

# EE3L11

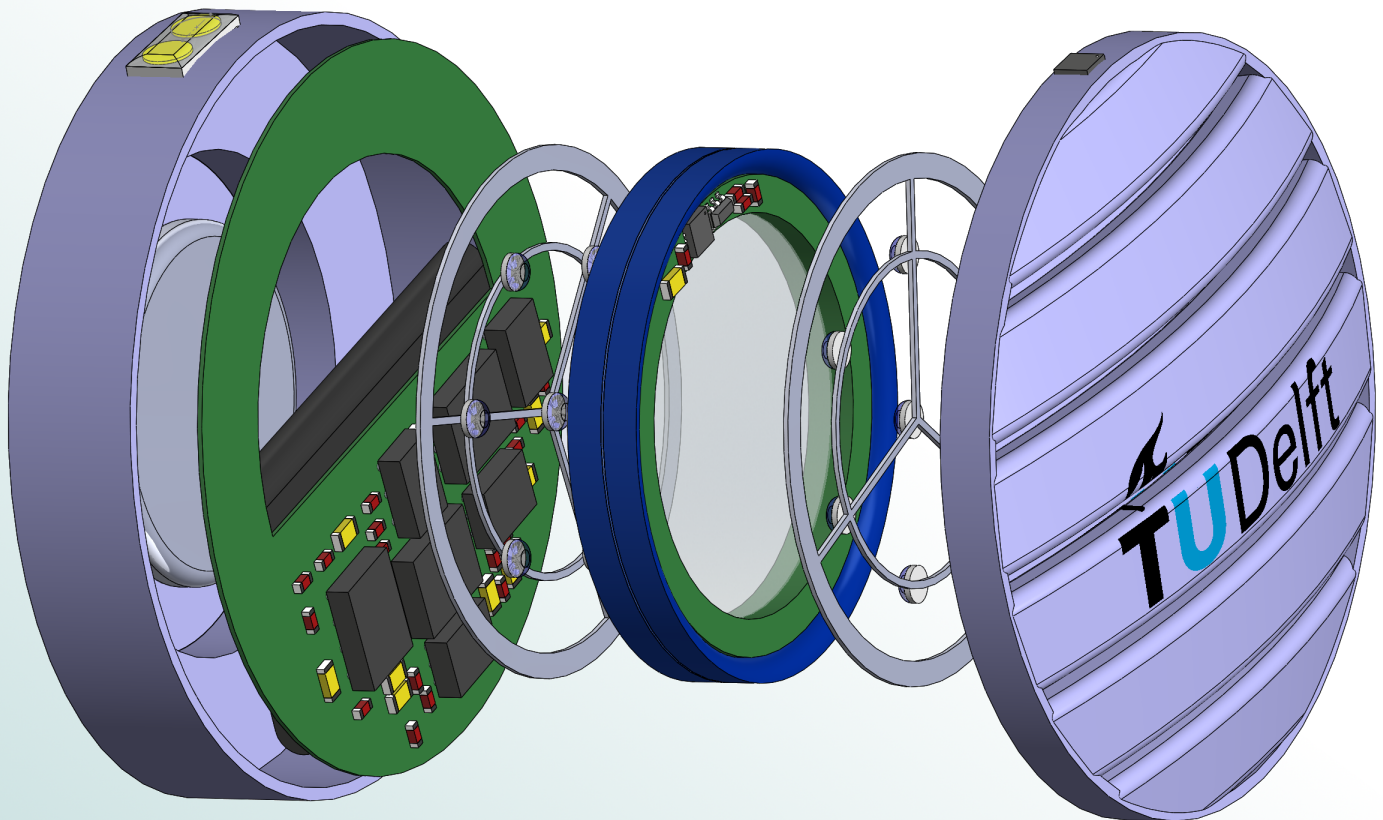
## SPPE

Smart Personal Protective Equipment

# UVGI Group

R. P. M. Bakker (4662482)

M. J. H. Brouwers (4728181)



# Smart Personal Protective Equipment

UVGI Group

by

Roy Bakker  
Marcel Brouwers

to obtain the degree of Bachelor of Science  
at the Delft University of Technology,

Authors:	R. P. M. Bakker	4662482
	M. J. H. Brouwers	4728181
Project duration:	April 20, 2020 – July 3, 2020	
Supervisor:	Dr. Ing. H. van Zeijl	
Thesis committee:	Prof. Dr. Ir. W. D. van Driel,	TU Delft
	Dr. Ing. H. van Zeijl,	TU Delft
	Dr. S. D. Cotofana,	TU Delft

With contribution from the entire SPPE group in Chapter 1.  
Bsc. Electrical Engineering 2020 BAP group C:

R. P. M. Bakker	4662482
M. J. H. Brouwers	4728181
H. J. Donkers	4475941
M. Goddijn	4666968
J. J. M. Lut	4698207
C. Zwart	4430867

# Abstract

The Smart Personal Protective Equipment (SPPE) is proposed as a result of the COVID-19 pandemic, which has led to shortages of standard face masks. This thesis describes one of the three subsystems of the SPPE, namely the Ultraviolet Germicidal Irradiation (UVGI). The UVGI subsystem provides the SPPE with an in situ disinfection system, in order to prolong the period in which the filters of the SPPE can be used to at least 8 hours.

The UVGI is implemented by the use of UV LEDs. This implementation is done in two steps. Step one is a simulation which allows for the optimization of the LED placement depending on a multitude of parameters, including: distance between the LEDs and the filter, and the LED tilt angle. The second step is the design of a driver circuit, to allow for the adjustment of the dose applied by the LEDs.

The simulation resulted in an LED array which offers the most optimal irradiation of the filter surface. The driver circuit has been designed, simulated to verify its functionality, and implemented in the form of a PCB design.

The UVGI subsystem provides the SPPE with an in situ disinfection system by delivering a base dose of 305 mJ/cm<sup>2</sup> and a driver circuit which allows for adjusting this dose, should this be desired. The UVGI subsystem should be able to extend the period in which the filters of the SPPE can be used to at least 8 hours. However, due to the restriction of not being allowed to create a prototype this has not yet been verified.

# Preface

This document contains the thesis about the Ultraviolet Germicidal Irradiation (UVGI) subsystem of the Smart Personal Protective Equipment (SPPE). This project is carried out for the course EE3L11 Bachelor Graduation project Electrical Engineering, to obtain the degree of Bachelor of Science at the Delft University of Technology.

The SPPE was not the original subject of this project but it was the subject which would be implemented, due to the restrictions put in place as a consequence of the COVID-19 pandemic, the creative idea of our supervisor, and the willingness of the group working on the project to take a challenge. The impact that the SPPE can potentially have on our society is an amazing prospect. And to have contributed to its early development truly feels like an honor.

The project has been a great learning experience, especially since UVGI is not a subject directly related to Electrical Engineering, therefore allowing for a small theoretical experience into a different field.

We want to thank our supervisor, Dr. Ing. H. van Zeijl, for his continued support throughout this Bsc. Thesis project. His comments, advice and positive state-of-mind have been a great motivation during the entire project and has helped us greatly to achieve this result. He is also responsible for designing 3D design impressions of the SPPE, which can be found within this thesis, and for this we would also like to thank him as it offers great visualizations of the product we strive to achieve.

We also like to express our gratitude to Ir. A. Montagne, who has been both a great consultant and inspiration. His advice during the implementation of our driver circuit has been a great boon for this project. And his views on Electrical Engineering has provided us with new inspiration for future projects.

Finally, we would like to thank our fellow group members from BAP group C. Huub Donkers, Michael Goddijn, Jasper-Jan Lut, and Lotte Zwart have all been superb during the entire project in their enthusiasm, energy, feedback, and commitment to this project. They have all provided a great drive to push this project to the best that we could make it, and we think it is safe to say that we could not have achieved this result without them.

We hope you enjoy the rest of this thesis.

Roy Bakker  
Marcel Brouwers  
June, 2020

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Preface</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The SPPE project overview . . . . .	2
1.1.1 Design scope . . . . .	2
1.1.2 Division in submodules . . . . .	3
1.1.3 Submodule interconnections . . . . .	4
1.2 State of the art analysis . . . . .	4
1.2.1 The current use of UVGI . . . . .	5
1.2.2 Personal Protective Equipment . . . . .	5
1.2.3 Alternative Disinfection Treatments . . . . .	6
1.3 Problem definition . . . . .	6
1.4 Thesis synopsis . . . . .	7
<b>2 Program of Requirements</b>	<b>8</b>
2.1 Requirements for the SPPE . . . . .	8
2.2 Requirements for the UVGI . . . . .	9
<b>3 Introduction into UVGI</b>	<b>10</b>
3.1 UVGI . . . . .	10
3.1.1 The working of UVGI . . . . .	10
3.1.2 Effect of environmental variables . . . . .	12
3.1.3 Structural degradation . . . . .	12
3.2 Safety risks . . . . .	12
3.2.1 Generation of harmful particles . . . . .	13
3.2.2 ISO/IEC safety regulations . . . . .	13
<b>4 LED Simulation</b>	<b>15</b>
4.1 Irradiance . . . . .	15
4.2 Irradiance on a discretized surface . . . . .	16
4.3 Irradiance of an LED structure . . . . .	18
4.4 LED tilt angle . . . . .	19
4.5 Reflectivity of filter housing . . . . .	22
4.6 Filter surface area . . . . .	24
4.7 Optimization . . . . .	25
4.7.1 UVC LED selection . . . . .	25
4.7.2 LED structure optimization theory . . . . .	26
4.7.3 LED structure optimization results . . . . .	27
<b>5 Driver Circuit</b>	<b>29</b>
5.1 Circuit implementations . . . . .	29
5.1.1 Basic transadmittance amplifier . . . . .	29
5.1.2 Transadmittance amplifier driver circuits . . . . .	31
5.1.3 Boost-mode driver circuit . . . . .	33
5.2 Circuit selection . . . . .	34
5.3 Component selection . . . . .	35
5.3.1 Assumed circuit specifications . . . . .	35
5.3.2 Passive component selection . . . . .	37
5.3.3 LED driver chip selection . . . . .	39

---

5.4	Circuit analysis . . . . .	40
5.4.1	SPICE simulations . . . . .	40
5.4.2	LED heat generation . . . . .	42
<b>6</b>	<b>Discussion, conclusions, and recommendations</b>	<b>43</b>
6.1	Discussion . . . . .	43
6.2	Conclusions. . . . .	45
6.3	Recommendations . . . . .	45
	<b>Bibliography</b>	<b>47</b>
	<b>Appendices</b>	<b>51</b>
<b>A</b>	<b>Comparison of UVC LEDs</b>	<b>51</b>
<b>B</b>	<b>PCB design</b>	<b>53</b>
<b>C</b>	<b>LTspice XVII Driver Simulation</b>	<b>57</b>
<b>D</b>	<b>Matlab Simulation and Optimization Scripts</b>	<b>59</b>
D.1	Irradiance by a single UVC LED . . . . .	59
D.2	Irradiance by a ring of UVC LED . . . . .	61
D.3	Irradiance by an UVC LED structure . . . . .	63
D.4	Optimal angle between UVC LED rings . . . . .	67
D.5	Tilted normalized intensity distribution. . . . .	69
D.6	Normalized intensity distribution: cross section . . . . .	71
D.7	Normalized intensity distribution: 3D image . . . . .	73
D.8	Irradiance reflections . . . . .	74
D.9	Irradiance uniformity . . . . .	79
D.10	Irradiance optimization. . . . .	80



# Introduction

The EE3P11 GRADUATION PROJECT forms the last course in the curriculum of the Bachelor Electrical Engineering of the Delft University of Technology. In the project, a group of six students investigates an electrical engineering challenge and develops a solution to it. Ideally, the solution is an electrical system, which is assembled into a prototype and tested by the students. However, due to the recent developments concerning the COVID-19 pandemic, the university decided that building the prototype is prohibited. Instead, the group must create a solution, i.e. a design, based on literature review and simulations.

In the time that this thesis is conducted the world is captured by the COVID-19 pandemic. The pandemic demonstrates the need for respiratory protection in health institutions to reduce the risk of infection for the workers. Particularly the shortage of face masks [1], generally designed for single use, leads to the question if microelectronics could be applied to make air filters smarter in order to extend their use and improve their protection.

To design a smart, self sterilizing air filter targeting a virus, the properties of that virus must be known. However, The virus that causes COVID-19 is a novel coronavirus, called SARS-CoV-2. As it is new to the scientific world, there is currently very limited data available. In this century, two other coronaviruses have caused epidemics. In 2003 the SARS-coronavirus was the cause of the *severe acute respiratory syndrome* (SARS) outbreak. From 2012 until present day, the MERS-coronavirus circulated and caused the Middle East respiratory syndrome (MERS) [2].

The authors aim to use the most relevant data in this thesis regarding the novel coronavirus. When we need to, we rely on information regarding the other coronaviruses, or viruses of different kinds. Which virus is considered in given data will be clearly indicated.

Bacteria, viruses and other pathogens can be killed or inactivated by *ultra violet germicidal irradiation* (UVGI) [3]. Today, several techniques are available to irradiate pathogens with ultra violet (UV) light in order to disinfect air, surfaces, and drinking water. These techniques created applications such as air filtering systems with UV lighting, cabinets in which hospitals can sterilize their face masks, and continuous overhead UV-lighting in laboratories [3]. However, none of these applications contain a small and mobile implementation of UVGI.

It is important to emphasize that the technique of UVGI sterilization can be implemented in different form factors, such as stationary applications where sterile air is crucial. Here, UVGI sources are installed as a filter module in an air conduct. These devices are beneficial in the medical and food industry.

Other domains of interest are for example military as a form of protection or water filtration systems to remove any unwanted pathogens.

We aim to design the SPPE such that its techniques are easily applied in other UVGI devices.

For this graduation project the group proposes the use of Smart Personal Protective Equipment, SPPE in short, in order to increase the effectiveness and lifetime of the filters used in standard PPE's. The SPPE is proposed as a filter module with an in-situ disinfection functionality based on UVGI. The key to the SPPE is the integration of a UV source in the filter combined with sensors to monitor the sterilisation process and

filter performance. Two SPPE modules are connected to a face mask to create a Smart Personal Protective face mask.

## 1.1. The SPPE project overview.

In this section an overview of the SPPE is provided to the reader. The aim is for the reader to obtain a clear overview of the entire project. How the project has been divided in subsystems with its mutual connections, and what are the parameters of each subsystem. For each subsystem, a separate report is written with this chapter as a general introduction. In the following chapters of this report, a detailed explanation is given on the work which has been done to tackle the respective problems of the thesis group.

### 1.1.1. Design scope

The final design of the SPPE consists of two smart filter modules which are applied to a face mask. Each module can operate autonomously and has a replaceable filter. When the SPPE is operational, the filter material is periodically irradiated with UVC light to neutralize pathogens that have been captured by the filter material. Sensors measure the air pressure, relative humidity, and temperature to compute the required irradiation intensity and monitor the quality of the filter. When the filter has reached the end of its life, the SPPE will notify the user to have it replaced. The filter module is battery powered and can operate for eight hours before needing to be recharged. Figure 1.1 shows the filter module's components implemented in a cross-sectional view.

Due to COVID-19 restrictions, a physical demonstrator of the SPPE was not realized. However, Figure 1.2 gives an impression of what the SPPE would look like.

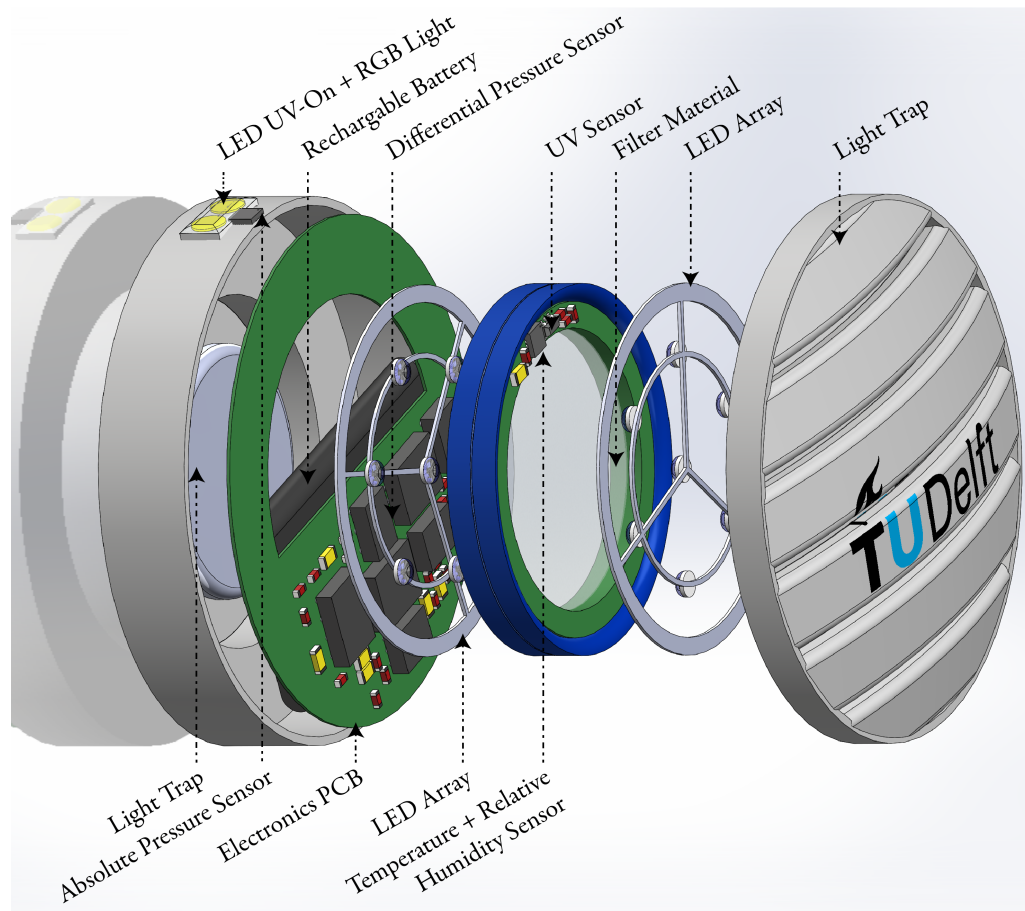


Figure 1.1: Exploded view of the SPPE. It includes the 2 air traps, electronics PCB, both LED arrays and filter material. Combined with integrated pressure, relative humidity, temperature and UV sensors.



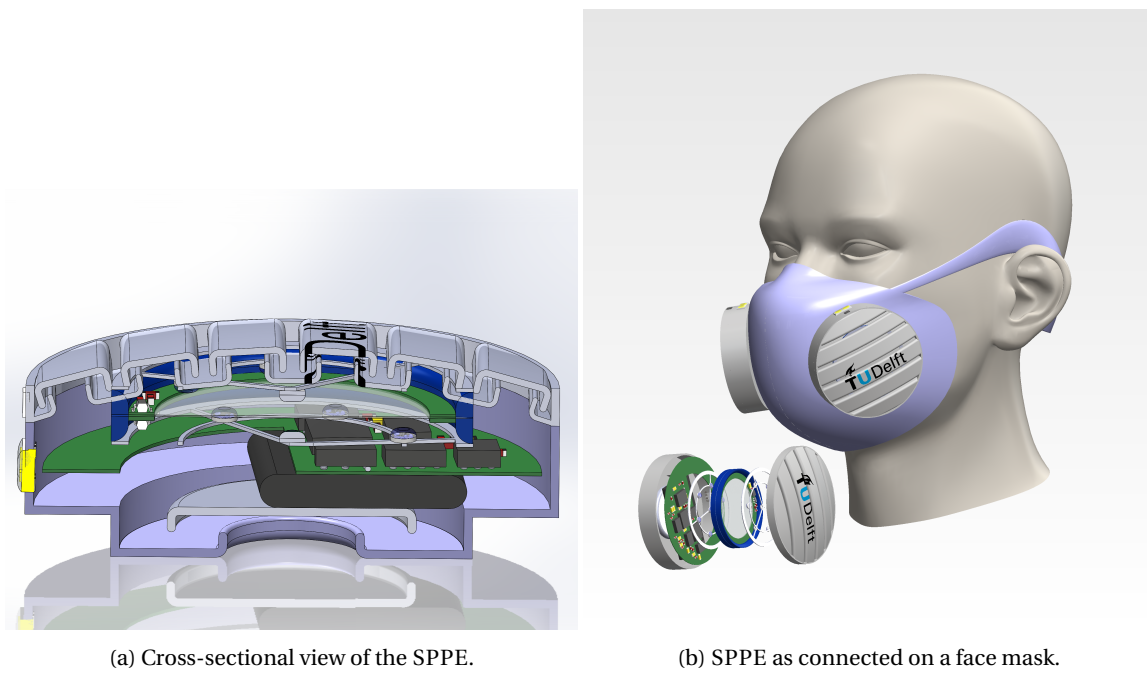


Figure 1.2: Artist impression of the SPPE.

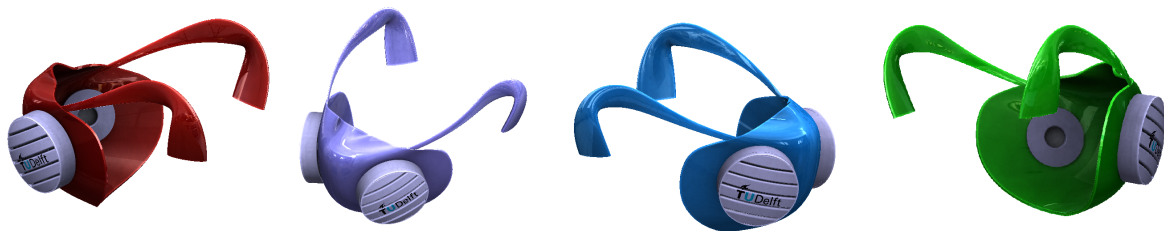


Figure 1.3: All-round view of the SPPE

### 1.1.2. Division in submodules

The graduation project group is divided into three teams consisting of two students each. Every team tackles a part of the SPPE design as described above. The teams are organised as follows:

Ultraviolet Germicidal Irradiation (UVGI)	R.P.M. Bakker M.J.H. Brouwers
Sensing and Control (SaC)	M. Goddijn J.J.M. Lut
On-Board Power Management (OBPM)	H.J. Donkers C. Zwart

Below follows a brief introduction on every subgroup.

#### Ultraviolet Germicidal Irradiation

The Ultraviolet Germicidal Irradiation group is responsible for providing the UV radiation in order to disinfect the filter material in the SPPE. This is achieved through the use of LEDs radiating in the so-called UVC spectrum (wavelengths between 200 and 280 nm). The UVGI group selects LEDs with the most effective wavelength, and ensures the most optimal placement of the LEDs to achieve uniform irradiation, with accuracy and power consumption in mind. Additionally the UVGI group will design the LED drivers. The LED drivers are controlled by signals provided by the SaC group.

### Sensing and Control

The Sensing and Control submodule is the controlling body of the SPPE. The submodule measures the temperature, relative humidity, air pressure, and the pressure drop over the filter material. It generates the control signal for the UVGI submodule, thereby regulating the intensity of the UVC radiation. Using a control loop, the Sensing and Control submodule ensures a safe (always sufficient UV intensity) and power-efficient (not an unnecessarily high UV intensity) operation of the SPPE. It also monitors the quality of the filter material.

### On-Board Power Management

The aim of the On-Board Power Management is to provide energy for the sensor circuitry and the LED modules. Since the SPPE is a wearable device, the system needs to have on-board power with an energy management system for low-power application and battery safety. This will be implemented by making use of a rechargeable battery and a battery management system that controls and protects the battery. The goal is to design a system that can supply the SPPE with required power in a safe and efficient manner.

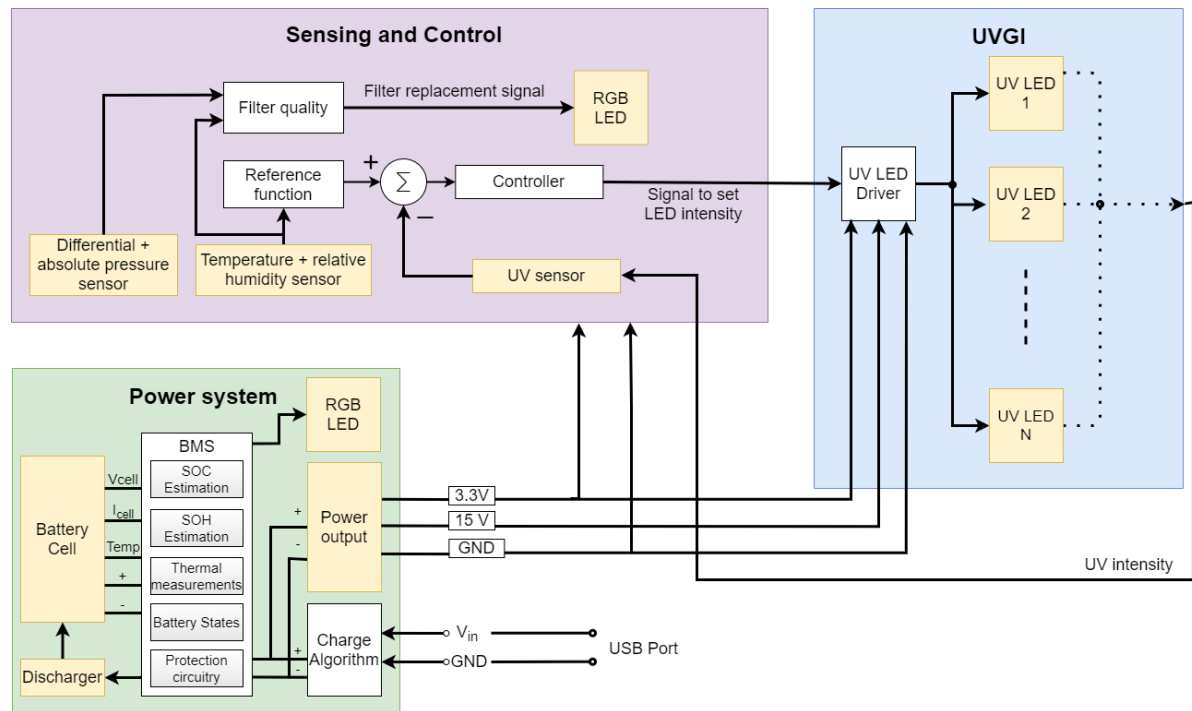


Figure 1.4: System overview divided in submodules.

### 1.1.3. Submodule interconnections

The submodules are interconnected in a physical way and by means of design constraints. Figure 1.4 shows the physical interconnections, which are the power outputs from the OBPM submodule to the other submodules, the control signal for the LEDs from the SaC module, and the feedback of UV intensity to the SaC module.

The non-physical interconnections between the modules are the design constraints. The diameter of the filter material influences the amount of LEDs needed to radiate the entire filter with an adequate dose. As one can imagine, this has an influence on how the control loop is calibrated. Using more LEDs also means that more power and higher peak currents are necessary, which influences the battery size. As the battery is the largest component, this has an influence on the filter diameter and the weight of the filter module itself. These interconnections illustrate the need for communication between the subgroups about these parameters throughout the design process. Communication is key to ensure an optimal SPPE system design.

## 1.2. State of the art analysis

People already know about the effectiveness of UVGI for a long time. In 1985, for example, there was a paper describing an experiment for comparing the effectiveness of UV light between inactivating various microorganisms [4]. A full introduction into UVGI will be given in Chapter 3, here an outline of the current applica-

tions of UVGI is given (Section 1.2.1). And, since the UVGI will be applied to a personal protective equipment (PPE), there will be a brief overview of different PPE's and their effectiveness (Section 1.2.2). Finally some alternative treatments to UVGI will be discussed (Section 1.2.3).

### 1.2.1. The current use of UVGI

The use of UVGI spans a wide range of applications. For example, it is used in air disinfection systems [5] [6], cleaning chambers [7] and room cleaners [8]. Recently, due to the COVID-19 pandemic, there are also implementations made of UVGI in biosafety cabinets (BSCs) [9]. The application that approximates the SPPE the closest is UVGI Sanitizing Cabinets [10]. This application can be compared to a cleaning chamber, but exclusively for face masks. The application has a small form factor and can fit on a normal table, but it is not a wearable application. In that regard the SPPE appears to be unique.

UVGI is commonly applied by the use of mercury vapor lamps, which emit UV radiation centred around 254 nm [3]. However, the development of LEDs over the last years has enabled UVC light sources with a small form factor and high energy efficiency to trigger new applications. There have been numerous studies done to confirm that UV LEDs are at a point in their development where they can be used for germicidal purposes [11] [12] [13]. Moreover, UV LEDs offer better control over the radiation spectrum which is emitted. For example, UV LEDs can emit around the optimum wavelength for germicidal effectiveness, which is around 265 nm [3] which can also be seen in Figure 1.5.

