da die Zeit zwischen dem periodischen Entleeren des Kondensators am Arduino eingestellt werden kann, ist der genaue Wert der Zeitkonstante nicht von großer Bedeutung. Wichtig ist jedoch, dass der Integrator beim Integrieren nicht an seine Grenze kommt, dann wäre der Kondensator bei jedem Abfragen auf 5 V aufgeladen und das Signal enthält keine Information über die Aerosoldichte mehr.

Dass dies nicht passiert, kann durch zwei verschiedene Maßnahmen sichergestellt werden: eine größere Zeitkonstante oder eine kleine Integrationszeit. Da die

Integrationszeit nicht zu klein sein darf, sonst würde der Integrator seinen Zweck der Glättung der Messungen nicht erreichen, ist es sinnvoll, einen großen Kondensator zu verwenden, um eine ausreichende Zeitkonstante zu erhalten. In unserem Fall haben wir uns für einen 100 nF Kondensator entschieden.

Der Widerstand, der sich auf dem Schaltplan unter dem Kondensator befindet und mit dem Transistor verschaltet ist, dient der Strombegrenzung durch den Transistor. Auch hier ist der genaue Wert nicht ausschlaggebend, es muss jedoch darauf geachtet werden, dass er die Funktion der Strombegrenzung erfüllt, aber nicht zu groß ist. Bei zu einem hohen Widerstand ist der Entladevorgang des Kondensators sehr langsam, was die Gesamtgeschwindigkeit des Systems beeinträchtigt, da der Kondensator bei jeder Messung vor der Integration entladen werden muss. Ein 100 Ω Widerstand erfüllt beide Bedingungen.

Es müssen also noch zwei verschiedene Werte gefunden werden: der Widerstand zur Strombegrenzung in der LED-Schaltung, sowie eine geeignete Integrationszeit. Diesen beiden Fragen widmen sich die nächsten zwei Unterkapitel.

4.3.1 Arbeitspunktanalyse in der LED-Schaltung

Der hier betrachtete Ausschnitt der Schaltung ist in Abbildung 4.12 dargestellt.

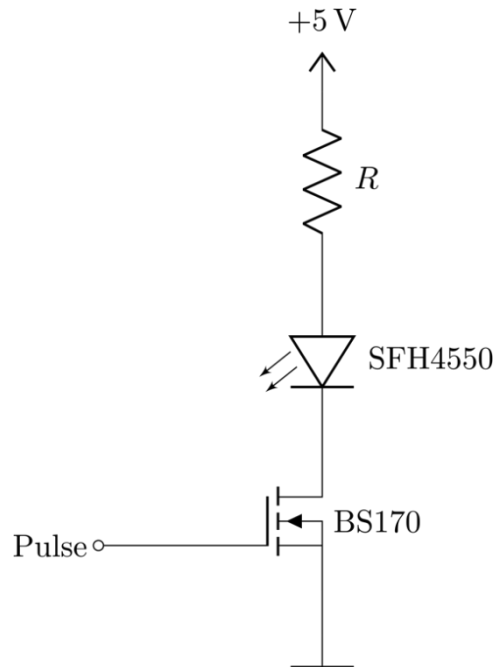


Abbildung 4.12: Die LED-Schaltung. Es muss ein geeigneter Wert für R gefunden werden.

Es soll der Widerstand R so bestimmt werden, dass der in dem Zweig fließende Strom knapp unter dem Maximalstrom der LED liegt, damit diese mit möglichst hoher Helligkeit leuchtet, aber nicht beschädigt wird. Der Widerstand lässt sich mit der folgenden Formel berechnet:

$$R = \frac{U_0 - U_{D,AP} - U_{DS,AP}}{I_{D,max}} \quad (4.7)$$

Hier ist $U_0 = 5\text{ V}$ die Versorgungsspannung und $I_{D,max} = 100\text{ mA}$ ist der zulässige Maximalstrom durch die LED, der aus [11] bestimmt wurde. Bei $U_{D,AP}$ und $U_{DS,AP}$ handelt es sich um die Spannungen, die sich im Arbeitspunkt an der LED bzw. am Transistor (zwischen Drain und Source), einstellen. Bei dem hier betrachteten Arbeitspunkt ist die LED an, es gilt also $\text{Pulse} = U_{GS} = 5\text{ V}$. Da sich die Spannungen $U_{D,AP}$ und $U_{DS,AP}$ aus den in den Datenblättern [11] und [13] zur Verfügung gestellten Kennlinien nicht genau abgelesen lassen, haben wir uns für eine SPICE-Simulation der Bauteile entschieden, um die Arbeitspunkte genau zu bestimmen.

Die SPICE-Modelle der LED und des Transistors sind [14] und [15] entnommen und der entsprechende Code ist im Anhang D.1 und D.2 hinterlegt. Wir haben die grafische SPICE-Umgebung LTspice verwendet. Die Schaltungen zur Bestimmung der Spannungen $U_{D,AP}$ und $U_{DS,AP}$ sind in Abbildung 4.13 (links und mittig) dargestellt. Die im oberen Bereich der Abbildung stehenden SPICE-Befehle gehören jeweils zur darunterliegenden Schaltung. Alle drei Befehle wurden unabhängig voneinander ausgeführt und beschreiben unterschiedliche Simulationen.

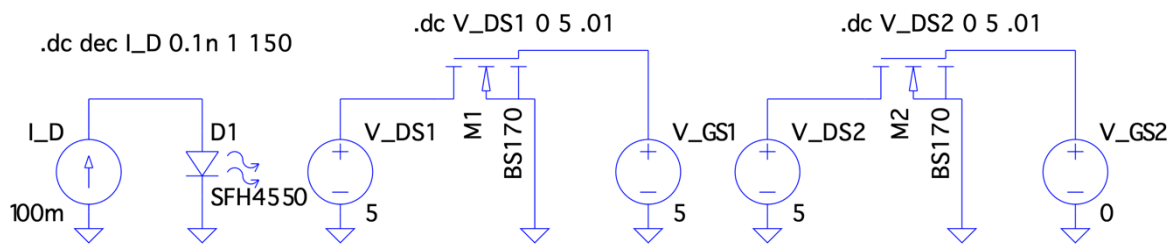


Abbildung 4.13: SPICE-Simulationsschaltungen zur Bestimmung der genauen Kennlinien von LED und FET. Links: Simulationsschaltung der LED, hier wird die Abhängigkeit Diodenspannung U_D von dem Diodenstrom I_D aufgenommen. Mitte: Simulationsschaltung des MOSFETs mit $U_{GS} = 5$ V, hier wird die Abhängigkeit des Drain-Stroms I_D von U_{DS} aufgenommen. Rechts: Hier wird dieselbe Abhängigkeit wie im mittleren Bild untersucht, jedoch bei $U_{GS} = 0$ V, also im „Aus“-Arbeitspunkt.

Um die Spannung U_D zu bestimmen, die bei 100 mA an der LED abfällt, wird der Strom durch die LED von 1 nA bis 1 A in logarithmischen Abständen variiert und die Resultierende Diodenspannung wird abgegriffen (Abb. 4.13, links). Das Ergebnis der Simulation ist in Abbildung 4.14 dargestellt. Natürlich hätte es in diesem Fall auch ausgereicht, die Schaltung nur mit dem konstanten Strom 100 mA zu simulieren. Stattdessen wird hier die gesamte U-I-Kennlinie erstellt, aus der abgelesen werden kann, dass sich bei der Diode im „Ein“-Arbeitspunkt eine Spannung von $U_{D,AP} = 1,495$ V einstellt.

Anschließend muss die Spannung, die sich im „Ein“-Arbeitspunkt am Transistor zwischen Drain und Source einstellt, bestimmt werden. Hier kann sehr ähnlich verfahren werden, die benutzte Simulationsschaltung ist mittig in Abbildung 4.13 dargestellt. U_{GS} wird konstant bei 5 V gehalten und U_{DS} wird von 0 V bis 5 V variiert. Das Ergebnis dieser Simulation ist die Ausgangskennlinie des MOSFETs, dargestellt in Abbildung 4.15. Es ist sehr leicht abzulesen, dass sich am Transistor im Arbeitspunkt eine Spannung von $U_{DS,AP} = 0,58$ V einstellt.

Somit sind alle notwendigen Werte bestimmt und es lässt sich der optimale Vorwiderstand berechnen:

$$\begin{aligned}
 R &= \frac{U_0 - U_{D,AP} - U_{DS,AP}}{I_{D,max}} \\
 &= \frac{5 \text{ V} - 1,495 \text{ V} - 0,58 \text{ V}}{100 \text{ mA}} \\
 &= 29,25 \, \Omega
 \end{aligned}$$

Damit der Maximalstrom nicht überschritten wird, haben wir den nächstgrößeren in der Widerstandsreihe verfügbaren Widerstand ausgewählt, in diesem Fall hat dieser den Wert $33 \, \Omega$.

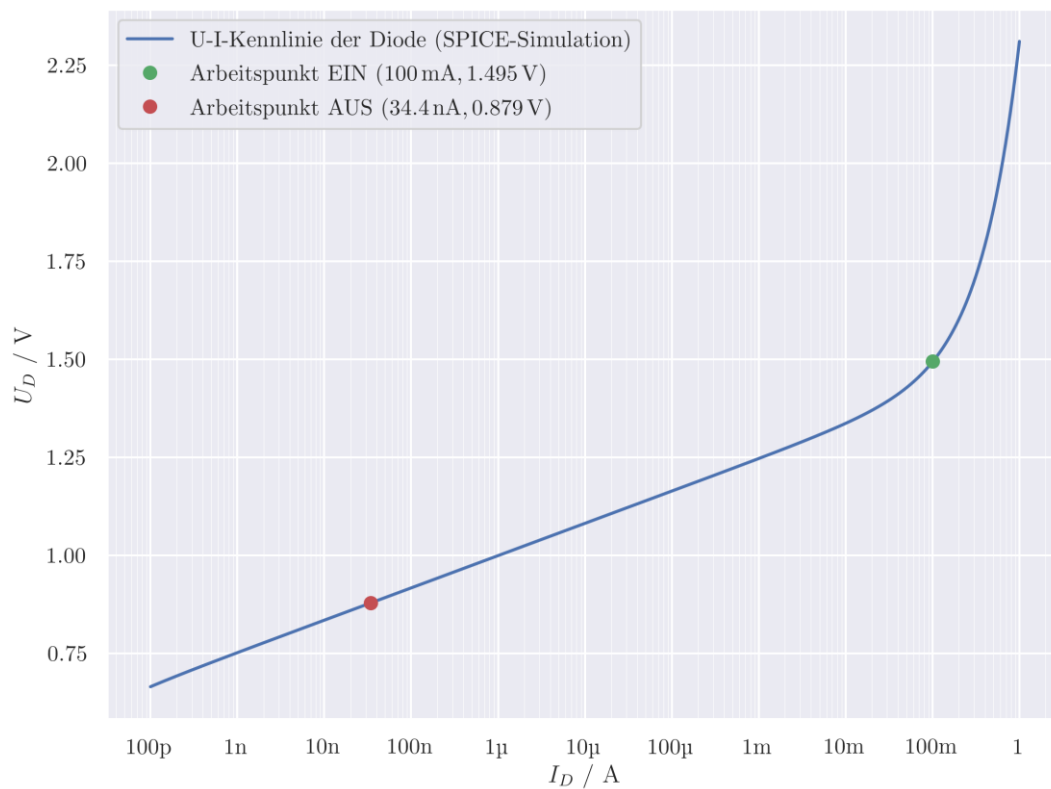


Abbildung 4.14: U-I-Kennlinie der LED. Der „Ein“-Arbeitspunkt kann direkt abgelesen werden: Bei dem soll-Strom $I_D = I_{D,max} = 100 \text{ mA}$ stellt sich eine Spannung von $U_D = 1,495 \text{ V}$ ein. Der „Aus“-Arbeitspunkt wird weiter unten behandelt.

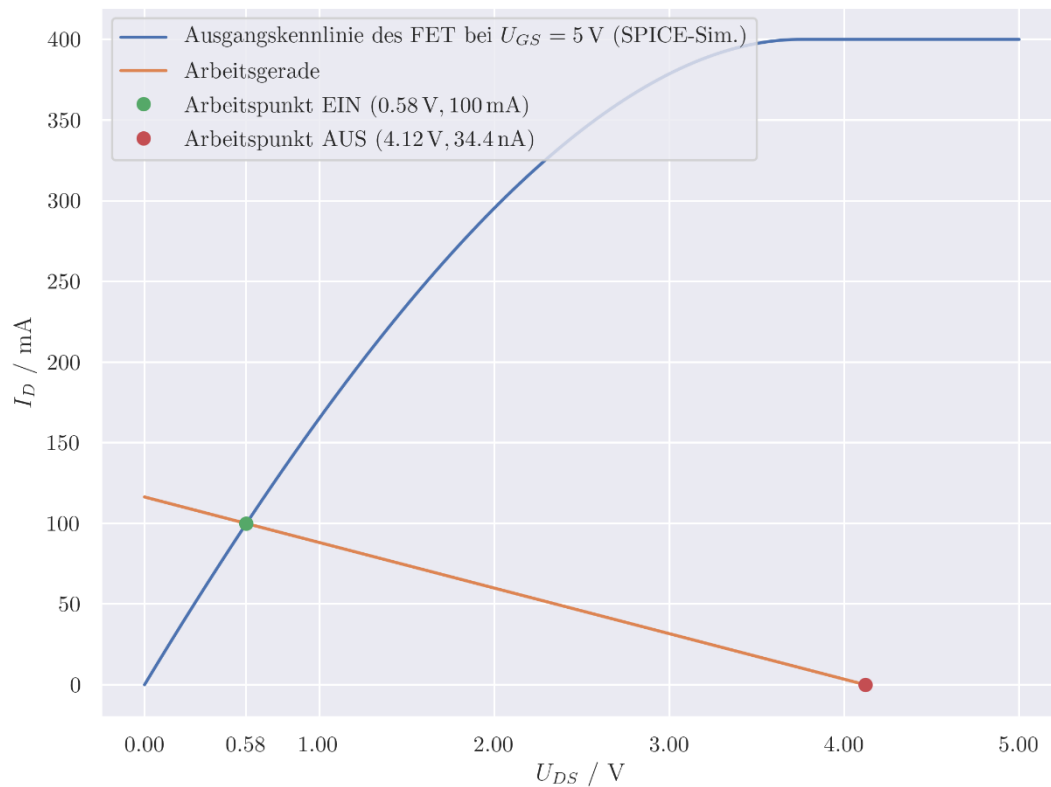


Abbildung 4.15: Ausgangskennlinie des Transistors ($U_{GS} = 5 \text{ V}$). Der „Ein“-Arbeitspunkt kann wieder direkt abgelesen werden: Bei dem soll-Strom $I_D = 100 \text{ mA}$ stellt sich eine Spannung von $U_{DS} = 0,58 \text{ V}$ ein. Der „Aus“-Arbeitspunkt, sowie die Arbeitsgerade wird weiter unten behandelt.

Bei der Simulation der U-I-Kennlinie der LED fällt eine Eigenart in auf, die in Abbildung 4.15 zu sehen ist. Selbst bei sehr niedrigen Strömen (nA-Bereich) fällt noch eine große Spannung an der LED ab. Auf den Betrieb im „Ein“-Modus (Pulse = 5 V) hat dies keinen Einfluss, da hier der Arbeitspunkt durch den Vorwiderstand festgelegt wird. Es wäre zu erwarten, dass der Arbeitspunkt „Aus“ (also Pulse = 0 V) sehr simpel ist, nämlich $U_{DS} = 5 \text{ V}$, $I_D = 0 \text{ A}$. Die in Abbildung 4.15 erkennbare starke Nichtlinearität der LED im unteren Strombereich sorgt jedoch dafür, dass sich ein anderer Arbeitspunkt einstellt.

Es ist zu erwarten, dass der Transistor selbst bei $U_{GS} = 0 \text{ V}$ noch einen sehr geringen Strom I_D durchlässt, wenn eine Drain-Source-Spannung von $U_{DS} = 5 \text{ V}$ anliegt. Tatsächlich scheint der Transistor bei einer Gate-Source-Spannung von 0 V eine lineare Abhängigkeit des Stroms I_D von U_{DS} aufzuweisen, sodass bei $U_{DS} = 5 \text{ V}$ statt des gewünschten Nullstroms ein Strom von $I_D \approx 40 \text{ nA}$ fließt.