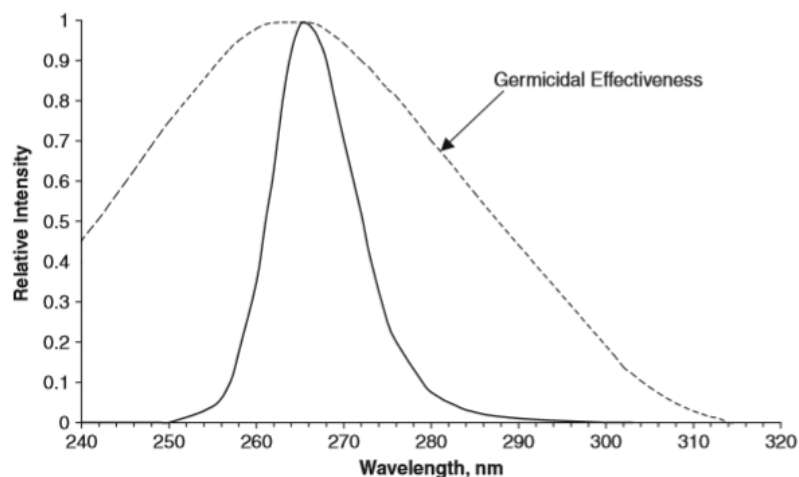


Figure 1.5: Spectral output of a nominal 265 nm LED superimposed on the germicidal effectiveness curve [3].

### 1.2.2. Personal Protective Equipment

Wearing personal protective equipment is usually not a standard used by the general public. However, especially during pandemics the effectiveness of PPE, or face masks, is once again evaluated. When considering the COVID-19 pandemic, it is reported that wearing a mask alone already reduces risk of infection by 68% [14]. This effectiveness in turn leads to a high demand in PPE, so high that shortages are starting to appear in the supply of PPE [1].

Normally there are strict rules regarding the categorization of PPE and when to use them. One common occurrence is the name of N95 respirator, which is an American standard. The N95 standard states that at least 95% of aerosol particles having a diameter equal or greater than 0.3 microns are removed [14]. The equivalent European standard FFP2 removes a minimum of 94% [15]. Both types are built for optimal fit around the mouth and nose, which should create a tight facial seal [14].

A different facial mask, which might sound more familiar, is the surgical mask. Contrary to common believe, the surgical mask was never meant to protect the wearer. Instead it was meant to protect the patient in case the surgeon sneezed or coughed [14]. Additionally the surgical mask does not offer a tight fit, leaving the wearer open for air flow from all sides of the mask [14].

Based on these findings it might seem obvious to only use N95 type masks, however there have been at least two independent studies comparing the effectiveness of N95 and surgical masks [16] [17]. Both studies confirm that there is no significant difference in preventing influenza among healthcare personnel with the use

of either N95 or surgical masks. Thus it would seem that both masks can be used.

Despite both masks being a valid option for use, thereby increasing the supply of PPE, both surgical masks and N95 respirators are intended for a one-time-use, after which they should be discarded [14]. This leads back to the previous comment about shortages and shows the need for the introduction of the SPPE, as it is intended for extended use.

### 1.2.3. Alternative Disinfection Treatments

Before the start of the design of the SPPE, multiple treatment methods were compared in order to validate the use of UV radiation as a disinfection method in the SPPE. Though not directly related to the rest of the design, it is deemed informative to present alternatives to the use of UV radiation.

#### Comparison studies

Some studies have compared UV irradiation with other possible treatments in terms of effectiveness and side-effects. One such study made a comparison between UV germicidal irradiation (UVGI), micro-wave generated steam, and moist heat [18]. The researchers concluded that the most effective treatment is UVGI. A different study discusses UVC radiation (260-285 nm); 70 °C heat; 70% ethanol; and vaporized hydrogen peroxide [19]. Note that this study is not yet peer-reviewed, but due to the COVID-19 pandemic it is deemed a useful source of information. Both the application of ethanol and the application of vaporized hydrogen peroxide resulted in rapid inactivation of SARS-CoV-2 on N95 masks. In comparison, UV and heat inactivated the same virus at a slower rate.

However, the experiment also included measurements on the filtration performance after each decontamination. It was observed that the ethanol treatment caused a significant drop in filtration performance after the first decontamination round, making it unsuitable for sustainable use. The same result applied to the heat treatment, although being applicable to a lesser degree. In contrast, the UV and vaporized hydrogen peroxide methods retained their filtration performance after two decontamination treatments, and maintained a "passable" performance even after three rounds. Vaporized hydrogen peroxide can cause serious health issues in case of overexposure and it is much harder to introduce safety measures for a portable application compared to UVGI. It is therefore not deemed a useful candidate for the SPPE.

#### Heat treatment

Heat treatment of the filters of a PPE seems to be the best alternative to UVGI. Therefore some additional research was done into this particular treatment. However, heat introduces serious issues when it is considered as an active sterilization technique in a portable application. First of all, the temperatures needed to effectively neutralize viruses exceed 56 °C for neutralization levels higher than 50% [20]. Other papers go even higher with 70 °C for a prolonged period of time [19] [21]. Such temperatures are not deemed safe with regards to the proximity to a persons face. Additionally, heat seems to need relatively long amounts of time before a virus is rendered inactive. For heat treatment the duration is in the region of hours, compared to minutes for UVGI, which can be found in Figure 2 and Figure 3 of [22] (page 172 and 173 respectively). This could seriously compromise the power consumption of the SPPE, reducing the increased duration for which a PPE can be worn.

Due to these drawbacks and the indication that UVGI is more effective at neutralizing viruses, heat treatment is not deemed a useful candidate for the SPPE and the rest of this thesis will focus on the implementation of UVGI.

## 1.3. Problem definition

This thesis focuses on the implementation of the UVGI subsystem. As was discussed in the previous section, the conventional application of UVGI employed mercury vapor lamps in big installations like cleaning chambers. Due to the recent improvements to LEDs, it is possible to employ UV LEDs for UVGI, thereby decreasing the form factor and allowing for more mobile applications like the UVGI Sanitizing Cabinets. In this thesis the possibility to implement UVGI in a wearable device is explored, in the form of the SPPE.

The SPPE is introduced due to the recent shortages of PPE as a consequence of the COVID-19 pandemic. These shortages demonstrate that a more sustainable use of PPE is needed. The SPPE offers this, by extending the usable period of a single filter, with the use of UVGI, and allowing it to be replaced after a full day of work. This thesis offers an implementation of the UVGI sterilization for a filter module in the SPPE.

The UVGI subsystem is designed based on uniform radiation of a filter surface area and minimal energy consumption while doing so. For the optimization a simulation is created in which the irradiation can be analyzed. Additionally, a driver circuit has been designed to ensure the LEDs irradiate the correct intensity. The main goal of this thesis is defined as follows:

*Designing an energy efficient implementation of UVGI for application in a wearable device.*

## **1.4. Thesis synopsis**

This thesis, about the UVGI subsystem, is divided into a number of chapters. To start of, the requirements set for the subsystem will be presented in Chapter 2. Since UVGI is not a standard part of Electrical Engineering there will be an introduction into UVGI in Chapter 3, which also includes safety precautions and regulations. Once the workings of UVGI are known, the main application for which it will be used is introduced in the form of a UVGI irradiation simulation discussed in Chapter 4. This application in turn needs to receive the correct input value, or in other words, it needs to be driven. This results in a driver circuit design, which is covered in Chapter 5. Finally, a discussion about the the main results of the thesis and the consequent conclusions following from this are discussed in Chapter 6. This chapter also includes recommendations for future work.

# 2

## Program of Requirements

The Smart Personal Protective Equipment is meant to be a face mask which is actively being disinfected by the use of in situ UVGI. The SPPE will consist of two filter modules, each being a separate package. This means that each filter module will have its own control circuit, power supply and UVGI installation. A thorough explanation of the SPPE can be found in Chapter 1.

The SPPE is meant to be used primarily by healthcare personnel, as they are currently suffering from shortages of normal PPE. These shortages of PPE forces people to use their PPE for prolonged periods of time and re-use it after it has been cleaned, for which they are not designed.

Key requirements for the SPPE design are: re-usability, power efficiency and safety. This chapter outlines the requirements for the SPPE as a whole (Section 2.1), to which all subsystems should comply. Additionally this chapter will present the requirements set for the UVGI subsystem (Section 2.2).

### 2.1. Requirements for the SPPE

The SPPE is one thesis-project divided into three separate subsystems. However, the SPPE does have objectives and requirements which should be met by every subsystem, in order to produce a satisfactory result. Therefore, the objectives and requirements for the SPPE as a whole will be taken into account first. The general design objectives for the SPPE are as follows:

#### SPPE design objectives

- The SPPE should be safe for use by humans.
- The SPPE should be designed power efficiently.
- The SPPE should be designed as light as possible.
- The SPPE is designed as in-depth as possible without creating prototypes.

Additionally there are general design requirements for the SPPE:

#### SPPE design requirements

- The SPPE uses UVGI to sterilize the filter material.
- The SPPE's mechanical filter is replaceable.
- The SPPE's electronics are reusable after the mechanical filter is replaced.
- The SPPE's battery capacity is sufficient for at least eight hours.
- The SPPE is designed preferably using off-the-shelf components, unless it is absolutely necessary.
- The SPPE's component lifetime is at least 2 years.

- The SPPE design is circular.
- The SPPE should communicate its status conditions to the user.
- The SPPE filter module should be airtight apart from the dedicated airflow pathways.
- The SPPE filter module should be small enough to fit on the side of a PPE.

## 2.2. Requirements for the UVGI

The UVGI subsystem is at the core of the SPPE, forming the active disinfection method of the filter material. The UVGI subsystem will be implemented in a portable device, which should work as efficient as possible. As a result, the performance of the UVGI will be dependent on the degree to which the key performance indicators are met. These indicators fall within the trade-off requirements and are bounded by the mandatory requirements.

### Key performance indicators

- Radiation uniformity
- Low energy consumption
- High power efficiency of the driver circuit

The mandatory requirements are distinct quantitative values which the UVGI must meet.

### Mandatory requirements

- The UVGI driver should be operational at a source voltage of 15V.
- The current drawn by the UVGI driver circuit must not exceed 500 mA when a dose is applied.
- The UVGI subsystem must provide an overdose ( $E_{\max}/E_{\min}$ ) on the filter material equal to, or lower than, 1.5.
- The UVGI subsystem must be able to deliver a UVC radiation dose of 305 mJ/cm<sup>2</sup> [3].
- The UVGI subsystem must be able to deliver said dose every 2 hours.
- The UVGI subsystem must be able to deliver said dose a minimum of 4 times a day on one operation cycle.
- The UVGI subsystem should be able to adjust the dosage based on signals originating from the Sensor and Control subsystem.
- The distance from the UVC LEDs to the filter material may not exceed 15 mm.

The trade-off requirements have no precise value, but should be as optimal as possible, without compromising the other requirements. As always, engineering is a trade-off.

### Trade-off requirements

- The UVGI subsystem should provide a radiation upon the filter surface as uniformly as possible.
- The UVGI subsystem should generate as little heat as possible.
- The UVGI subsystem should have minimal energy consumption.
- The implementation of the UVGI subsystem should have a minimal form factor.

# 3

## Introduction into UVGI

The core of the SPPE is the real-time disinfection of the internal filter by the use of ultraviolet germicidal irradiation or UVGI, thereby prolonging the time for which the PPE can effectively be used. UVGI is widely applied in the healthcare sector in the form of disinfection rooms for PPE or air filtration installations for example. This subject is however not directly related to electrical engineering and it can therefore be that information and terms used within the sector are unknown. For this reason, a brief overview will be given regarding the use of UVGI and what it actually is (Section 3.1). Additionally the possible risks that arise from the use of UV radiation will be discussed (Section 3.2).

### 3.1. UVGI

Ultraviolet germicidal irradiation is used for disinfecting air, surfaces and water. UV radiation is divided into 3 classes: UV-A (315-400 nm), UV-B (280-315 nm) and UV-C (100-280 nm) [3] [23]. UVGI is the spectrum between wavelengths of 100 and 315 nm, or UV-C and UV-B combined. Additionally, UV-C is the part of the spectrum which is the most effective to use against viruses and bacteria [3] [23] [24]. The radiation used in the SPPE falls within the UV-C spectrum and will be used to irradiate a filter surface, thereby disinfecting that surface. The attribute of disinfection is derived from the term germicidal, which implies that UVGI is able to "destroy, kill, or inactivate microorganisms such as viruses, bacteria, and fungi" [3]. Other terms which often return in literature are pathogens and allergens. These are defined as follows: "Pathogens are any microbes that cause infections in humans and animals, and these include viruses, bacteria, and fungi." [3]. "Allergens are microbes, biological products, and compounds that induce allergic reactions in atopic, or susceptible, individuals." [3]. The UV-C irradiation is expressed in the units  $W/m^2$  and the UV dose applied due to this irradiance is expressed in the units  $J/m^2$ . These are the preferred units in air and surface disinfection [3]. The irradiance and the dose can be related to each other through Equation (3.1), where  $D$  is the dose delivered to the irradiated material,  $I$  is the irradiance arriving at the irradiated material, and  $t$  is the time over which said irradiance is applied to the material.

$$D = I \cdot t \quad (3.1)$$

#### 3.1.1. The working of UVGI

UVGI is going to be used against viruses and bacteria, which are both build-up out of nucleic acids. There are two types of nucleic acid: ribonucleic acid (RNA) and deoxyribonucleic acid (DNA). "Deoxyribonucleic acid (DNA) is a large, high molecular weight macromolecule composed of subunits called nucleotides. Each nucleotide subunit has three parts: deoxyribose, phosphate, and one of four nitrogenous bases (nucleic acid bases). The four bases are thymine (T), adenine (A), cytosine (C), and guanine (G). These four bases form base pairs of either thymine bonded to adenine or cytosine bonded to guanine." [3]. "The specific sequences formed by these base pairs make up the genetic code that forms the chemical basis for heredity" [3]. In RNA the base thymine (T) is replaced with a different base called uracil (U), which bonds with adenine (A). Bacteria contain both DNA and RNA. Viruses contain DNA or RNA, but not both. The bonds between the bases of DNA and RNA consist of hydrogen bonds. UV radiation can cause a cross-link between two thymine bases that is more stable than a hydrogen bond, called thymine dimers, thereby damaging the nucleic acid. For RNA



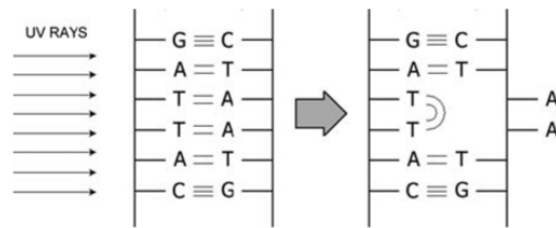


Figure 3.1: Formation of thymine dimers caused by UV absorption [3]

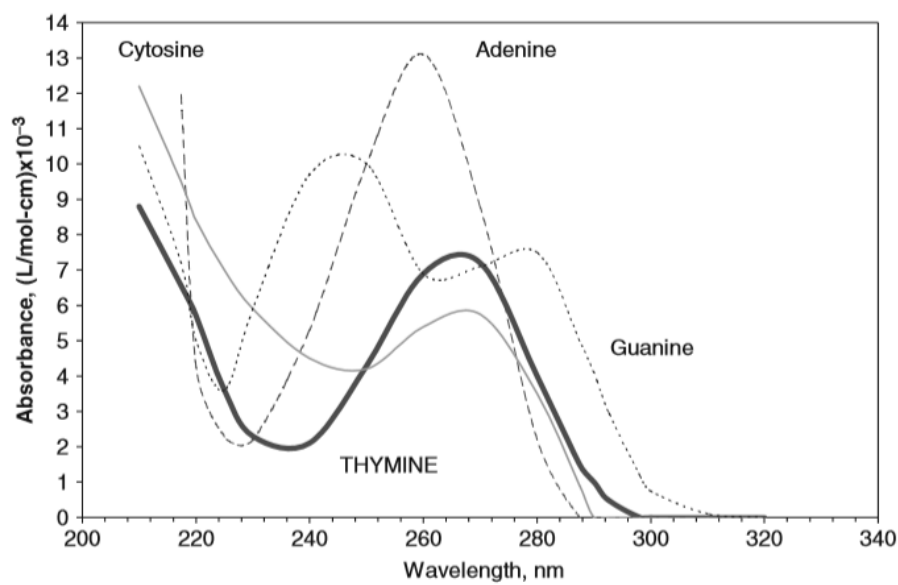


Figure 3.2: Absorption spectra of the nucleotides of DNA [3]

this effect occurs between uracil nucleotides, forming uracil dimers. Since DNA and RNA are responsible for cell replication and protein synthesis, damage to these nucleic acids results in inactivation or the failure to reproduce [3]. An image displaying the forming of the thymine dimers is displayed in Figure 3.1.

In order to give a better impression of the absorptance of UV radiation by the bases, their absorption spectra are shown in Figure 3.2. An absorption spectrum is a quantitative way to display the absorptive capacity of a molecule over an electromagnetic frequency range. In Figure 3.2 one can observe that the peaks of all nucleotides are present in the UVC region. Thymine and cytosine both have peaks in the region between 260 and 280 nm, which is the region typically used for UVGI applications. This also means that the LED selected for UVGI should radiate in the range of 260 to 280 nm.

At the moment of writing, the world is subject to the COVID-19 pandemic. COVID-19 stands for CORonaVirus Disease 2019 and is caused by the the SARS-CoV-2 virus. Viruses come in several types: ssRNA, ssDNA, dsRNA and dsDNA, where ss stands for single stranded and ds for double stranded [3] [25]. COVID-19 is a ssRNA type and this type will require the smallest irradiation intensity, compared to the other types of viruses, for disinfection purposes [3] [25]. There have been several other studies regarding the effectiveness of UV irradiation on viruses, both on surfaces and in air. All agree that UV irradiation does work very effectively, provided that the right intensity is applied and the irradiation is carried out for a sufficient amount of time. In other words, the dose applied to inactivate the virus needs to be high enough [5] [6] [19].

### 3.1.2. Effect of environmental variables

The effectiveness of UV radiation to inactivate viruses and other pathogens changes for varying environmental conditions. One such condition is air temperature. Inactivation rates of viruses increase with increasing air temperature [26], meaning that a higher ambient temperature is beneficial for the effectiveness of UV radiation.

Another metric which defines the effectiveness of UV radiation is the relative humidity. In contrast to air temperature however, relative humidity shows no linear relation to the inactivation rate. It was found that at low relative humidity (< 20%) and high relative humidity (> 80%) the inactivation rate decreases significantly. Research shows that the highest inactivation rate appears around 60% relative humidity for both viruses and bacteria [26] [27]. These two environmental variables do not receive further attention in this thesis and are instead incorporated into the Sensing and Control subsystem, described in Chapter 1.

Lastly, the effectiveness of UV radiation is inversely proportional to the distance of the UV source from the exposed surface [27], which can be of use when the distance between the filter material and the LEDs is considered, but more on this in Chapter 4.

### 3.1.3. Structural degradation

Apart from the effect of the environment on the effectiveness of UVGI, UVGI also has effects on the environment in which it is applied. UV radiation does not only disinfect the object it irradiates, but it can also cause degradation of the structural integrity of said object [28]. The different parts of PPE will be differently affected by the UV irradiation [29], but can vary based on material choice, applied dose, etc. However, as was also mentioned in Chapter 1, UVGI does not compromise the effectiveness of the filter material, until after multiple decontamination treatments [19]. Although this would have to be examined, once a prototype may actually be produced, for now the assumption will be made that the structural integrity will remain of sufficient quality.

## 3.2. Safety risks

Since the SPPE will be a medical device, safety is a primary concern during the design. UV radiation can cause harmful effects when it comes into contact with the skin. In the case of high doses it can even lead to skin cancer. As a result it is of the utmost importance that the UV radiation used by the SPPE cannot come into contact with the skin.

UV radiation has a relatively low penetration depth of only 20  $\mu\text{m}$  at a wavelength of 290 nm (varying depending on the exact location on the human body) [30]. For comparison purposes, the thickness of the outermost layer of skin is 0.1 millimeters on average [31]. An overview of skin penetration, for a part of the UV spectrum, is displayed in Figure 3.3, here the volar and dorsal aspect of the forearm are displayed as squares and circles respectively and the thenar is displayed as triangles. The range displayed is slightly outside the range of the UV radiation spectrum which is going to be used in the SPPE, but if the trend is followed to lower values for the wavelength, one can assume that the skin penetration around 270-280 nm will be even lower than 20  $\mu\text{m}$ .

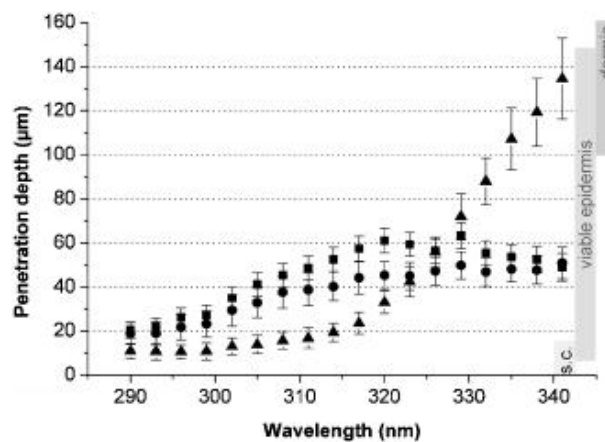


Figure 3.3: Penetration depths of UV radiation in human skin [30]

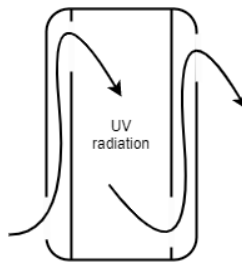


Figure 3.4: A trap structure in the filter module, to reduce the risk of UV radiation coming into contact with the skin

Despite this relative low penetration, there are still precautions within the SPPE to ensure that there will be no contact between the UV radiation and human skin. First of all, there will be a coating on the inside of the filter module, made from black lacquer paint. This paint has a reflectivity coefficient of 5% in the UVC/UVB range [3]. This means that the paint will absorb 95% of the UV radiation that arrives at its surface. Secondly, there is a structure in place in the filter module similar to the effect of a trap, such as used in sanitary drains. The basic idea is displayed in Figure 3.4, where the arrows indicate the airflow. This structure should prevent any UV radiation which is not absorbed by the filter itself, or the lacquer paint, to remain trapped inside the filter module.

### 3.2.1. Generation of harmful particles

Apart from the risk of the radiation itself, UV radiation can also generate various byproducts at certain wavelengths. Of particular interest is the generation of ozone [3] [5] [32] [33], and so-called Secondary Organic Aerosols (SOA) [34].

The ozone generation is a problem mentioned in multiple sources, already listed above. Ozone is generated by the conversion of oxygen into ozone by exposure to UV radiation. This generation of ozone occurs between wavelengths of 175 and 210 nm, which is outside the spectral range for the SPPE, as it uses wavelengths above 250 nm. Instead of generation, ozone is actually destroyed by wavelengths between 250 nm and 260 nm [32] [33].

The other byproducts generated were associated with SOA, leading to the generation of fine particulates (PM<sub>2.5</sub> or particulate with diameter less than 2.5µm) associated with negative health effects. This process is more pronounced at wavelengths between 300 nm and 400 nm, but not unheard of in 254 nm [34]. However, though it is not unheard of, it is concluded in [34] that the generation of SOA's due to UV radiation in realistic environments is not deemed to be of large impact. Therefore, it is assumed that the generation of SOA's will not cause a risk for the safety during the use of the SPPE. Should the SPPE ever be taken into production, it is advised to research the generation of SOA's due to the use of UV radiation in the SPPE and confirm the assumption made.

### 3.2.2. ISO/IEC safety regulations

The SPPE will be medical equipment. As such, there is not only the consideration of the risks that might arise from the use of UV radiation, but there are also safety regulations which should be adhered to within the design of the SPPE. Safety regulations are made on a national level, and can differ from country to country. There are however organizations which help with the globalization of such regulations. The IEC (International Electrotechnical Commission) and the ISO (International Organization for Standardization) both develop regulations which are internationally recognized and these regulations can be used as a baseline during the design process.

Of particular interest to the SPPE are the regulations produced by the IEC TC 62 committee (Electrical equipment in medical practice) and the ISO/TC 198 committee (Sterilization of health care products). The following regulations are deemed important for the design of the SPPE:

- IEC 60601-1:2020 SER Series, "IEC 60601 series is widely recognized as the fundamental safety standards for medical electrical equipment." [35].
- ISO 11137-1:2006, "Sterilization of health care products — Radiation — Part 1: Requirements for development, validation and routine control of a sterilization process for medical devices" [36].

- ISO 11137-1:2006/AMD 1:2013, "Sterilization of health care products — Radiation — Part 1: Requirements for development, validation and routine control of a sterilization process for medical devices — Amendment 1" [36].
- ISO 11137-1:2006/AMD 2:2018, "Sterilization of health care products — Radiation — Part 1: Requirements for development, validation and routine control of a sterilization process for medical devices — Amendment 2: Revision to 4.3.4 and 11.2" [36].
- ISO 11137-2:2013, "Sterilization of health care products — Radiation — Part 2: Establishing the sterilization dose" [36].
- ISO 11137-3:2017, "Sterilization of health care products — Radiation — Part 3: Guidance on dosimetric aspects of development, validation and routine control" [36].

Unfortunately, the regulations themselves are behind paywalls, which are considered too high for the purpose of this thesis, as it is only a proof of concept. Should one consider to actually take the SPPE into production, the above named regulations should be taken into account. The design steps and decisions made in this thesis should be retraced and checked for compliance to these regulations.

# 4

## LED Simulation

The main functionality of the SPPE module is to disinfect a filter material fitted inside the filter head module. As explained in the previous chapter, the preferred method of disinfection is using UVC radiation. UVC radiation can be efficiently produced in a relatively small package. Due to its advantages over other technologies (refer to Section 1.2), UVC LEDs will be used in the SPPE module. The question then arises, how many LEDs are actually needed in the module? Moreover, what are the locations of the LEDs? And which one of the many available types of UVC LEDs is suitable for this application? The answers mainly depend on the power requirements (Refer to Chapter 2). That is to say the LEDs should be structured such that power efficiency and consumption are optimized. Note that the definition of an LED structure used throughout the thesis is the physical layout of the UVC LEDs in the SPPE module. Furthermore, there are multiple constraints which have to be met by the UVGI subsystem. These include radiation uniformity and a minimum UVC irradiation over the whole area of the filter material. The entire optimization problem can be solved by performing simulations of UVC radiation upon the filter material generated by UVC LEDs. As dedicated UV ray tracing software was not readily available, Matlab code is developed to optimize the UVC LED structure.

This chapter discusses the simulation and optimization solutions, starting with the mathematics behind the irradiance of a single LED on a continuous flat surface (Section 4.1) and on a discretized flat surface (Section 4.2). This is then extended to the irradiance of multiple LEDs in the form of an LED structure (Section 4.3). Next, the tilt angle of the LEDs as additional parameter of the LED structure is explained in detail (Section 4.4). To increase efficiency, the reflectivity of the filter housing is analyzed within the simulations (Section 4.5). Finally, the last important parameter of the LED structure is analyzed, namely the filter surface area (Section 4.6). Once all these parameters have been covered, the optimization problem and its solution are explained (Section 4.7), after which the final results are discussed (Section 4.7.3).

### 4.1. Irradiance

UVC LEDs in the SPPE module are used as a means to apply a given UVC dose to the filter material, thereby inactivating pathogens on the filter material (refer to Chapter 3). The applied UVC dose, as provided in Equation (3.1), is determined by two factors: the on-time of the UVC LEDs [s] and the irradiance received by the filter [ $\text{W}/\text{m}^2$ ]. It is important to recognize that the irradiance, and therefore the dose, changes depending on the location on the filter. Because a certain minimum dose needs to be applied, the location with the minimum irradiance determines the on-time of the LEDs. Therefore, it is key to be able to determine the minimum irradiance for every LED structure. Due to the complexity of the many parameters involved in the LED structure, it becomes impossible to find an analytical solution for the minimum irradiance. For this reason the thesis aims for a numerical analysis by means of irradiation simulations.

In physics, specifically the field of radiometry, irradiance is the incident radiant flux per unit area of a surface [37] and has as unit  $\text{W}/\text{m}^2$  or  $\text{mW}/\text{cm}^2$ . The formal mathematical definition is  $E = \frac{\partial\Phi}{\partial A}$ , where  $\Phi$  is the radiant flux and  $A$  is the surface area. Equation (4.1) elaborates on this definition and gives an expression of the irradiance on an area. The denominator gives the irradiated area defined by  $r$ , the distance between the UVC LED and the surface, and the solid angle  $\Omega = \int I(\theta, \phi) d\Omega$ , where  $I(\theta, \phi)$  is the normalized intensity

distribution [38]. The numerator gives an expression for the radiant flux, which depends on the radiant flux of the LED and also the normalized intensity distribution.

$$E(r, \theta, \phi) = \frac{\Phi_0 I(\theta, \phi)}{r^2 \int I(\theta, \phi) d\Omega} \quad (4.1)$$

The normalized intensity distribution is an important property unique to each LED. A common distribution is shown in Figure 4.1a. This shows the typical normalized intensity distribution for the LA KL120U8F LED by Light Avenue [39]. Notice that the plot only shows the distribution for a varying zenith angle  $\theta$ , which is symmetric for  $[0^\circ, 180^\circ]$  and  $[0^\circ, -180^\circ]$ . This is because the distribution along the azimuth dimension  $\phi$  is constant for any zenith angle  $\theta$ . For the rest of this chapter, all simulations are assumed to use the LA KL180U8F normalized intensity distribution unless otherwise specified.