Dieses Ergebnis kommt aus einer weiteren Simulation, in der $U_{GS} = 0\text{ V}$ konstant gehalten wurde und U_{DS} von 0 V bis 5 V erhöht wurde (Abb. 4.13, rechts). In dem verwendeten SPICE-Modell ist der Drain-Strom I_D bei $U_{GS} = 0\text{ V}$ (Sperrbereich) linear von U_{DS} abhängig. Da es sich hier um einen Edge-Case (Ströme im nA-Bereich) handelt, kann es gut sein, dass das Modell die Realität nicht akkurat widerspiegelt. Dasselbe gilt für das SPICE-Modell der LED. Da diese kleinen Ströme aber dennoch große Spannungsabfälle an der LED zu bewirken scheinen, ist die Analyse nicht unbegründet.

Wie in Abbildung 4.14 zu erkennen ist, bedeutet der eben bestimmte Strom $I_D \approx 40\text{ nA}$, dass eine erhebliche Spannung an der LED abfallen würde, die somit nicht mehr am Transistor abfallen kann. Die Ströme sind hier so gering, dass die Spannung am Widerstand als vernachlässigbar klein angenommen werden kann, es gilt also näherungsweise $U_0 = U_{LED} + U_{DS}$. Diese Änderung der Spannung am Transistor führt wiederum dazu, dass der Strom durch den Transistor auch abfällt, was wieder die Spannung an der LED beeinflusst. Dieses dynamische System lässt sich leicht iterativ lösen, mit dem Anfangsschätzwert $U_{DS} = 5\text{ V}$ und $I_D = 0\text{ A}$. Die Iteration ist in Tabelle 4.2 zu sehen, eine graphische Darstellung ist in Abbildung 4.16 gezeigt. Die Iteration konvergiert sehr schnell und es stellt sich ein „Aus“-Arbeitspunkt von $(U_{DS}; I_D) = (4,12\text{ V}; 34,4\text{ nA})$ ein. Dieser Arbeitspunkt ist auch in Abbildung 4.14 und Abbildung 4.15 eingezeichnet. In Abbildung 4.15 ist auch die Arbeitsgerade eingezeichnet, die beim Ein- und Ausschalten des Transistors passiert wird.

Um das Schaltverhalten mit dem berechneten Widerstandswertes von $R = 33\ \Omega$ zu überprüfen, wird noch eine letzte Simulation der Gesamtschaltung durchgeführt, wobei U_{GS} (Pulse) nicht, wie in der praktischen Umsetzung der Schaltung, binär zwischen 0 V und 5 V geschaltet wird, sondern kontinuierlich geändert wird, damit sich die glatte Kurve bildet, die in Abbildung 4.18 zu sehen ist. Die verwendete Simulationsschaltung ist in Abbildung 4.17 dargestellt.

Iterationsschritt	0	1	2	3
$I_D[\text{nA}]$	0	41,7	34,3	34,4
$U_{\text{LED}}[\text{V}]$	0	0,886	0,879	0,879
$U_{DS}[\text{V}]$	5	4,11	4,12	4,12

Tabelle 4.2: Der „Aus“-Arbeitspunkt wird durch eine numerische Iteration gefunden. Die Iteration wird so durchgeführt, dass erst aus U_{DS} , mithilfe der linearen Kennlinie $I_D(U_{DS})$, die im Sperrbereich des Transistors gültig ist, der Strom I_D bestimmt wird. Aus I_D wird dann mithilfe der in Abbildung 4.14 gezeichneten Kennlinie der LED die Spannung $U_D = U_{\text{LED}}$, die an der LED abfällt bestimmt. Aus dem Verhältnis $U_{DS} = 5 \text{ V} - U_{\text{LED}}$ wird der neue Wert für U_{DS} bestimmt und die Iteration fängt von vorne an.

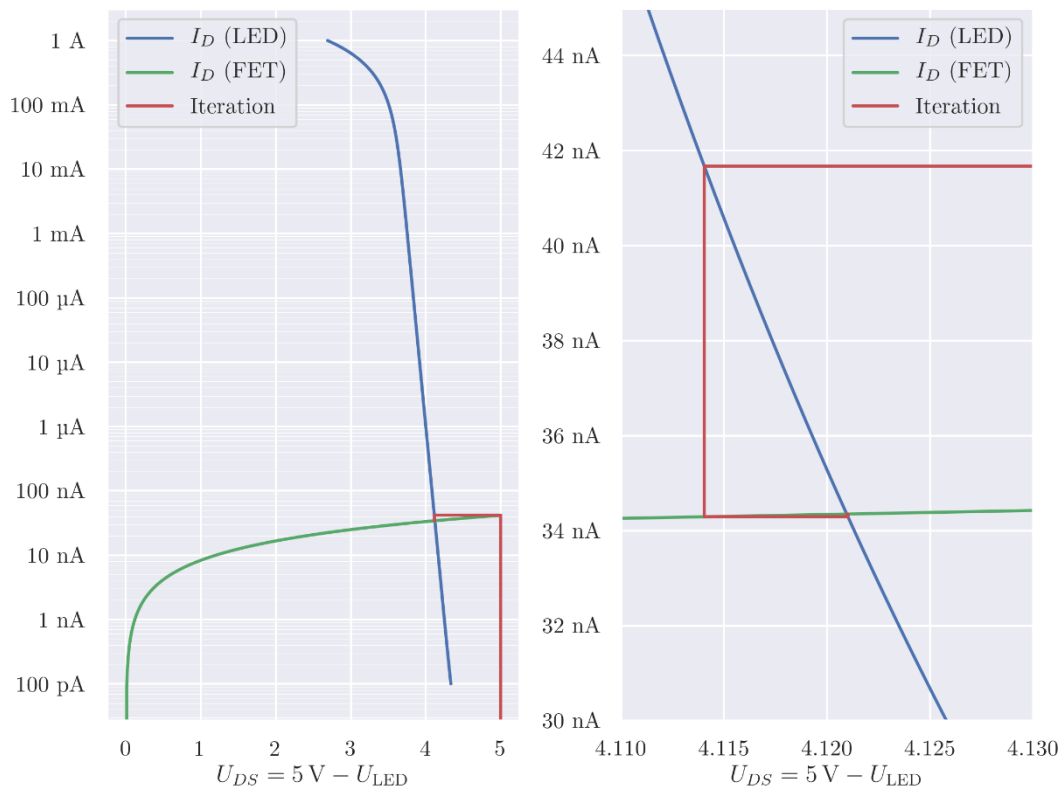


Abbildung 4.16: Dieselbe Iteration aus Tabelle 4.2 in graphischer Darstellung. Die blaue Kurve zeigt $I_D(U_{\text{LED}})$, die Umkehrfunktion der in Abbildung 4.14 dargestellten U-I-Kurve der LED. Hierbei wird verwendet, dass $U_{\text{LED}} = 5 \text{ V} - U_{DS}$ gilt. Die grüne Kurve zeigt das lineare Verhältnis $I_D(U_{DS})$, das beim Transistor im Fall $U_{GS} = 0 \text{ V}$ gilt. Der rechte Graph zeigt eine Vergrößerung des linken im Bereich um den „Aus“-Arbeitspunkt. Die Iteration hat die graphische Interpretation, den Schnittpunkt der zwei Gleichungen zu finden.

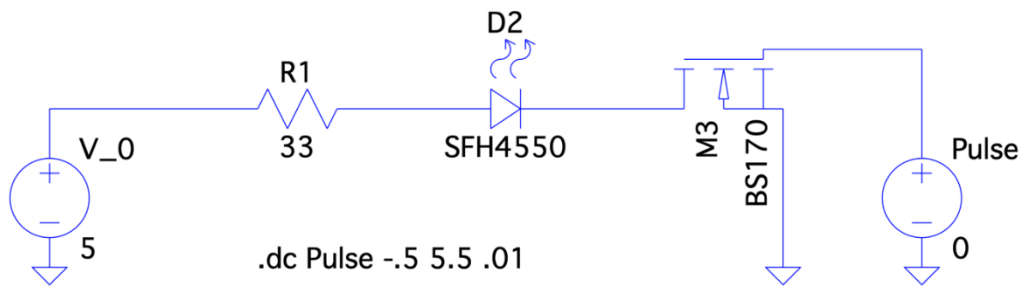


Abbildung 4.17: Simulationsschaltung mit der der Graph in Abbildung 4.18 erstellt wurde.

Wie zu sehen ist, erreicht der Diodenstrom nicht den gewünschten Wert von 100 mA, sondern nur knapp über 90 mA. Dies liegt daran, dass statt dem berechneten Widerstandswert von $29,25\ \Omega$ ein $33\ \Omega$ Widerstand verwendet wird. Unsere Tests haben gezeigt, dass der so erreichte Strom durch die LED für unseren Anwendungszweck ausreichend ist.

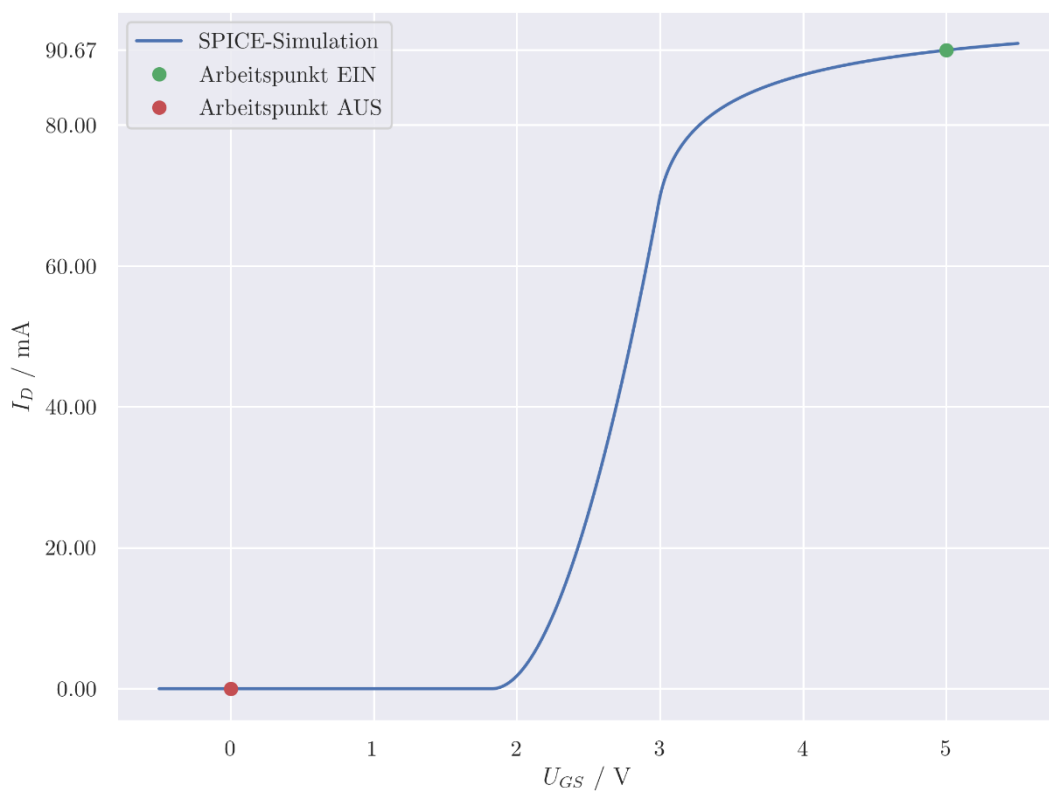


Abbildung 4.18: Schaltverhalten mit $R = 33\ \Omega$ bei $U_0 = 5\ V$.

4.3.2 Auswahl einer geeigneten Integrationszeit

Wie bereits erwähnt, darf die Integrationszeit weder zu klein noch zu groß sein. Wird zu lange integriert, tritt eine Sättigung auf. Der Kondensator lädt sich auf 5 V auf und eine genaue Messung ist nicht mehr möglich. Wird zu kurz integriert, sinkt das Detail der Messergebnisse. Der verwendete Arduino unterstützt nur 1024 diskrete Spannungswerte; bei kleiner Integrationszeit liegen alle Messungen im unteren Wertebereich und die Auflösung kommt an ihre Grenzen, es kann nicht mehr zwischen verschiedenen Werten differenziert werden. Ist die Integrationszeit beispielsweise so klein, dass alle im Normalfall auftretenden Messungen im unteren Viertel des Messbereichs liegen, sind nur noch 256 diskrete Spannungswerte unterscheidbar (die oberen 768 treten nie auf). Eine optimale Integrationszeit nutzt also den kompletten Wertebereich (0 V bis 5 V) so weit wie möglich aus, ohne dass im normalen Betrieb eine Sättigung an der oberen Spannungsgrenze auftritt.

In Abbildung 4.19 ist eine Messung dargestellt, bei der 6 verschiedene Integrationszeiten miteinander verglichen werden. Zur Referenz ist auch noch eine Messung aufgetragen, bei der die LED nicht eingeschaltet wurde (dunkelblaue Kurve). Mit dieser Messung kann das konstante Störsignal abgeschätzt werden. Der Arduino-Code für diese Messung ist in Anhang D.3 hinterlegt. Das Nephelometer wurde in dem Gestell platziert und die Nebelmaschine wurde etwa 70 s nach Beginn der Messung eingeschaltet. Bei etwa $t = 140$ s wurde die Nebelmaschine wieder ausgeschaltet und die Filteranlage eingeschaltet. Der Peak, der zu diesem Zeitpunkt zu sehen ist, ist durch die temporäre Verdichtung des Nebels innerhalb der Dunkelkammer entstanden, hervorgerufen durch Turbulenzen, die vom Sog der Filteranlage ausgingen.

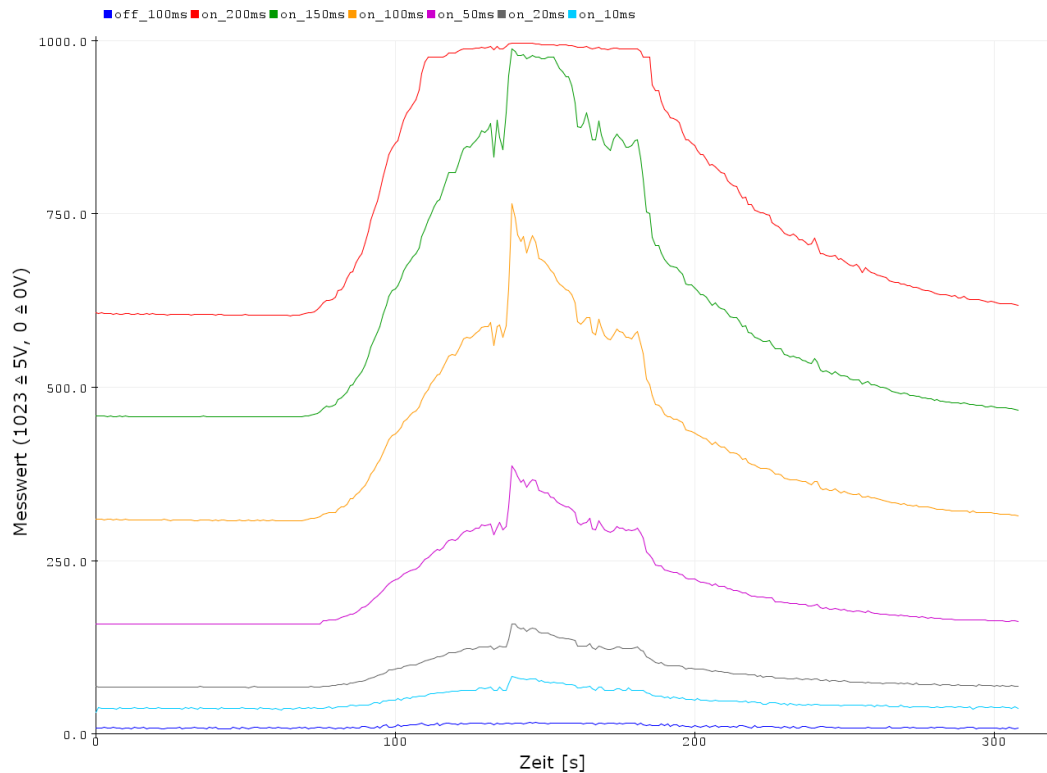


Abbildung 4.19: Testmessung zur Bestimmung der optimalen Integrationszeit T . Diese Abbildung stellt die unveränderte Ausgabe des Programms in Anhang D.3 dar und wurde mit dem Arduino Serial Plotter erstellt.

Es ist zu erkennen, dass die oberen beiden Integrationszeiten (T) zu groß sind. Bei $T = 200$ ms (rote Kurve) ist die Kurve bereits vor dem oben genannten Peak gesättigt. Bei $T = 150$ ms (grüne Kurve) kommt die Kurve bei dem Peak an ihre Grenzen und es ist hier bereits weniger Detail zu erkennen als in den unteren Kurven. Die naheliegende Wahl ist deshalb $T = 100$ ms (gelbe Kurve), da wie oben erläutert der verfügbare Wertebereich so weit wie möglich ausgenutzt werden soll. In dem Programm für das Nephelometer, hinterlegt in Anhang D.5, wird deshalb auch $T = 100$ ms einprogrammiert.

Der bei allen Messungen zu erkennende Konstantwert entsteht dadurch, dass nicht die gesamte Strahlung der LED vom Lichtfang eingefangen wird, sondern ein erheblicher Teil auch ohne in der Luft gelöste Partikel zur PD gelangt. Dieses Phänomen schränkt den verwendbaren Wertebereich und damit die effektive Auflösung weiter ein. Bei $T = 100$ ms ist zu erkennen, dass das untere Viertel des Wertebereichs nie erreicht wird.

In Abbildung 4.19 ist gut die Linearität des Integrators zu erkennen: die gelbe Kurve ($T = 100$ ms) lässt sich konstruieren, indem die violette Kurve ($T = 50$ ms) verdoppelt wird. Diese Eigenschaft wird im nächsten Abschnitt bei der Referenzmessung ausgenutzt, wo aber auch gezeigt wird, dass die Linearität aufgrund des Programms nicht perfekt ist.

Eine denkbare Verbesserung des Programms ist die Verwendung einer variablen Integrationszeit. So könnte bei einer niedrigen Aerosoldichte automatisch eine große Integrationszeit und bei einer hohen Dichte eine kleine Integrationszeit gewählt werden. Dies würde bewirken, dass die Auflösung immer bestmöglich ausgenutzt wird. Diese Verbesserung kann direkt in der Arduino-Software implementiert werden. Es müsste dann jedoch auch die aktuelle Integrationszeit auf die serielle Schnittstelle geschrieben werden, damit diese im Bedienungsprogramm (siehe unten) berücksichtigt werden kann. Zusätzlich müsste ein Modell erstellt werden, dass die λ -Faktoren (siehe unten) für Integrationszeiten berechnet, die nicht bei der Referenzmessung auftreten. Die Referenzmessung wird im nächsten Abschnitt behandelt.

4.3.3 Referenzmessung zur Umrechnung der Messwerte in dB

Die Ausgabe des Nephelometers soll die Dämpfung (in dB) innerhalb der Dunkelkammer sein. Bisher wird jedoch nur eine ganze Zahl zwischen 0 und 1023 ausgegeben, die den Spannungswert am Eingang des Arduinos zwischen 0 V und 5 beschreibt. Um die Abhängigkeit zwischen diesem Messwert und der Dämpfung zu bestimmen, wird eine Referenzmessung durchgeführt.

Bei der Referenzmessung wurde das Nephelometer in einen Rauchkanal gelassen. Hier wurde die Rauchdichte kontrolliert erhöht und die Dämpfung wurde von einem Referenzmessgerät (MIREX) aufgezeichnet. Das Nephelometer hat hierbei dasselbe Programm wie in der obigen Messung ausgeführt. Die Ergebnisse sind in Abbildung 4.20 dargestellt.

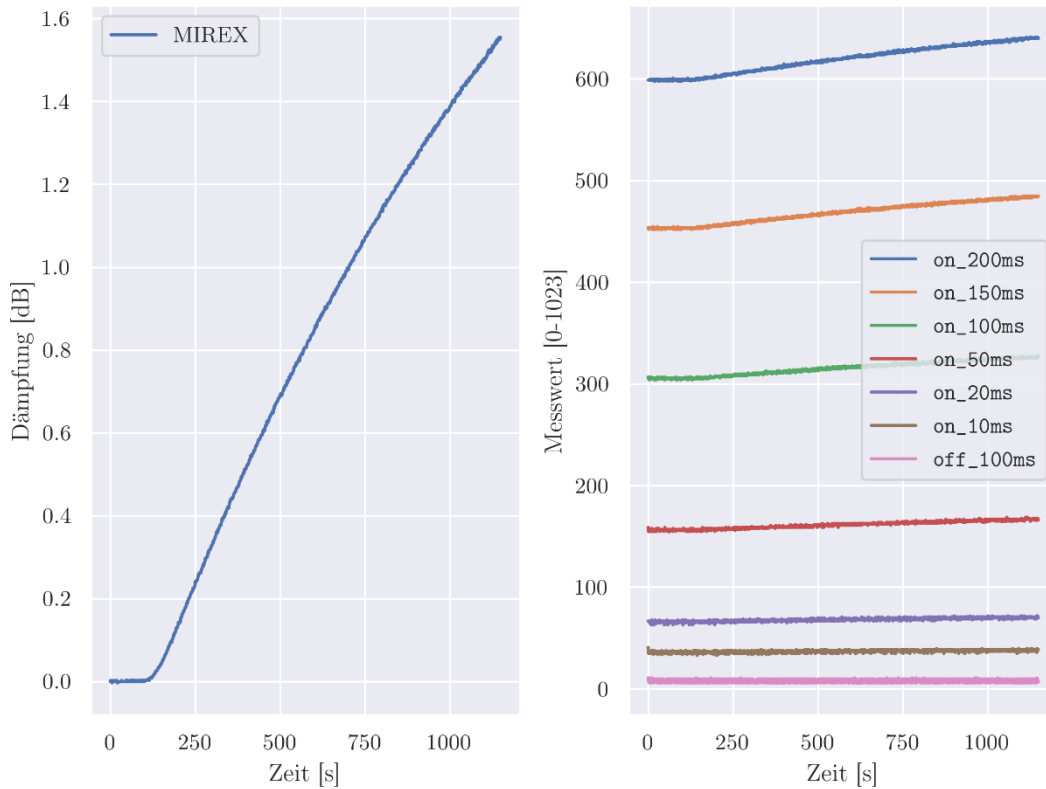


Abbildung 4.20: Ergebnis der Referenzmessung. Links: Messwerte des Referenzmessgeräts (MIREX) in dB. Rechts: Messwerte des Nephelometers mit verschiedenen Integrationszeiten. Geplottet ist genau wie in Abbildung 4.19 die unveränderte Ausgabe des Programms, das in Anhang D.3 zu finden ist.

Zur Auswertung der Messergebnisse müssen als erstes die Verhältnisse zwischen den verschiedenen Integrationszeiten untereinander untersucht werden. Beim verwendeten Integrator wird, wie bereits ausführlich beschrieben, die Integrationszeit T vom Arduino-Programm festgelegt. Bei einem Eingangssignal $x(t)$ wird deshalb das folgende folgendes Ausgangssignal zurückgegeben:

$$I_T[x](t) = c \int_{t-T}^t x(\tau) d\tau, \quad (2.4)$$

wobei c eine Konstante ist. Es kann erkannt werden, dass das Integral mit dem T -fachen des im Kapitel Theoretischer Hintergrund beschriebenen Mittelwerts äquivalent ist:

$$I_T[x](t) = cT\bar{x}(t). \quad (4.8)$$

Daraus folgt, dass das Verhältnis zweier unserer Messungen bei verschiedenen Integrationszeiten durch die folgende Relation ausgedrückt werden kann:

$$I_{T_1}[x](t) = \frac{T_1}{T_2} I_{T_2}[x](t). \quad (4.9)$$

Dieses Verhältnis wurde bereits im letzten Abschnitt durch die Linearität des Integrators begründet. In unserem Fall ist dieses Verhältnis jedoch nicht exakt, da die tatsächlichen Integrationszeiten nicht genau mit den angegebenen übereinstimmen. Dies liegt an dem Arduino-Programm, welches die Messungen steuert. Die Integrationszeit wird im Programm dadurch umgesetzt, dass der Prozessor nach Beginn der Integration die vorgegebene Zeit T wartet und anschließend die Integration stoppt. Durch die Umsetzung solcher Befehle auf dem Arduino kommt es zu Nichtlinearitäten zwischen der angegebenen Zeit T und der tatsächlichen Integrationszeit. Bei jeder Messung wird zum Beispiel zusätzlich zur vorgegebenen Zeit T eine weitere, konstante Zeit integriert, die von den ausgeführten Befehlen abhängt. Je größer T ist, desto weniger fällt diese konstante Zeit ins Gewicht, bei kleinem T ist sie jedoch ausschlaggebend. Wir modellieren diese Nichtlinearitäten in den Verhältnissen zwischen Messungen verschiedener Integrationszeiten T_1 und T_2 mit dem folgenden allgemeinen Modell:

$$I_{T_1}[x](t) = \lambda_{T_2 \rightarrow T_1} \cdot I_{T_2}[x](t). \quad (4.10)$$

Im Idealfall gilt also $\lambda_{T_2 \rightarrow T_1} = \frac{T_1}{T_2}$. Die richtigen Werte für λ wurden durch eine einfache numerische Optimierung des quadratischen Fehlers $(I_{T_1}[x](t) - \lambda_{T_2 \rightarrow T_1} \cdot I_{T_2}[x](t))^2$ gefunden. Die Ergebnisse sind in Abbildung 4.22 dargestellt. Es gilt $\lambda_{T_2 \rightarrow T_1} = \frac{1}{\lambda_{T_1 \rightarrow T_2}}$, deshalb mussten λ -Faktoren mit $\lambda > 1$ (die unteren Dreiecksmatrizen) nicht dargestellt werden.

Nachdem die λ -Faktoren berechnet wurden, kann mit der Auswertung fortgefahren werden. Als erstes wird von jeder aufgenommenen Messung der konstante Hintergrundstörwert abgezogen. Dies ist der „Bias“-Wert, der auch ohne eingeschaltete LED aufgenommen wird. Im Fall $T = 100$ ms ist dies leicht, da das Hintergrund-

signal hier aufgenommen wurde („off_100ms“). Es wird also einfach der Mittelwert des Hintergrundsignals von der Messung abgezogen. Da das Hintergrundsignal ohnehin konstant ist, wird durch das Mitteln nur das Rauschen entfernt. Bei allen anderen Signalen muss das Hintergrundsignal anhand der Messung bei $T = 100$ ms abgeschätzt werden. Dies ist jedoch mit den oben bestimmten λ -Parametern leicht und es wird für jede Messung bestimmt:

$$\tilde{x}_T(t) := x_T(t) - \lambda_{100\text{ms} \rightarrow T} \cdot h(t), \quad (4.11)$$

wobei $\tilde{x}_T(t)$ die verschobene Messung der Integrationszeit T ist, $x_T(t)$ die entsprechende Originalmessung und $h(t)$ das Hintergrundsignal („off_100ms“). Das Ergebnis der Verschiebungen ist in Abbildung 4.21 dargestellt.

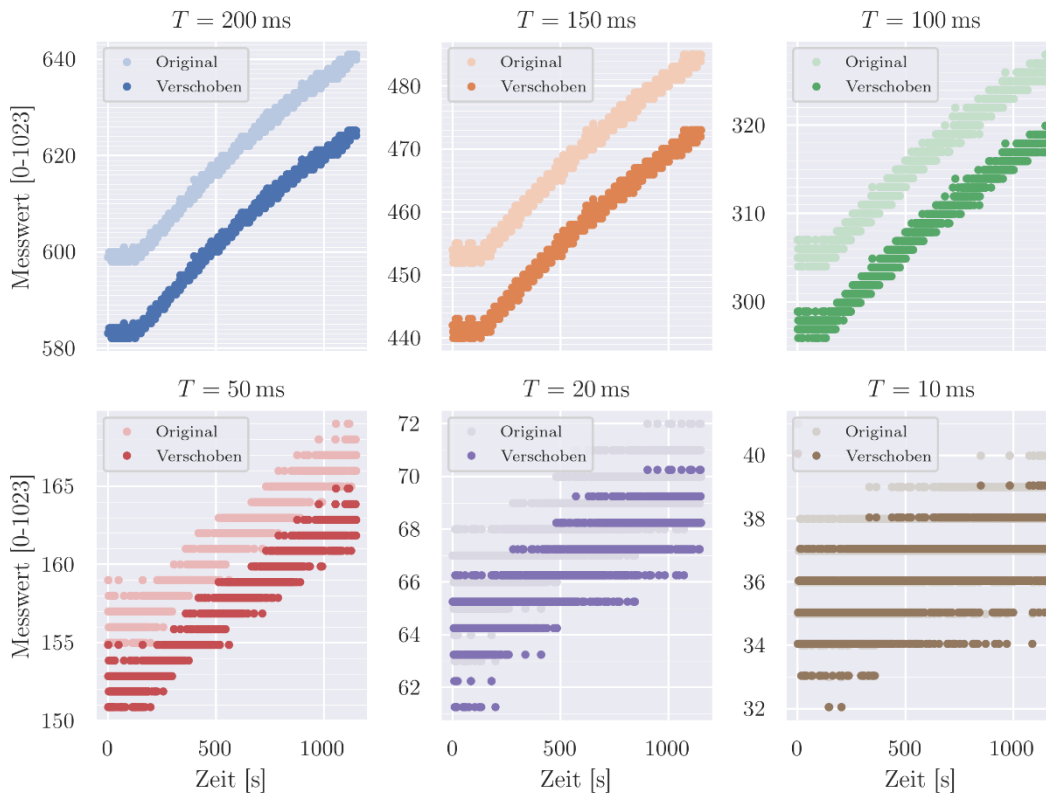


Abbildung 4.21: Die Messdaten $x(t)$ und die jeweils verschobenen Daten $\tilde{x}(t)$, bei denen der Hintergrundstörwert abgezogen wurde.

Hier kann erkannt werden, dass die Messwerte jetzt natürlich immer noch nur diskrete Werte, aber nicht mehr wie vorher ganze Zahlen annehmen. Sehr auffällig ist, was im vorherigen Abschnitt ausführlich diskutiert wurde: je kleiner die Integrationszeit, desto schlechter ist die effektive Auflösung. Deshalb wird für die restliche

Analyse die Messung bei $T = 200$ ms verwendet und die Ergebnisse werden zum Schluss auf die eigentlich wichtige Integrationszeit von $T = 100$ ms übertragen. Es ist ersichtlich, dass bei der Referenzmessung eine Messung bei einer noch größeren Integrationszeit, etwa $T = 300$ ms noch besser Ergebnisse hätte liefern können. Da in der Testmessung (Abb. 4.19) aber bereits bei $T = 150$ ms Sättigung aufgetreten ist, haben wir für die Referenzmessung keine solche größere Integrationszeit eingebaut.

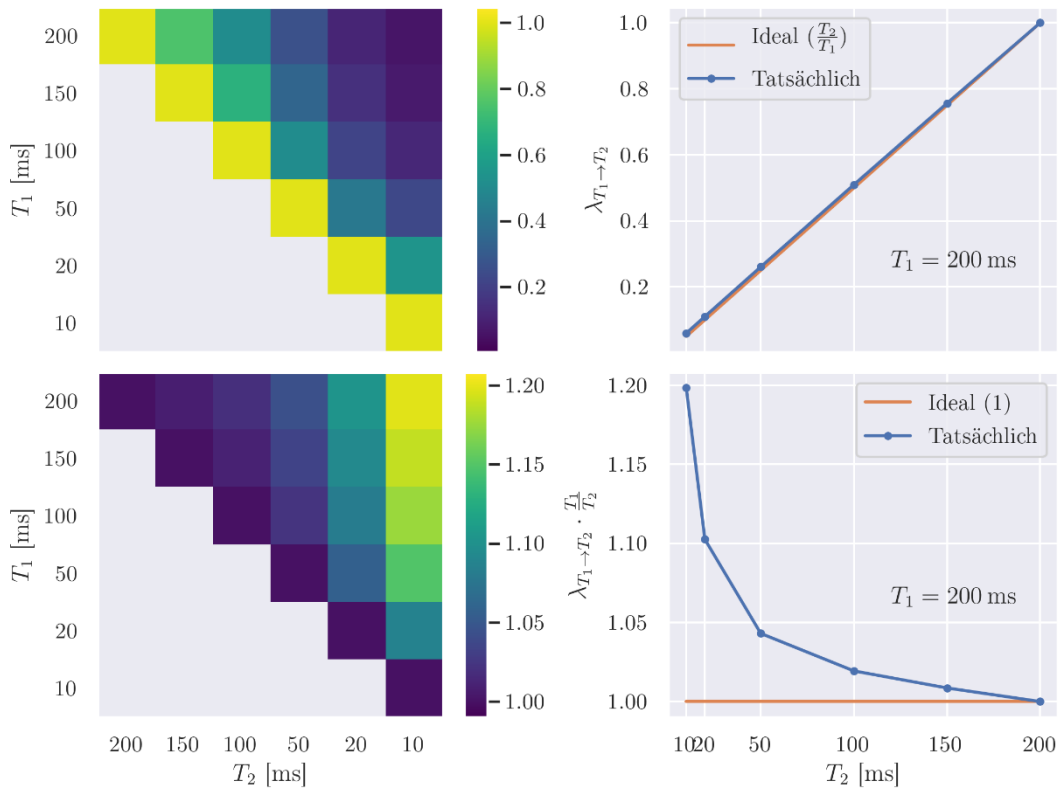


Abbildung 4.22: Umrechnungsfaktoren zwischen den Messwerten verschiedener Integrationszeiten. Im Idealfall gilt $\lambda_{T_1 \rightarrow T_2} = \frac{T_2}{T_1}$. In den oberen zwei Graphen sind die Faktoren $\lambda_{T_1 \rightarrow T_2}$ selbst dargestellt, links für alle Kombinationen, bei denen $\lambda_{T_1 \rightarrow T_2} > 1$ gilt und rechts nur für $T_1 = 200$ ms, im Vergleich mit dem Idealfall $\frac{T_2}{200 \text{ ms}}$. Unten sind die relativen Fehler der λ -Faktoren dargestellt. Links wieder für alle notwendigen Kombinationen und rechts für den Spezialfall $T_1 = 200$ ms. Hier ist gut zu erkennen, dass die entstehenden Nichtlinearitäten bei kleinen Integrationszeiten besonders ausschlaggebend sind.