Equation (4.1) presents the irradiance on a unit area of a sphere  $dA$ . To obtain the irradiance on a flat surface, such as the filter material, the irradiance on the projected unit area  $dA_{\text{proj}}$  needs to be found. The relation between  $dA$  and  $dA_{\text{proj}}$  is shown in Figure 4.1b. Mathematically, this translates to  $dA_{\text{proj}} = dA / \cos \theta$ , with  $\theta$  being the zenith angle. The resulting expression for the projected irradiance is shown in Equation (4.2).

$$E_{\text{proj}}(r, \theta, \phi) = \frac{\Phi_0 I(\theta, \phi) \cos \theta}{r^2 \int I(\theta, \phi) d\Omega} \quad (4.2)$$

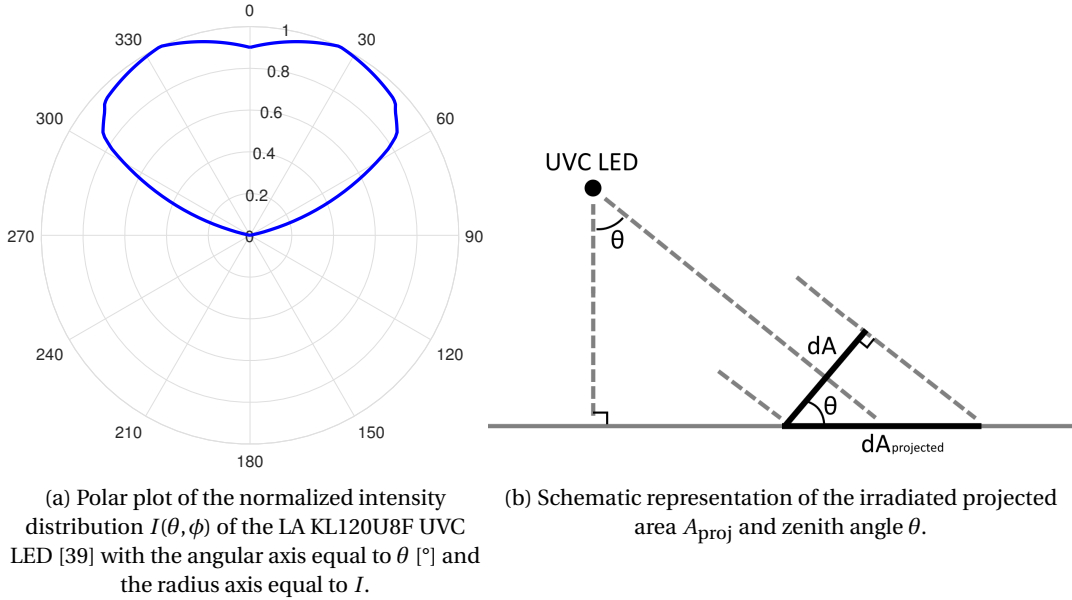


Figure 4.1

## 4.2. Irradiance on a discretized surface

With Equation (4.2) the irradiance on every part of a flat surface can be determined. This expression can thus be used to simulate the irradiance of an LED shining on a flat surface through the use of MATLAB. The irradiance then needs to be computed for every position on the surface. To obtain a finite solution, the surface needs to be discretized by dividing the surface into many small squares with area  $\Delta A$ . The more squares inside a certain area, the higher the resolution, but the more computing power is necessary to calculate the irradiance on every square. Assuming the surface is square itself, the result is a grid of  $n \times n$  elements, which can be represented by a matrix  $\mathbf{S}$ . Each element  $s_{ij}$  contains the irradiance corresponding to a position on the surface  $(x, y)$ . The conversion to its position is dependent on the length of the array  $n$  and the length of the surface  $L$ , resulting in:

$$x_j = L \left( \frac{j}{n} - \frac{1}{2} \right), \quad y_i = L \left( \frac{i}{n} - \frac{1}{2} \right) \quad (4.3)$$

To be able to use Equation (4.2), the distance between the LED and surface element  $r$ , the zenith angle  $\theta$   $[0, \pi]$  and the azimuth angle  $\phi$   $[0, 2\pi]$  both relative to the LED, are to be determined. Assuming the position of the

LED is given by  $(x_{\text{LED}}, y_{\text{LED}}, z_{\text{LED}})$ , with  $z_{\text{LED}}$  the height of the LED relative to the surface, these variables are given by:

$$\begin{aligned} r &= \sqrt{(x_j - x_{\text{LED}})^2 + (y_i - y_{\text{LED}})^2 + z_{\text{LED}}^2} \\ \theta &= \tan^{-1} \left( \frac{\sqrt{(x_j - x_{\text{LED}})^2 + (y_i - y_{\text{LED}})^2}}{z_{\text{LED}}} \right) \\ \phi &= \tan^{-1} \left( \frac{y_i - y_{\text{LED}}}{x_j - x_{\text{LED}}} \right) \end{aligned} \quad (4.4)$$

Finally, the normalized intensity distribution  $I(\theta, \phi)$  needs to be addressed. Even though this function ought to be continuous, the distribution is measured at certain intervals in the characteristic, which is included in the datasheet of the LED in question [39]. Therefore, the normalized intensity distribution is represented by a matrix  $I$  with each element  $i_{kl}$  corresponding to the normalized intensity distribution at  $\theta = l - 1$  [°],  $\phi = k - 1$  [°]. Note that the values for  $\theta$  and  $\phi$  as expressed in Equation (4.4) are not discrete values. Therefore, these values need to be rounded to whole integers with units [°], such that they can be used to index the normalized intensity distribution matrix  $I$ .

Equation (4.5) provides the final expression for the irradiance at grid element  $s_{ij}$ . It is assumed that  $\theta$  and  $\phi$  are calculated using equations from (4.3) and (4.4). Moreover, the summations in the denominator of Equation (4.5) is the result of discretizing the integral from Equation (4.2), where  $\int I(\theta, \phi) d\Omega = \int \int I(\theta, \phi) \sin\theta d\theta d\phi$ .

$$s_{ij} = \frac{\Phi_0 i_{kl}|_{k=\phi+1, l=\theta+1} \cos\theta}{r^2 \left[ \sum_{\phi=0}^{\phi=359} \sum_{\theta=0}^{\theta=180} i_{kl}|_{k=\phi+1, l=\theta+1} \sin\theta \right]} \quad (4.5)$$

After implementing the above expression in MATLAB (refer to Appendix D.1), the images in Figure 4.2 were obtained. The image on the right (Figure 4.2b) clearly shows the grid structure consisting of individual square elements. To check whether this implementation is correct, the radiant flux on each grid element should be summed. Intuitively, the result should be equal to the radiant flux  $\Phi_0$  of the UVC LED. This method and its result are shown in Equation (4.6). As can be seen, the obtained radiant flux is equal to the radiant flux emitted by the UVC LED used in this example, namely 3.5 mW.

$$\Phi = \sum_{i=1}^n \sum_{j=1}^n s_{ij} \Delta x \Delta y \approx 3.5 \text{ mW} \quad (4.6)$$

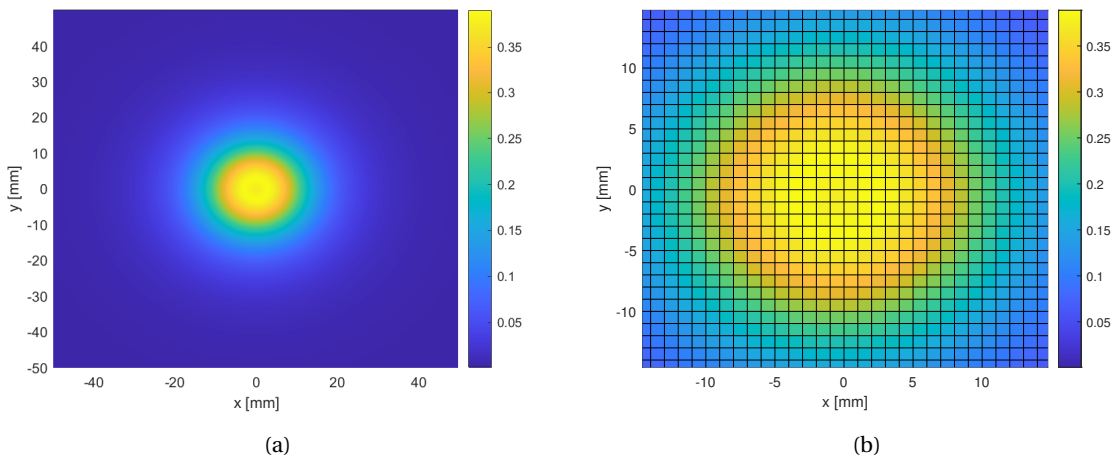


Figure 4.2: Both figures present the irradiance [ $\text{mW}/\text{cm}^2$ ] on a  $100 \times 100$  mm flat surface by a single UVC LED with position  $(0, 0, 15)$  mm, radiant flux  $\Phi_0 = 3.5$  mW, and normalized intensity distribution as shown in Figure 4.1a. The figure on the right shows a zoom-in on the irradiated spot by the LED, such that the grid elements are clearly visible.

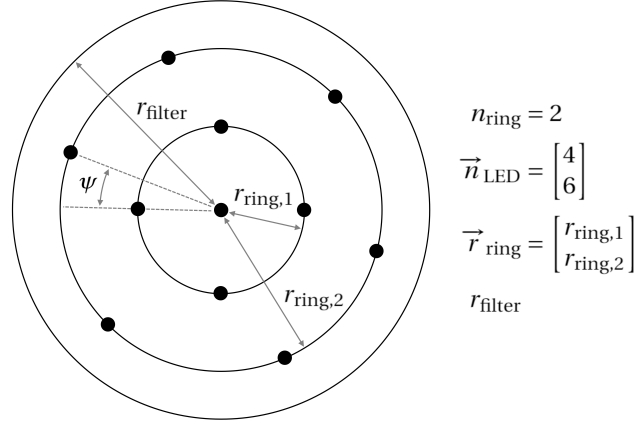


Figure 4.3: Example of an LED structure consisting of 2 rings and 11 LEDs total.

### 4.3. Irradiance of an LED structure

In the introduction of this chapter an LED structure was mentioned on multiple occasions. It stated that in this thesis an LED structure refers to the physical layout of the LEDs. It is defined by the following parameters:

- $n_{\text{ring}}$ : number of rings in the LED structure, excluding the center LED.
- $\vec{n}_{\text{LED}}$ : vector in which the position of the element corresponds to the ring number (from the inner ring outwards) and every element represents the number of LEDs in that ring.
- $\vec{r}_{\text{ring}}$ : vector in which the position of the element corresponds to the ring number (from the inner ring outwards) and every element represents the radius of that ring.
- $\vec{\theta}_{\text{tilt}}$ : vector in which the position of the element corresponds to the ring number (from the inner ring outwards) and every element represents the tilt angle of the LEDs in that ring.
- $h_{\text{LED}}$ : height of all LEDs with respect to the filter surface.
- $r_{\text{filter}}$ : radius of the filter surface.

Figure 4.3 provides an example of an LED structure with two rings, consisting of 4 and 6 LEDs respectively. Notice that the structural parameters can only define circular LED structures which irradiates a round filter with radius  $r_{\text{filter}}$ . This shape was chosen on the grounds that the radiation pattern of an LED is usually round itself. Consequently, introducing corners in the filter makes it harder to apply uniform radiation on the filter material, thereby decreasing uniformity. This in turn increases the rate of structural degradation of the filter material, which is an undesired side-effect (Refer to the Program of Requirements, Chapter 2).

An important property introduced by the use of LED rings is the angle between two adjacent rings. Notice that to conform to the uniformity requirement, the LEDs should be spaced out as much as possible, while keeping their original radius. This optimization problem should be solved for each pair of adjacent rings. The key is to maximize the angle between the two closest LEDs from the two rings in question. In the case of the example shown in Figure 4.3, this angle is indicated by  $\psi$ . Maximizing  $\psi$  ensures minimal overlap of radiation patterns between LEDs of adjacent rings, and thereby increasing radiation uniformity. For the implementation of this solution refer to Appendix D.4.

With the structural parameters as listed above, the irradiance of such a structure can be easily constructed. The reason is because ideally the irradiance on each grid element generated by the individual LEDs can be summed. Mathematically, this translates to the expression in Equation (4.7). Here,  $N$  corresponds to the number of LEDs in the structure and  $s_{ij,n}$  refers to the irradiance on grid element  $(i, j)$  generated by UVC LED  $n$ . An example of the resulting irradiance is shown in Figure 4.4. Notice the increase in the maximum irradiance found on the surface when compared to Figure 4.2, which is precisely the result of the summation. The implementation of the irradiance on a flat filter surface by a parameterized LED structure can be found in Appendix D.3.



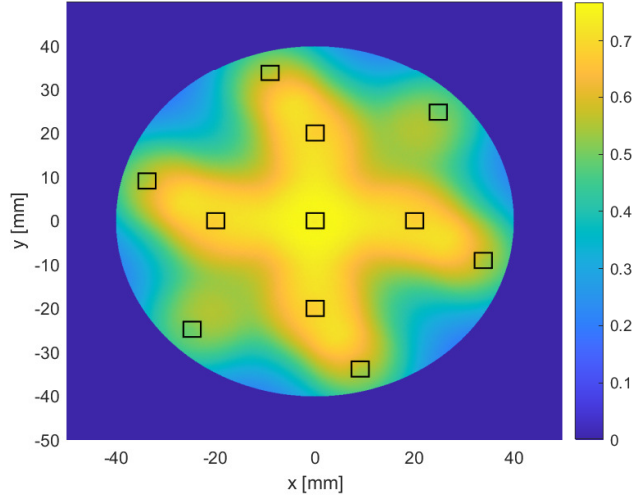


Figure 4.4: Irradiance [ $\text{mW}/\text{cm}^2$ ] on a flat surface with radii of 20 mm, 35 mm, and 40 mm for ring 1, ring 2, and the filter material respectively. The LEDs have a radiant flux  $\Phi_0 = 3.5 \text{ mW}$ , normalized intensity distribution as shown in Figure 4.1a, and their positions are indicated by the black square outlines. The LEDs are all positioned at 15 mm from the surface.

$$s_{ij,\text{total}} = \sum_{n=1}^N s_{ij,n} \quad (4.7)$$

#### 4.4. LED tilt angle

One of the LED structure parameters listed in Section 4.3 has not yet been discussed in much detail, and that is the tilt angle vector  $\vec{\theta}_{\text{tilt}}$ . Due to its complexity, this parameter is discussed separately in this section. Consider the example shown in Figure 4.5. This figure shows a cross section of an LED structure in which the LEDs in the outer ring have a tilt angle  $\theta_{\text{tilt},1}$ . Notice that the tilt angle equals the zenith angle  $\theta$  of the LED. The zenith angle causes the radiation pattern on the surface to transform from circular to a more ellipse-like pattern, thereby increasing the irradiated area. Following this logic, tilting the LEDs might require fewer LEDs to irradiate a given area compared to not tilting any LED. This hypothesis is tested in Section 4.7 where the LED structure is optimized for power efficiency and consumption.

The implementation of the tilt angle is done by transforming the normalized intensity distribution matrix  $I$  (refer to Section 4.2), or more precisely, tilting the normalized intensity distribution. An example of the result of such an implementation is shown in Figure 4.6. Here, the matrix  $I$  has been transformed by tilting it by  $30^\circ$ , and a subset of the resulting matrix is shown in this figure.

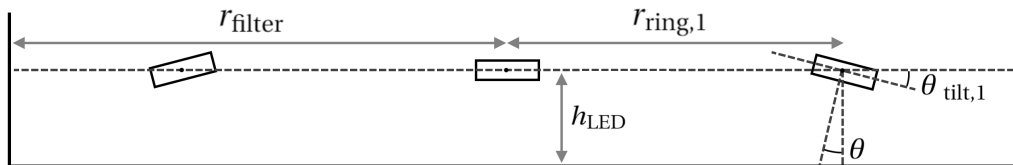


Figure 4.5: Cross section of an LED structure with consisting of a single ring. This ring has a radius  $r_{\text{ring},1}$  and tilt angle  $\theta_{\text{ring},1}$ .

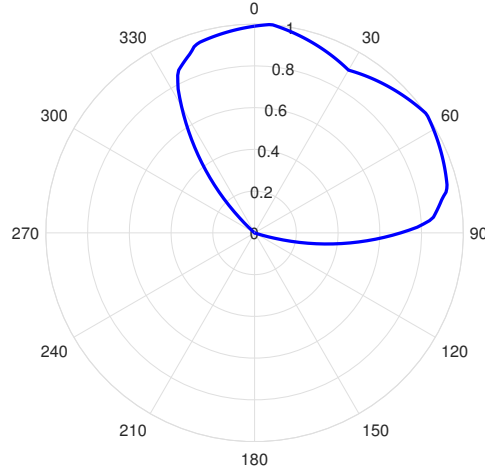


Figure 4.6: Polar plot of the normalized intensity distribution matrix  $I$  of a tilted UVC LED with  $\theta_{\text{tilt}} = 30^\circ$ . The figure shows a range of  $I$  elements with index ranges  $\{\phi = 0^\circ, \theta = [0^\circ, 180^\circ]\}$  and  $\{\phi = 180^\circ, \theta = [0^\circ, 180^\circ]\}$ .

$$r = i_{k_1 l_1} \quad (4.8)$$

$$\phi_1 = k_1 - 1 \quad [^\circ]$$

$$\theta_1 = l_1 - 1 \quad [^\circ]$$

The transformation is done as follows. Consider a single element  $i_{k_1 l_1}$  of the original normalized intensity distribution matrix  $I$ . This can be imagined as a point in the spherical coordinate system  $(r, \phi_1, \theta_1)$ . Figure 4.7a visualizes this idea as point  $p_1$ . The values of the coordinates are expressed in Equation (4.8). The objective of the transformation is to tilt all elements by  $\theta_{\text{tilt}}$  in the direction of  $-y$ , which in the case of Figure 4.7a, translates to shifting point  $p_1$  to  $p_2$ . This is achieved by moving  $p_1$  around the circular cross-section along the  $yz$ -plane, as shown in Figure 4.7b. Before being able to do so, the coordinates of  $p_1$  have to be transformed to the cylindrical coordinate system,  $p_1 = (\rho, \psi_1, x)$ , with an intermediate step of converting them into Cartesian coordinates  $(x_1, y_1, z_1)$ . This transformation is demonstrated in Equation (4.9) and Equation (4.10).

$$x_1 = r \sin \theta_1 \cos \phi_1 \quad (4.9)$$

$$y_1 = r \sin \theta_1 \sin \phi_1$$

$$z_1 = r \cos \theta_1$$

$$\rho = \sqrt{y_1^2 + z_1^2} \quad (4.10)$$

$$\psi_1 = \tan^{-1} z_1 / y_1$$

$$x = x_1$$

Now that  $p_1$  has been translated to cylindrical coordinates, moving it along the radius of the circular cross-section becomes trivial. As can be seen in Figure 4.7b, tilting point  $p_1$  is done by adding the tilt angle  $\theta_{\text{tilt}}$  to the angular position  $\psi_1$ . This is shown in the equation below.

$$\psi_2 = \psi_1 + \theta_{\text{tilt}} \quad (4.11)$$

After the transformation of the angular position, point  $p_1$  has now been shifted to the location of point  $p_2$  with coordinates  $(\rho, \psi_2, x)$ . Notice that only the angular position has changed. The cylindrical coordinates of  $p_2$  are not useful by themselves, as the normalized intensity distribution matrix  $I$  requires spherical coordinates to be able to index its elements. Therefore, the cylindrical coordinates are transformed back into spherical coordinates  $(r, \phi_2, \theta_2)$ , again with an intermediate step of converting them into Cartesian coordinates  $(x_2, y_2, z_2)$ . These steps are outlined in Equation (4.12) and Equation (4.13).

$$x_2 = x \quad (4.12)$$

$$y_2 = \rho \cos \psi_2$$

$$z_2 = \rho \sin \psi_2$$

$$r = \sqrt{x_2^2 + y_2^2 + z_2^2} \quad (4.13)$$

$$\phi_2 = \tan^{-1} \left( \frac{y_2}{x_2} \right)$$

$$\theta_2 = \tan^{-1} \left( \frac{\sqrt{x_2^2 + y_2^2}}{z_2} \right)$$

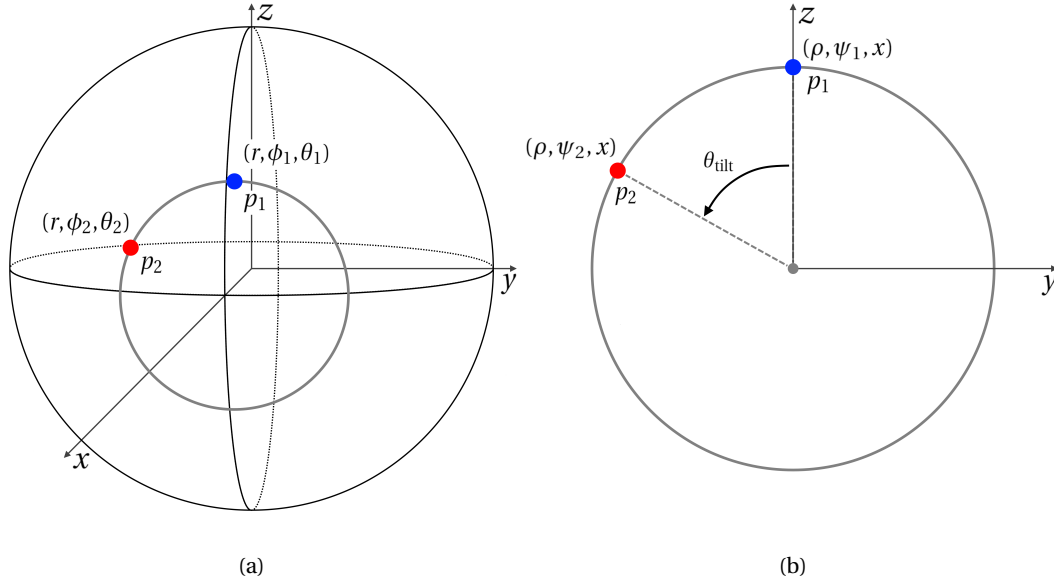


Figure 4.7: Visualization of the tilt transformation with  $\theta_{\text{tilt}}$ , which is applied to a single normalized intensity distribution matrix element  $i_{k_1 l_1}$ , represented in both figures by the blue point  $p_1$ . The result is a new matrix element  $i_{k_2 l_2}$  which has a different position in the normalized distribution matrix  $\mathbf{I}_{\text{tilt}}$ , but has the same value as element  $i_{k_1 l_1}$ .  $i_{k_2 l_2}$  is represented by the red point  $p_2$ .

Analyzing the coordinate transformations in this section, it can be proven that the radius  $r$  of point  $p_2$  in spherical coordinates has the same value as  $r$  of point  $p_1$ . Intuitively, this is true because point  $p_1$  has only been moved along the surface of the sphere shown in Figure 4.7a.

The transformations result in the value of matrix element  $i_{k_1 l_1}$  being moved to a different matrix element  $i_{k_2 l_2}$  in a new matrix  $\mathbf{I}_{\text{tilt}}$ . This is expressed in Equation (4.14), where  $\phi_2$  and  $\theta_2$  have their units in  $[\circ]$ , and are rounded to whole integers.

$$\begin{aligned} i_{k_2 l_2} &= r = i_{k_1 l_1} & (4.14) \\ k_2 &= \phi_2 + 1 \\ l_2 &= \theta_2 + 1 \end{aligned}$$

To tilt the normalized intensity distribution, the entire normalized intensity distribution matrix  $\mathbf{I}$  has to be transformed into  $\mathbf{I}_{\text{tilt}}$ . This means that the steps described in this section have to be performed for every matrix element  $i_{k l}$ . Figure 4.8a shows a 3D-visualization of the original normalized intensity distribution  $\mathbf{I}$ . After applying the tilt transformation with  $\theta_{\text{tilt}} = 30^\circ$  to this matrix, a new matrix  $\mathbf{I}_{\text{tilt}}$  is obtained, which is similarly visualized in Figure 4.8b. Note that Figure 4.6 is a cross-section along the  $xz$ -plane of Figure 4.8b.

Finally, one last problem has to be addressed still. On occasion it could happen that two  $\mathbf{I}$  matrix elements map to the same new indices  $k_2$  and  $l_2$  after rounding the obtained  $\phi_2$  and  $\theta_2$ . Consequently, this leaves some  $\mathbf{I}_{\text{tilt}}$  matrix elements without any assigned value. To overcome this problem, a linear interpolation function is applied to the columns (the  $\phi$ -dimension) of  $\mathbf{I}_{\text{tilt}}$  after completion of the tilt transformation. The used interpolation function is a standard function found in MATLAB [40].

The LED structure parameters listed in Section 4.3 included the vector  $\vec{\theta}_{\text{tilt}}$ . Each vector element presents the tilt of an entire ring of UVC LEDs. Important to note is that a positive  $\theta_{\text{tilt}}$  means the LEDs are tilted in the  $-\rho$ -dimension (cylindrical coordinate system), meaning towards the center of the LED structure. This is depicted in Figure 4.5.

The implementation in MATLAB of the methods discussed in this section can be found in Appendix D.5. For the MATLAB code to generate Figure 4.6 and Figure 4.8b, refer to Appendix D.6 and Appendix D.7 respectively.

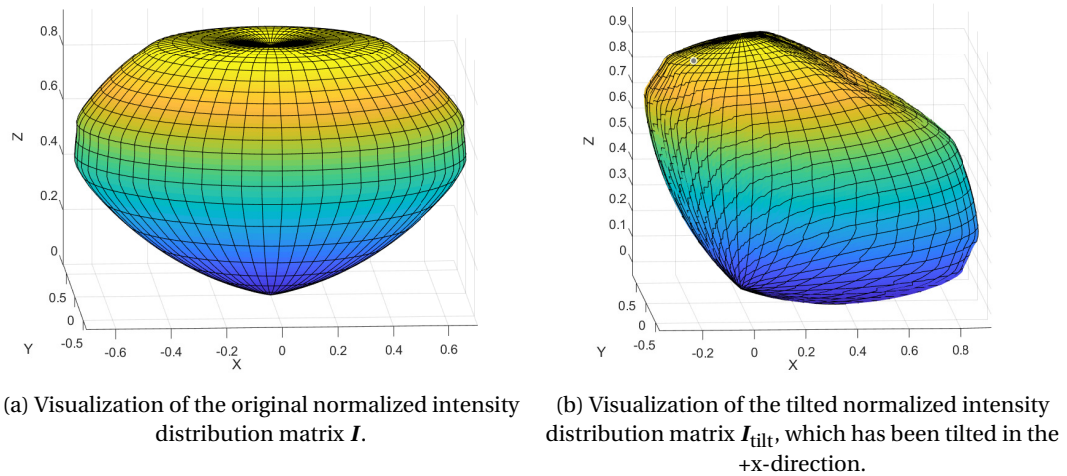


Figure 4.8

## 4.5. Reflectivity of filter housing

Besides the parameters listed in Section 4.3, which are used to describe an LED structure, there is one more property that is useful to include in the simulations. This is the reflectivity of the "walls" surrounding the filter material. Consider Figure 4.4. To create this figure, the simulation only accounted for UVC radiation that has a single uninterrupted path from the LEDs to the flat surface. This implicitly means radiation from LEDs that have a direct path to the perimeter wall (instead of to the filter material) are completely absorbed. For this to happen, a special material is needed which has an absorptance of 100% at the UVC radiation range of the selected LEDs. However, in regards to the requirements listed in Section 2.2, efficiency is an important consideration in the design of the UVGI subsystem.

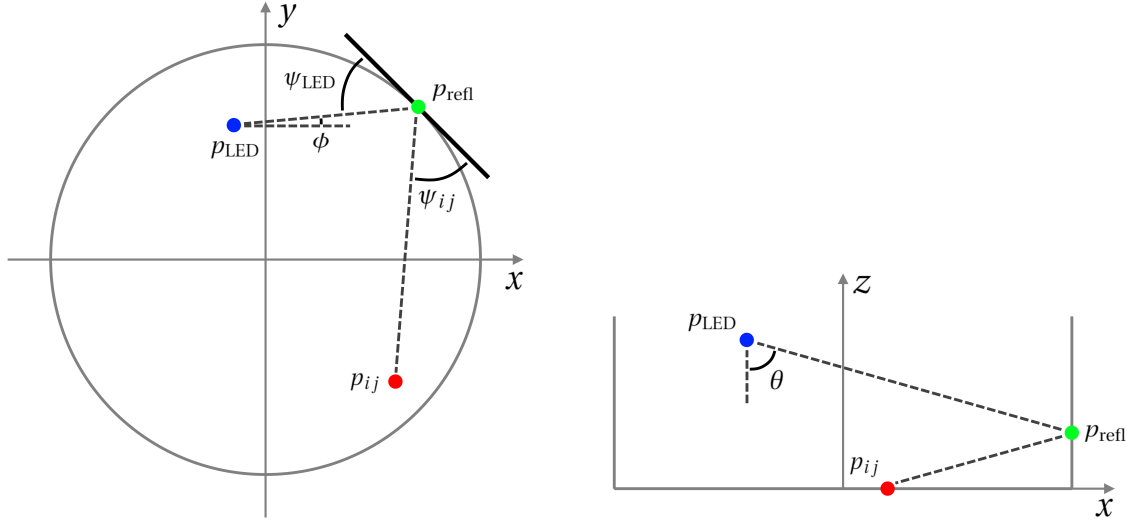
Instead of absorbing the energy radiated onto the perimeter walls of the SPPE module, reflecting said energy would be much more efficient, as most radiated energy would be able to reach the filter material surface. Instead of using a highly absorbing material, UVC reflective materials need to be considered for the perimeter wall. An important property of a reflective material is whether its reflection is diffuse or specular. Diffuse reflection is caused by rough surfaces, which results in light being reflected through a broad distribution of angles. On the other hand, specular reflection is caused by smooth surfaces. An incident light ray reflected on these surfaces, result in a single ray of reflected light with the incident angle equal to the reflection angle [41].