Als nächstes soll die Messung geglättet werden; durch das verbinden mehrerer Messwerte kann so mehr Detail im Signal erreicht werden. Normalerweise kann so eine Kurve gut geglättet werden, indem sie mit einer symmetrischen Fensterfunktion (z.B. Gauß-Glocke) gefaltet wird. Da die Glättung, die hier angewendet wird, aber auch später im Echtzeitbetrieb durchgeführt werden soll, muss eine Funktion

verwendet werden, die zur Berechnung des Funktionswerts am Zeitpunkt t nur die vorherigen Zeitpunkte ($< t$) heranzieht. Berechnet wird ein durch diese Fensterfunktion gewichteter Mittelwert aller im Zeitraum $t - T_{\text{glätt}}$ bis t aufgenommener Messwerte. Konkret bedeutet das, dass die Funktion nur auf dem Intervall $[-T_{\text{glätt}}, 0]$ von Null verschieden sein darf. Drei solche Funktionen sind links in Abbildung 4.23 dargestellt. Alle drei haben die Form:

$$g(t; \alpha) = \begin{cases} \frac{\alpha + 1}{2T_{\text{glätt}}^{\alpha+1}} (t + T_{\text{glätt}})^{\alpha}, & \text{für } -T_{\text{glätt}} \leq t < 0 \\ 0, & \text{sonst} \end{cases} \quad (4.12)$$

mit $T_{\text{glätt}} = 20$ s. Diese Funktion wurde wegen ihrer Einfachheit gewählt, die Form ist t^{α} , wobei α ein frei wählbarer Parameter ist. Die Funktionen werden um $T_{\text{glätt}}$ verschoben und mit dem Faktor $\frac{\alpha+1}{T_{\text{glätt}}^{\alpha+1}}$ normalisiert. Der zusätzliche Faktor $\frac{1}{2}$ ist notwendig, da die Messung in $\Delta t = 500$ ms Intervallen aufgenommen wird, die Zeit aber in Sekunden gemessen wird. Die geglätteten Messwerte können dann wie folgt berechnet werden (gewichteter Mittelwert):

$$\bar{x}(t) = \sum_{i=0}^{\frac{T_{\text{glätt}}}{\Delta t}} g(t - i\Delta t) \tilde{x}(t - i\Delta t), \quad (4.13)$$

oder, da $g(t)$ so definiert ist, dass es außerhalb des betrachteten Bereichs Null wird, ist die folgende Rechnung äquivalent:

$$\bar{x}(t) = \sum_{i=-\infty}^{\infty} g(i\Delta t - t) \tilde{x}(i\Delta t). \quad (4.14)$$

Diese Berechnung entspricht einer diskreten Faltung von $\tilde{x}(t)$ mit $g(-t)$, und wird auch programmatisch als solche beschrieben.

In Abbildung 4.23 (links) ist $g(t)$ mit drei verschiedenen Werten für α gezeichnet. Auffällig ist, dass ein kleines α (hier 0,2) eine sehr flache Kurve erzeugt. Der Extremfall $\alpha \rightarrow 0$ stellt den normalen, ungewichteten Mittelwert dar (Rechteckfunktion).

Ein großer Wert für α (hier 5) hingegen, bedeutet eine sehr steile Kurve. Hier ist der Extremfall $\alpha \rightarrow \infty$ ein Dirac-Impuls, womit gar kein Mittelwert mehr gebildet wird, sondern nur der aktuellste Wert mit dem Faktor 1 in das Ergebnis einfließt.

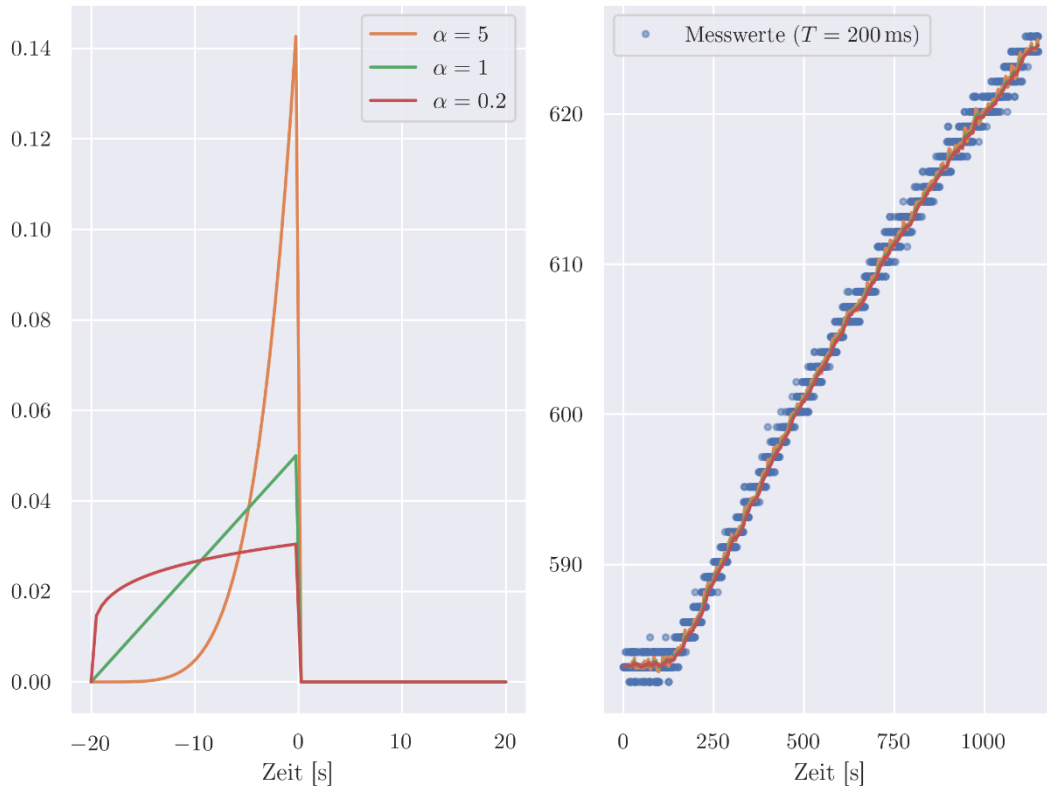


Abbildung 4.23: Links: Die Fensterfunktion $g(t)$ mit verschiedenen Werten des Parameters α . Rechts: Die verschobenen Messdaten $\tilde{x}(t)$ ($T = 200$ ms) und die geglätteten Daten $\bar{x}(t)$. Für das Glätten wurden die drei Funktionen verwendet, die links dargestellt sind, die farbliche Kennzeichnung in der Legende gilt für beide Graphen.

In Abbildung 4.23 sind rechts die Ergebnisse der Glättung mit diesen drei Filtern gezeigt. Es ist zu erkennen, dass ein kleines α eine glattere Kurve bedeutet, wohingegen ein großes α noch viel Rauschen durchlässt. Ein kleines α hat jedoch auch eine entsprechend größere Reaktionszeit, schließlich hinkt der Mittelwert der letzten 20 s dem Momentanwert hinterher. Als Mittelweg, mit dem Bonus, dass es sehr leicht zu berechnen ist, entscheiden wir uns für $\alpha = 1$.

Jetzt wo die Messung als glatte Funktion betrachtet werden kann, kann versucht werden, die Parameter zu finden, die $x(t)$ nach einem vorgegebenen Modell in eine Funktion $y(t)$ transformieren, welche möglichst nah an der Referenzmessung $m(t)$ (Abb. 4.20, links) liegt. Dazu muss als erstes das Modell aufgestellt werden. Da sich

die Kurven bereits sehr ähnlich sehen (siehe auch Abbildung 4.24, oben und mittig), sollte das folgende simple Modell hinreichen:

$$y(t; t_0, k, b) = k\bar{x}(t - t_0) + b. \quad (4.15)$$

Die Zeitverschiebung um t_0 kommt daher, dass die Reaktionszeiten der zwei Messgeräte (MIREX und Nephelometer) identisch sind. Ansonsten sollen also nur ein linearer Faktor k und ein konstanter additiver Parameter b gefunden werden.

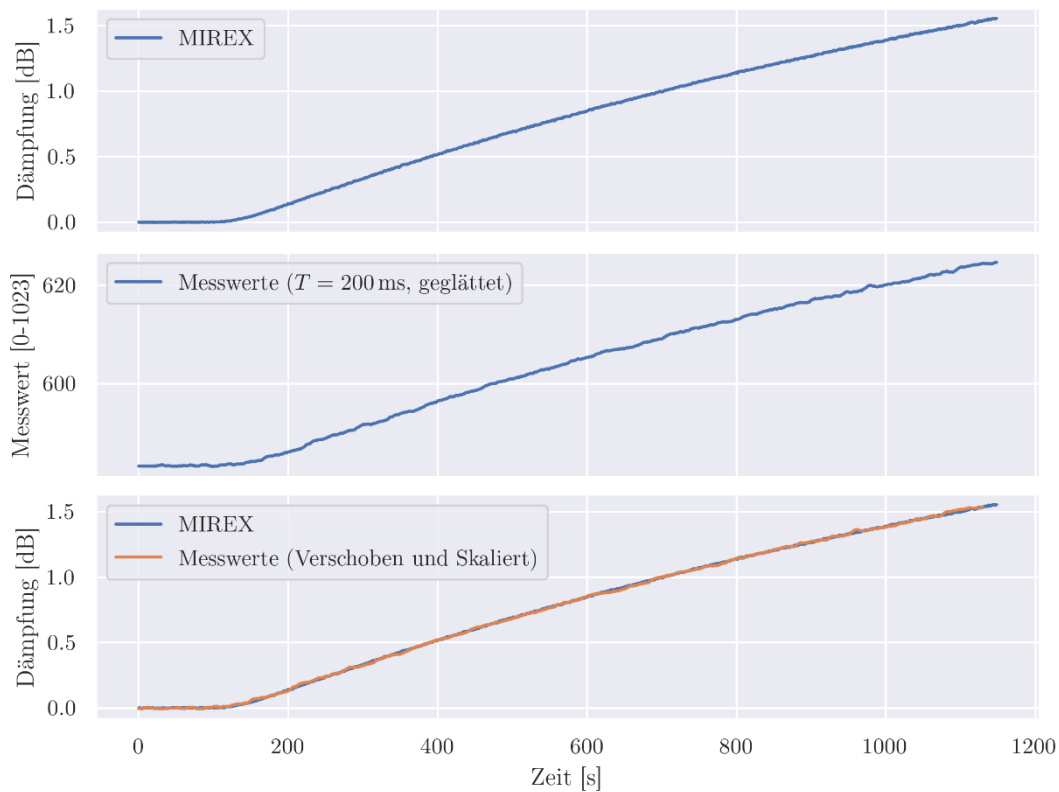


Abbildung 4.24: Oben: Die Messung $m(t)$ des Referenzmessgeräts. Mittig: Die verschobene und geglättete Messung $\bar{x}(t)$ des Nephelometers bei der Integrationszeit $T = 200$ ms. Es ist gut zu erkennen, dass die zwei Kurven eine sehr ähnliche Form aufweisen, die absoluten Werte jedoch sehr unterschiedlich sind. Unten: Die zwei Kurven $m(t)$ und $y(t)$ mit den gefundenen Parametern. $y(t)$ entspricht einer im Zeit- und Wertebereich verschobenen, sowie skalierten Funktion $\bar{x}(t)$.

Für die numerische Optimierung, mit der nach den Parametern gesucht wird, wird die folgende Fehlerfunktion verwendet:

$$E(t_0, k, b) = \sum_{i=0}^{\frac{T_{\text{Messung}}}{\Delta t}} (m(i\Delta t) - y(i\Delta t; t_0, k, b))^2 \begin{cases} 2, & i\Delta t < 100 \text{ s} \\ 1, & \text{sonst} \end{cases}, \quad (4.16)$$

wobei T_{Messung} die gesamte Zeit der Messung (ca. 20 min) und $\Delta t = 500$ ms der zeitliche Abstand zwischen zwei Messwerten ist. Es wird also die Methode der kleinsten Quadrate verwendet. Zusätzlich wird von der Fehlerfunktion der Zeitbereich unter 100 s doppelt gewichtet, damit es möglichst keine systematische Abweichung im 0 dB-Bereich gibt. Damit t_0 bestimmt werden kann, werden die verwendeten Funktionen so erweitert, dass sie außerhalb des Definitionsbereichs einen Fehler von 0 zurückgeben. Damit dies nicht dazu führt, dass der Optimierungsalgorithmus t_0 riesig wählt, sodass die resultierende Funktion nirgendwo definiert ist (und der Fehler Null ist), wird der Suchbereich für t_0 auf das Intervall $[-50 \text{ s}, 50 \text{ s}]$ begrenzt. Die gefundenen Parameter sind in Tabelle 2 zu sehen. Das Resultat der Transformation ist in Abbildung 4.24 zu erkennen, wo $y(t)$ und $m(t)$ im gleichen Graphen gezeichnet sind.

t_0	k	b
18,25 s	0,0372 dB	-21,70 dB

Tabelle 4.3. Die gefundenen Parameter für das Modell aus Gl. 4.15

Als letztes muss der Parameter k noch auf die zu benutzende Integrationszeit $T = 100$ ms angepasst werden. Dies ist leicht mit dem oben definierten λ -Faktor gemacht:

$$k_{100\text{ms}} = \lambda_{100\text{ms} \rightarrow 200\text{ms}} \cdot k = 0,073 \text{ dB} \quad (4.17)$$

In Abbildung 4.25 ist rechts das Ergebnis der Transformation für die originalen (um den Mittelwert des gemessenen Hintergrundstörsignals verschobenen) Messdaten, sowie die geglätteten Messdaten mit $T = 100$ ms dargestellt. Es ist zu sehen, dass die transformierte Kurve, abgesehen von etwas Rauschen, gut auf der Referenzkurve liegt. Links in Abbildung 4.25 ist dasselbe für alle Integrationszeiten geplottet. Wie zu erwarten war, sind die kleinsten Integrationszeiten $T = 10$ ms und

$T = 20$ ms sehr stark verrauscht. Dennoch scheint es bei allen Kurven der Fall zu sein, dass die Parameter, welche nur mit den $T = 200$ ms Messwerten gefunden wurden, nah am Optimum liegen.