The reflectivity of a range of materials is listed in [3]. It is mentioned that the material with the highest reflectivity is called PTFE (polytetrafluoroethylene), which has a listed reflectivity of 95%-99% around the UVC/UVB range of wavelengths. However, the reflectivity of this material greatly depends on the thickness [42]. Furthermore, the light reflectance distribution for PTFE is mostly diffuse. The second highly reflective material is aluminum, with a reflectivity of around 88%. Whether the light reflectance distribution is diffuse or specular depends on the finish of the aluminum surface. For example etched aluminum has a mostly diffuse distribution, whereas polished aluminum has a mostly specular distribution [43].

For the simulation, computing the path of the reflected beam from a reflection of a single incident light beam (specular reflection) has been estimated to be less complicated to implement than diffuse reflection. This is because for diffuse reflection, the light distribution (multiple reflected beams) on the filter material from a single incident beam has to be computed. For this reason, the reflectivity of aluminum has been assumed (88%) for the perimeter wall, and the rest of this section thus discusses the implementation of specular reflection.

As was already mentioned in this section, for specular reflection the following statement is always true: the angle of the incident light beam, relative to the wall it reflects off, is equal to the angle of the reflected light beam, relative to the same wall. This is visualized in Figure 4.9a, where  $\psi_{\text{LED}} = \psi_{ij}$ .

The source of the beam is a single LED represented by the blue point,  $p_{\text{LED}} = (x_{\text{LED}}, y_{\text{LED}}, z_{\text{LED}})$ . Secondly, the point on the filter surface which is analyzed (grid element  $s_{ij}$ ) is represented by the red point,  $p_{ij} = (x_{ij}, y_{ij})$  ( $z_{ij} = 0$ ). The simulation then moves along the perimeter of the filter surface, and tries to locate the points on the perimeter for which  $\psi_{\text{LED}} = \psi_{ij}$ . Such a point is called a point of reflection,  $p_{\text{refl}} = (x_{\text{refl}}, y_{\text{refl}}, z_{\text{refl}})$ .



(a) Top-view of the filter surface, in which the reflection of a single light beam is shown along the  $xy$ -plane. (b) Side-view of the filter surface, in which the reflection of a single light beam is shown along the  $xz$ -plane.

Figure 4.9: Visualization of UVC light reflection off the perimeter wall. In both figures, the blue point  $p_{LED}$  represents the location of the LED; the red point  $p_{ij}$  represents the location of the grid element  $s_{ij}$ ; and the green point  $p_{refl}$  represents the location of the point of reflection.

Using the above method, when a point of reflection has been found (the  $xy$ -coordinates only), the irradiance on the current grid element  $s_{ij}$  can be calculated using Equation (4.5). However, in order to do so, several terms need to be determined first. For example, the distance  $r$  that the light beam travels from  $p_{LED}$  to  $p_{ij}$  via  $p_{refl}$ . For this, the  $z$ -coordinate of  $p_{refl}$  is needed, which is computed as in Equation (4.15). The distance  $r$  can then be found using Equation (4.16). The two other terms that still need to be determined are the zenith angle  $\theta$  and azimuth angle  $\phi$  of the light beam relative to the LED, as depicted in Figure 4.9. These are used to index the normalized intensity distribution as discussed in Section 4.2. The expressions used to compute  $\theta$  and  $\phi$  are shown in Equation (4.17) and Equation (4.18) respectively.

$$z_{refl} = z_{LED} \cdot \frac{\sqrt{(x_{refl} - x_{LED})^2 + (y_{refl} - y_{LED})^2}}{\sqrt{(x_{refl} - x_{LED})^2 + (y_{refl} - y_{LED})^2} + \sqrt{(x_{refl} - x_{ij})^2 + (y_{refl} - y_{ij})^2}} \quad (4.15)$$

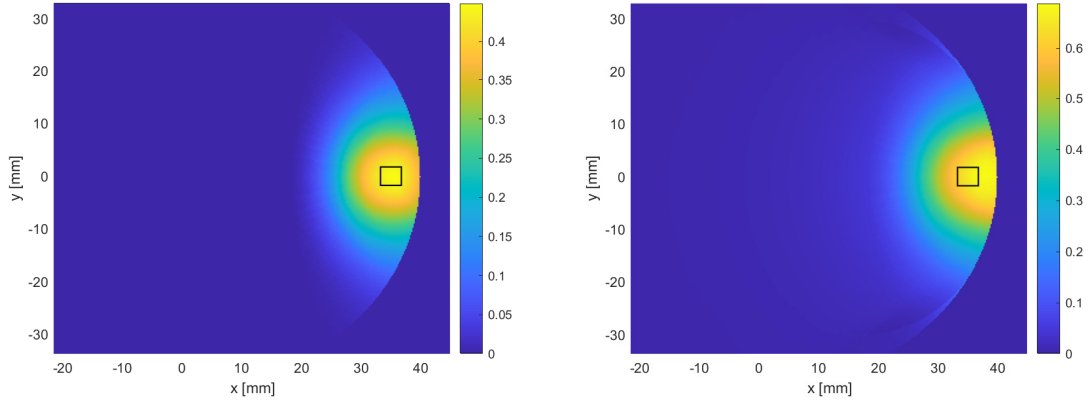
$$r = \sqrt{(x_{refl} - x_{LED})^2 + (y_{refl} - y_{LED})^2 + (z_{refl} - z_{LED})^2} + \sqrt{(x_{refl} - x_{ij})^2 + (y_{refl} - y_{ij})^2 + z_{refl}^2} \quad (4.16)$$

$$\theta = \tan^{-1} \left( \frac{\sqrt{(x_{refl} - x_{LED})^2 + (y_{refl} - y_{LED})^2}}{z_{refl}} \right) \quad (4.17)$$

$$\phi = \tan^{-1} \left( \frac{y_{refl} - y_{LED}}{x_{refl} - x_{LED}} \right) \quad (4.18)$$

Using the above method, the reflected irradiance on a single grid element can be computed. Important to note is that the computed irradiance from reflections should always be multiplied by the reflectivity of the filter housing material. As mentioned earlier in this section, for this project the reflectivity of aluminum is assumed, namely  $\rho_{aluminum} = 0.88$ .

The calculated results for the irradiance without and with reflections for all  $n \times n$  grid elements could look something as shown in Figure 4.10a and Figure 4.10b respectively. Compared to Figure 4.10a, which shows the irradiance without reflections for the exact same LED position, Figure 4.10b shows higher irradiance values for a larger area. It can be concluded that more of the radiated energy is used to irradiate the filter material, thereby increasing the radiation efficiency of the LED structure.



(a) Irradiance with perimeter wall reflectivity equal to 0%. (b) Irradiance with perimeter wall reflectivity equal to 88% (aluminum).

Figure 4.10: Irradiance [ $\text{mW}/\text{cm}^2$ ] on a flat surface ( $r_{\text{filter}} = 40 \text{ mm}$ ) by a single LED located at (35, 0, 15) mm. Moreover, the LED has a radiant flux  $\Phi_0 = 3.5 \text{ mW}$  and normalized intensity distribution as shown in Figure 4.1a.

It is important to note that the method discussed in this section only accounts for first order reflections. That is, the only light beams that reflect off the perimeter wall once are considered. Generally speaking, the more reflections, the larger the beam travel distance, which in turn decreases the intensity of said beam. This is because the light intensity is inversely proportional to the square of the travel distance (refer to Equation (4.5)). Nevertheless, when higher order reflections are considered, the irradiance values on the filter surface only increase, thereby increasing the efficiency even more.

The implementation in MATLAB of the methods discussed in this section can be found in Appendix D.8.

## 4.6. Filter surface area

The design for the filter surface area is a trade-off. On one hand the filter surface size should be large enough in order for the wearer of the SPPE to breath comfortably. On the the other hand, a smaller surface area would mean fewer LEDs are needed to irradiate said area, thereby lowering the power consumption. The institute responsible for regulating PPE in the United States is known as the National Institute for Occupational Safety and Health, or NIOSH for short. This institute keeps regulations in place for the ease of breathing while using a PPE. These regulations take the form of limits on the inhalation resistance of the filter. In 2003, the maximum resistance to inhalation airflow was set at 50 mm water column, for a 85 L/min continuous airflow [44]. The conversion from water column to Pascal is as follows: 1 mm water column = 9.80665 Pa. Therefore, 50 mm water column is equal to 490.3325 Pa. Since 2003 there have been revisions to the maximum resistance. The most recent edition was found to be an inhalation resistance smaller or equal than 343 Pa at a flow rate of 85 L/min [15]. The resistance against airflow can be described with the use of the Poiseuille's law [45]:

$$R = \frac{8 \cdot \eta \cdot l}{\pi \cdot r^4} \quad (4.19)$$

Here,  $\eta$  is viscosity,  $l$  is the tube length, and  $r$  is the tube radius. This equation is normally used to determine the resistance against the airflow in a cylindrical pipe. In order to use this equation for determining the filter surface area, the radius  $r$  needs to be known. However, this radius is not equal to the radius of the filter, but instead this radius is equal to the effective open area of the filter material. In other words, the combined area of the pores present in the filter needs to be known. Such data is rarely found in literature for specific materials, but can be experimentally determined by use of the airflow.

The airflow through an area can be defined as in Equation (4.20) [46][47].

$$Q = A \cdot \sqrt{2\rho \cdot \Delta p} \cdot CD \quad (4.20)$$

In Equation (4.20),  $A$  represents the area,  $\rho$  is the flow density,  $\Delta p$  is the difference between total and static flow pressures at the measurement element, and  $CD$  is the discharge coefficient. Assuming that  $\rho$ ,  $\Delta p$  and  $CD$  are known, and  $Q$  is determined through measurements, it is possible to determine  $A$ . It is important to

note that  $CD$  can be related to the ratio between the filter area and the total open area, as is done in Table D.7 in [47]. This ratio is expressed as  $d/D$ , where  $d$  is the diameter of the hole through which the air flows and  $D$  is the diameter of the pipe [47].

Unfortunately at the moment of writing this thesis, the COVID-19 regulations result in an inability to determine the effective open area of the filter material in an experimental setup. Therefore, the decision has been made to use the surface size of a known PPE, which also makes use of filter modules, namely the Spasciani 2030 A1B1E1K1. These filter modules have a diameter of 8 cm. This diameter is probably too large due to the PPE module also containing chemicals for filtering gasses, while the implementation presented in this thesis will not use such methods. Hence, the filter area is likely to be overestimated. Once experimental data becomes available, this design decision should be reconsidered.

## 4.7. Optimization

Now that the irradiance simulation of an LED structure has been thoroughly explained, the problem of selecting the optimal LED structure can be tackled. In this case, the optimal solution is an LED structure with the least amount of energy consumption while still being able to provide a UVC radiation dose of at least  $305 \text{ mJ/cm}^2$ . Furthermore, the UVC overdose on the filter material should be lower or equal to 1.5, to reduce structural degradation to the filter material. Note that the optimal solution is entirely based on the requirements for the UVGI subsystem as listed in Section 2.2.

For the optimization to work, a single variable is needed, that is either to be maximized or minimized. As listed above, for this project the energy consumption needs to be minimized, and thus this is the variable to be optimized. The expression of the energy consumption per dose by the LED structure is given in Equation (4.21).

$$E_{\text{dose}} = \frac{N_{\text{LED}} P_{\text{total}} D_{\text{min}}}{\Phi_0 F_{\text{min}}} \quad (4.21)$$

- $E_{\text{dose}}$ : energy consumption per dose of the entire LED structure [J].
- $N_{\text{LED}}$ : number of UVC LEDs in the LED structure.
- $P_{\text{total}}$ : power consumption of a single UVC LED [W].
- $D_{\text{min}}$ : minimum UVC dose required to apply to the filter [ $\text{J/m}^2$ ].
- $\Phi_0$ : radiant flux of a single UVC LED [W].
- $F_{\text{min}}$ : minimum irradiance factor [ $1/\text{m}^2$ ].

The above equation consists of several components. The first of these determines the UVC LED selection and is discussed in Section 4.7.1. The second component determines the LED structure, which uses the previous selected UVC LED. This component is discussed in more detail in Section 4.7.2.

### 4.7.1. UVC LED selection

Equation (4.22) shows the first component, which is named the first optimization variable  $V_{\text{opt},1}$ . Both  $P_{\text{total}}$  and  $\Phi_0$  are specifications that can be found in the datasheet of a UVC LED. Minimizing the energy consumption per dose requires this variable to be minimized. Therefore, for this project, the UVC LED selection is almost entirely based on the lowest  $V_{\text{opt},1}$  value.

However, another important property of UVC LEDs, and especially for UVGI, that needs to be considered is the peak wavelength  $\lambda_p$  of the UVC radiation. As mentioned in Section 3.1.1, the optimal wavelength for UVGI applications range from 260 to 280 nm. Most UVC LEDs specifically intended for UVGI have a peak wavelength of around 270 to 280 nm however.

With the optimal UVGI wavelengths in mind, a range of UVC LEDs from various manufacturers are considered. These LEDs are all listed in the table in Appendix A, along with their most important characteristics, which include  $P_{\text{total}}$  and  $\Phi_0$ . The last column of this table shows the value of  $V_{\text{opt},1}$  for each of these LEDs. Based on data from the datasheets, it can be seen that the UVC LED with the lowest  $V_{\text{opt},1}$  value is the LA KL120U8F from Light Avenue. This UVC LED has thus been selected for this project.

$$V_{\text{opt},1} = \frac{P_{\text{total}}}{\Phi_0} \quad (4.22)$$

### 4.7.2. LED structure optimization theory

The second component of Equation (4.21) is named the second optimization variable  $V_{\text{opt},2}$  and its value is shown in Equation (4.23). As stated before,  $N_{\text{LED}}$  is the number of LEDs in the LED structure,  $\Phi_0$  is the radiant flux of the selected UVC LED, and  $F_{\text{min}}$  is the minimum irradiance factor. Note that the minimum irradiance  $E_{\text{min}}$  relates to the radiant flux and minimum irradiance factor by the expression shown in Equation (4.23).

$$V_{\text{opt},2} = \frac{N_{\text{LED}}}{\Phi_0 F_{\text{min}}} = \frac{N_{\text{LED}}}{E_{\text{min}}} \quad (4.23)$$

As with  $V_{\text{opt},1}$ , minimizing the second optimization variable  $V_{\text{opt},2}$  in turn minimizes the energy consumption per dose. Note that  $E_{\text{min}}$  depends on the outcome of the irradiance simulation, and thus on the LED structure. Not surprisingly,  $N_{\text{LED}}$  depends on the LED structure as well. The LED structure itself is entirely constructed from the variables listed at the beginning of Section 4.3. Optimizing the LED structure thus means finding the LED structure parameters, which result in the minimum  $V_{\text{opt},2}$  value.

The LED structure optimization problem can be defined in mathematical optimization terms [48] [49]. The following terminology is used in this section:

- $\vec{v}$ , **decision variables**: the variables that are under the control of the optimization program.
- $\vec{v}_{\text{min}}$  and  $\vec{v}_{\text{max}}$ , **decision variable bounds**: bounds outlining the subset of each decision variable over which the objective function is optimized.
- $\vec{p}$ , **input parameters**: fixed data which control the optimization program.
- $\vec{c}$ , **constraints**: restrictions posed on the solution of the optimization program.
- $f(\vec{v}, \vec{p})$ , **objective function**: a real-valued function, whose value is to be minimized (or maximized in some other cases) by varying the set of decision variables within their bounds.

The elements in the decision variables vector  $\vec{v}$  are listed LED structure parameters (excluding the filter radius  $r_{\text{filter}}$ ), as these are varied while minimizing the objective function. The elements of the decision variables vector are displayed in Equation (4.24). The contents of the decision variable bounds vectors are the same sequence of elements as in  $\vec{v}$ , the only difference is that the bounds vectors contain fixed values representing the boundary values of the decision variables.

There is one exception to this rule however, and that is in regards to the LED ring radius vector  $\vec{r}_{\text{ring}}$ . This is because the minimum bound is not considered to be the absolute minimum radius, but the minimum difference in radii between two adjacent rings. For example, if the inner ring has a radius of 10 mm, the minimum radius of the adjacent ring is 10 mm +  $r_{\text{ring},\text{min}}$ .

$$\vec{v} = [n_{\text{ring}}, \vec{n}_{\text{LED}}, \vec{r}_{\text{ring}}, \vec{\theta}_{\text{tilt}}, h_{\text{LED}}] \quad (4.24)$$

The elements in the input parameters vector  $\vec{p}$  consist of fixed values that are needed to compute  $V_{\text{opt},2}$  for each LED structure iteration. The content of  $\vec{p}$  is shown in Equation (4.25). Here,  $r_{\text{filter}}$  is the radius of the irradiated filter material,  $\Phi_0$  is the radiant flux of the selected UVC LED,  $I$  is the non-tilted normalized intensity distribution of the selected UVC LED, and  $\rho$  is the reflectivity of the filter housing material.

$$\vec{p} = [r_{\text{filter}}, \Phi_0, I, \rho] \quad (4.25)$$

$$r_{\text{filter}} = 40 \text{ mm} \quad \Phi_0 = 3.5 \text{ mW/cm}^2 \quad I = I_{\text{LA KL120U8F}} \quad \rho = 0.88$$

The elements in the constraints vector  $\vec{c}$ , shown in Equation (4.26), consist of three main constraints imposed on the the optimal LED structure solution. The first of these is the minimum irradiance found on the filter material. Section 2.2 of the Program of Requirements states that the dose that the UVGI subsystem has to deliver to the filter material per 2 hours is equal to 305 mJ/cm<sup>2</sup>. From this, the minimum irradiance on the filter material can be computed, which would still be able to apply said dose within the stated 2 hours. The minimum irradiance is calculated as in Equation (4.27).

The second constraint is the maximum overdose. The overdose is a ratio of the maximum irradiance over the minimum irradiance found on the filter. Its mathematical definition is given in Equation (4.28). Here, the value of the maximum overdose is provided as well, which is based on the UVGI requirements in Section 2.2 of the Program of Requirements.

Lastly, the third constraint is that of the maximum height of the UVC LEDs with respect to the filter surface.



Due to size limitations of the SPPE module, the maximum height has been set to 15 mm, which was obtained from Section 2.2 of the Program of Requirements. The constraint can be seen in Equation (4.29).

$$\vec{c} = [E_{\min}, O_{\max}, h_{\text{LED,max}}] \quad (4.26)$$

$$E_{\min} = \frac{D}{T_{\max}} = \frac{305 \text{ mJ/cm}^2}{7200 \text{ s}} \approx 42.4 \text{ } \mu\text{W/cm}^2 \quad (4.27)$$

$$O = \frac{E_{\max}}{E_{\min}} \quad O_{\max} = 1.5 \quad (4.28)$$

$$h_{\text{LED,max}} = 15 \text{ mm} \quad (4.29)$$

For optimizing the LED structure, the value of the objective function  $f$  is equal to the second optimization variable  $V_{\text{opt},2}$ . This is mathematically stated in Equation (4.30).

$$V_{\text{opt},2} = f(\vec{v}, \vec{p}) = \frac{N_{\text{LED}}}{E_{\min}} \quad (4.30)$$

With all variables and parameters of the optimization program explained, the optimization process can be described. The optimization program has as input the two decision variable bounds,  $\vec{v}_{\min}$  and  $\vec{v}_{\max}$ , and the input parameters  $\vec{p}$ . It then computes the value of the objective function  $f$  for all values of  $\vec{v}$  within the range specified by the bounds  $\vec{v}_{\min}$  and  $\vec{v}_{\max}$ . Note from Equation (4.30) that the objective function is directly dependent on the result of the irradiance simulation. Summarized, this means that the objective function is computed for all permutations of the LED structure set by the bounds. The LED structure with the lowest objective function value is said to be optimal.

Lastly, the resulting optimal LED structure can still be discarded if the constraints set by  $\vec{c}$  are not met. The results of the most important optimization runs are explained in the next Section.

The complete optimization code written in MATLAB can be found in Appendix D.10.

### 4.7.3. LED structure optimization results

As described in the previous section, optimizing the LED structure is done by going through a set of LED structure permutations and simulating the irradiance on the filter surface for each one of these. Intuitively, having a large set of permutations increases the duration of the optimization process. In contrast, having a small set of permutations may lead to a non-optimal LED structure. Therefore, a number of optimization runs have been performed with different decision variable bounds, such that the optimal LED structure can be narrowed down to a smaller set of permutations each time. Still, the duration of the optimization runs were sometimes in the order of weeks. For this reason, a TU Delft server was tasked with performing these optimization runs.

The most important optimization runs along with their results are shown in Table 4.1. On the left side of the table, the optimization input is shown in the form of the decision variables bounds. The values of the

Table 4.1: Four optimization runs with their respective inputs and results.

Run [#]	Optimization input					Optimization results							
	$[\vec{v}_{\min}, \vec{v}_{\max}]$					$\vec{v}_{\text{optimal}}$					$f_{\min}$	$E_{\min}$	$O$
	$n_{\text{ring}}$	$n_{\text{LED}}$	$r_{\text{ring}}$ [mm]	$\theta_{\text{tilt}}$ [°]	$h_{\text{LED}}$ [mm]	$n_{\text{ring}}$	$\vec{n}_{\text{LED}}$	$\vec{r}_{\text{ring}}$ [mm]	$\vec{\theta}_{\text{tilt}}$ [°]	$h_{\text{LED}}$ [mm]			
1	[1, 2]	[2, 32]	[3, 37]	[0, 0]	[10, 10]	1	15	30	0	10	34.7	0.461	3.07
2	[1, 1]	[4, 22]	[3, 37]	[-30, 30]	[8, 22]	1	12	29	12	18	19.2	0.677	1.25
3	[1, 1]	[10, 18]	[3, 37]	[0, 20]	[12, 15]	1	12	29	10	15	19.6	0.663	1.42
4	[1, 1]	[10, 20]	[3, 37]	[0, 20]	[8, 10]	1	12	30	10	10	25.0	0.520	2.43

input parameters were constant, and are shown in Equation (4.25). The exception is the first run, which had its reflectivity parameter set to  $\rho = 0$ . On the right side of Table 4.1, the optimization results are shown. These consist of the optimal value for each decision variable, the corresponding minimum objective function value  $f_{\min}$ , and their constraints values  $E_{\min}$  and  $O$ .

From the result of the first run (see Figure 4.11a), it became apparent that an LED structure with only a single ring would be optimal. Because of this, the  $n_{\text{ring}}$  bounds for the runs thereafter were reduced to only a single ring. The result of the second run listed in Table 4.1 showed that adding a tilt angle and using a reflective housing material made for a significant improvement in the minimum objective function value. Moreover, the minimum irradiance jumped up to  $0.677 \text{ mW/cm}^2$  and the overdose was reduced to only 1.25. The irradiance of this structure is shown in Figure 4.11b. Unfortunately, the value for  $h_{\text{LED}}$  of this LED structure is higher than the  $h_{\text{LED,max}}$  constraint value, as shown in Equation (4.29). Therefore, optimization run 3 reduced the maximum height boundary to 15 mm. Unsurprisingly, this resulted in an optimum LED structure with its height set to 15 mm. Fortunately, its minimum objective function and overdose value increased only slightly compared to the previous run. In addition, this LED structure meets all constraints set in the previous section. The irradiation of this LED structure can be observed in Figure 4.11c. Finally, because a minimal UVGI form factor is favored, a last optimization run was performed with the upper boundary for the height set to 10 mm. As can be seen in Table 4.1 however, the resulting LED structure had an overdose much larger than the constraint of 1.5.

From the above text, it is clear that, without taking the constraints into account, the overall optimal LED structure is the one from run 2. However, the constraints do have to be met. Consequently, for this project the LED structure from run 3 is deemed as the optimal LED structure for the application of UVGI on a 40 mm radius filter material.

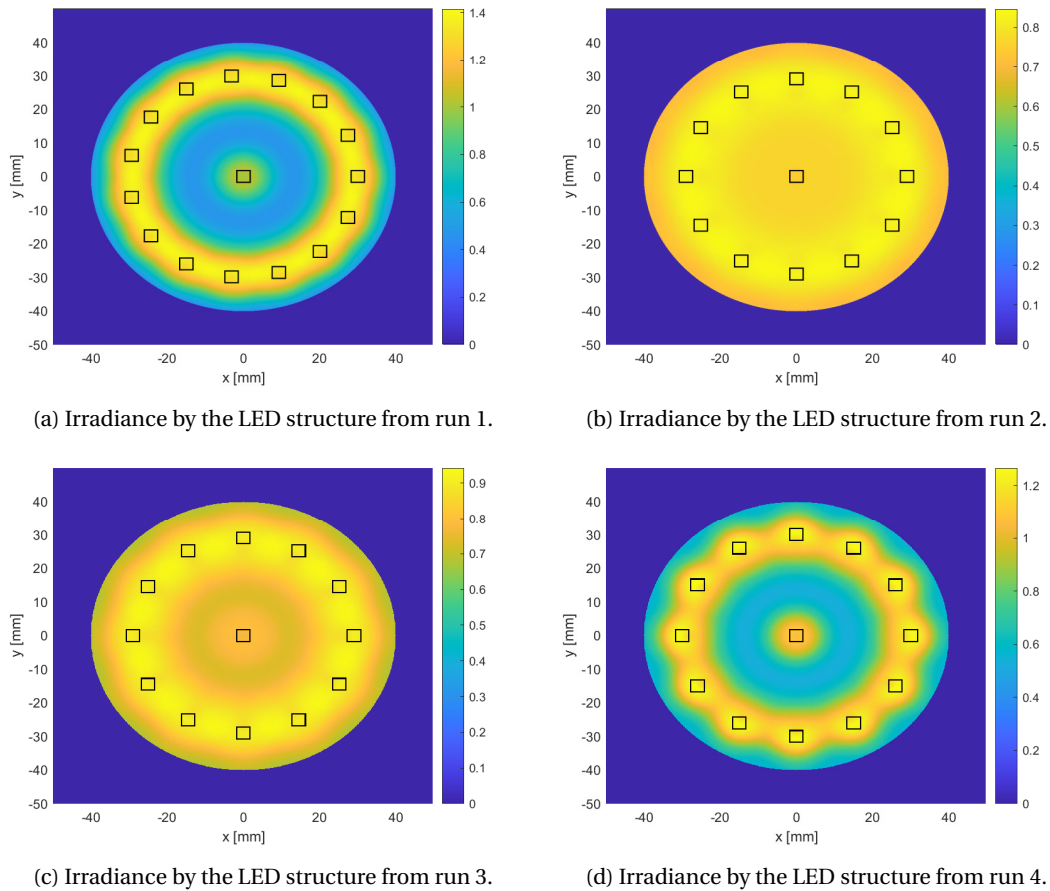


Figure 4.11: Irradiance [ $\text{mW/cm}^2$ ] on a flat surface by the LED structures resulting from the optimization runs shown in Table 4.1.

# 5

## Driver Circuit

One of the design requirements for the UVGI subsystem is the ability to control and adjust the UVC dose applied by the LEDs. The intensity of the UVC LEDs is based on the electrical current going through the p-n junction. In order to control the dose, a driver circuit has been designed, which controls the LED current towards a given UVC intensity value. To adjust the dose, the on-time of the UVC LEDs can be adjusted accordingly. Important to note is that for this chapter, the LA KL120U8F UVC LED [39] is assumed to be used for all implementations that are discussed.

This chapter will cover the design process of the driver circuit, starting by discussing different circuit options (Section 5.1). Following this, a list of design criteria is compared between the different implementation options and a base typology will be selected (Section 5.2). The selected circuit is then further analyzed and the best components are chosen for this application (Section 5.3). Finally, the functionality of the selected circuit is tested by running various simulations (Section 5.4).

### 5.1. Circuit implementations

As mentioned above, the intensity of the UVC LEDs is approximately linearly dependent on the current through their p-n junction [39]. Moreover, the current-voltage relation of a diode is strongly non-linear. Therefore, it is much easier to accurately control the intensity of LEDs with a controllable current source, than it is with a controllable voltage.

The driver circuit itself will be controlled based on input signals from the microcontroller of the Sensing and Control group. As the availability of controlled voltage sources is higher than that for controlled current sources for the microcontroller, it stands to reason that the input of the driver circuit will thus be a voltage.