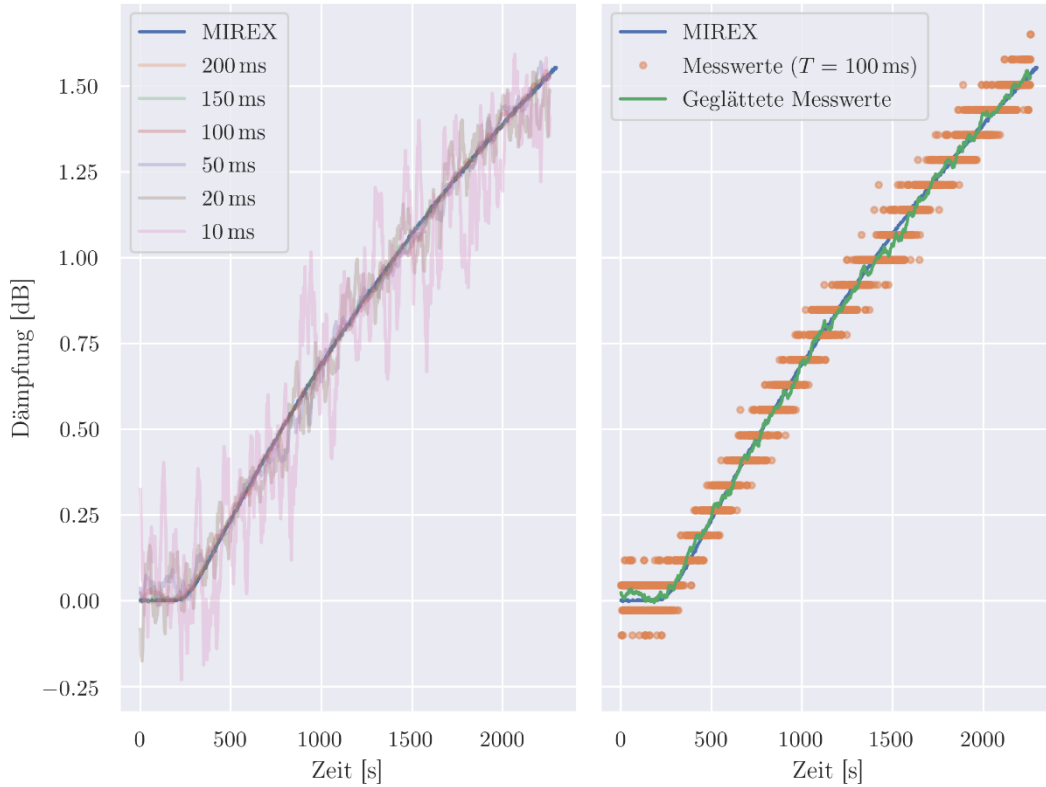


Abbildung 4.25: Die resultierenden Transformationen (nach Gleichung 4.15) mit den gefundenen Parametern (Tab. 2). Links: Transformation der geglätteten Messwerte aller Integrationszeiten. Rechts: Transformation der Messwerte bei $T = 100$ ms.

Der Parameter t_0 wird im Folgenden keine Rolle mehr spielen. Er war nur ein Hilfsmittel, um die anderen beiden Parameter zu finden, selbst hat er aber keinen Einfluss auf den Funktionswert von $y(t)$ und kann deshalb ignoriert werden. Dennoch liefert t_0 die Information, dass bei Messungen eine Verzögerung von ca. 20 s zu erwarten ist: die Zeit, die zwischen dem Signal des Nephelometers und dem des Referenzmessgeräts liegt. Im Bedienungsprogramm, welches gleich beschrieben wird, werden nur die zwei Parameter $k_{100\text{ms}}$ und b verwendet.

Der Python-Code zur Erstellung aller Grafiken dieses und des vorletzten Abschnittes, sowie zur Bestimmung des „Aus“-Arbeitspunktes, der λ -Parameter und der Parameter k , b und t_0 ist in Anhang D.4 hinterlegt.

4.3.4 Bedienungsprogramm und Endprodukt

Im Betrieb läuft auf dem Arduino ein anderes Programm als in den obigen Abschnitten, da nun nur noch die Integrationszeit $T = 100$ ms eine Rolle spielt. Es wird aber weiterhin neben der Messung, in der die LED eingeschaltet ist, der Hintergrundwert aufgenommen. Dieser wird dann wie oben von der eigentlichen Messung abgezogen. Das Programm ist in Anhang D.5 hinterlegt.

Die Ausgabe des Programms wird vom Arduino über die serielle Schnittstelle an den per USB verbundenen Computer weitergegeben. Hier wird die Schnittstelle von dem Bedienungsprogramm eingelesen. Es werden dann die Messungen ausgewertet und grafisch dargestellt. Der komplette Code des Bedienungsprogramms ist in Anhang D.6 hinterlegt. Ein Screenshot des Programms ist in Abbildung 4.26 dargestellt. Vom Arduino werden alle 500 ms der Hintergrundmesswert $h(t)$ (LED ausgeschaltet, $T = 100$ ms) und der eigentliche Messwert $x(t)$ (LED eingeschaltet, $T = 100$ ms) aufgenommen und an den Computer geschickt. Um die Dämpfung zu bestimmen, wird die Rechnung aus dem vorherigen Abschnitt mit denselben Parametern durchgeführt, wobei t_0 nicht beachtet wird. Zur Glättung wird wieder $\alpha = 1$ gesetzt und es wird ein gewichteter Mittelwert über $T_{\text{glätt}} = 20$ s, bei $\Delta t = 500$ ms also $M = 40$ Messwerte berechnet. Insgesamt wird berechnet:

$$y(t) = b + k(\bar{x}(t) - \bar{h}(t)) \quad (4.18)$$

Mit $\bar{x}(t)$ den geglätteten Messwerten:

$$\bar{x}(t) = \sum_{i=1}^M x(i\Delta t - T_{\text{glätt}})g(i\Delta t - T_{\text{glätt}}) \quad (4.19)$$

und $\bar{h}(t)$ den geglätteten Hintergrundwerten:

$$\bar{h}(t) = \sum_{i=1}^M h(i\Delta t - T_{\text{glätt}})g(i\Delta t - T_{\text{glätt}}). \quad (4.20)$$

Die Fensterfunktion $g(t)$ ist in Abbildung 4.23 zu sehen (die grüne Linie). Die Ergebnisse werden anschließend geplottet, wie in Abbildung 4.26 erkennbar. Im obersten Diagramm sind die Werte $y(t)$ der letzten Minute über der Zeit geplottet. Darunter sind alle Werte $y(t)$ seit Beginn der Messung geplottet. Unten im Fenster sind vier Diagramme zu sehen, die die aktuellen Messungen mit der Referenzmessung vergleichen. In dem Diagramm ganz links ist in blau der Teil der Referenzmessung, bei dem die LED aus war (bei $T = 100$ ms), aufgetragen. Diese Werte entsprechen der pinken Linie in Abbildung 4.20. Zusätzlich sind zwei horizontale Linien zu sehen; die rote ist der aktuellste Wert von $h(t)$ und die grüne ist ein „korrigierter“ Wert, der im Folgenden noch besprochen wird. In dem zweiten Diagramm von links sind diese Werte in Histogrammform dargestellt. Das blaue Histogramm beschreibt die relative Häufigkeit einzelner Messwerte bei der Referenzmessung, das rote Histogramm beschreibt die relative Häufigkeit der Messwerte bei der aktuellen Messung und das dünne bzw. gelbe Histogramm beschreibt dasselbe bei den „korrigierten“ Werten. Rechts daneben sind zwei weitere Diagramme, die exakt analog zu diesen beiden sind, nur dass hier die Messwerte, bei denen die LED an ist, betrachtet werden. Die entsprechende Linie in Abbildung 4.20 (und in Abbildung 4.21) ist grün.

Wie schon in Abbildung 4.26 zu sehen ist, liegen die Messwerte im Normalfall, angedeutet durch die roten horizontalen Linien, bzw. durch die roten Histogramme, außerhalb des Bereichs, der bei der Referenzmessung aufgetreten ist. Der Grund hierfür ist uns nicht bekannt. Es scheint jedoch nicht an der Temperatur zu liegen, wie aus Abbildung 4.27 ersichtlich wird.

Es ist zu erkennen, dass die Temperatur scheinbar einen kleinen Unterschied macht, jedoch keinen Hinweis darauf liefert, warum die Messung so stark von der Referenzmessung abweicht.

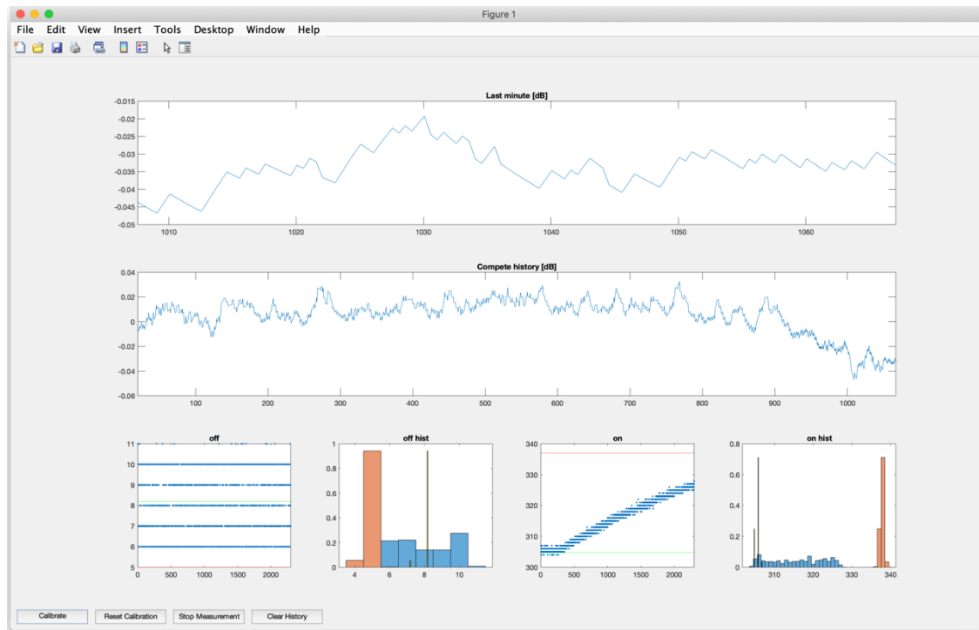


Abbildung 4.26: Die grafische Oberfläche des Bedienungsprogramms.

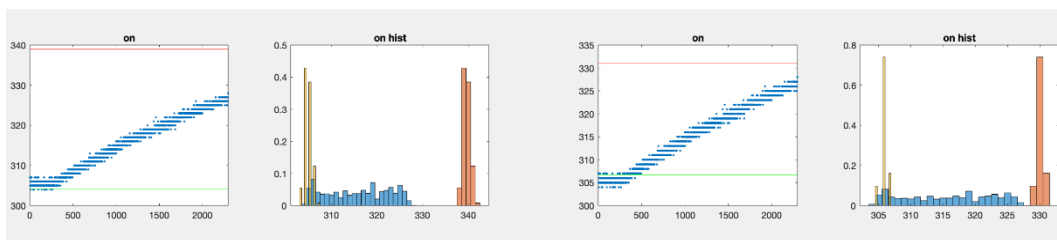


Abbildung 4.27 Links: Messung bei niedriger Temperatur (Nephelometer im Eisfach). Rechts: Messung bei hoher Temperatur (Nephelometer auf der Heizung)

Um das Problem zu beheben, ist in das Programm eine Ruhewertmitführung eingebaut. Wenn die aktuelle Aerosoldämpfung 0 dB betragen soll, kann der „Calibrate“-Button dafür benutzt werden. Solange dieser betätigt ist, wird ein Mittelwert der Messungen (LED an und LED aus) aufgenommen. Das Kalibrieren wird durch erneutes betätigen des Buttons wieder beendet. Nach Kalibration sind zwei Mittelwerte x_{0dB} und h_{0dB} verfügbar, die nach der folgenden Gleichung in die Rechnung eingehen:

$$y(t) = b + k(\bar{x}(t) - x_{0dB} - \bar{h}(t) + h_{0dB}). \quad (4.21)$$

In den vier unteren Diagrammen auf der Bedienoberfläche sind auch die „korrigierten“ Werte $\bar{x}(t) - x_{0dB}$ und $\bar{h}(t) - h_{0dB}$ als grüne horizontale Linien bzw. als dünne / gelbe Histogramme dargestellt.

Mit dem Button „Reset Calibration“ werden die Mittelwerte x_{0dB} und h_{0dB} wieder gelöscht. Um das Programm zu beenden muss der Button „Stop Measurement“ bestätigt werden, anschließend kann das Fenster geschlossen werden. Mit dem Button „Clear History“ wird die gesamte aufgezeichnete Messung gelöscht. Dies kann eventuell notwendig werden, wenn die Messung eine sehr lange Zeit gelaufen ist, dann kann das Programm theoretisch an die Leistungsgrenzen des Computers kommen. Im Normalfall sollte es aber nicht notwendig sein.

5 Fazit und Ausblick

Das Ziel der Bachelor-Projektarbeit ist es, einen Demonstrator zu entwerfen, der die optischen Umgebungsbedingungen eines Brandes durch Rauch simuliert und die Dichte dieses Rauchs mithilfe eines Nephelometers misst. In diesem Demonstrator sollen in späteren Projekten mit Terahertzstrahlung Objekte im Rauch erkannt werden. Es wird die Funktionsweise eines Nephelometers dargestellt und die Theorie hinter Terahertzstrahlung erklärt. Auf Ziele und Herangehensweisen innerhalb des Projektes wird ebenfalls eingegangen. Es folgt der Aufbau des Gestells und die Erklärung der Komponenten. Dann wird erläutert, wie das Nephelometer aufgebaut ist und wie es elektrisch und programmatisch umgesetzt wurde.

Während des Projektes hat dabei das Nephelometer den größten Teil der wissenschaftlichen Ausarbeitung ausgemacht. Die Entwicklung des Nephelometergehäuses aus den Ideen des Fachbereiches, die elektrische Auslegung der Platine und nicht zuletzt die Programmierung, Messung und Kalibrierung des Nephelometers erfordern eine ausgiebige Dokumentation. Der Aufbau der Wandstruktur des Nephelometers kann genutzt werden, um weitere beliebige lichtdichte, aber luftdurchlässige Körper zu erzeugen. Das ist durch die vorhandenen Modelle sichergestellt. Es ist durch die 3D-Drucktechnik möglich, das Gehäuse jederzeit auch in anderer Form erneut zu drucken. Der Schaltplan der Platine ist digitalisiert und ist somit anpassbar. Eine besondere Neuerung für das Fachgebiet ist das MATLAB-Programm, welches die Messung des Arduinos über die serielle Schnittstelle abgreift. Damit ist es möglich, im Betrieb auf die Daten zuzugreifen, die das Nephelometer erzeugt, und mit diesen dann weiterzuarbeiten.

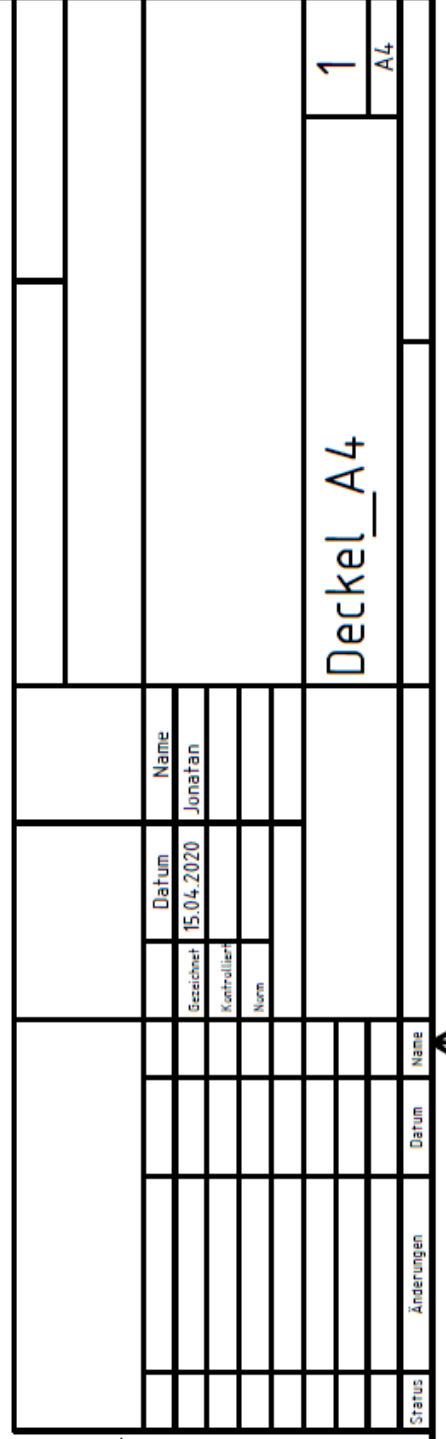
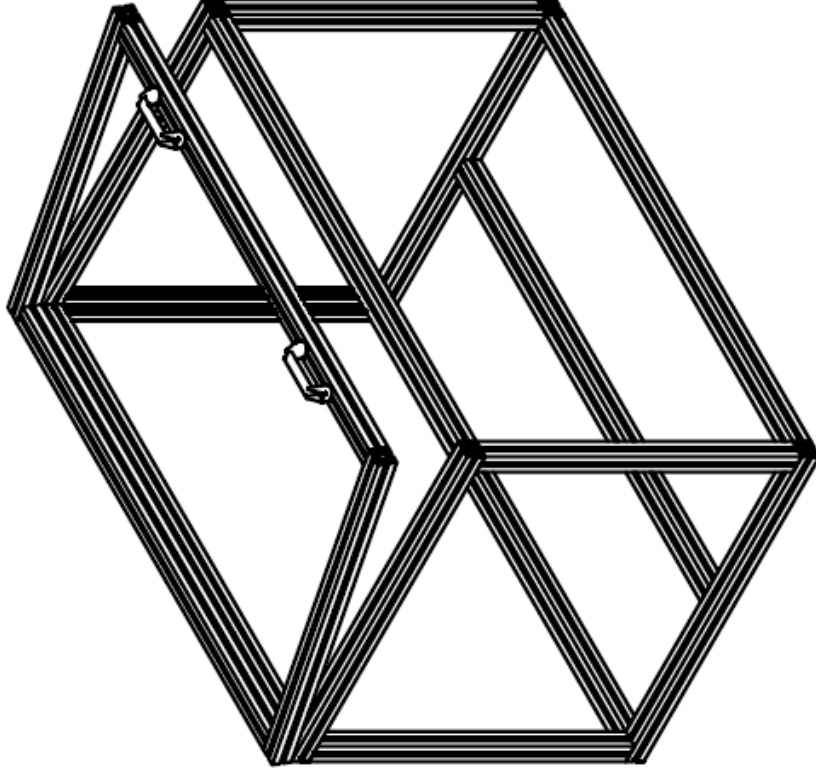
Das Projekt kann in Zukunft in verschiedene Richtungen erweitert werden. So ist es möglich, das Gehäuse des Nephelometers so anzupassen, dass damit Richtungsabhängig die Lichtbrechung durch Rauch untersucht werden kann. Ebenfalls ist das Programm des Nephelometers erweiterbar. In Zukunft ist der Einsatz des Nephelometers mit variablen Integrationszeiten denkbar, um die bestmögliche Auflösung der Messung für jede Rauchkonzentration sicherstellen zu können.