From the above analysis, the basic format for the UVGI driver circuit should be a transadmittance amplifier (voltage in, current out).

#### 5.1.1. Basic transadmittance amplifier

A transadmittance amplifier has a voltage input quantity and a current output quantity. A basic implementation of the driver circuit with a transadmittance amplifier configuration is displayed in Figure 5.1. The circuit consists of an operational amplifier (op-amp), transistor, and input signal generated by a DAC, which is controlled by the controller.

The circuit needs to drive the UVC LED array, as shown in Figure 5.1. For the LA KL120U8F, the typical forward current and voltage is 20 mA and 5.7 V respectively. An additional voltage supply,  $V_{CC}$ , is provided by the On-Board Power Management team, which provides the relatively high voltage necessary to drive the LED array. Note that  $V_{CC}$  should be large enough such that it exceeds the voltage drop over resistor  $R_1$ , the voltage drop over two UVC LEDs in series, and the base-emitter voltage of the transistor. Mathematically, this is shown in Equation (5.1).

$$V_{CC} > V_{R1} + 2V_{LED} + V_{BE} \quad (5.1)$$

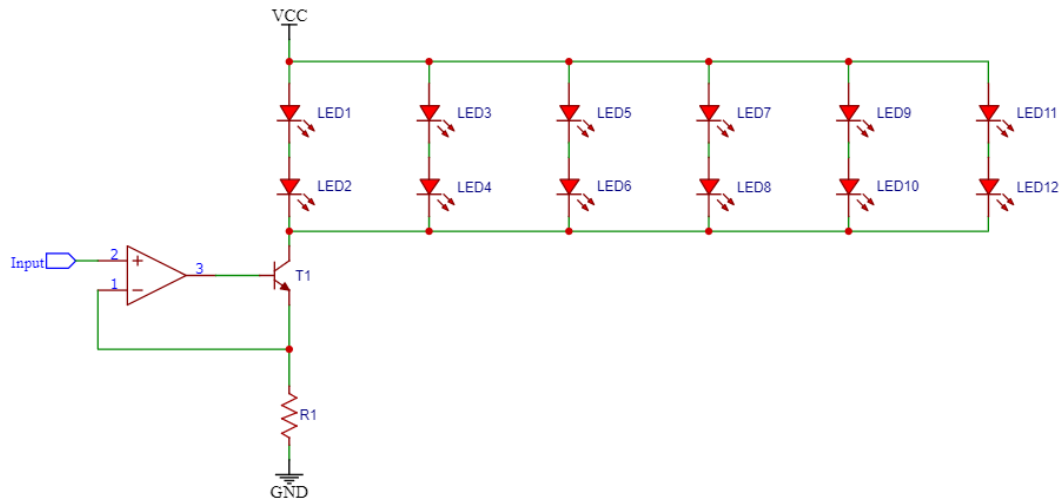


Figure 5.1: The UVC LED driver circuit implemented with a transadmittance amplifier.

The reason that there are two LEDs placed in series per parallel string of LEDs and not more, is that the value of  $V_{cc}$  is 15 V. With two LEDs in series, this means that the voltage drop over these two LEDs is already 11.4 V. Longer strings of UVC LEDs are therefore not possible with the available 15 V.

The current through each string of LEDs is determined as follows. Due to the application of negative feedback, the op-amp generates a voltage at its output terminal, such that the voltage at its inverting input is equal to that of the non-inverting input. Applying Ohm's law, it can be said that the current through  $R1$ , and therefore the emitter current, is equal to  $V_{input}/R1$ . The current gain of a transistor from the collector terminal to the emitter terminal ( $I_C/I_E$ ) is called  $\alpha$ . If it is assumed that  $\alpha \approx 1$ , then the emitter current is equal to the collector current of  $T1$ . Finally, the current through each string of LEDs is equal to the collector current  $I_C$  divided by the number of parallel LED strings  $N_{string}$ . In conclusion, the current through each UVC LED is mathematically shown in Equation (5.2).

$$I_{LED} \approx \frac{V_{input}}{N_{string} R1} \quad (5.2)$$

Note that the current sourced by  $V_{cc}$  is linearly dependent on the number of parallel LED strings.

The number of parallel strings displayed in Figure 5.1 can be extended to reach the 26 required LEDs (two LED structures for irradiation of both sides) as was determined in Chapter 4. Note that for clarity, not all the 26 LEDs are shown.

### Binning

As mentioned earlier, the LED selected for the application of UVC radiation is the LA KL120U8F. As with other LEDs, a binned supply of this LED is available. The meaning of binning is that the LEDs are sorted into multiple ranges of component characteristics. For example, the LA KL120U8F has 4 bins mentioned in the datasheet for its typical radiant power and 5 bins for its typical forward voltage [39]. These differences in their current-voltage characteristic means that there could be different currents drawn per LED string, which in turn leads to different radiation intensities. This effect is clarified with the following example.

Assume the UVC LEDs used, all originate from voltage bin 2 stated in the datasheet [39]:  $V_{typ,min} = 5.1$  V, and  $V_{typ,max} = 5.5$  V at 20 mA. Furthermore, assume there are two of these LEDs connected in parallel. The first LED has a typical voltage of 5.1 V and the second LED has a typical voltage of 5.5 V. The graphs displaying the characteristics of the LA KL120U8F are shown in Figure 5.2. If a voltage of 5.1 V is applied over this parallel connection, the first LED then draws its typical 20 mA. However, the second LED has a different current-voltage characteristic, which is shown in the left plot of Figure 5.2. With a forward voltage of 5.1 V, this LED only draws about 3 mA. Assuming the relation between the forward current and radiant flux is purely linear, the second LED has a  $([20 \text{ mA} - 3 \text{ mA}]/20 \text{ mA} \cdot 100\% =)$  85% drop in radiant flux compared to the first LED.

It can thus be concluded that differences in LED characteristics of LEDs in the same bin is a serious issue. The

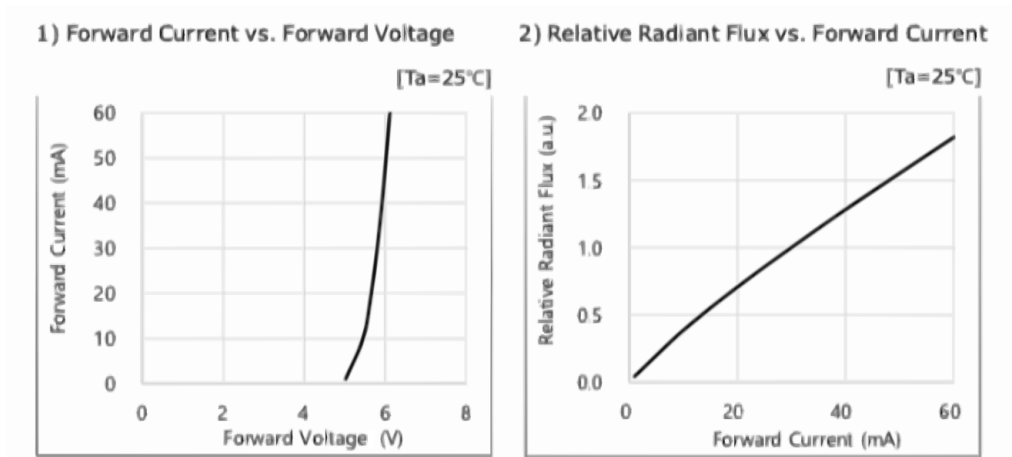


Figure 5.2: Graphs of the LA KL120U8F characteristics [39]

result is that the implementation shown in Figure 5.1 is not a valid LED driver circuit, as it does not account for these differences.

However, there are some adjustments which can be made to improve the current instability. One such improvement is to place more LEDs in series. If more LEDs are placed in series, the different typical voltages of each LED will be averaged out. But, as stated earlier, two LEDs in series already cause a voltage difference of roughly 11.4 V, and there is only 15 V available as per the requirements listed in Section 2.2. With only 3.6 V headroom in the driver design, the transadmittance configuration as shown in Figure 5.1 is not further considered.

The next section discusses two implementations that can be implemented to overcome this problem.

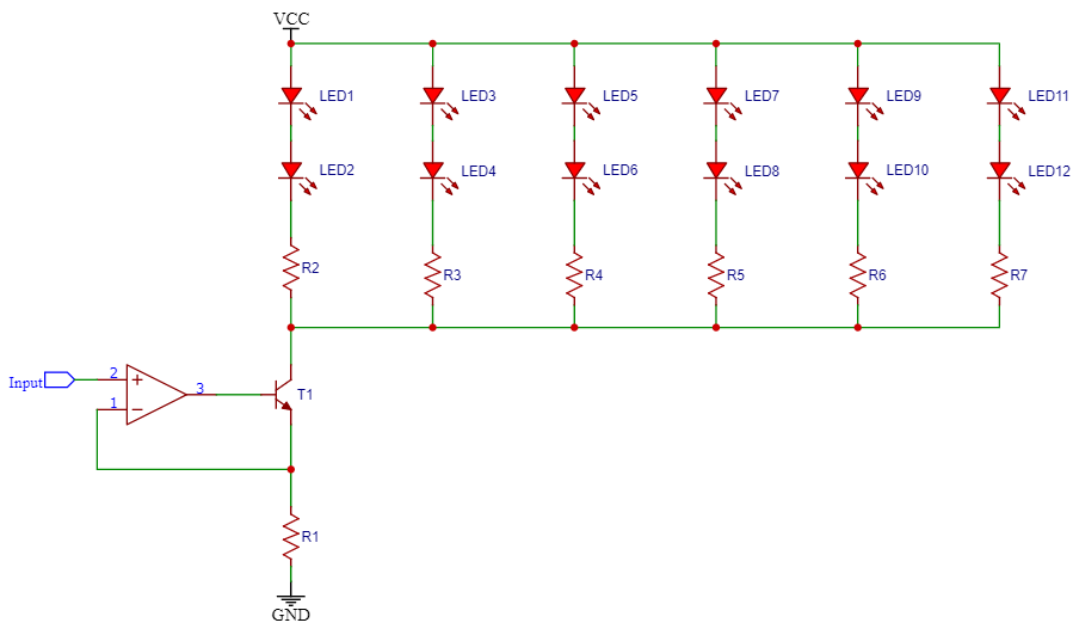


Figure 5.3: UVC LED array driver circuit 1.

### 5.1.2. Transadmittance amplifier driver circuits

The transadmittance amplifier circuit can be adjusted to significantly reduce the problem of varying LED current-voltage characteristics. One such implementation is displayed in Figure 5.3. In this circuit, every LED string is extended by the introduction of an additional resistor ( $R_2$  to  $R_7$ ). Consider the situation in which the voltage drop across the resistors is relatively large compared to the voltage drop over the two LEDs

per string. Then any variation in voltage drop over the individual LEDs is nullified by the much larger voltage over the resistors, which almost does not change. The current match across the strings would thus be much higher than for the circuit shown in Figure 5.1.

There are quite a few downsides to this implementation however. Most importantly, with a supply voltage of only 15 V, it becomes impossible to create the relatively large voltage drop across the resistors in each string. Therefore, the current matching ability is reduced significantly, resulting in a decrease in accuracy. Even if this problem was to be disregarded, the power dissipation in each resistor reduces the efficiency of the LED driver considerably. Moreover, as the current directly relates to the power dissipation, both increase linearly with each extra string of LEDs.

A different approach to minimize the problem of varying LED characteristics is displayed in Figure 5.4. In this circuit, every LED string now has its own transistor. In doing so, the LEDs are no longer directly in parallel, thereby removing the negative effect of varying LED characteristics. The current through each string is almost only determined by the value of resistors  $R7$  through  $R12$ . If it is assumed that all these resistors have the same resistance  $R$ , an approximation of the LED current can be given as shown in Equation (5.3).

$$I_{\text{LED}} \approx \frac{V_{\text{input}}}{R} \quad (5.3)$$

Note that the transistors are all driven by a single operational amplifier. This reduces the component count, but also introduces a small error. If  $R1$  through  $R6$  were to be shorted, the variation in electrical characteristics among the transistors (most obviously variations in  $V_{\text{BE}}$ ) might introduce a variation in LED currents among the parallel strings. Resistors  $R1$  through  $R6$  cause the voltage at the emitter of each transistor to be averaged, which is then directed to the inverting input of the op-amp. This reduces the effect of electrical variations in the transistors.

Similarly to the previous implementation, this circuit has some significant downsides. Most noticeably, this circuit uses a lot more components than the previous circuit. Furthermore, the power dissipation, and thus efficiency, of this circuit has not improved either. This is because all LED current has to go through every transistor and resistor in series with the strings of LEDs.

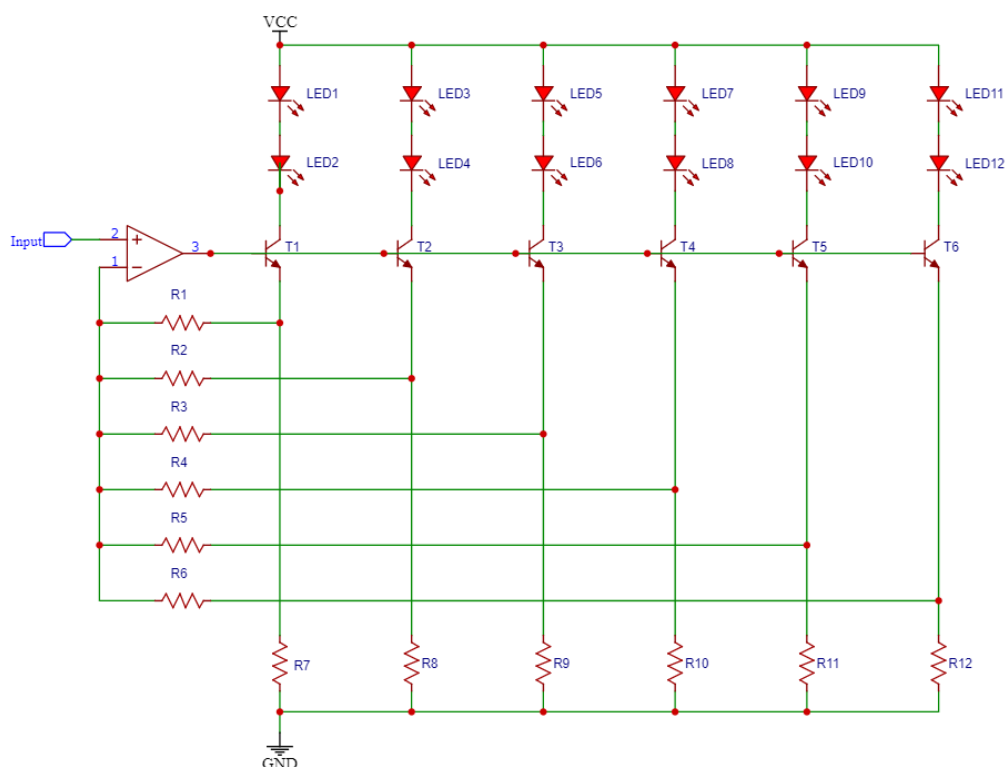


Figure 5.4: UVC LED array driver circuit 2.

It is important to mention that there will always be differences in radiant power generated by each UVC LED. This is because of the variations in electrical characteristics mentioned earlier. The LA KL120U8F has a number of bins for its typical radiant power at 20 mA forward current [39]. However, the differences in radiant power within a certain bin are not as significant as previously calculated due to parallel LED strings from the implementation in Section 5.1.1.

### 5.1.3. Boost-mode driver circuit

There is also the possibility for a different driver typology altogether. Figure 5.5 displays a boost-mode implementation of the driver circuit.

This circuit functions as follows. When the switch in the circuit is closed, a current starts flowing through the inductor and from switch terminal 1 to terminal 2. The inductor thus starts storing energy in the form of a magnetic field. After a period of  $DT_s$  seconds, the switch is opened. Here,  $D$  is the duty-cycle and  $T_s$  is the switching period. The moment the switch is opened, the original current through the switch is redirected through the diode, thereby charging the capacitor and turning on the LEDs. The inductor sources the current by discharging its stored energy. After a period of  $(1 - D)T_s$  seconds, the switch is closed, which causes the inductor to be charged again. This time however, the current through the LEDs is provided by output capacitor  $C1$ . This cycle repeats itself every period  $T_s$ .

The voltage and current at the output (the anode of LED1) is given by Equation (5.4) and Equation (5.5) respectively.

$$V_{\text{out}} = \frac{V_{\text{in}}}{1 - D} \quad (5.4)$$

$$I_{\text{out}} = (1 - D)I_{\text{in}} \quad (5.5)$$

Regulations which limit the maximum voltage in a consumer electronic device are reported in the IEC 62369-1 [50]. The voltage is being limited to 42.4 V, but this value differs per type of electronic device. Since no specific value for an SPPE was found, the decision was made to round this number down to a maximum voltage of 40 V. This maximum voltage is listed in Section 2.2 of the Program of Requirements.

There are several advantages of this circuit compared to the ones previously discussed. For one, this circuit is much more power efficient as there are no resistors in series with the LEDs for example. There is still power dissipation inside the diode, switch, inductor, and capacitor, but this is significantly less than with the other driver circuits. Secondly, a boost-mode circuit such as this one could produce a high voltage gain. This

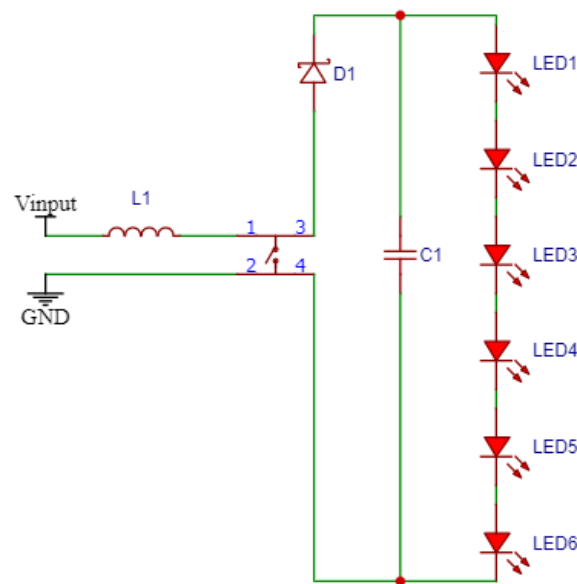


Figure 5.5: UVC LED array driver circuit 3.

makes it possible to put a larger number of UVC LEDs in series per string, as seen in Figure 5.5. The number of LEDs is limited by the output voltage regulation as mentioned before. Note that this circuit can generate electromagnetic interference (EMI). The reason is that the switching frequency of the switch is usually in a relatively high frequency range (100 kHz - 10 MHz). To reduce the effect of EMI, the traces connecting to the switch should be kept as short as possible [51].

Another design constraint is the input current, which is limited by the requirement set in the Program of Requirements. The requirement states that the input current is limited to 500 mA.

## 5.2. Circuit selection

The circuits discussed in the previous sections all have certain advantages and disadvantages when it comes to their implementation. To select the best circuit for this project, they are compared based on various common design criteria. These design criteria are as follows:

- **Power dissipation:** to reduce wasted energy in the form of heat, the power dissipation should be as low as possible. For the comparison, "+" indicates low power dissipation, "+-" indicates medium power dissipation, and "-" indicates high power dissipation.
- **Efficiency:** to conform to the requirement of low energy consumption, a driver circuit with a high efficiency is favored. For the comparison, "+" indicates high efficiency, "+-" indicates medium efficiency, and "-" indicates low efficiency.
- **Circuit size:** as the form factor of the UVGI subsystem and the SPPE module as a whole should be kept as small as possible, the aim is for a small driver circuit size. For the comparison, "+" indicates a small sized driver circuit, "+-" indicates a medium sized driver circuit, and "-" indicates a large sized driver circuit.
- **Accuracy:** to control the UVC dose applied to the filter material, it is important that the radiant flux is controlled accurately. Clearly, any variation in the LED current-voltage characteristic not accounted for in the circuit negatively affects the accuracy. For the comparison "+" indicates high accuracy, "+-" indicates medium accuracy, and "-" indicates low accuracy.
- **Safety:** safety of the driver circuit mostly comes down to conforming to safety regulations. For comparison, "+" indicates the circuit is extremely safe, "+-" indicates the circuit is safe, and "-" indicates the circuit nears the border of (some) safety regulations.
- **EMI:** as previously mentioned, electromagnetic interference should be kept as small as possible so as to comply with EMC standards. For comparison, "+" indicates low potential for EMI generation, "+-" indicates medium potential for EMI generation, and "-" indicates high potential for EMI generation.

In Table 5.1 the different design criteria for the driver circuit have been placed with their relative scores (+, +-, -) for each implementation option. It can be observed in this table that the boost-mode driver circuit is ranked the highest on most design criteria, except for three criteria.

For one, the circuit size could potentially be relatively large. This is because of the large inductor size and because a driver chip is needed that automatically regulates the duty-cycle (refer to Section 4.7.1).

The second low scoring design criteria is safety. Whereas the other two driver circuits mostly have to deal

Table 5.1: Table in which the design criteria of the various UVC LED array drivers are compared.

Design criteria	Circuit 1	Circuit 2	Circuit 3
Power dissipation	-	-	+
Efficiency	-	-	+
Circuit size	+-	-	+-
Accuracy	-	+	+
Safety	+-	+-	-
EMI	+	+	-



with regulating the heat due to their relatively low efficiency, the boost-mode circuit has the risk of high voltage signals. As mentioned in Section 5.1.3, the output voltage should be limited to 40 V. Due to the potential high-voltage gain of the boost-mode circuit, it is important to ensure this limit is not exceeded.

Finally, driver circuit 3 scored relatively low on the EMI design criteria. This has already been discussed in Section 5.1.3.

Taking these arguments into consideration, it was decided to further expand the implementation of the boost-mode driver circuit and develop a complete implementation. This is discussed in the rest of the chapter.

### 5.3. Component selection

The aim for this section is to select components for the boost-mode driver circuit, which was chosen in the previous section. The driver circuit with the entire UVC LED array is shown in Figure 5.6. The figure does not yet show the LED driver chip, as this is discussed later in the chapter.

Note that there are four LED strings each containing 6 LEDs. This makes for a total of 24 LEDs and not the 26 LEDs which was found to be optimal ( $2 \times 13$  UVC LEDs per optimal LED structure, as discussed in Chapter 4). The reason is that 24 is divisible by 4, such that each string contains an even 6 UVC LEDs. If 26 LEDs are to be driven, there has to be 13 strings of 2 LEDs in series. Alternatively, two driver chips could be used, each with 2 LED strings: one with  $2 \times 6$  LEDs and the other with  $2 \times 7$  LEDs. However, this would increase the circuit size and power dissipation. Therefore, the choice was made to create two LED structures with 12 UVC LEDs each, thereby driving 4 strings of 6 UVC LEDs (see Figure 5.6).

The optimal LED structure with 12 UVC LEDs at a height of 15 mm from the filter surface has the following values for the optimal decision variables (refer to Section 4.7):

$$\begin{aligned} n_{\text{ring}} &= 1 \\ \vec{n}_{\text{LED}} &= 11 \\ \vec{r}_{\text{ring}} &= 30 \text{ mm} \\ \vec{\theta}_{\text{tilt}} &= 14^\circ \\ h_{\text{LED}} &= 15 \text{ mm} \end{aligned}$$

The corresponding objective function value, minimum irradiance, and overdose values are:

$$\begin{aligned} f &= 20.2 \\ E_{\text{min}} &= 0.593 \text{ mW/cm}^2 \\ O &= 1.43 \end{aligned}$$

All these values still align with the requirements set in Section 2.2. Therefore this reduction in the number of LEDs is deemed appropriate.

With the UVC LED array now fully complete, the circuit specifications can be determined, as is done in Section 5.3.1. After this, the main components for the boost-mode LED driver can be selected. The selection procedure for the passive components is outlined in Section 5.3.2. Finally, an LED driver chip is selected and discussed in Section 5.3.3.

#### 5.3.1. Assumed circuit specifications

To compute all component values, several circuit specifications have to be assumed beforehand. Then, with the components selected, the actual circuit specifications can be computed. In this section, the assumed circuit specifications are discussed.

##### Input voltage

As has been mentioned earlier in this chapter, the voltage available to the LED driver is 15 V. Therefore, the following can be stated:

$$V_{\text{in}} = 15 \text{ V} \quad (5.6)$$

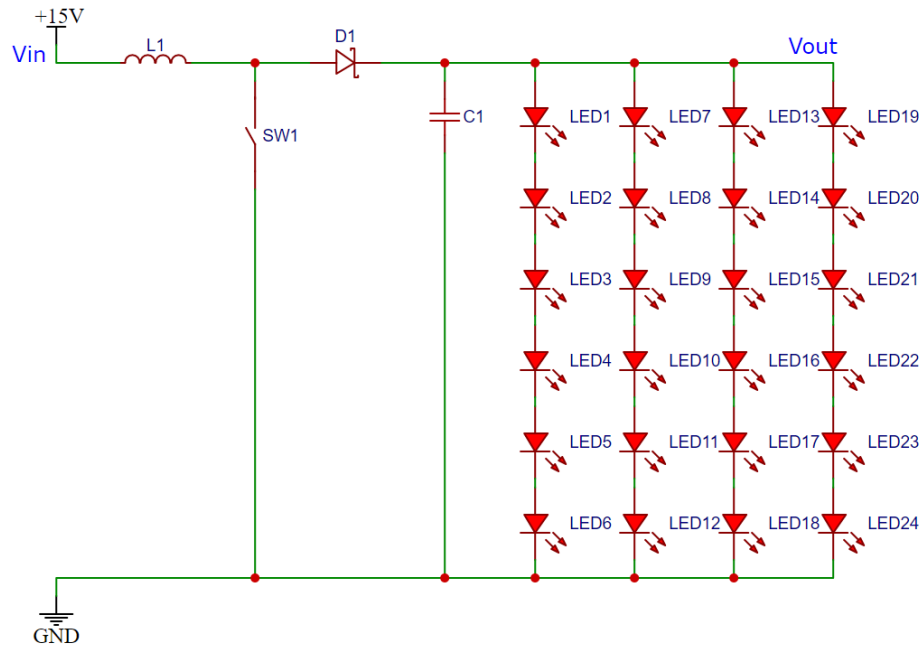


Figure 5.6: The final boost-mode LED array driver excluding an LED driver chip.

### UVC LED operating conditions

The boost-converter circuit, as shown in Figure 5.6, controls the UVC dose provided by the LEDs differently than the transmittance amplifiers (discussed in Section 5.1.2). It was explained that the transmittance amplifiers change the intensity of the LEDs by controlling the current through them. This is harder to do effectively for the boost-converter circuit. Instead, the on-time of the LEDs is changed, thereby in turn changing the applied UVC dose. This means that the current through the LED strings and voltage over them are always held constant during the on-time. For this, an operation condition has to be determined for the UVC LEDs, such that these are (ideally) always operating at the same voltage and current. For this, the typical forward voltage and forward current of the LA KL120U8F are selected [39]. In conclusion, it is assumed that:

$$V_{\text{LED}} = 5.7 \text{ V} \quad (5.7)$$

$$I_{\text{LED}} = 20 \text{ mA} \quad (5.8)$$

### Output voltage and output current

At the beginning of this section it was concluded that 24 UVC LEDs were to be used in the LED array. These were divided into 4 strings with each 6 LEDs. From the LED operating conditions determined above, it can be concluded that the voltage over the strings of LEDs is computed as in Equation (5.9). Moreover, the current through the LED array is computed as in Equation (5.10).

$$V_{\text{out}} = 6 \times 5.7 \text{ V} = 34.2 \text{ V} \quad (5.9)$$

$$I_{\text{out}} = 4 \times 20 \text{ mA} = 80 \text{ mA} \quad (5.10)$$

### Schottky diode forward voltage

To determine the correct electrical characteristics of the boost-mode converter, the forward voltage of Schottky diode *D1* (Figure 5.6) needs to be accounted for. The estimated forward voltage is:

$$V_{\text{F}} = 0.4 \text{ V} \quad (5.11)$$

### Switching frequency

The switching frequency of the boost-converter circuit is relatively complex to determine. This is because there are a multitude of variables dependent on the frequency. It is based on trade-offs between size, cost, and efficiency. In general, the higher the frequency, the smaller and cheaper the inductors. However, a higher

frequency gives rise to greater switching losses [51]. As will become apparent during the inductor selection, a frequency of 1 MHz results in an adequately sized inductor, and acceptable efficiency.

$$f_s = 1 \text{ MHz} \quad (5.12)$$

### Converter efficiency

As with the diode forward voltage selection, the converter efficiency has to be estimated to compute reasonably accurate component values. Typically, the efficiency can be determined by looking into the datasheet of the LED driver. The LED driver selection is explained in a later section, but its datasheet is used to get a rough estimate of the driver efficiency [51]. At an output current of 80 mA, the efficiency is approximately:

$$\eta = 87\% \quad (5.13)$$

### Switch duty-cycle

With the previous determined values, the required duty-cycle can be calculated to produce an output voltage listed in Equation (5.9). This is done as follows [52]:

$$D = 1 - \frac{\eta \times V_{in}}{V_{out} + V_F} \approx 0.62 \quad (5.14)$$

### Average inductor current

The average inductor current is needed to compute various component characteristics in Section 5.3.2. As with the duty-cycle, it can be determined using previously calculated specifications, which is shown in Equation (5.15).

$$I_L = \frac{V_{out} \times I_{out}}{\eta \times V_{in}} \approx 210 \text{ mA} \quad (5.15)$$

### Inductor ripple current

The inductor ripple current is usually determined as a percentage of the average inductor current. In general, the ripple current is taken to be 20% to 40% of  $I_L$  [52]. The main reason is that the output voltage ripple increases with increasing inductor ripple current. The increase of voltage ripple is due to the equivalent series resistance (ESR) of the output capacitor  $C1$  [53], which is further explained when selecting the output capacitor. If a relatively low ESR is assumed, the inductor ripple current can be increased slightly, thereby reducing the inductance value of the inductor. The final ripple current is thus assumed to be 50% of the average inductor current:

$$\Delta i_{L,max} \approx 105 \text{ mA} \quad (5.16)$$

### Output voltage ripple

The last specification to be determined is at the same time one of the most important ones. This is due to the fact that the output voltage ripple directly effects the ripple current through the LED strings. The objective is to keep the LED current as constant as possible, such that the UVC radiant flux does not vary too much. Therefore the maximum output voltage ripple is assumed to be:

$$\Delta v_{out,max} = 20 \text{ mV} \quad (5.17)$$

## 5.3.2. Passive component selection

With all circuit specifications outlined in the previous subsection, the components of the circuit shown in Figure 5.6 can be determined. Chronologically, the components discussed are the inductor  $L1$ , the Schottky diode  $D1$ , and finally the output capacitor  $C1$ .

### Inductor

The main properties of the inductor to be determined are the inductance  $L$ , the saturation current  $I_{sat}$ , and the equivalent series resistance,  $ESR_L$ .

Using the previously obtained circuit specifications, the minimum inductance can be computed as shown

in Equation (5.18) [52]. Here it can be seen that the computed inductance is approximately 86  $\mu\text{H}$ . This is rounded up to 100  $\mu\text{H}$ , such that a slightly lower inductor ripple current is obtained.

$$L = \frac{D \times V_{\text{in}}}{\Delta i_L \times f_s} \approx 86 \mu\text{H} \rightarrow 100 \mu\text{H} \quad (5.18)$$

Next is the inductor saturation current. As the name implies, this is the current at which the inductor saturates, thereby losing its inductance [54]. As a rule of thumb, the minimum saturation current is taken as 30% above the maximum inductor current. Summarized, this leads to the following expression:

$$I_{\text{sat,min}} = 1.3 \left( I_L + \frac{1}{2} \Delta i_L \right) \approx 340 \text{ mA} \quad (5.19)$$

Lastly, the equivalent series resistance (ESR) should be kept to a minimum, such that the power dissipation inside the inductor is reduced. Typically an ESR of 300  $\text{m}\Omega$  is favored:

$$ESR_{L,\text{min}} = 300 \text{ m}\Omega \quad (5.20)$$

The inductor that was chosen, which complies with all these specifications, is the *SRR7045-101M* from Bourns [55]. This inductor has the following properties:  $L = 100 \mu\text{H}$ ;  $I_{\text{sat}} = 500 \text{ mA}$ ;  $ESR_L = 300 \text{ m}\Omega$ . Finally, the size of the inductor is  $7 \times 7 \times 4.5 \text{ mm}$ , which is a reasonably sized inductor for this application.

With the newly calculated inductance, a new estimate can be made of the inductor ripple current by rewriting Equation (5.18). The resulting ripple current is shown in the equation below.

$$\Delta i_L = \frac{D \times V_{\text{in}}}{L \times f_s} \approx 93 \text{ mA} \quad (5.21)$$

### Schottky diode

There are various conditions involved in selecting the right diode *D1* in the boost-mode circuit shown in Figure 5.6. One of the most important diode characteristics is the forward voltage  $V_F$ . In the Assumed circuit specifications section, a diode forward voltage of 0.4 V was selected. A typical rectifier diode would have a forward voltage around 0.6 - 0.7 V. However, to decrease power dissipation, the forward voltage should be kept as small as possible. The reason why Schottky diodes are often used in a boost-mode converter, is that they generally have a lower forward voltage than the average rectifier diode.

The selected Schottky diode should be able to withstand the maximum current through the diode during an entire switching cycle [52]. As a rule of thumb, the rated forward current of the diode should be 30% higher than the maximum current:

$$I_{F,\text{min}} = 1.3 \left( I_L + \frac{1}{2} \Delta i_L \right) \approx 340 \text{ mA} \quad (5.22)$$

The second diode characteristic to take into account is the reverse breakdown voltage, which is the maximum rated reverse voltage of the diode. The maximum reverse voltage in a boost-mode converter circuit is equal to the maximum output voltage. Using the 30% rule of thumb mentioned earlier, the minimum reverse breakdown voltage is:

$$V_{R,\text{min}} = 1.3 \left( V_{\text{out}} + \frac{1}{2} \Delta v_{\text{out}} \right) \approx 45 \text{ V} \quad (5.23)$$

The Schottky diode that was chosen for this application and that complies with the above specifications is the *MBRA160T3G* from ON Semiconductor [56]. This diode has the following properties:  $V_F = 0.4 \text{ V}$ ;  $I_F = 1 \text{ A}$ ;  $V_R = 60 \text{ V}$ .

### Capacitor

The last component still to be selected is the output capacitor *C1* as shown in Figure 5.6. To do so, various capacitor properties need to be determined first.

The capacitance is established as seen in Equation (5.24) [52]. Note that the minimum capacitance required

to produce an output voltage ripple of 20 mV is approximately 2.5  $\mu\text{F}$ . To account for any extra losses, the capacitance is rounded up to 4.7  $\mu\text{F}$ .

$$C = \frac{D \times I_{\text{out}}}{\Delta V_{\text{out}} \times f_s} \approx 2.5 \mu\text{F} \rightarrow 4.7 \mu\text{F} \quad (5.24)$$

The equivalent series resistance of the capacitor ( $ESR_C$ ) is the next characteristic that needs to be determined. From Equation (5.25) it can be seen that the output voltage ripple depends linearly on the ESR [52]. Rewriting this, the minimum equivalent series resistance can be computed for the output voltage ripple assumed in Section 5.3.1. This computation is shown in Equation (5.26).

$$\Delta v_{\text{out}} = ESR_C \left( I_L + \frac{1}{2} \Delta i_L \right) + \frac{D \times I_{\text{out}}}{C \times f_s} \text{ [V]} \quad (5.25)$$

$$ESR_{C,\text{min}} = \frac{\Delta V_{\text{out}} - \left( \frac{D \times I_{\text{out}}}{C \times f_s} \right)}{I_L + \frac{1}{2} \Delta i_L} \approx 40 \text{ m}\Omega \quad (5.26)$$

The final property of the output capacitor that needs to be taken into account is its rated voltage, or the absolute maximum voltage that can be applied. As before, the rated voltage should be at least 30% above the maximum voltage to be applied to the capacitor during operation. This comes down to:

$$V_{\text{rated,min}} = 1.3 \left( V_{\text{out}} + \frac{1}{2} \Delta v_{\text{out}} \right) \approx 45 \text{ V} \quad (5.27)$$

The capacitor that was chosen and that complies with all the listed specifications is the *GRM21BR61H475KE51L* from Murata [57]. This capacitor has the following properties:  $C = 4.7 \mu\text{F}$ ;  $ESR_C = 10 \text{ m}\Omega$ ;  $V_{\text{rated}} = 50 \text{ V}$ .

Now that the a capacitor has been selected, a new estimate can be made of the output voltage ripple using Equation (5.17). The new estimate is calculated using the selected capacitance and the previous estimate of the inductor ripple current. The result is as follows:

$$\Delta v_{\text{out}} \approx 13 \text{ mV} \quad (5.28)$$

### 5.3.3. LED driver chip selection

With all components selected for the boost-converter circuit shown in Figure 5.6, a working LED driver can be built. However, a problem with this circuit by itself is that the current has to be set by computing the correct output voltage and the corresponding duty-cycle. This can be improved by using a dedicated LED driver chip. Such a chip regulates the duty-cycle of  $SW1$  based on the current through the individual LED strings. The circuit from Figure 5.6 can stay as it is, along with the selected passive components. However, only the switch is swapped out for an LED driver chip.

To select the right LED driver chip for this application, some properties of the driver chip should be determined, such that the range of available chips is narrowed down. The properties discussed hereafter are the channel count, maximum output voltage, switch current limit, and soft-start option.

#### Channel count

At the beginning of this section it was determined to use 4 strings of LEDs. Therefore, the LED driver chip should have four channels for measuring and controlling the current through the individual strings.

#### Maximum output voltage

In Subsection 5.3.1, it was found that an output voltage of 34.2 V is necessary to drive 6 LA KL120U8F UVC LEDs in series. Then, the maximum output voltage that the chip should be able to withstand is this output voltage plus half the output voltage ripple. As a rule of thumb, the maximum output voltage is taken as 30% above this value:

$$V_{\text{chip,max}} = 1.3 \left( V_{\text{out}} + \frac{1}{2} \Delta v_{\text{out}} \right) \approx 44 \text{ V} \quad (5.29)$$

### Switch current limit

As mentioned at the beginning of this section, the switch in Figure 5.6 is substituted by the LED driver chip. However, this chip then should be able to switch the current going through the inductor when it is charging. Taking the same 30% rule of thumb, the maximum switch current limit is:

$$I_{\text{sw,max}} = 1.3 \left( I_L + \frac{1}{2} \Delta i_L \right) \approx 340 \text{ mA} \quad (5.30)$$

### Soft-start

A soft-start option is a very useful feature for an LED driver chip to have. The reason is that during startup, the output capacitor is assumed to be empty. This being the case, the capacitor acts as a short circuit for a short duration. However, this is enough to cause an inrush current that may damage any components in its way. Soft-start is a method to limit this inrush current by programming a switch current ramp-up time [51]. This feature is thus taken to be a requirement for the driver chip.

With these properties now determined, a chip can be selected that complies with all of them. In the end, the *LT3599* from Linear Technology was chosen [51]. This chip features 4 LED string channels; an output voltage up to 44 V; 2.5 A switch current limit; and a soft-start option. The total schematic for the boost-mode UVC LED array driver that features the *LT3599* is included in Appendix B.

Before moving on to testing the circuit, three properties of the driver circuit need to be briefly explained.

### Current matching

The current matching ability of the *LT3599* is a great advantage of this LED driver chip, as it increases the radiant flux accuracy. Because, the four LED channels of the *LT3599* are controlled such that there is minimum variation of current among them. The datasheet states that the LED string current match is at most  $\pm 1.5\%$  between the channels [51]. This serves to reduce any variation in radiant intensity due to the variation in LED characteristics.

### OPENLED and SHORTLED

For extra security, the *LT3599* provides two output signals which can be interpreted by the main controller in the SPPE. The first of these is the OPENLED signal, which is pulled down when any LED string is detected to be open. The second flag is the SHORTLED signal, which is pulled down when a short circuit is detected in one of the LED strings. In addition, the LED string, in which the short circuit was detected, is immediately disabled [51].

### Shutdown

The third interesting feature of the *LT3599* is the shutdown pin. As the name implies, when this pin is pulled down, the entire chip is shutdown, which reduces the quiescent current to only  $1 \mu\text{A}$  [51]. This function is used whenever the required dose has been provided to the filter material. It has been mentioned at the beginning of Section 5.3.1, that the UVC radiant intensity is not changed in the boost-mode driver circuit. Instead, the period that UVC radiation is applied is varied in order to change the UVC dose. The radiation time is timed by the controller and when the correct dose has been applied, the driver chip is put into shutdown mode. This way, both the dose can be accurately controlled and the chip is only active when needed, thereby increasing efficiency.

## 5.4. Circuit analysis

At this point, the complete LED driver circuit has been designed and all its components have been selected. For this reason, an analysis is done of the total circuit. The first part of this section describes the results from SPICE simulations of the driver circuit. The second part of this section discusses the heat generation of the UVC LEDs.

### 5.4.1. SPICE simulations

To test if the total LED driver circuit functions correctly, simulations of the circuit are performed in LTspice XVII simulator software. For this, all components have been imported as SPICE models, such that accurate simulation results are obtained. The total schematic in LTspice is shown in Appendix C.

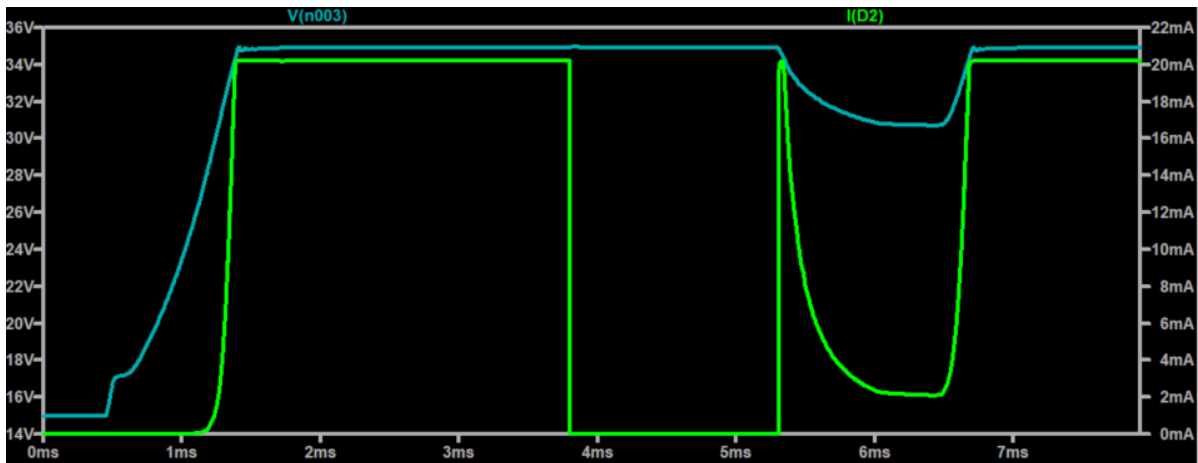


Figure 5.7: Two SPICE simulation plots showing the voltage across the LED strings (cyan) and the current through one of the strings (green).

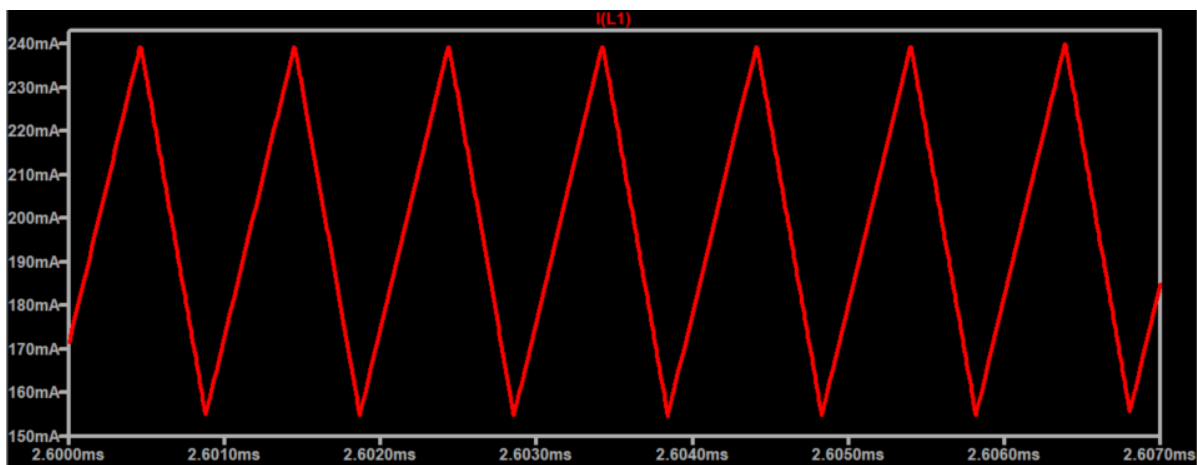


Figure 5.8: SPICE simulation plot showing the current through the inductor.

The first electrical properties that are simulated are the general waveforms for the LED string current and LED string voltage. These are shown in Figure 5.7. First, it should be observed that the current through the LED strings is at a steady level around 1.3 ms, which is due to the soft-start function inside the LT3599. Both voltage and current are then steady for a period of around 2.6 ms, after which the LED channels are disabled. This is exactly the time at which the SHUTDOWN pin is pulled down, and thus when the LT3599 shuts down. Note that the voltage stays constant until the SHUTDOWN pin is pulled high again at 5.3 ms. The reason is that the charge stored in the output capacitor does not have a closed path such that the capacitor is discharged. When the SHUTDOWN pin is pulled high at 5.3 ms, the charge stored in the capacitor finally does have a closed path through the LED strings. This causes the short current spike in the LED string current. Finally, after the 1.3 ms soft-start, normal operation continues at 6.6 ms.

Note that the average output voltage over the LED strings during normal operation is about 35 V, which is slightly higher than the estimated 34.2 V (refer to Section 5.3.1). This is most likely due to the current controller inside the LT3599, such that there is a small voltage present at the LED string inputs of the chip. When measuring the voltage across the LED strings directly, a voltage of approximately 34.2 V is found, which matches the estimated value almost exactly. Furthermore, it can be observed that the current through a single LED string is approximately 20 mA, which closely matches the estimated LED current.

The second waveform of interest is the average current and ripple current through the inductor. Figure 5.8 shows the current through the inductor under normal operation of the LED driver. As estimated, the average inductor current is close to 210 mA. Next, the inductor ripple current can be determined from this plot, which is equal to the peak to peak current:  $240 \text{ mA} - 155 \text{ mA} = 85 \text{ mA}$ . This is about 8 mA lower than the estimate computed in Section 5.3.2.

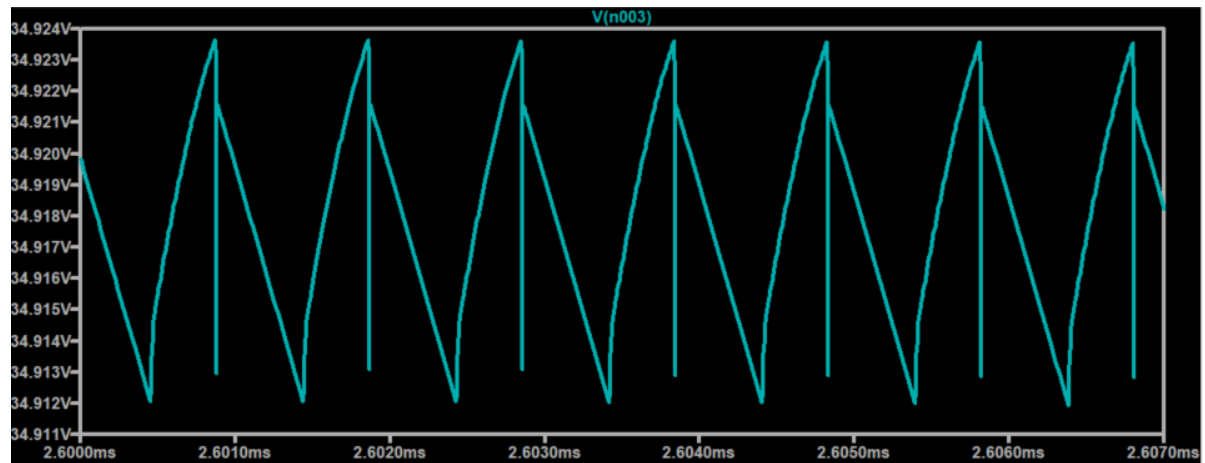


Figure 5.9: SPICE simulation plot showing a zoom-in of the LED string voltage.

The third interesting waveform is that of the LED string voltage, or rather the voltage ripple thereof, which can be seen in Figure 5.9. As with the ripple current, the output voltage ripple is equal to the peak to peak voltage:  $34.923\text{ V} - 34.911\text{ V} = 12\text{ mV}$ . This is almost exactly equal to the voltage ripple estimate in Section 5.3.2.

With the simulation in place, the efficiency of the power transfer to the UVC LEDs can be computed. This computation is shown in Equation (5.31). The efficiency in this simulation appears to be slightly higher than the estimated 87% found in Equation (5.13). It should be mentioned however that this calculation does not include the power consumption of the LED driver itself. Unfortunately, this cannot be determined using the SPICE simulation and SPICE models currently used.

$$\eta = \frac{V_{\text{out}} \times I_{\text{out}}}{V_{\text{in}} \times I_L} \times 100\% \approx 91.2\% \quad (5.31)$$

#### 5.4.2. LED heat generation

The heat generation in the LEDs will undoubtedly cause an increase in the ambient temperature within the SPPE. This increase is investigated to ensure that there are no safety risks introduced in the SPPE.

There is no direct translation from the the power dissipated in watt, to the heat generated in °C, without knowing the heat capacity of the object. This parameter is not known for the LA KL120U8F. However, it is possible to determine the increase in ambient temperature at the junction of a resistor, and by extend the LED. By making use of the thermal resistance [K/W] (a parameter found in the datasheet of the LA KL120U8F [39]), it is possible to determine the increase in temperature for one LED at the junction.

First the assumption is made that all power delivered to an LED is either used to generate the radiant flux, or converted into heat. The total power delivered to a single LED is equal to  $(5.7\text{ V} \times 20\text{ mA}) = 114\text{ mW}$ . And according to the datasheet [39], the radiant flux of the LA KL120U8F is equal to 3.5 mW. Thus, the average power converted into heat will be  $114\text{ mW} - 3.5\text{ mW} = 110.5\text{ mW}$ , for a single LED.

This is then multiplied by the 'junction-ambient thermal resistance' ( $\Theta_{JA}$  [K/W]) of the LED, to get an expression for the increase in temperature (refer to Equation 5.32).

$$\Delta T = \Theta_{JA} \times P_{\text{heat}} \quad [\text{K}] \quad (5.32)$$

For the LA KL120U8F, this results in  $110.5\text{ mW} \times 21\text{ K/W} \approx 2.3\text{ K}$ .

It is unknown how this temperature increase at the junction will translate to the ambient temperature of the SPPE. This could also not be tested due to the restrictions imposed on the project as a result of the COVID-19 pandemic. However, the total power dissipation of all 24 LEDs will be about 2.7 W, which is not deemed to be excessive. Additionally the LED will be periodically turned on over an interval of 2 hours. This leaves room for the LEDs to cool-down in between delivering bursts of UV radiation.

Should experiments performed at a later time indicate that the temperature increase may be uncomfortable for the wearer, then it is possible to adjust the control algorithm for the LEDs. This would entail ensuring that the heat is only generated during the exhalation of the wearer, to immediately remove said heat.



# 6

## Discussion, conclusions, and recommendations

This thesis outlines the work done on the in situ disinfection system for the SPPE by use of UVGI. There is a brief introduction into the entire project of the SPPE and the current state of the art regarding UVGI in Chapter 1. Following this are the requirements for the SPPE as a whole, and the UVGI subsystem specifically. Chapter 3 gives an introduction into the concept and workings of UVGI, as well as presenting several safety regulations.

Chapter 4 and Chapter 5 contain the main design implementations of the UVGI. Chapter 4 provides a detailed description of the UV LED simulation. Herein, all parameters used are discussed including the intensity distribution, the LED tilt angle, the reflectivity of the filter housing, and the filter surface area. Additionally the chapter provides the LED selection and the optimization of the entire LED structure.

The LEDs in the optimal structure also need to be driven for which the LED driver circuit is designed, refer to Chapter 5. The basic implementation of the driver circuit is discussed first, followed by a few more refined designs. These are then compared, after which one will be used for the implementation of the driver circuit. The circuit is designed, the components for the circuit are selected, and the entire circuit is simulated in LTspice. This also leads to some alterations in the LED structure. The final design step includes the design of a PCB, which is visualized in Appendix B.

### 6.1. Discussion

With all the design steps completed, the project is returned to the program of requirements to assess the design steps made. First, the general requirements for the entire SPPE are listed in Table 6.1, including whether the requirement has been met (Yes) or not (No), or if it is not applicable (N/A).

Two requirements are not applicable for this thesis. Replacing the mechanical filter is determined by the design of the filter module, not the LED structures. Additionally the battery life is covered by the OBPM group and falls outside the scope of this thesis.

Then there are two requirements which have not yet been met. The component lifetime of the UVGI subsystem has not been researched in-depth. The component lifetime was not provided in the datasheet of the LA KL120U8F, which is the main component of the UVGI subsystem. Therefore it was deemed that a valid argumentation could not be provided at this point, more on this in Section 6.3. Secondly, communicating the internal status of the UVGI subsystem to the user has not been implemented yet. Again, more on this in Section 6.3. The rest of the general requirements for the SPPE have all been met by the UVGI subsystem.

Table 6.1: Assessment of the SPPE design requirements.

Requirement	Pass
The SPPE uses UVGI to sterilize the filter material.	Yes
The SPPE's mechanical filter is replaceable.	N/A
The SPPE's electronics are reusable after the mechanical filter is replaced.	Yes
The SPPE's battery capacity is sufficient for at least eight hours.	N/A
The SPPE is designed preferably using off-the-shelf components, unless it is absolutely necessary.	Yes
The SPPE's component lifetime is at least 2 years or 8000 hours.	No
The SPPE design is circular.	Yes
The SPPE should communicate internal status to the user.	No
The SPPE filter module should be airtight apart from the dedicated airflow pathways.	Yes
The SPPE filter module should be small enough to fit on the side of a PPE.	Yes

In the assessment of the mandatory and trade-off requirements of the UVGI subsystem the respective section where the requirement is discussed is listed as well. The mandatory requirements are listed in Table 6.2. The mandatory requirements include the boundary conditions for the UVGI subsystem to function properly. The requirements with regards to the irradiation of the filter have been covered in Chapter 4, where the simulation was designed to take the mandatory requirements into account. The rest of the requirements are related to the circuit design, which is covered in Chapter 5. Here, the circuit is designed keeping the mandatory requirements as boundary conditions. As can be seen in the table, all these requirements have been met.

Table 6.2: Assessment of the UVGI mandatory requirements.

Requirement	Pass	Section
The UVGI driver should be operational at a source voltage of 15V.	Yes	5.3.1
The current drawn by the UVGI driver circuit must not exceed 500 mA when a dose is applied.	Yes	5.3.1
The UVGI subsystem must provide an overdose ( $E_{\max}/E_{\min}$ ) on the filter material equal to, or lower than, 1.5.	Yes	4.7.3
The UVGI subsystem must be able to deliver a UVC radiation dose of 305 mJ/cm <sup>2</sup> .	Yes	4.7.2
The UVGI subsystem must be able to deliver said dose every 2 hours.	Yes	4.7.2
The UVGI subsystem must be able to deliver said dose a minimum of 4 times a day on one operation cycle.	Yes	4
The UVGI subsystem should be able to adjust the dose based on signals originating from the Sensor and Control subsystem.	Yes	5.2
The distance from the UVC LEDs to the filter material may not exceed 15 mm.	Yes	4.7.3

Finally, there are the trade-off requirements in Table 6.3. The trade-off requirements have no set values, except for the boundaries provided by the mandatory requirements. However, since the SPPE is a wearable, there is only a limited amount of space and resources available to make the device function properly. Therefore, it is important to take these requirements into account during every design decision. All requirements have been met, meaning that none exceed the maximum limits and all are deemed to be of sufficient quality.

Table 6.3: Assessment of the UVGI trade-off requirements

Requirement	Pass	Section
The UVGI subsystem should provide a radiation upon the filter surface as uniformly as possible.	Yes	4.7.3
The UVGI subsystem should generate as little heat as possible.	Yes	5.4.2
The UVGI subsystem should have minimal energy consumption.	Yes	4.7.3
The implementation of the UVGI subsystem should have a minimal form factor.	Yes	5.2

## 6.2. Conclusions

Overall the the UVGI subsystem appears to be a proper functioning system which can provide the in situ disinfection for the SPPE. The UVGI subsystem has fulfilled almost all requirements that were set for it, including uniform radiation and control over the applied dose. To return to the original goal of this thesis:

*Designing an energy efficient implementation of UVGI for application in a wearable device.*

We conclude that this goal has been achieved with the design presented in this thesis. With the design being verified to the best of our abilities without using a prototype.

This thesis provides a complete simulation in MATLAB which can be used to determine the radiation on a round surface with a multitude of parameters which provides a wide range of customization. Additionally we have implemented a circuit design which can be used to control the LEDs. But, most importantly, this thesis shows that it is possible to use UVGI in a very small form factor. UVGI had not been used in anything smaller than a sterilization box for face masks, but the SPPE makes use of the concept in just a single filter module. This also provides new entry ways for different applications, like a ventilator for example. Sterilization is going to be needed a lot more over the coming years, especially if another pandemic might break out. UVGI can be a great method to provide this sterilization and now it can be used even more.

## 6.3. Recommendations

Despite the overall good results for the UVGI subsystem, there are still topics which can be expanded upon, or more thoroughly investigated. Some of these have already been briefly mentioned in the preceding chapters. However, there will be a complete overview presented here.

One of the main drawbacks regarding the Bachelor Graduation Project as a whole was the restriction on the development of a prototype. While this has allowed the entire project to be better supported by theoretical research, it did also inhibit research into a couple design choices. The first of which is the effect of structural degradation caused by UV radiation. As was discussed in Section 3.1, different materials will degrade differently depending on the dose of UV radiation which they receive. Should a prototype ever be constructed, it is advised to perform tests on the structural integrity of the SPPE per material used for extended periods of time.