A Technische Zeichnungen

A.1 Grundgestell

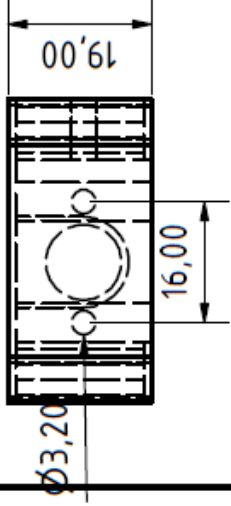
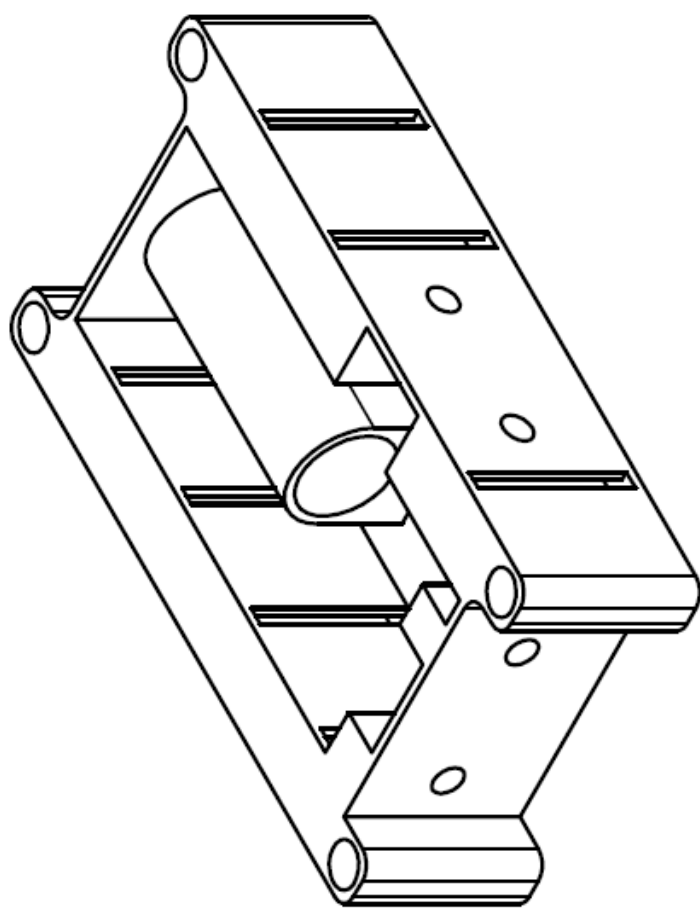
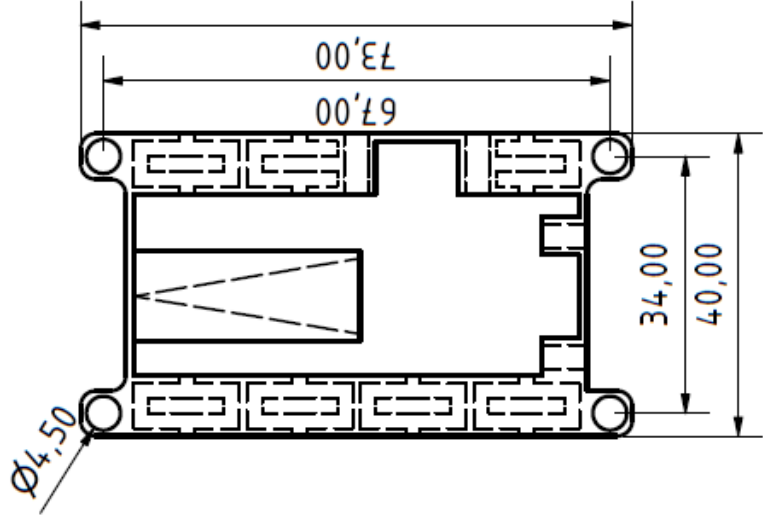
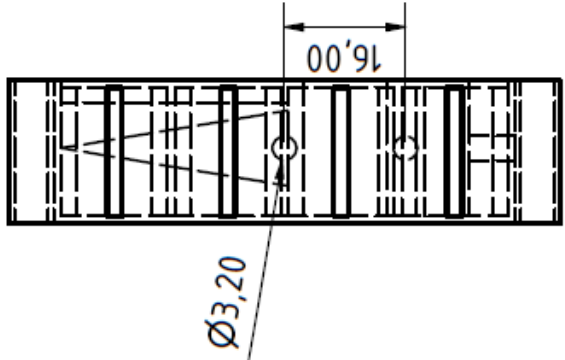
A.1.1 Dreitafelprojektion: Grundgestell

A.1.2 Dreitafelprojektion: Deckel & Zusammenbau



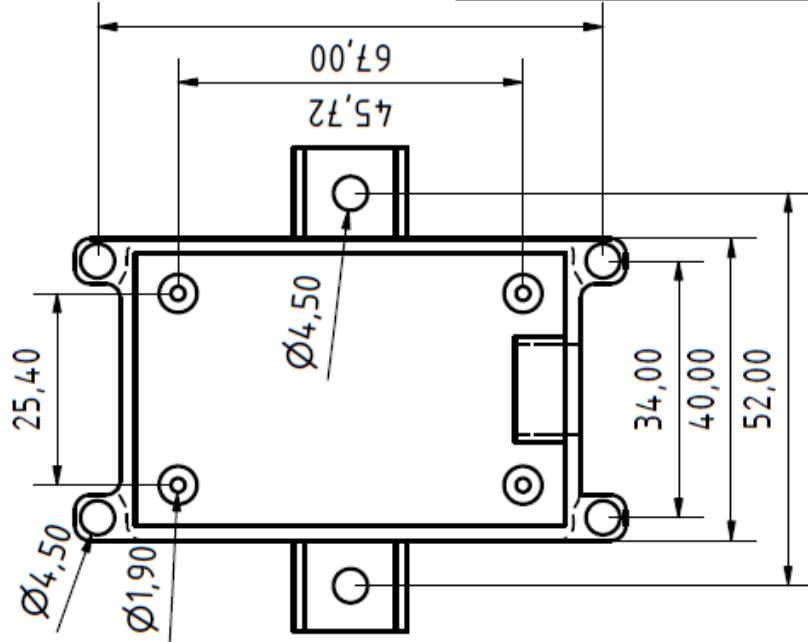
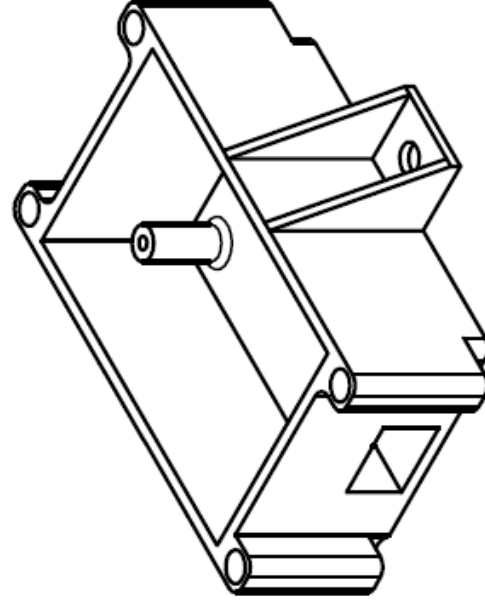
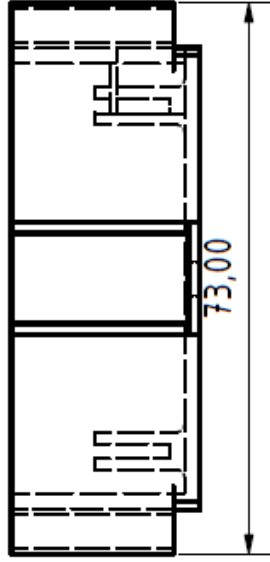
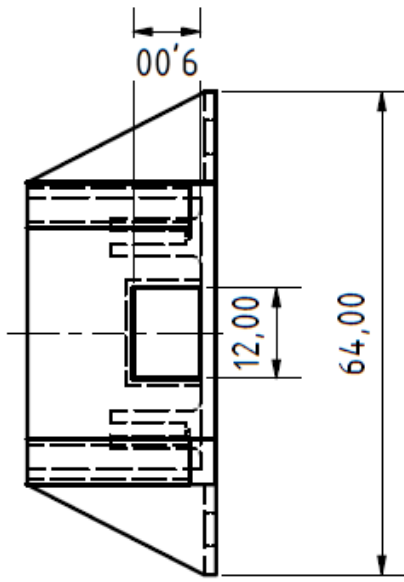
A.2 Nephelometer

A.2.1 Dreitafelprojektion: Dunkelkammer



A.2.2 Dreitafelprojektion: Deckel

A.2.3 Dreitafelprojektion: Arduino Gehäuse



B Projektmanagement

B.1 GANTT-Diagramm

Datum	Teilnehmer	Unterlagen	Inhalt
18.09.2019	J, O, N, T, B,		Vorbesprechung und grobes Umreißen der Aufgabenstellung
08.10.2019	A, B		Besichtigung der Labore, Einführung in die verwendeten Bauteile
15.10.2019	A,T		Powertiny ausprobieren
22.10.2019	A		Organisatorisches-Unterweisungen
29.10.2019	A,B		Organisatorisches-Laufzettel, Vorstellung,
11.11.2019	A,T		Besprechung Rauchsensor, Erhalten ergänzender Unterlagen, Bestellung Gestell fertigmachen
19.11.2019	A		Einweisung 3D-Drucker, Einweisung Werkzeuge
03.12.2019	A		Einkauf Schrauben f. Gestell
21.01.2019	A,T		Messungen Kalibrierung im Rauchkanal f. Nephelometer

B.2 ToDo-Liste

- 0. Vorplanung
 - 0.1. Arbeitstag in der Woche festlegen
 - 0.2. Programme für Kooperation festlegen
 - 0.3. Programme zum Zeichnen und Programmieren festlegen
- 1. Demonstrator
 - 1.1. Box grob dimensionieren
 - 1.1.1. Dimensionen und Spezifikationen mit Betreuern besprechen
 - 1.1.2. Außenbau/Umgebung: Fragestellungen
 - 1.1.2.1. Wagen kaufen oder selbst bauen?
 - 1.1.2.2. Wie soll das Gestänge montiert sein?
 - 1.1.2.3. Was muss am Kasten befestigt sein?
 - 1.1.3. Innenbau klären
 - 1.1.3.1. Was muss in den Kasten, was kann draußen bleiben?
 - 1.1.3.2. Wo muss das, was in den Kasten kommt, montiert werden?
 - 1.1.3.3. Wie bekommt man etwas von innen nach außen?
 - 1.1.4. Material klären
 - 1.1.4.1. Woraus werden Stützstrukturen gefertigt?
 - 1.1.4.2. Decke/Boden/Wände?
 - 1.1.4.3. Wie wird die Box selbst dicht?
 - 1.1.4.4. Wie wird die Box selbst stabil?
 - 1.2. Box planen
 - 1.2.1. Herunterladen und festlegen der Zeichnungen für die Profile
 - 1.2.2. In gewünschten Dimensionen zeichnen
 - 1.2.3. Bohrungen und Verbindungen in der Zeichnung festlegen
 - 1.2.4. Stückliste aus Zeichnung erstellen
 - 1.3. Box in Auftrag geben
 - 1.3.1. Angebot einholen
 - 1.3.2. Mit Betreuer über Angebot sprechen
 - 1.3.3. Angebot bestellen lassen
 - 1.3.4. Schrauben zusätzlich kaufen
 - 1.4. Box aufbauen

1.4.1. Grundkonstrukt aufbauen

1.4.1.1. Bodenplatte aufbauen

1.4.1.2. Seiten aufbauen

1.4.1.3. Deckel bauen

1.4.1.4. Deckel an Grundkonstrukt montieren

1.4.1.5. Bohrung/Schnitt in Seite für Leitungsauslässe

1.5. Box testen

1.5.1. Sichtprüfung

1.5.2. Dichtheitsprüfung mit Dampf

Box erweitern

1.5.3. Kabeleinführungen bohren/schneiden/montieren

1.5.4. nachdichten, eventuell Silikon

1.5.5. POWER-TINY einsetzen

1.5.5.1. Fernbedienungsmodul kaufen

1.5.6. HEPA-Filter-Sauger einbauen

1.5.7. Motor einsetzen

1.5.7.1. Motorschiene montieren

1.5.7.2. Motor aufsetzen

1.5.7.3. Kabelführung des Motors ausrichten

1.5.7.4. Arm montieren

1.5.8. Nephelometer einsetzen

1.5.9. Antenne anbringen

1.6. Box testen

1.6.1. Motor in Betrieb nehmen

1.6.2. Powertiny in Betrieb nehmen

1.6.3. Nephelometer in Betrieb nehmen- Siehe 2

1.6.3.1. Signaltest

1.6.3.2. Kalibrierungsmessung

1.6.4. Lüfter mit Hepa-Filter in Betrieb nehmen

1.6.5. Antenne testen

1.7. Box auf Wagen stellen

2. Nephelometer
 - 2.1. Grundgeometrie überlegen für Außenbox
 - 2.2. Box für Arduino entwerfen
 - 2.3. Wichtige Elemente festlegen
 - 2.3.1. Lichtfalle
 - 2.3.2. Stützbeine
 - 2.3.3. Platinenanordnung
 - 2.3.4. Kabelausführung
 - 2.3.5. Lichtdichter Lufteinlass
 - 2.4. Zusammenstellen der Platine
 - 2.4.1. Einordnen der Bauelemente
 - 2.4.2. Auslegen der Platine mit Pspice
 - 2.4.3. Simulation der Platine
 - 2.4.4. Ausrechnen der Widerstände
 - 2.5. Anfertigen der ersten Platinen
 - 2.5.1. Löten der LED-Platine
 - 2.5.2. Löten der Fotodioden-Platine
 - 2.6. Anfertigen der letzten Platine
 - 2.6.1. Auslegen der Platine
 - 2.6.2. Auflöten der Bauteile auf die Lochrasterplatine
 - 2.7. Kabeldurchführung fertigmachen
 - 2.8. Arduino an Platine montieren
 - 2.9. Box zusammenschrauben-Siehe 1.7.3
3. Programm für Nephelometer schreiben
 - 3.1. Altes Programm sichten
 - 3.2. Neues Programm einrichten
 - 3.3. Kennlinien aufnehmen, Programm dahingehend anpassen
 - 3.4. Nephelometer Programmatisch kalibrieren
 - 3.5. Ausleitung der Daten in Matlab o.ä. ermöglichen
 - 3.6. Programm etablieren, testen und dokumentieren
4. Dokumentation
 - 4.1. Dokumentation
 - 4.2. Präsentation