There is also the limiting factor of costs. The safety regulations listed in Section 3.2 are placed behind paywalls which are deemed to high for a proof-of-concept bachelor thesis. However, safety is, and should always be, a priority which cannot be overlooked. Therefore, should the SPPE ever be taken into production, it is advised to retrace the design decisions made in this work and confirm that they are in line with the appropriate regulations. Additionally, should there be aspects which have not been considered in this thesis, which should be present for a legal approval, then they should be investigated upon and added where necessary.

There has also been work done on a different implementation of the tilt angle calculation for the LED simulation. Unfortunately there was not enough time to cross check the code for this implementation, upon which it was decided to not include it in this thesis. Therefore it is unclear if there are advantages compared to the current implementation (regarding computation time and precision). This implementation is something which could be further explored in future works, or determined experimentally when possible.

One important parameter for the entire filter module is the radius of the filter. The radius determines the area which needs to be irradiated, and thus by extend the number of LEDs which need to be used, but is also one of the major form factor limitations. There are regulations in place regarding the maximum air resistance of the filter, which are covered in Section 4.6, which directly influences the filter surface size. Unfortunately it is not possible to do the proper experiments required for determining the appropriate radius of the filter. Should this project ever be expanded upon, it is highly recommended to research a possible decrease in radius.

The driver circuit presented in Chapter 5 works properly, according to the simulation. Of course this would still need to be verified by a prototype, but there is also one adjustment which should be made in case the design of the driver circuit is redone. Instead of the approach taken now, the first step should be to design a nullor feedback circuit which regulates the current by changing the duty cycle accordingly and add different

functionalities from there. This might lead to different circuit implementation than covered in this thesis. It is certainly a design step which will be applied in future works.

The heat generated by the entire driver circuit is hard to determine without a prototype or in-depth knowledge of thermodynamics. In Section 5.4.2 it was found that the total power dissipated as heat will be equal to 2.7 W. However, to what extent this results in an ambient temperature increase is unknown. What is known is that at the junction, a single LED will generate 2.3 K above ambient temperature. It is advised to perform tests on the increase in ambient temperature due to the UVGI subsystem, once such experiments can be performed. Should the temperature increase pose a problem, it is possible to adjust the algorithm controlling the LEDs to only radiate UV during exhalation.

Then there are two general SPPE requirements which have not been met yet. The first of which is the component lifetime. Since this information was not readily available for the LA KL120U8E, and since this LED is the main component of the UVGI subsystem, the main conclusion which should be drawn is that this lifetime should be investigated upon in future works.

The second design requirement of the SPPE which has not been implemented yet in the UVGI subsystem, is the communication of the internal status to the user. This is an aspect of the design which was planned to be done with the use of LEDs at the surface of the filter module, indicating when the UVGI subsystem was providing a UV dose. However it has not yet been implemented in the driver circuit. This is an addition which certainly should be added in a second version.

# Bibliography

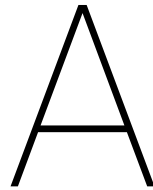
- [1] M.-W. Wang, M.-Y. Zhou, G.-H. Ji, L. YE, Y.-R. Cheng, Z.-H. Feng, J. Chen, "Mask crisis during the COVID-19 outbreak," *European Review for Medical and Pharmacological Sciences*, vol. 24, pp. 3397–3399, 2020.
- [2] D. Wang, M. Zhou, X. Nie, W. Qiu, M. Yang, X. Wang, T. Xu, Z. Ye, X. Feng, Y. Xiao, W. Chen, "Epidemiological characteristics and transmission model of Corona Virus Disease 2019 in China," *Journal of infection*, vol. 80, no. 5, pp. 25–27, 2020.
- [3] W. Kowalski, *Ultraviolet Germicidal Irradiation Handbook, UVGI for Air and Surface Disinfection*. Berlin, Heidelberg: Springer-Verlag, 2009.
- [4] J. C. H. Chang et al, "UV inactivation of pathogenic and indicator microorganisms," *Applied and Environmental Microbiology*, vol. 49, no. 6, pp. 1361–1365, June 1985.
- [5] J. Kim, J. Jang, "Inactivation of airborne viruses using vacuum ultraviolet photocatalysis for a flow-through indoor air purifier with short irradiation time," *Aerosol Science and Technology*, vol. 52, no. 5, pp. 557–566, February 2018.
- [6] D.-K. Kim, D.-H. Kang, "UVC LED Irradiation Effectively Inactivates Aerosolized Viruses, Bacteria, and Fungi in a Chamber-Type Air Disinfection System," *Applied and Environmental Microbiology*, vol. 84, no. 17, September 2018.
- [7] BioShift. BioShift® Pass-Through UV-C Chamber – Large. (2020, June 15). [Online]. Available: <https://www.once.lighting/product/bioshift-large/>
- [8] Germicidallamp. UV4 AIR MOBIL 30 UV-C Lamp. (2020, June 15). [Online]. Available: [https://www.germicidallamp.nl/webshop/producten/detail/36/uv4-air-mobil-30--uv-c-lamp.html?gclid=Cj0KCQjwuJz3BRDTARIsAMg-HxVzgz8Yp6jUzpf2IP7gJUWsyKs0OGM2SjRn4PrSw\\_ZNO4PQWYXeVEaAh8pEALw\\_wcB](https://www.germicidallamp.nl/webshop/producten/detail/36/uv4-air-mobil-30--uv-c-lamp.html?gclid=Cj0KCQjwuJz3BRDTARIsAMg-HxVzgz8Yp6jUzpf2IP7gJUWsyKs0OGM2SjRn4PrSw_ZNO4PQWYXeVEaAh8pEALw_wcB)
- [9] K. J. Card et al, "UV sterilization of personal protective equipment with idle laboratory biosafety cabinets during the COVID-19 pandemic," *medRxiv*, April 2020.
- [10] Spectroline. UVGI SANITIZING CABINETS. (2020, June 15). [Online]. Available: <https://spectroline.com/germicidal-uv-cabinet/>
- [11] P.O. Nyangaresi, Y. Qin, G.L. Chen, B.P. Zhang, Y.H. Lu, L. Shen, "Comparison of the performance of pulsed and continuous UVC-LED irradiation in the inactivation of bacteria," *Water Research*, vol. 157, pp. 218–227, 2019.
- [12] S.S. Nunayon, H.H. Zhang, A.C.K. Lai, "Comparison of disinfection performance of UVC-LED and conventional upper-room UVGI systems," *Indoor Air*, vol. 30, no. 1, pp. 180–191, 2019.
- [13] G. Messina, M. Fattorini, S. Burgassi, M. Tani, Nante, G. Cevenini, "Novel UVC LED approach for disinfecting contact lenses," *European Journal of Public Health*, vol. 26, p. 459, 2016.
- [14] S. S. Salvi, "In this pandemic and panic of COVID-19 what should doctors know about masks and respirators?" April 2020, accessed: 07.05.2020. [Online]. Available: <http://apiindia.org/wp-content/uploads/pdf/corona-virus/review-article-on-mask.pdf>
- [15] 3M, "Comparison of ffp2, kn95, and n95 and other filtering facepiece respirator classes," May 2020, accessed: 07.0.2020. [Online]. Available: <https://multimedia.3m.com/mws/media/1791500O/comparison-ffp2-kn95-n95-filtering-facepiece-respirator-classes-tb.pdf>
- [16] L. J. Radonovich Jr et al, "N95 respirators vs medical masks for preventing influenza among health care personnel: A randomized clinical trial," *JAMA*, vol. 322, pp. 824–833, September 2019.

- [17] M. Loeb et al, "Surgical mask vs n95 respirator for preventing influenza among health care workers," *JAMA*, vol. 302, pp. 1865–1871, November 2009.
- [18] M. B. Lore, B. K. Heimbuch, T. L. Brown, J. D. Wander, S. H. Hinrichs, "Effectiveness of three decontamination treatments against influenza virus applied to filtering facepiece respirators," *The Annals of Occupational Hygiene*, vol. 56, pp. 92–101, January 2012.
- [19] R. J. Fischer et al, "Assessment of N95 respirator decontamination and re-use for SARS-CoV-2," *medRxiv*, April 2020.
- [20] Q. Zhang, Q. Zhao, "Inactivating porcine coronavirus before nuclei acid isolation with the temperature higher than 56 °c damages its genome integrity seriously," *bioRxiv*, February 2020.
- [21] K. Alger, H. Ip, J. Hall, S. Nashold, K. Richgels, C. Smith, "Inactivation of viable surrogates for the select agents virulent newcastle disease virus and highly pathogenic avian influenza virus using either commercial lysis buffer or heat," *Applied Biosafety*, vol. 24, no. 4, pp. 189–199, December 2019.
- [22] Author Unknown, "Guidelines on viral inactivation and removal procedures intended to assure the viral safety of human blood plasma products," World Health Organisation, WHO Technical Report 924, November 2004.
- [23] (2020) An introduction to uv-c led lighting for germicidal, sterilization and disinfection applications. Accessed: 22.04.2020. [Online]. Available: <https://www.waveformlighting.com/uv-c-led/uv-c-led-lighting-for-germicidal-sterilization-and-disinfection-applications>
- [24] M. E. R. Darnell, K. Subbarao, S. M. Feinstone, D. R. Taylor, "Inactivation of the Coronavirus that Induces Severe Acute Respiratory Syndrome, SARS-CoV," *Journal of Virological Methods*, vol. 121, pp. 85–91, August 2004.
- [25] C.-C. Tseng, C.-S. Li, "Inactivation of viruses on surfaces by ultraviolet germicidal irradiation," *Journal of Occupational and Environmental Hygiene*, vol. 4, no. 6, pp. 400–405, November 2007.
- [26] L. M. Casanova, S. Jeon, W. A. Rutala, D. J. Weber, M. D. Sobsey, "Effects of air temperature and relative humidity on coronavirus survival on surfaces," *Applied and Environmental Microbiology*, vol. 76, pp. 2712–2717, May 2010.
- [27] G. Byrns et al, "The Uses and Limitations of a Hand-Held Germicidal Ultraviolet Wand for Surface Disinfection," *Journal of Occupational and Environmental Hygiene*, vol. 14, no. 10, pp. 749–757, September 2017.
- [28] G. Zhang, Y. Xiao, J. Yan, N. Xie, R. Liu, Y. Zhang, "Ultraviolet light-degradation behaviour and antibacterial activity of polypropylene/zno nanoparticles fibres," *Polymers*, vol. 11, p. 1841, November 2019.
- [29] W. G. Lindsley et al, "Effects of ultraviolet germicidal irradiation (UVGI) on N95 respirator filtration performance and structural integrity," *Journal of Occupational and Environmental Hygiene*, vol. 12, pp. 509–517, August 2015.
- [30] M. Meinhardt, R. Krebs, A. Anders, U. Heinrich, and H. Tronnier, "Wavelength-dependent penetration depths of ultraviolet radiation in human skin," *Journal of Biomedical Optics*, vol. 13, no. 4, pp. 1 – 5, 2008. [Online]. Available: <https://doi.org/10.1117/1.2957970>
- [31] (2020) Skin anatomy and physiology. Accessed: 09.06.2020. [Online]. Available: [https://www.nuskin.com/en\\_ZA/corporate/company/science/skin\\_care\\_science/skin\\_anatomy\\_andphysiology.html#:~:text=The%20epidermis%20is%20the%20outermost,of%20one%20sheet%20of%20paper](https://www.nuskin.com/en_ZA/corporate/company/science/skin_care_science/skin_anatomy_andphysiology.html#:~:text=The%20epidermis%20is%20the%20outermost,of%20one%20sheet%20of%20paper).
- [32] J. R. Vig, "Uv /ozone cleaning of surfaces," *Journal of Vacuum Science & Technology*, vol. 3, no. 3, pp. 1027–1034, October 1984.
- [33] S. T. Summerfelt, "Ozonation and uv irradiation-an introduction and examples of current applications," *Aquacultural Engineering*, vol. 28, no. 2, pp. 21–36, June 2003.
- [34] E. Choi, Z. Tan, W. A. Anderson, "Formation of secondary organic aerosols by germicidal ultraviolet light," *Environments*, vol. 6, January 2019.

- [35] (2020) Examples by industry sector: Medical devices. Accessed: 10.06.2020. [Online]. Available: [https://www.iec.ch/perspectives/government/sectors/medical\\_devices.htm](https://www.iec.ch/perspectives/government/sectors/medical_devices.htm)
- [36] (2020) Standards by iso/tc 198, sterilization of health care products. Accessed: 10.06.2020. [Online]. Available: <https://www.iso.org/committee/54576/x/catalogue/p/1/u/0/w/0/d/0>
- [37] M. Bass, *Handbook of Optics: Volume I - Geometrical and Physical Optics, Polarized Light, Components and Instruments, Third Edition*. New York, Chicago, San Francisco, Lisbon, London, Madrid, Mexico City, Milan, New Delhi, San Juan, Seoul, Singapore, Sydney, Toronto: McGraw-Hill Professional, 2010, ch. 24.
- [38] I. Moreno, C.-Y. Tsai, D. Bermúdez, C.-C. Sun, "Simple function for intensity distribution from leds," *Proceedings of SPIE*, vol. 6670, September 2007.
- [39] *LA KL120U8F 3.5mW 275nm UVC LED Datasheet*, Light Avenue, 2019. [Online]. Available: <https://www.chips4light.com/assets/files/products/LA-KL120U8F-Version-1-v3.0.pdf>
- [40] (2020) Matlab 'fillmissing' documentation. Accessed: 17.06.2020. [Online]. Available: <https://www.mathworks.com/help/matlab/ref/fillmissing.html#d120e360281>
- [41] P. M. Janecek, "Optical reflectance measurements for commonly used reflectors," June 2009.
- [42] T. Quill, S. Weiss, C. Hirschler, V. Pankadzh, G. DiBattista, M. Arthur, J. Chen. Ultraviolet reflectance of microporous ptf. Porex Corporation. Accessed: 17.06.2020. [Online]. Available: <https://www.porex.com/wp-content/uploads/2020/04/Ultraviolet-Reflectance-of-Microporous-PTFE.pdf>
- [43] Reflective aluminum. Accessed: 17.06.2020. [Online]. Available: <http://www.anomet.com/reflective-aluminum>
- [44] NIOSH, "Statement of standard for chemical, biological, radiological, and nuclear (cbrn) full facepiece air purifying respirator (apr)," March 2003.
- [45] L. R. Johnson, *Essential Medical Physiology, Third Edition*. 84 Theobald's Road, London WCIX 8RR, UK: Elsevier Academic Press, 2003, ch. 19, p. 284.
- [46] T. Nagy, A. Jilek, J. Pecinka, "Air flow rate measurement with various differential pressure methods," in *2017 International Conference on Military Technologies (ICMT)*, June 2017, pp. 535–541.
- [47] D. W. Hendricks, *Water Treatment Unit Processes: Physical and Chemical*. 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, Florida, USA: CRC Press, Taylor & Francis Group, 2018, ch. D, p. 1153.
- [48] J. Burke, "Linear optimization, definitions: Sections 1–3," University of Washington, Tech. Rep., 2019. [Online]. Available: <https://sites.math.washington.edu/~burke/crs/407/PS/defn1-3.pdf>
- [49] T. L. M. S. P. Bradley, A. C. Hax, *Applied Mathematical Programming*. Addison-Wesley, 1977, ch. 5, pp. 180–183.
- [50] (2020) Norm: Iec 62368-1:2018 en. audio/video, informatietechnologie- en communicatietechnologie-apparatuur - deel 1: Veiligheidseisen. Accessed: 18.06.2020. [Online]. Available: <https://www.nen.nl/NEN-Shop/Norm/IEC-6236812018-en.htm>
- [51] *LT3599 4-string 120 mA LED driver datasheet*, Linear Technology, Milpitas, CA, 2009. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/3599ff.pdf>
- [52] *SLVA372C - Basic Calculation of a Boost Converter's Power Stage*, Texas Instruments, Dallas, TX, 2014. [Online]. Available: <https://www.ti.com/lit/an/slva372c/slva372c.pdf?ts=1587975346240>
- [53] *SLVA061 - Understanding Boost Power Stages in Switchmode Power Supplies*, Texas Instruments, Dallas, TX, 1999. [Online]. Available: <https://www.ti.com/lit/an/slva061/slva061.pdf?ts=1591189841911>
- [54] *AN-1696 - Designing a Boost LED Driver Using the LM5022*, Texas Instruments, Dallas, TX, 2013. [Online]. Available: <https://www.ti.com/lit/an/snva282b/snva282b.pdf?ts=1591193284623>

- 
- [55] *SRR7045-101M Shielded SMD Power Inductor datasheet*, Bourns, Riverside, CA, 2018. [Online]. Available: <https://www.bourns.com/docs/Product-Datasheets/SRR7045.pdf>
- [56] *MBRA160T3G Schottky Power Rectifier datasheet*, ON Semiconductor, Phoenix, AZ, 2019. [Online]. Available: <https://www.onsemi.com/pub/Collateral/MBRA160T3-D.PDF>
- [57] *GRM21BR61H475KE51L SMD Multilayer Ceramic Capacitor datasheet*, Murata, Kyoto, Japan, 2016. [Online]. Available: [http://www.farnell.com/datasheets/2079186.pdf?\\_ga=2.263431876.978926630.1591706248-77083702.1576354984](http://www.farnell.com/datasheets/2079186.pdf?_ga=2.263431876.978926630.1591706248-77083702.1576354984)





## Comparison of UVC LEDs

Table A.1: Comparison table with all evaluated UVC LEDs. Based on the data shown in this table, the LA KL120U8F was selected for the UVGI subsystem. The selection procedure is describes in detail in Section 4.7.1.

Name	Company	Wavelength $\lambda_p$ [nm]	Radiant Flux $\Phi_0$ [mW]	View Angle $\theta$ [°]	Forward Voltage $V_F$ [V]
LEUVK37B50HF00	LG Innotek	278	2	125	5.3 - 7.3
LEUVA66G00HF00	LG Innotek	278	10	126	7
LA KL120U8F	Light Avenue	275	3.5	125	5
LA KM120U8F	Light Avenue	275	10	125	5.4
LA KH120U8F	Light Avenue	275	100	125	16.5
UVR270-SC12	Roithner LT	270	3	120	8
UVR270-4F12	Roithner LT	270	48	120	32
UVTOP270-FW-SMD	Roithner LT	275	2	125	6
UVTOP270H-FW-SMD	Roithner LT	275	12	125	-
CUD8AF1C	Roithner LT	275	3	115	6.5
CUD8AF1D	Roithner LT	275	19	120	-
CUD8AF4C	Roithner LT	275	13	116	6.5
CUD8AF4D	Roithner LT	275	50	118	6.5
UVR280-SC12	Roithner LT	280	3	120	8
UVR280-4F12	Roithner LT	280	48	120	32

Table A.2: UVC LED comparison table continued.

<b>Name</b>	<b>Company</b>	<b>Power Consumption</b> $P_{total}$ [mW]	<b>Size</b> ( $L \times W \times D$ ) [mm]	<b>Price (stk)</b> [€]	$P_{total}/\Phi_0$
LEUVK37B50HF00	LG Innotek	225	3.5 x 3.5 x 1.43	-	112.5
LEUVA66G00HF00	LG Innotek	1155	6.0 x 6.0 x 1.35	-	115.5
LA KL120U8F	Light Avenue	180	3.5 x 3.5 x 1.4	-	51.4
LA KM120U8F	Light Avenue	600	3.5 x 3.5 x 1.4	-	60
LA KH120U8F	Light Avenue	5500	6.0 x 6.0 x 1.8	-	55
UVR270-SC12	Roithner LT	500	4.4 x 4.4 x 1.5	7.6	166.7
UVR270-4F12	Roithner LT	7000	6.0 x 6.0 x 2.2	49.1	145.8
UVTOP270-FW-SMD	Roithner LT	210	3.5 x 3.5 x 1.0	13.2	105
UVTOP270H-FW-SMD	Roithner LT	-	-	42.5	-
CUD8AF1C	Roithner LT	325	6.4 x 6.4 x 1.8	15.13	108.3
CUD8AF1D	Roithner LT	-	-	24.65	-
CUD8AF4C	Roithner LT	1300	6.4 x 6.4 x 1.8	65.45	100
CUD8AF4D	Roithner LT	5100	6.4 x 6.4 x 1.8	89.25	102
UVR280-SC12	Roithner LT	500	4.4 x 4.4 x 1.5	7.6	166.7
UVR280-4F12	Roithner LT	7000	6.0 x 6.0 x 2.2	49.1	145.8

# B

## PCB design

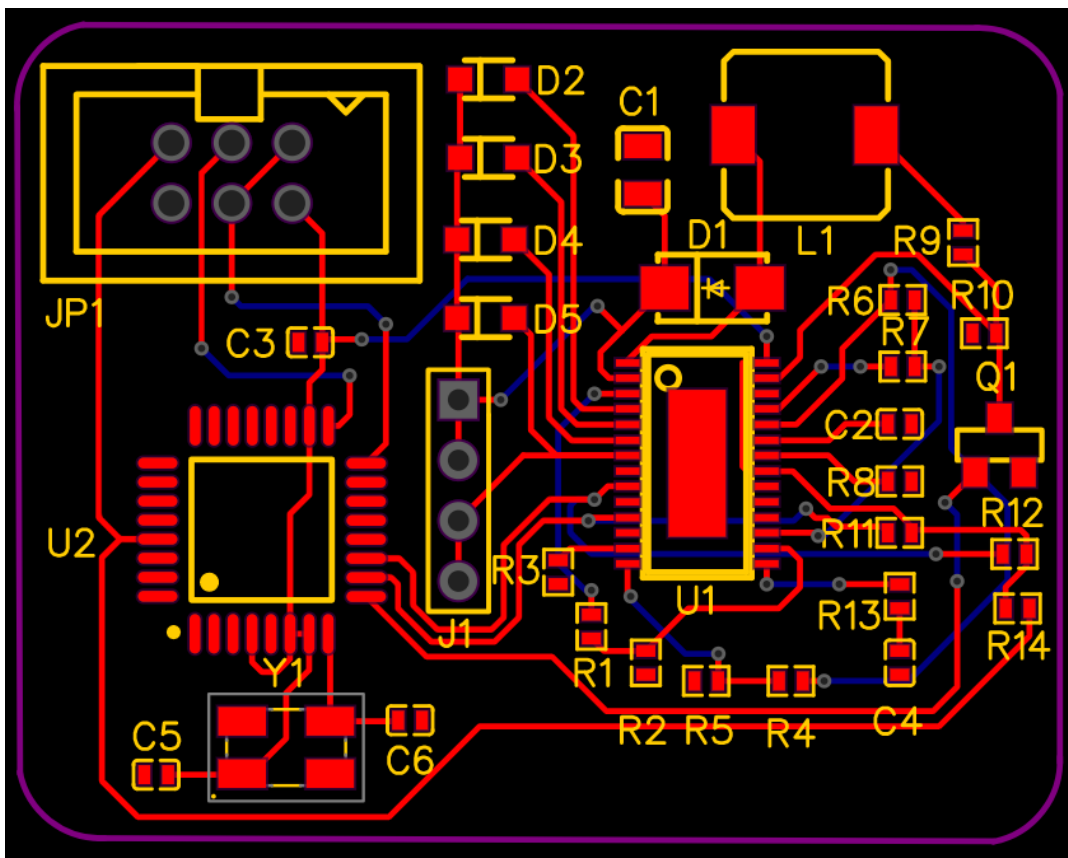


Figure B.1: The PCB design, as implemented in EasyEDA.

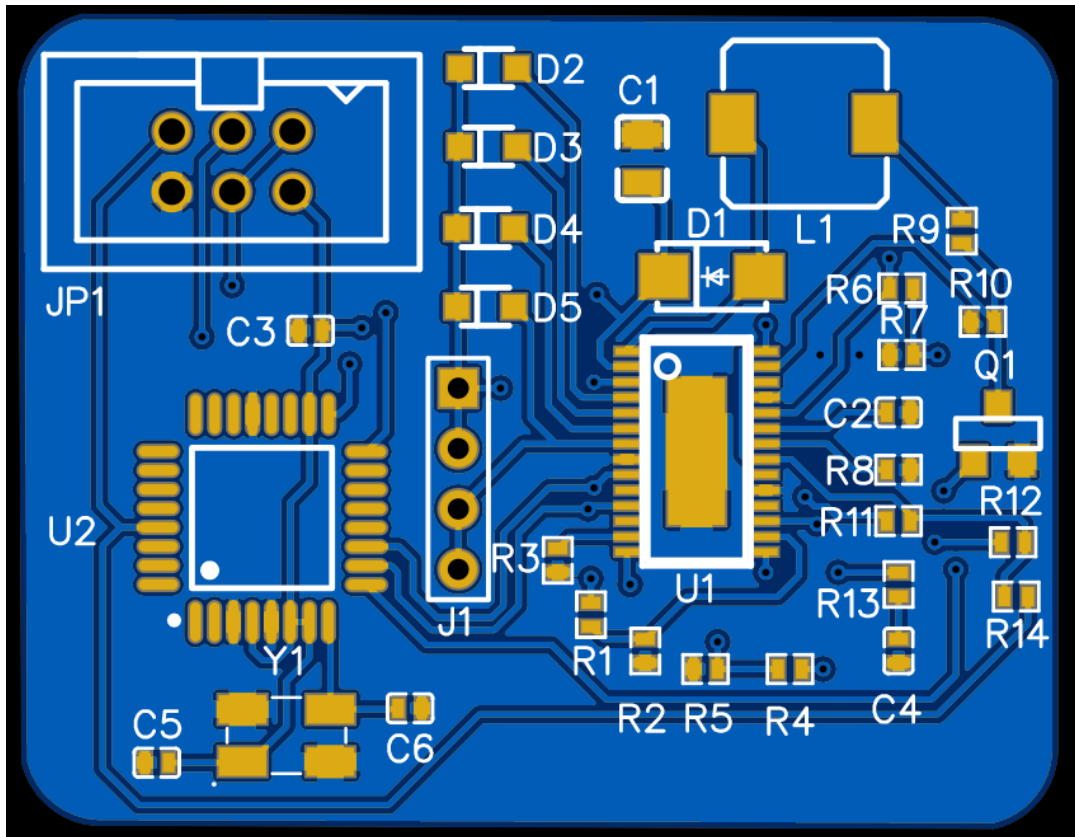
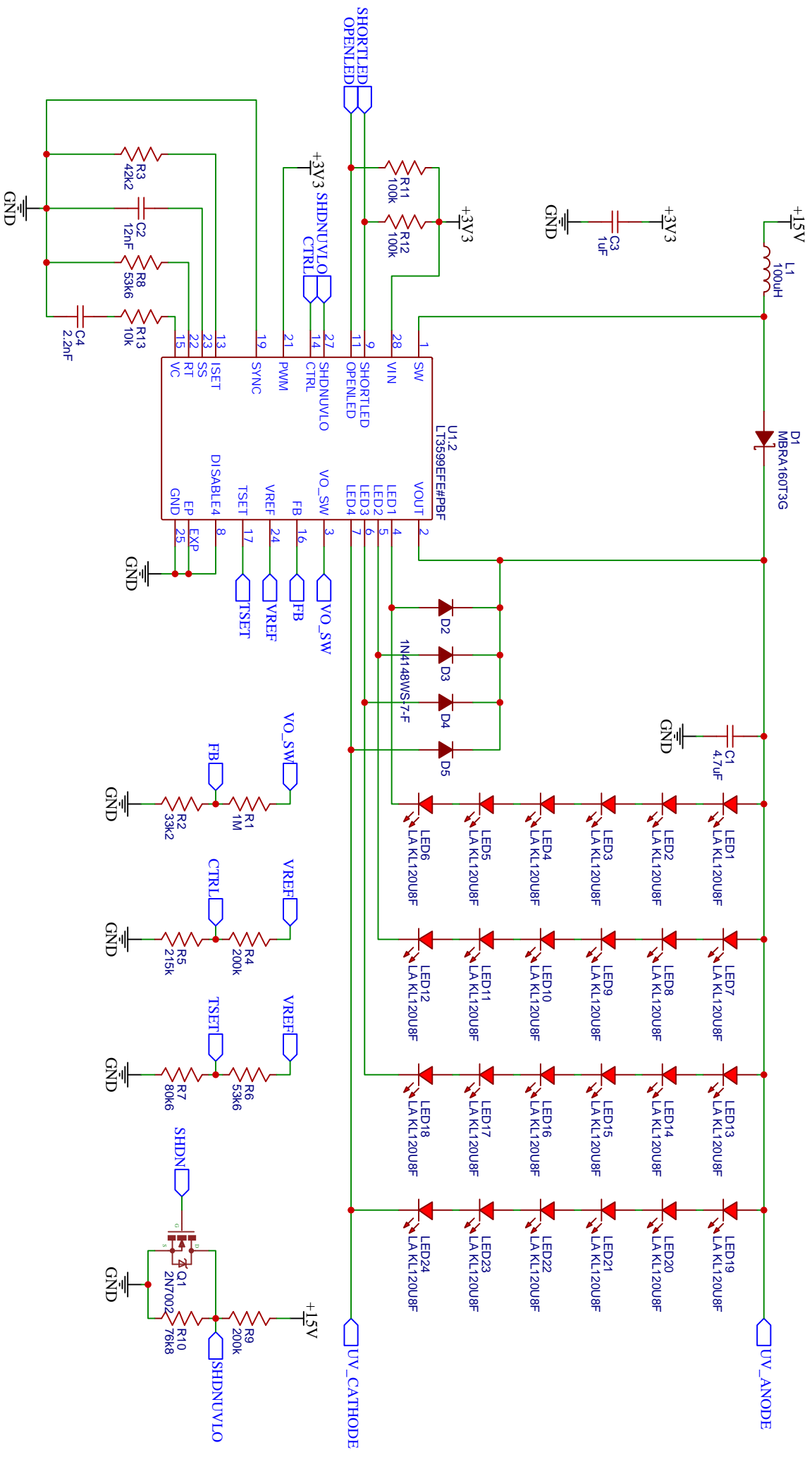
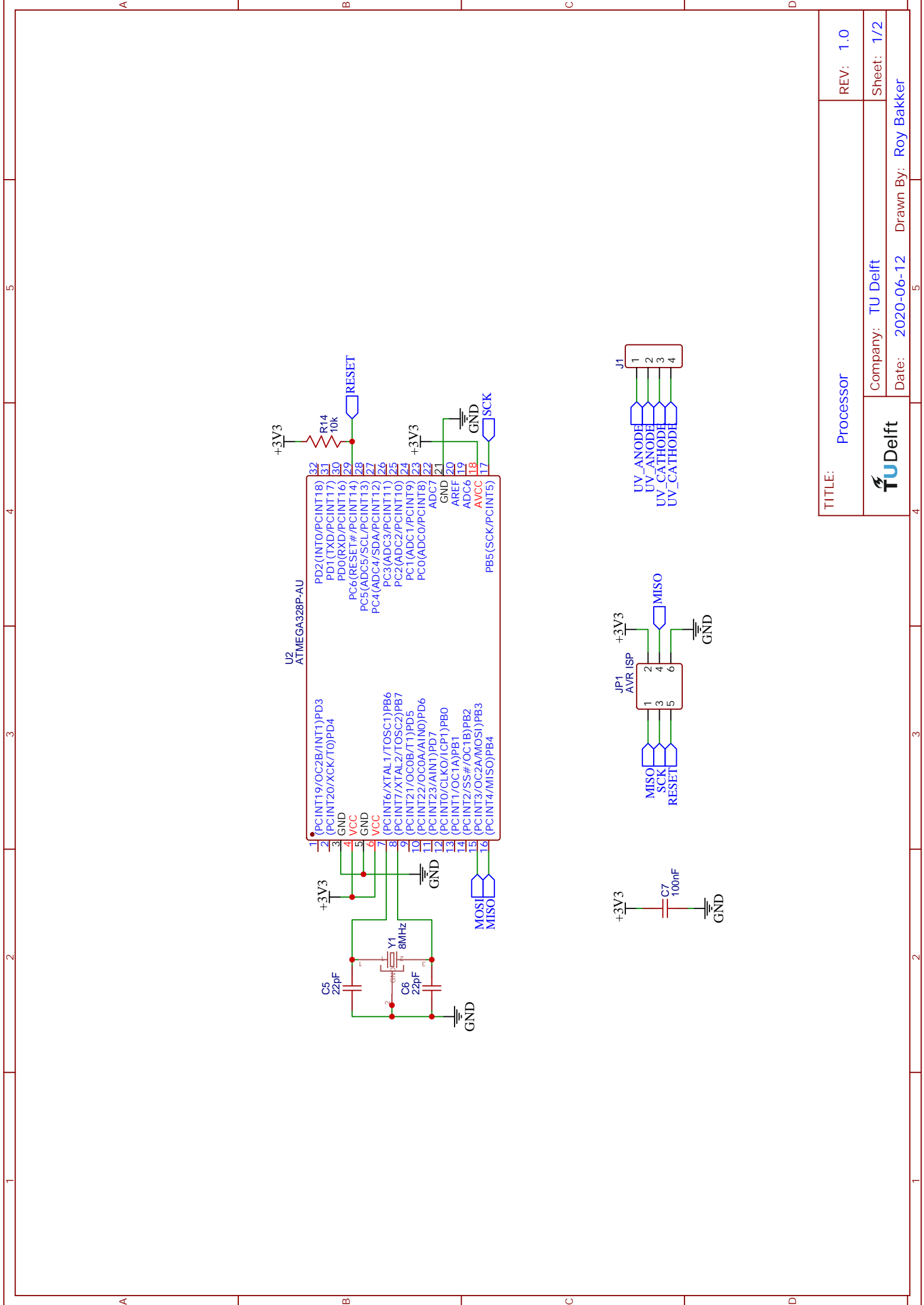