C Schaltplan

D Quellcodes

D.1 Spice-Modell: LED

```
.MODEL SFH4550 D
+ IS    = 7.06922E-19
+ N      = 1.381203754
+ RS     = 0.775721745
+ IKF    = 0.341537995
+ IBV    = 0.00000003
+ NBV    = 2
+ IBVL   = 0.000000003
+ NBVL   = 150
+ BV     = 14
+ CJO    = 4.19E-11
+ VJ     = 0.23
+ M      = 0.07
+ FC     = -0.33
+ TT     = 0.000000009
+ EG     = 1.46
```

D.2 Spice-Modell: Transistor

```
.MODEL BS170 VDMOS
+ VTO    = 1.824
+ RG     = 270
+ RS     = 1.572
+ RD     = 1.436
+ RB     = .768
+ KP     = .1233
+ Cgdmax = 20p
+ Cgdmin = 3p
+ CGS    = 28p
+ Cjo    = 35p
+ Rds    = 1.2E8
+ IS     = 5p
+ Bv     = 60
+ Ibv    = 10u
+ Tt     = 161.6n
```

D.3 Messung / Plot (Arduino)

```
// Pins

#define RESET 7

#define PULSE 8

#define SENSOR A1


void setup() {

    pinMode(RESET, OUTPUT);    // Pin for discharging the capacitor

    pinMode(PULSE, OUTPUT);    // Pin for turning the LED on/off

    Serial.begin(9600);

    Serial.println("off_100ms,on_200ms,on_150ms,on_100ms,on_50ms,on_20ms,on_10ms");

}


int record(int T, bool led_on) {

    digitalWrite(RESET, HIGH);    // Reset capacitor (close bypass connection)

    delayMicroseconds(100);        // Let capacitor discharge (R = 100 Ohm, C = 100 nF => tau = 10 us)

    if (led_on)

        digitalWrite(PULSE, HIGH);    // Turn LED on!

    digitalWrite(RESET, LOW);    // Start integration (open bypass connection)

    delay(T);                    // Integrate for T milliseconds

    if (led_on)

        digitalWrite(PULSE, LOW);    // Turn LED off!

    return analogRead(SENSOR);    // Record sensor measurement

}


void loop() {

    char str[64];

    sprintf(str, "%d,%d,%d,%d,%d,%d", record(100, false), record(200, true), record(150, true),

        record(100, true), record(50, true), record(20, true), record(10, true));

    Serial.println(str);

}
```

D.4 Auswertung und Grafiken (Python)

```
import numpy as np

import matplotlib as mpl

import matplotlib.pyplot as plt

from matplotlib.ticker import EngFormatter

import seaborn as sns

import pandas as pd

from scipy.interpolate import interp1d

from scipy.signal import convolve

from scipy.optimize import minimize

from sklearn.linear_model import LinearRegression


# Configuration

sns.set()

plt.rcParams.update({

    'figure.figsize': (4*1.8, 3*1.8),

    'figure.dpi': 300,

    'font.family': 'serif',

    'font.serif': ['Computer Modern Roman'],

    'text.usetex': True,

    'text.latex.preamble': [

        r'\usepackage[T1]{fontenc}',

        r'\usepackage{siunitx}',

        r'\usepackage{physics}',

        r'\usepackage{amsmath}'

    ]

})


## Resistor and operating points

# Load spice data

ledvi = pd.read_csv('led_v_i.txt', '\t').values

afk_on = pd.read_csv('fet_id_uds_ugs5.txt', '\t').values

afk_off = pd.read_csv('fet_id_uds_ugs0.txt', '\t').values

operation = pd.read_csv('operation.txt', '\t').values


# Compute "off" operating point

led_v = interp1d(ledvi[:, 0], ledvi[:, 1])

fet_id = interp1d(afk_off[:, 0], afk_off[:, 1])

fet_v = 5

i_d = 0

points = np.zeros((4, 2))

print("Arbeitspunkt-Iteration:")

for i in range(4):

    print(f"{i_d:.5e} {5 - fet_v:.5e} {fet_v:.5e}")

    points[i, :] = np.r_[fet_v, i_d]

    i_d = fet_id(fet_v)

    fet_v = 5 - led_v(i_d)


# Plot LED V-I diagram and operating points

fig, ax = plt.subplots(constrained_layout=True)
```

```

ax.semilogx(ledvi[:, 0], ledvi[:, 1], label="U-I-Kennlinie der Diode (SPICE-Simulation)")
ax.plot([0.1], [1.495], 'go', label=r'Arbeitspunkt EIN $\text{\SI{100}{\mA}}, \text{\SI{1.495}{\V}}$')
ax.plot(points[-1][1], 5 - points[-1][0], 'ro', label=r'Arbeitspunkt AUS $\text{\SI{34.4}{\nA}}, \text{\SI{0.879}{\V}}$')
ax.set(xlabel='$I_D$ / A', ylabel='$U_D$ / V')
ax.set_xticks(10.*np.arange(-10, 1))
ax.xaxis.grid(True, which='minor', linewidth=.3)
ax.xaxis.set_minor_locator(mpl.ticker.LogLocator(base=10., subs=(.1, .2, .3, .4, .5, .6, .7, .8, .9), numticks=12))
ax.xaxis.set_major_formatter(EngFormatter(places=0, sep=""))
ax.legend()
fig.savefig('AP.png')

# Plot FET output curve and operating points / line
fig, ax = plt.subplots(constrained_layout=True)
ax.plot(afk_on[:, 0], afk_on[:, 1]*1000, label=r"Ausgangskennlinie des FET bei $U_{GS} = \text{\SI{5}{\V}}$ (SPICE-Sim.)")
p = 100 + (100 - points[-1, 1]) / (points[-1, 0] - 0.58) * 0.58
ax.plot([0, points[-1, 0]], [p, points[-1, 1]], label='Arbeitsgerade')
ax.plot([0.58], [100], 'o', label=r'Arbeitspunkt EIN $\text{\SI{0.58}{\V}}, \text{\SI{100}{\mA}}$')
ax.plot(points[-1, 0], points[-1, 1], 'o', label=r'Arbeitspunkt AUS $\text{\SI{4.12}{\V}}, \text{\SI{34.4}{\nA}}$')
ax.set_xticks(list(range(6)) + [0.58])
ax.set(xlabel='$U_{DS}$ / V', ylabel='$I_D$ / mA')
ax.legend(loc='upper left')
fig.savefig('AKF.png')

# Plot switching characteristic
fig, ax = plt.subplots(constrained_layout=True)
ax.plot(operation[:, 0], operation[:, 1]*1000, label="SPICE-Simulation")
ax.plot([5], [90.67], 'go', label='Arbeitspunkt EIN')
ax.plot([0], [0], 'ro', label='Arbeitspunkt AUS')
ax.set_yticks(list(range(0, 90, 20)) + [90.67])
ax.set(xlabel='$U_{GS}$ / V', ylabel='$I_D$ / mA')
ax.legend()
fig.savefig('sim.png')

# Plot iteration
fig, ax = plt.subplots(1, 2, constrained_layout=True)
ax[0].semilogx(5 - ledvi[:, 1], ledvi[:, 0], label="$I_D$ (LED)")
ax[0].semilogx(afk_off[:, 0], afk_off[:, 1], "C2", label="$I_D$ (FET)")
ax[0].step(points[:, 0], points[:, 1], 'C3', label="Iteration")
ax[0].set_xticks(np.arange(0, 6))
ax[0].set_yticks(10.*np.arange(-10, 1))
ax[0].yaxis.grid(True, which='minor', linewidth=.3)
ax[0].yaxis.set_minor_locator(mpl.ticker.LogLocator(base=10., subs=(.1, .2, .3, .4, .5, .6, .7, .8, .9), numticks=12))
ax[0].yaxis.set_major_formatter(EngFormatter(unit="A", places=0))
ax[0].set_xlabel("$U_{DS} = \text{\SI{5}{\V}} - U_{\text{\mathrm{LED}}}$")
ax[0].legend(loc='upper left')
ax[1].plot(5 - ledvi[len(ledvi)//2, 1], ledvi[len(ledvi)//2, 0], label="$I_D$ (LED)")
ax[1].plot(afk_off[len(afk_off)//2, 0], afk_off[len(afk_off)//2, 1], "C2", label="$I_D$ (FET)")
ax[1].step(points[:, 0], points[:, 1], 'C3', label="Iteration")
ax[1].yaxis.set_major_formatter(EngFormatter(unit="A", places=0))
ax[1].set_xlim(4.11, 4.13)
ax[1].set_ylim(3e-8, 4.5e-8)

```

```

ax[1].set_xlabel("$U_{DS} = \SI{5}{\V} - U_{\mathrm{LED}}$")
ax[1].legend(loc='upper right')
fig.savefig('iteration.png')

## Reference measurement
# Import and process MIREX data
reference = pd.read_csv('nephelometer_002.dat', '\t;', index_col=0, skiprows=19, engine='python')
reference = reference.apply(lambda c: c.str.replace(',', '.')).astype(float).rename(columns=lambda c: c.strip())
reference.index = pd.to_datetime(reference.index) - pd.to_datetime(reference.index[0])
reference = reference.resample('500ms').interpolate('time')
reference.index.name = None
reference['sec'] = reference.index.values.astype(float) * 1e-9

# Import and process nephelometer data
nephelometer = pd.read_csv('messung.txt', '\ |,', index_col=0, engine='python').iloc[:, 1:] # Import data
nephelometer.index = (pd.to_datetime(nephelometer.index) # Normalize timestamps
                     - pd.to_datetime(nephelometer.index[0]))
nephelometer = nephelometer.resample('500ms').ffill() # Upsample for synchronous data
nephelometer.index.name = None
nephelometer['sec'] = nephelometer.index.values.astype(float) * 1e-9

# The relation between integration times is not precisely linear (due to arduino program)
def scale(data_from, data_to):
    return LinearRegression(fit_intercept=False).fit(data_from[:, np.newaxis], data_to[:, np.newaxis]).coef_[0][0]

# Remove background measurement from measurement with LED turned on and normalize values.
for i in ['200', '150', '100', '50', '20', '10']:
    nephelometer[f'i{i}'] = (nephelometer[f'on_{i}ms']
                           - (scale(nephelometer.on_100ms, nephelometer[f'on_{i}ms'])
                              * nephelometer.off_100ms.mean()))

# Make both the MIREX and nephelometer measurements the same length
reference = reference[:len(nephelometer)]
assert len(reference) == len(nephelometer)

# Plot measurements
fig, ax = plt.subplots(1, 2, constrained_layout=True)
reference.plot(x='sec', y='MIREX', ax=ax[0])
ax[0].set_xlabel("Zeit [s]")
ax[0].set_ylabel("Dämpfung [dB]")
nephelometer.plot(x='sec', y=['on_200ms', 'on_150ms', 'on_100ms', 'on_50ms', 'on_20ms', 'on_10ms', 'off_100ms'],
                 ax=ax[1])
ax[1].legend([r'\texttt{on_200ms}', r'\texttt{on_150ms}', r'\texttt{on_100ms}', r'\texttt{on_50ms}',
             r'\texttt{on_20ms}', r'\texttt{on_10ms}', r'\texttt{off_100ms}'])
ax[1].set_xlabel("Zeit [s]")
ax[1].set_ylabel("Messwert [0-1023]")
fig.savefig('messung.png')

# Plot shifted measurements
fig, ax = plt.subplots(2, 3, sharex=True, constrained_layout=True)

# Lighter colors

```

```

c = np.zeros((6, 3))

for i in range(6):

    x = mpl.colors.rgb_to_hsv(mpl.colors.to_rgb(f"C{i}"))

    x[1] **= 3

    x[2] **= 1/3

    c[i, :] = mpl.colors.hsv_to_rgb(x)

nephelometer.plot(x='sec', y='on_200ms', c=c[0, :], style='.', ax=ax[0, 0],
                  title=r"$T = \SI{200}{\ms}$", label="Original", alpha=1)

nephelometer.plot(x='sec', y='on_150ms', c=c[1, :], style='.', ax=ax[0, 1],
                  title=r"$T = \SI{150}{\ms}$", label="Original", alpha=1)

nephelometer.plot(x='sec', y='on_100ms', c=c[2, :], style='.', ax=ax[0, 2],
                  title=r"$T = \SI{100}{\ms}$", label="Original", alpha=1)

nephelometer.plot(x='sec', y='on_50ms', c=c[3, :], style='.', ax=ax[1, 0],
                  title=r"$T = \SI{50}{\ms}$", label="Original", alpha=1)

nephelometer.plot(x='sec', y='on_20ms', c=c[4, :], style='.', ax=ax[1, 1],
                  title=r"$T = \SI{20}{\ms}$", label="Original", alpha=1)

nephelometer.plot(x='sec', y='on_10ms', c=c[5, :], style='.', ax=ax[1, 2],
                  title=r"$T = \SI{10}{\ms}$", label="Original", alpha=1)

nephelometer.plot(x='sec', y='i200', style='C0.', ax=ax[0, 0], title=r"$T = \SI{200}{\ms}$", label="Verschoben")

nephelometer.plot(x='sec', y='i150', style='C1.', ax=ax[0, 1], title=r"$T = \SI{150}{\ms}$", label="Verschoben")

nephelometer.plot(x='sec', y='i100', style='C2.', ax=ax[0, 2], title=r"$T = \SI{100}{\ms}$", label="Verschoben")

nephelometer.plot(x='sec', y='i50', style='C3.', ax=ax[1, 0], title=r"$T = \SI{50}{\ms}$", label="Verschoben")

nephelometer.plot(x='sec', y='i20', style='C4.', ax=ax[1, 1], title=r"$T = \SI{20}{\ms}$", label="Verschoben")

nephelometer.plot(x='sec', y='i10', style='C5.', ax=ax[1, 2], title=r"$T = \SI{10}{\ms}$", label="Verschoben")

ax[0, 0].set_ylabel("Messwert [0-1023]")

ax[1, 0].set_ylabel("Messwert [0-1023]")

for a in ax.flat:

    a.yaxis.grid(True, 'minor', linewidth=.3)

    a.yaxis.set_minor_locator(mpl.ticker.MultipleLocator(1))

    a.yaxis.set_major_formatter(mpl.ticker.FormatStrFormatter('%d'))

    a.set_xlabel("Zeit [s]")

    a.legend(fontsize=8, loc='upper left')

fig.savefig('verschiebung.png')

# Window function
def window(T=40, alpha=1):

    w = np.r_[np.linspace(0, 1, T)**alpha, T*[0]]

    return w / w.sum() # Normalization

assert np.allclose([window(alpha=a).sum() for a in [.2, 1, 5]], [1]*3) # Test normalization

# Smoothing function
def conv(data, T=40, alpha=1, pad=True):

    win = window(T, alpha) # Window function

    res = np.r_[T*[np.nan], convolve(data, win[::-1], mode='same')][T:] # Convolution

    if pad:

        res = np.r_[T*[data[T].mean()], res[T:]] # Pad beginning

    return res

```



```

# Plot window functions and smoothed measurements
fig, ax = plt.subplots(1, 2, constrained_layout=True)

ax[0].plot(np.linspace(-20, 20, 80), window(alpha=5), 'C1', label=r'$\alpha = 5$')      # almost f (dirac)
ax[0].plot(np.linspace(-20, 20, 80), window(alpha=1), 'C2', label=r'$\alpha = 1$')    # linear
ax[0].plot(np.linspace(-20, 20, 80), window(alpha=.2), 'C3', label=r'$\alpha = 0.2$')  # almost mean (boxcar)
ax[0].set_xlabel(r"Zeit [s]")
ax[0].legend()

ax[1].plot(nephelometer.sec, nephelometer.i200.values, '.', alpha=.5, label='Messwerte ($T = \text{SI}\{200\}\{\text{ms}\}$)')
ax[1].plot(nephelometer.sec, conv(nephelometer.i200.values, alpha=5))
ax[1].plot(nephelometer.sec, conv(nephelometer.i200.values, alpha=1))
ax[1].plot(nephelometer.sec, conv(nephelometer.i200.values, alpha=.2))
ax[1].set_xlabel(r"Zeit [s]")
ax[1].legend()

fig.savefig('glatt.png')

# Find parameters using the 200ms curve to get the cleanest fit.
neph = conv(nephelometer.i200.values)
ref = reference['MIREX'].values

# Optimization to find actual parameters
t = np.arange(len(neph))
f_ref = interp1d(t, ref, bounds_error=False)
f_neph = interp1d(t, neph, bounds_error=False)

f_model = lambda t, a, b, c: c*f_neph(t - a) + b

def error(x):
    E = (f_ref(t) - f_model(t, x[0], x[1], x[2])) ** 2      # Least squares error
    E[:200] *= 2                                           # Especially penalize bias at 0
    return E[~np.isnan(E)].sum()

# Find optimal parameters using `error` as the objective function
x = None
for a in range(-50, 50):
    x_ = *minimize(error, np.r_[a, 0, 1], bounds=((-100, 100), (None, None), (None, None))).x,
    if x is None or error(x_) < error(x):
        x = x_

a, b, c = *x,      # Optimal parameters
print("\nGefundene Parameter:")
print(a, b, c)

# Plot fit with found parameters
fig, ax = plt.subplots(3, 1, sharex=True, constrained_layout=True)
ax[0].plot(t/2, ref, label='MIREX')
ax[0].legend()
ax[0].set_ylabel("Dämpfung [dB]")
ax[1].plot(t/2, neph, label='Messwerte ($T = \text{SI}\{200\}\{\text{ms}\}$, geglättet)')
ax[1].legend()
ax[1].set_ylabel("Messwert [0-1023]")
ax[2].plot(t/2, f_ref(t), label='MIREX')

```

```

ax[2].plot(t/2, f_model(t, a, b, c), label='Messwerte (Verschoben und Skaliert)')
ax[2].legend()
ax[2].set_ylabel("Dämpfung [dB]")
ax[2].set_xlabel("Zeit [s]")
fig.savefig('optim.png')

# Rescale parameters for T = 100ms
k = c*scale(nephelometer.i100.values, nephelometer.i200.values)
b = b
t0 = a

print(f"\nParameters for 100ms integration:\nScale factor: k = {k}\nOffset: b = {b}\nTime delay: t0 = {t0}"
      "\nf(t) [dB] = k*m(t - t_0) + b")

# Test model on all measurements
def model(data):
    f = interp1d(t, conv(scale(data, nephelometer.i200.values)*data), bounds_error=False)
    return lambda t, a, b, c: c*f(t - a) + b

# Plot results
fig, ax = plt.subplots(1, 2, sharey=True, constrained_layout=True)
ax[0].plot(t, f_ref(t), label='MIREX')
ax[0].plot(t, model(nephelometer.i200.values)(t, a, b, c), alpha=.3, label=r'$\SI{200}{ms}$')
ax[0].plot(t, model(nephelometer.i150.values)(t, a, b, c), alpha=.3, label=r'$\SI{150}{ms}$')
ax[0].plot(t, model(nephelometer.i100.values)(t, a, b, c), alpha=.3, label=r'$\SI{100}{ms}$')
ax[0].plot(t, model(nephelometer.i50.values)(t, a, b, c), alpha=.3, label=r'$\SI{50}{ms}$')
ax[0].plot(t, model(nephelometer.i20.values)(t, a, b, c), alpha=.3, label=r'$\SI{20}{ms}$')
ax[0].plot(t, model(nephelometer.i10.values)(t, a, b, c), alpha=.3, label=r'$\SI{10}{ms}$')
ax[0].legend()
ax[1].plot(t, f_ref(t), label='MIREX')
ax[1].plot(k*nephelometer.i100.values[int(-t0):] + b, '.', # Result without smoothing
           alpha=.5, label=r'Messwerte ($T = \SI{100}{ms}$)')
ax[1].plot(k*interp1d(t, conv(nephelometer.i100.values,
                               bounds_error=False)(t - t0) + b, label='Geglättete Messwerte')
           # Result with smoothing

ax[1].legend()
ax[0].set_ylabel("Dämpfung [dB]")
ax[0].set_xlabel("Zeit [s]")
ax[1].set_xlabel("Zeit [s]")
fig.savefig('final.png')

# Scale factors (lambda-factors)
sizes = [200, 150, 100, 50, 20, 10]
scales = np.full((6, 6), np.nan)
scales2 = np.full((6, 6), np.nan)
for i, a_ in enumerate(sizes):
    for j, b_ in enumerate(sizes[i:], i):
        scales[i, j] = scale(nephelometer[f{i{a_}"}], nephelometer[f{i{b_}"}])
        scales2[i, j] = scales[i, j] / (sizes[j] / sizes[i])

# Plot scale factors
fig, ax = plt.subplots(2, 2, sharex='col', constrained_layout=True)
ax[0, 1].plot(sizes, np.r_[sizes] / 200, 'C1-', label=r'Ideal ($\frac{T_2}{T_1}$)')

```

```

ax[0, 1].plot(sizes, scales[0, :], 'C0.-', label="Tatsächlich")
ax[0, 1].text(120, 0.25, r"$T_1 = \SI{200}{\ms}$")
ax[0, 1].legend()
ax[0, 1].set_ylabel(r"$\lambda_{T_1 \rightarrow T_2}$")
ax[1, 1].plot(sizes, len(sizes)*[1], 'C1-', label="Ideal (1)")
ax[1, 1].plot(sizes, scales2[0, :], 'C0.-', label="Tatsächlich")
ax[1, 1].text(120, 1.06, r"$T_1 = \SI{200}{\ms}$")
ax[1, 1].legend()
ax[1, 1].set_ylabel(r"$\lambda_{T_1 \rightarrow T_2} \cdot \frac{T_1}{T_2}$")
ax[1, 1].set_xticks(sizes)
ax[1, 1].set_xlabel(r"$T_2$ [ms]")
im = ax[0, 0].imshow(scales, cmap='viridis', vmin=ax[0, 1].get_ylim()[0], vmax=ax[0, 1].get_ylim()[1])
cbar = fig.colorbar(im, ax=ax[0, 0])
ax[0, 0].grid(False)
ax[0, 0].set_xticks(np.arange(0, 6))
ax[0, 0].yaxis.set_major_formatter(mpl.ticker.IndexFormatter(sizes))
ax[0, 0].set_ylabel(r"$T_1$ [ms]")
im = ax[1, 0].imshow(scales2, cmap='viridis', vmin=ax[1, 1].get_ylim()[0], vmax=ax[1, 1].get_ylim()[1])
cbar = fig.colorbar(im, ax=ax[1, 0])
ax[1, 0].grid(False)
ax[1, 0].set_xticks(np.arange(0, 6))
ax[1, 0].yaxis.set_major_formatter(mpl.ticker.IndexFormatter(sizes))
ax[1, 0].xaxis.set_major_formatter(mpl.ticker.IndexFormatter(sizes))
ax[1, 0].set_xlabel(r"$T_2$ [ms]")
ax[1, 0].set_ylabel(r"$T_1$ [ms]")
fig.savefig('linear.png')

# Export 100ms data for error checking in matlab
export = nephelometer[['off_100ms', 'on_100ms']]
export.columns = ['off', 'on']
export.to_csv('reference_100ms.csv')

```

D.5 Nephelometer-Steuerung (Arduino)

```
/*
    Nephelometer Steuerung
    Bachelorprojekt Brinkmann, Eberhard, Stock 2020
*/

// Pins
#define RESET 7
#define PULSE 8
#define SENSOR A1

// Integrationszeit [ms]
#define T 100

void setup() {
    pinMode(RESET, OUTPUT);    // Pin for discharging the capacitor
    pinMode(PULSE, OUTPUT);    // Pin for turning the LED on/off
    Serial.begin(9600);
}

int record(bool led_on) {
    digitalWrite(RESET, HIGH);    // Reset capacitor (close bypass connection)
    delayMicroseconds(100);       // Let capacitor discharge (R = 100 Ohm, C = 100 nF => RC = 10 us)
    digitalWrite(RESET, LOW);     // Start integration (open bypass connection)
    if (led_on)
        digitalWrite(PULSE, HIGH);    // Turn LED on!
    delay(T);                      // Integrate for T milliseconds
    if (led_on)
        digitalWrite(PULSE, LOW);     // Turn LED off!
    return analogRead(SENSOR);        // Record sensor measurement
}

void loop() {
    // Record time to adjust delay for next measurement
    unsigned long time_start = millis();

    // Make the measurements
    int led_off = record(false);    // Background measurement
    int led_on  = record(true);     // Measurement with LED on

    // Print result to Serial
    char str[64];
    sprintf(str, "%010lu,%04d,%04d", time_start, led_off, led_on);
    Serial.println(str);

    // Repeat exactly every 500ms
    unsigned long time_diff = (millis() - time_start);
    delay(500 - time_diff);
}
```

D.6 Bedienungsprogramm (MATLAB)

```
clear all; close all; clc;

% TODO: Change port accordingly, e.g. 'COM3' or '/dev/tty.usbserial-A106PX8C'
port = '/dev/tty.usbserial-A106PX8C';
arduino = serialport(port, 9600);

% Global variables
global run cali offset_m offset_b off_count X Y c;

% Control variables
run = true;
cali = false;
offset_m = 0;
offset_b = 0;
off_count = 0;

X = []; % Matrix to store measurements
T = 40; % Amount of datapoints to use for convolution / smoothing (T/2 seconds setup time!)
Y = []; % Matrix to store smoothed values

% Window function for smoothing
w = linspace(0, 1, T);
w = w / sum(w); % Normalization, the sum is theoretically T/2, but it needs to be exact

% Parameters for 100ms integration
k = 0.07300424260104722;
b = -21.70479394467475;
t0 = -36.5046091840224; % This is the delay between our measurement and MIREX (response time in 500ms -> ca. 20s)

% Import reference data
ref = readtable('reference_100ms.csv');
ref = ref(:, 2:end);

% Setup figure with calibrate and reset button
c = uicontrol('Style', 'togglebutton', 'String', 'Calibrate', 'Callback', @calibrate, 'Position', [10 10 100 20]);
r = uicontrol('String', 'Reset Calibration', 'Callback', @reset, 'Position', [120 10 100 20]);
q = uicontrol('String', 'Stop Measurement', 'Callback', @stop, 'Position', [230 10 100 20]);
h = uicontrol('String', 'Clear History', 'Callback', @clear, 'Position', [340 10 100 20]);

while run

    % Read data from serial port
    x = strip(arduino.readline());

    disp(x);

    x = str2num(x);

    X = [X ; x];

    % Compute offset when in calibration mode
    if cali
        off_count = off_count + 1;
    end
end
```

```

offset_m = mean(X(end-off_count+1:end, 3)) - mean(ref(1:200, 2));
offset_b = mean(X(end-off_count+1:end, 2)) - mean(ref(1:200, 1));
end

% Superimpose current measurements onto reference data
subplot(3, 4, 9);
plot(ref(:, 1), '.');
yline(x(2), 'r');
if offset_b ~= 0
    yline(x(2) - offset_b, 'g');
end
xlim([0 inf])
title('off');
subplot(3, 4, 10);
histogram(ref(:, 1), 'Normalization', 'probability');
hold on;
histogram(X(:, 2), 'Normalization', 'probability');
if offset_b ~= 0
    histogram(X(:, 2) - offset_b, 'Normalization', 'probability');
end
hold off;
title('off hist');
subplot(3, 4, 11);
plot(ref(:, 2), '.');
yline(x(3), 'r');
if offset_m ~= 0
    yline(x(3) - offset_m, 'g');
end
xlim([0 inf])
title('on');
subplot(3, 4, 12);
histogram(ref(:, 2), 'Normalization', 'probability');
hold on;
histogram(X(:, 3), 'Normalization', 'probability');
if offset_m ~= 0
    histogram(X(:, 3) - offset_m, 'Normalization', 'probability');
end
hold off;
title('on hist');

% At least T measurements are needed to start evaluation
if length(X) < T
    continue;
end

% Smooth measurements
z1 = sum(X(end-T+1:end, 3)' .* w); % Smoothed measurement
z2 = sum(X(end-T+1:end, 2)' .* w); % Smoothed background
Y = [Y ; [x(1), z1, z2]]; % Store newly smoothed values

% Calculate dB values. For greater efficiency, this matrix could be

```

```

% updated row by row for every new measurement but would have to be
% recomputed every time the offset values are changed.
Z = [Y(:, 1), (Y(:, 2) - offset_m - (Y(:, 3) - offset_b))*k + b];

% Plot dB values
if length(Z) < 120 % Plot everything in one graph until 60s have passed
    subplot(3, 4, 1:8);
    plot(Z(:, 1)/1000, Z(:, 2));
    xlim([20 inf])
    title('dB')
else % After 60s plot complete history and last minute separately
    subplot(3, 4, 1:4);
    plot(Z(end-120+1:end, 1)/1000, Z(end-120+1:end, 2));
    title('Last minute [dB]')
    xlim([Z(end-120+1, 1)/1000 inf])
    subplot(3, 4, 5:8);
    plot(Z(:, 1)/1000, Z(:, 2));
    xlim([20 inf])
    title('Complete history [dB]')
end
end

function calibrate(~, ~)
    global cali off_count c;
    cali = ~cali;
    if cali
        c.String = 'Calibrating...';
        off_count = 0;
    else
        c.String = 'Calibrate';
    end
end

function reset(~, ~)
    global offset_b offset_m;
    offset_b = 0;
    offset_m = 0;
end

function stop(~, ~)
    global run;
    run = false;
end

function clear(~, ~)
    global X Y;
    X = [];
    Y = [];
end

```

Literaturverzeichnis

- [1] H.-R. Tränkler und L. M. Rindl, Sensortechnik Handbuch für Praxis und Wissenschaft, 2 Hrsg., Grünwald; Freiburg: Springer Vieweg, 2014.
- [2] Deutscher Feuerwehrverband e. V. (DFV), „feuerwehrverband.de,“ [Online]. Available: <http://www.feuerwehrverband.de/statistik.html>. [Zugriff am 26 März 2020].
- [3] Bundesamt für Strahlenschutz, „www.bfs.de,“ 31 03 2020. [Online]. Available: https://www.bfs.de/DE/themen/opt/opt_node.html.
- [4] R. M. P. R. Timothy Hitchcock, Radio-Frequency and ELF Electromagnetic Energies: A Handbook for Health Professionals, Wiley, 1950.
- [5] J. Beyerer, F. P. León und C. Frese, Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung, Springer-Verlag, 2016.
- [6] T. W. Frank Ellrich, „secupedia,“ [Online]. Available: <https://www.secupedia.info/wiki/Terahertz>. [Zugriff am 01 04 2020].
- [7] J. Schuer, „Schottky-Dioden-Mischer im Sub-Millimeter-Wellenlängenbereich für bildgebende Anwendungen,“ 2013. [Online]. [Zugriff am 01 04 2020].
- [8] T. Feldmann, „Das Rixin Magazin,“ Kunststoffhandel Rixin GmbH, 26 Mai 2017. [Online]. Available: <https://blog.rexin-shop.de/impressum-datenschutz/>. [Zugriff am 14 April 2020].
- [9] Wikibooks, „wikibooks.org,“ [Online]. Available: https://de.wikibooks.org/wiki/Tabellensammlung_Chemie/_Dichte_fester_Stoffe. [Zugriff am 14 April 2020].
- [10] K. AG, „Produktkatalog Kanya,“ [Online]. [Zugriff am 01 04 2020].

- [11] Osram Opto Semiconductors GmbH, „osram.de,“ 10 03 2020. [Online]. Available: https://dammedia.osram.info/media/resource/hires/osram-dam-5580407/SFH%204550_EN.pdf. [Zugriff am 19 03 2020].
- [12] T. Schultze, L. Sichma und M. E. Meyer, „A Smoke Detector to Prevent False Alarms in Lunar Missions by Dust Discrimination,“ Lisbon, 2020.
- [13] ON Semiconductor, „onsemi.com,“ November 2017. [Online]. Available: <https://www.onsemi.com/pub/Collateral/BS170-D.PDF>. [Zugriff am 14 April 2020].
- [14] Osram Opto Semiconductors GmbH, „osram.de,“ [Online]. Available: https://www.osram.com/ecat/Radial%20T1%203-4%20SFH%204550/com/en/class_pim_web_catalog_103489/global/prd_pim_device_2219775/. [Zugriff am 14 April 2020].
- [15] LTwiki, „ltwiki.org,“ 5 August 2016. [Online]. Available: <http://ltwiki.org/index.php?title=BS170>. [Zugriff am 14 April 2020].

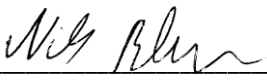
Versicherung an Eides statt

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit – mit Ausnahme der Anleitung durch die Betreuer – selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides statt nach § 156 und nach § 163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

Duisburg, 26.04.2020



Unterschrift (Nils Brinkmann)


Versicherung an Eides statt

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit – mit Ausnahme der Anleitung durch die Betreuer – selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides statt nach § 156 und nach § 163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

Duisburg, 26.04.2020



Unterschrift (Onno Eberhard)

Versicherung an Eides statt

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit – mit Ausnahme der Anleitung durch die Betreuer – selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides statt nach § 156 und nach § 163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

Duisburg, 26.04.2020



Unterschrift (Jonatan Stock)