Figure B.2: The PCB board.



TITLE:		LT3599	
Company:		TU DELFT	
Date:		2020-06-10	
Drawn By:		Roy Bakker	
REV:		1.0	
Sheet:		2/2	



TITLE: Processor	REV: 1.0
 TU Delft	Company: TU Delft
Date: 2020-06-12	Sheet: 1/2
	Drawn By: Roy Bakker

C

## LTspice XVII Driver Simulation

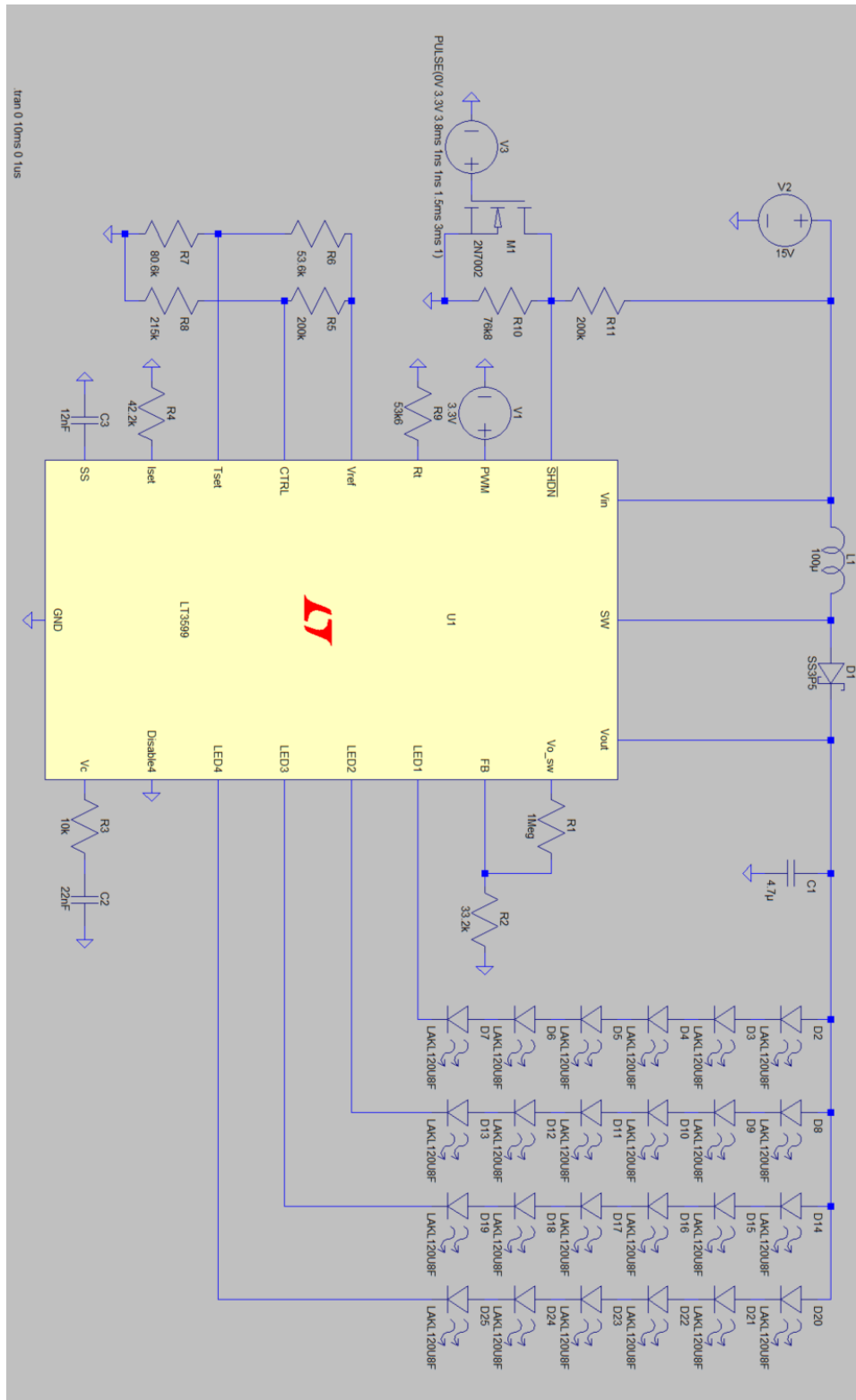
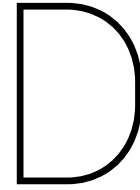


Figure C.1: Schematic of the boost-mode LED driver circuit in LTspice.





# Matlab Simulation and Optimization Scripts

## D.1. Irradiance by a single UVC LED

```
% -----  
% NAME: get_irradiance_grid_single  
% INPUT:  
% > radiant_intensity:    the absolute radiant intensity with a  
%    particular weighted normalized intensity distribution [mW/sr]  
% > led_position:         3-dimensional vector indicating the  
%    position of the UV LED (x, y, z) [mm]  
% > led_direction:        angle that the UV LED is directed at  
%    (referenced to the horizontal x-axis) [rad]  
% > perimeter_radius:     radius of the perimeter lining out the  
%    surface on which the UV LEDs provide radiation [mm]  
% > grid_dim:             size of the grid  
% > grid_length:          width and height of the grid [mm]  
% > is_reflected:        logical value (true/false) indicating if  
%    reflections from the walls surrounding the irradiated surface  
%    should be accounted for  
% > reflection_coefficient:  
%    coefficient indicating the fraction of  
%    reflected UV light intensity  
% > reflection_resolution:  
%    resolution with which the position of  
%    reflection is computed [°]  
% OUTPUT:  
% > irradiance_grid:     2-dimensional vector indicating the  
%    irradiance of each grid element [mW/cm^2]  
% FUNCTION:  
% This function computes the irradiance of a single UV led with  
% defined radiation_pattern, radiant_flux, and three dimensional  
% position. The irradiance is determined on a two dimensional grid  
% for each grid element.  
% AUTHOR:    Roy Bakker & Marcel Brouwers  
% DATE:      7-5-2020  
% -----  
function irradiance_grid = ...  
    get_irradiance_grid_single( radiant_intensity, ...  
                                led_position, ...
```

```

        led_direction, ...
        perimeter_radius, ...
        grid_dim, ...
        grid_length, ...
        is_reflected, ...
        reflection_coefficient, ...
        reflection_resolution)

% declare and define variables:
% grid with dimensions of grid_dim
irradiance_grid = zeros(grid_dim, grid_dim);

% compute the irradiance on the grid [mW/cm^2]
for y = 1:grid_dim
    for x = 1:grid_dim

        % only compute the irradiance on a grid element within surface
        % outlined by perimeter with radius perimeter_radius: if the
        % current grid element is not within this perimeter, skip it
        if sqrt((grid_length * x/grid_dim - grid_length/2)^2 + ...
            (grid_length * y/grid_dim - grid_length/2)^2) > ...
            perimeter_radius
            continue;
        end

        % compute the distance from grid element to projected led position
        % on the grid [mm]
        el_to_led_proj = sqrt(...
            (grid_length * y/grid_dim - grid_length/2 ...
            - led_position(2))^2 + ...
            (grid_length * x/grid_dim - grid_length/2 ...
            - led_position(1))^2);

        % compute the distance from grid element to led position [mm]
        el_to_led = sqrt(el_to_led_proj^2 + led_position(3)^2);
        % compute the zenith angle of the grid element to the led position
        % with respect to the surface [°]
        theta = round(atan2d(el_to_led_proj/led_position(3)));

        % compute the azimuth angle of the grid element to the led position
        % with respect to the surface [rad]
        phi = atan((grid_length * y/grid_dim - ...
            grid_length/2 - led_position(2)) / ...
            (grid_length * x/grid_dim - ...
            grid_length/2 - led_position(1)));
        % map the resulting azimuth angle to [-pi, pi]
        if (grid_length * x/grid_dim - grid_length/2 - ...
            led_position(1)) < 0
            if (grid_length * y/grid_dim - grid_length/2 - ...
                led_position(2)) < 0
                phi = phi - pi;
            else
                phi = phi + pi;
            end
        end
    end
end

```

```

% compute the difference between the above obtained azimuth angle
% and the angle of the UV LED radiation direction [rad]
phi = led_direction - phi;
% check and correct for abnormal values
if isnan(phi)
    phi = 0;
elseif round(phi*180/pi) < 0
    phi = 2*pi + phi;
end
% convert the angle into degrees and map it to [0, 359]
phi_d = round(phi*180/pi);
if phi_d > 359
    phi_d = phi_d - 360;
elseif phi_d < 0
    phi_d = 360 + phi_d;
end

% compute the irradiance on the grid element [mW/cm^2]
irradiance_grid(y, x) = cos(pi*theta/180) * ...
    radiant_intensity(phi_d+1, theta+1) / el_to_led^2 * 100;

% if reflections should be accounted for (is_reflection is true),
% compute the reflected irradiance at the current grid element
% position [mW/cm^2]
if is_reflected == true
    irradiance_grid(y, x) = irradiance_grid(y, x) + ...
        get_reflected_irradiance(    radiant_intensity, ...
                                     reflection_coefficient, ...
                                     reflection_resolution, ...
                                     led_position, ...
                                     led_direction, ...
                                     [grid_length * ...
                                      x/grid_dim - ...
                                      grid_length/2, ...
                                      grid_length * ...
                                      y/grid_dim - ...
                                      grid_length/2], ...
                                     perimeter_radius, ...
                                     false);
end
end
end
end
end

```

## D.2. Irradiance by a ring of UVC LED

```

% -----
% NAME: get_irradiance_grid_ring
% INPUT:
%     > ring_radius:      radius of the ring formed by UV LEDs [mm]
%     > ring_angle:      angle added to all UV LEDs in ring [rad]
%     > ring_tilt_angle:  zenith angle of each UV LED in the ring [°]
%     > n_leds:          number of UV LEDs in the ring
%     > led_distance_to_grid: distance from LEDs to the grid [mm]
%     > radiation_pattern: normalized intensity distribution table

```

```

%           corresponding to degrees [0°-90°]
% > led_radiant_flux:      radiant flux of the UV LED [mW]
% > perimeter_radius:     radius of the perimeter lining out the
%           surface on which the UV LEDs provide radiation [mm]
% > grid_dim:             size of the grid
% > grid_length:          width and height of the grid [mm]
% > is_reflected:        logical value (true/false) indicating if
%           reflections from the walls surrounding the irradiated surface
%           should be accounted for
% > reflection_coefficient:
%           coefficient indicating the fraction of
%           reflected UV light intensity
% > reflection_resolution:
%           resolution with which the position of
%           reflection is computed [°]
% OUTPUT:
% > irradiance_grid:      2-dimensional vector indicating the
%           irradiance of each grid element [mW/cm^2]
% > led_position_array:   array of size n_leds x 3 containing the
%           positions of the UV LEDs (x, y, z) [mm]
% FUNCTION:
% This function computes the irradiance of a ring of UV LEDs
% consisting of n_leds spaced with equal angle, each having the same
% defined radiation_pattern, radiant_flux, and distance from the
% grid. The irradiance is determined on a two dimensional grid
% for each grid element.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:    7-5-2020
% -----
function [irradiance_grid, led_position_array] = ...
    get_irradiance_grid_ring( ring_radius, ...
                             ring_angle, ...
                             ring_tilt_angle, ...
                             n_leds, ...
                             led_distance_to_grid, ...
                             radiation_pattern, ...
                             led_radiant_flux, ...
                             perimeter_radius, ...
                             grid_dim, ...
                             grid_length, ...
                             is_reflected, ...
                             reflection_coefficient, ...
                             reflection_resolution)

% declare and define variables:
led_position_array = zeros(n_leds, 3);
led_position = [0, 0, led_distance_to_grid];
irradiance_grid = zeros(grid_dim, grid_dim);

% compute the solid angle of the UV LED weighted with its normalized
% intensity distribution (figure 1) [sr]
weighted_solid_angle = 0;
for theta = 0:90
    weighted_solid_angle = weighted_solid_angle + ...
        2*pi * radiation_pattern(theta+1, 2) * ...
        sin(theta/180*pi) * 1/180*pi;
end

```

**end**

```
% compute the normalized intensity distribution table I(phi, theta); the
% rows correspond to the azimuth angle (phi) with range [0°, 359°] and
% steps of 1°; the columns correspond to the zenith angle (theta) with
% range [0°, 180°] and steps of 1°; the range of the normalized intensity
% values is [0, 1]
```

```
tilted_radiation_pattern = ...
    get_tilted_radiation_pattern( radiation_pattern, ...
                                abs(ring_tilt_angle));
```

```
% compute the absolute radiant intensity [mW/sr]
radiant_intensity = led_radiant_flux .* ...
    tilted_radiation_pattern ./ weighted_solid_angle;
```

```
% compute the irradiance on the grid for all UV LEDs combined [mW/cm^2]
```

```
for n = 1:n_leds
```

```
% compute the x and y coordinates of the current UV LED [mm]
```

```
led_position(1) = ring_radius * cos(n*2*pi/n_leds + ...
    ring_angle);
```

```
led_position(2) = ring_radius * sin(n*2*pi/n_leds + ...
    ring_angle);
```

```
% compute the azimuth angle of the light direction of the UV LED [rad]
```

```
led_direction = n*2*pi/n_leds + ring_angle - pi;
```

```
% the situation in which the zenith angle of the UV LED is negative, is
% the same as when the azimuth angle is shifted by pi rad and the
% zenith angle is taken to be positive
```

```
if ring_tilt_angle < 0
    led_direction = led_direction + pi;
```

```
end
```

```
% store the position of the current UV LED [mm]
```

```
led_position_array(n, :) = led_position;
```

```
% compute the irradiance on the grid for the current UV LED and add
% this to the irradiance on the grid for the previous UV LEDs
% [mW/cm^2]
```

```
irradiance_grid = irradiance_grid + ...
    get_irradiance_grid_single( radiant_intensity, ...
                                led_position, ...
                                led_direction, ...
                                perimeter_radius, ...
                                grid_dim, ...
                                grid_length, ...
                                is_reflected, ...
                                reflection_coefficient, ...
                                reflection_resolution);
```

**end**

**end**

### D.3. Irradiance by an UVC LED structure

```
% -----
% NAME: get_irradiance_grid_sum_ring
% INPUT:
```

```

% > n_rings:          number of rings encircling the center LED
% > leds_per_ring:    vector of size n_rings indicating the
%                   number of UV LEDs in each ring (in to out)
% > center_distance:  vector of size n_rings indicating the
%                   radius of each ring (in to out) [mm]
% > tilt_angle:       vector of size n_rings indicating the
%                   zenith angle of each UV LED in each ring (in to out) [°]
% > led_distance_to_grid: distance from LEDs to the grid [mm]
% > led_size:         size of the UV LED (width and height) [mm]
% > radiation_pattern: normalized intensity distribution table
%                   corresponding to degrees [0°-90°]
% > led_radiant_flux: radiant flux of the UV LED [mW]
% > perimeter_radius: radius of the perimeter lining out the
%                   surface on which the UV LEDs provide radiation [mm]
% > grid_dim:         size of the grid
% > grid_length:      width and height of the grid [mm]
% > is_reflected:    logical value (true/false) indicating if
%                   reflections from the walls surrounding the irradiated surface
%                   should be accounted for
% > reflection_coefficient:
%                   coefficient indicating the fraction of
%                   reflected UV light intensity
% > reflection_resolution:
%                   resolution with which the position of
%                   reflection is computed [°]
% > create_plot:      logical value (true/false) indicating if a
%                   plot is to be created of the irradiance on the grid [mW/cm^2]
% OUTPUT:
% > irradiance_grid:  2-dimensional vector indicating the
%                   irradiance of each grid element [mW/cm^2]
% > led_position_array: array of size
%                   n_rings x max(leds_per_ring x 3) containing the positions of the
%                   UV LEDs in each ring (x, y, z) [mm]
% FUNCTION:
% This function computes the irradiance of all n_rings rings of UV
% LEDs consisting of n_leds(i). The LEDs in each ring are spaced with
% equal angle, and the LEDs of two adjacent rings are spaced as far
% apart. Each UV LED has the same defined radiation_pattern,
% radiant_flux, and distance from the grid. The irradiance is
% determined on a two dimensional grid for each grid element.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:    10-5-2020
% -----
function [irradiance_grid, led_position_array] = ...
    get_irradiance_grid_sum_ring( n_rings, ...
    leds_per_ring, ...
    center_distance, ...
    tilt_angle, ...
    led_distance_to_grid, ...
    led_size, ...
    radiation_pattern, ...
    led_radiant_flux, ...
    perimeter_radius, ...
    grid_dim, ...
    grid_length, ...
    is_reflected, ...

```

```

        reflection_coefficient, ...
        reflection_resolution, ...
        create_plot)

% declare and define variables
ring_angle_shift = 0;
led_position_array = zeros(n_rings + 1, max(leds_per_ring), 3);

% compute the irradiance on the grid for the center UV LED [mW/cm^2]
[irradiance_grid, led_position] = ...
    get_irradiance_grid_ring( 0, 0, 0, 1, ...
        led_distance_to_grid, ...
        radiation_pattern, ...
        led_radiant_flux, ...
        perimeter_radius, ...
        grid_dim, ...
        grid_length, ...
        is_reflected, ...
        reflection_coefficient, ...
        reflection_resolution);

% store the position of the center UV LED in the LED position array [mm]
led_position_array(1, 1, :) = led_position;

% go through all rings of UV LEDs [1, n_rings]
for n = 1:n_rings

    % if the current ring is not the inner most ring...
    if n > 1
        % compute the optimal absolute angle of the current ring using the
        % number of the number of LEDs of the current and previous ring
        % [rad]
        ring_angle_shift = ring_angle_shift + ...
            optimal_angle_between_rings( leds_per_ring(n-1), ...
                leds_per_ring(n), ...
                pi/1000, ...
                false);
    end

    % compute the irradiance on the grid for the current ring of UV LEDs
    % [mW/cm^2]
    [irradiance_grid_tmp, led_position] = ...
        get_irradiance_grid_ring( center_distance(n), ...
            ring_angle_shift, ...
            tilt_angle(n), ...
            leds_per_ring(n), ...
            led_distance_to_grid, ...
            radiation_pattern, ...
            led_radiant_flux, ...
            perimeter_radius, ...
            grid_dim, ...
            grid_length, ...
            is_reflected, ...
            reflection_coefficient, ...
            reflection_resolution);

    % add the irradiance on the grid of the current ring to that of the

```

```

% previous rings [mm]
irradiance_grid = irradiance_grid + irradiance_grid_tmp;
% store the position of all UV LEDs in the current ring in the LED
% position array
led_position_array(n+1, 1:leds_per_ring(n), :) = led_position;
end

% if create_plot is true, plot the irradiance on the grid of the whole UV
% LED system defined by the parameters of this function
if create_plot == true

figure();
% map the grid to the right dimensions, namely a square grid with
% x = [-grid_length/2, grid_length/2] and
% y = [-grid_length/2, grid_length/2] [mm]
mapped_coordinates = -grid_length/2: ...
    grid_length/grid_dim: grid_length/2-grid_length/grid_dim;
% create a pseudocolor plot of the irradiance
s = pcolor(mapped_coordinates, mapped_coordinates, ...
    irradiance_grid);
% turn off the grid outline
s.EdgeColor = 'none';
% turn on the colorbar legend
colorbar;
% create an appropriate title for the current UV LED system
title(sprintf(...
    "Irradiance of %d UV LEDs @ %dmm distance [mW/cm^2]", ...
    sum(leds_per_ring)+1, led_distance_to_grid));
% label the axes
xlabel("x [mm]");
ylabel("y [mm]");
hold on;

% using the LED position array, indicate the position of the UV LEDs by
% drawing a square of size led_size at every LED position [mm]
for ring = 1:n_rings+1
    if ring == 1
        rectangle('Position', ...
            [(led_position_array(1, 1, 1)-led_size/2), ...
            (led_position_array(1, 1, 2)-led_size/2), ...
            led_size, led_size], ...
            'LineWidth', 1);
    else
        for led = 1:leds_per_ring(ring-1)
            rectangle('Position', ...
                [(led_position_array(ring, led, 1)-led_size/2), ...
                (led_position_array(ring, led, 2)-led_size/2), ...
                led_size, led_size], ...
                'LineWidth', 1);
        end
    end
end
end
end
end
end

```



## D.4. Optimal angle between UVC LED rings

```

% -----
% NAME: optimal_angle_between_rings
% INPUT:
%     > n_points_ring1:      number of points on ring 1
%     > n_points_ring2:      number of points on ring 2
%     > radial_resolution:    the resolution with which the optimal angle
%                             is tuned [rad]
%     > create_figure:       logical value (true/false) indicating if a
%                             plot is to be created of all points on both rings with one of
%                             the rings shifted by the optimal angle
% OUTPUT:
%     > optimal_angle:       the angle that shifts either ring in a way
%                             such that the points of the two rings are spaced as far apart
%                             from each other as possible [rad]
% FUNCTION:
%     This function computes the so called optimal angle. That is, the
%     angle that shifts either ring in a way such that the points of
%     the two rings are spaced as far apart from each other as
%     possible.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:     10-5-2020
% -----
function optimal_angle = ...
    optimal_angle_between_rings(    n_points_ring1, ...
                                   n_points_ring2, ...
                                   radial_resolution, ...
                                   create_figure)

% declare and define variables
optimal_angle = 0;
maximal_smallest_angle = 0;
% compute the angle of each point on the rings such that they are spaced
% equally
angle_points_ring1 = 0: 2*pi/n_points_ring1: 2*pi-2*pi/n_points_ring1;
angle_points_ring2 = 0: 2*pi/n_points_ring2: 2*pi-2*pi/n_points_ring2;

% find the minimal angle between two points on either ring; this is the
% upper limit of optimal_angle: optimal_angle < minimal_angle
if angle_points_ring1(1) < angle_points_ring2(1)
    minimal_angle = angle_points_ring1(2);
else
    minimal_angle = angle_points_ring2(2);
end

% go through all possible values of optimal angle: [0, minimal_angle)
for current_angle = 0: radial_resolution: minimal_angle-radial_resolution

    % reset the smallest angle for this iteration
    smallest_angle = 2*pi;

    % go through all points on ring 1
    for n = 1:n_points_ring1

        % compute the angle between the current point on ring 1 and all
        % points on ring 2 shifted by current_angle
    
```

```

    angles = angle_points_ring2 + ...
              current_angle - angle_points_ring1(n);

    % map all obtained angles to (-pi, pi]
    % div_factor = { 0   for |angles(i)| < pi
                   { 1   for |angles(i)| >= pi
    div_factor = floor(abs(angles/pi));
    for i = 1:n_points_ring2
        if div_factor(i) > 0 % |angles(i)| >= pi
            % map 'angles(i)' into the domain [-pi, pi]
            angles(i) = pi - abs(rem(angles(i), pi));
        end
    end
    % find the smallest angle: the angle between the current point on
    % ring 1 and the closest point on ring 2
    angle_nearest = min(abs(angles));

    % find the smallest angle_nearest for all points on ring 1
    if angle_nearest < smallest_angle
        smallest_angle = angle_nearest;
    end
end

% check if the smallest angle found in this iteration is larger than
% that of previous iterations, thereby indicating a better spacing
% between the points on the rings
if smallest_angle > maximal_smallest_angle
    optimal_angle = current_angle;
    maximal_smallest_angle = smallest_angle;
end
end

% if create_figure is true, plot the points of both rings with the optimal
% shift
if create_figure == true

    % declare and define variables
    theta = 0:pi/100:2*pi;
    % shift the points on ring 2 by the computed optimal angle
    angle_points_ring2 = angle_points_ring2 + optimal_angle;

    figure();
    % draw blue circles indicating the points on ring 1
    plot(cos(angle_points_ring1), sin(angle_points_ring1), ...
         'o', 'Color', 'Blue', 'LineWidth', 6);
    hold on;
    % draw a circle indicating ring 1
    plot(cos(theta), sin(theta), ...
         '--', 'Color', 'Black', 'LineWidth', 1);

    % draw red circles indicating the points on ring 2
    plot(1.2 * cos(angle_points_ring2), 1.2 * sin(angle_points_ring2), ...
         'o', 'Color', 'Red', 'LineWidth', 6);
    % draw a circle indicating ring 2
    plot(1.2 * cos(theta), 1.2 * sin(theta), ...
         '--', 'Color', 'Black', 'LineWidth', 1);
end

```

```

% make the axes invisible
set(gca, 'visible', 'off');
% set the position, width, and height of the figure
set(gcf, 'position', [200, 200, 500, 450]);
end
end

```

## D.5. Tilted normalized intensity distribution

```

% -----
% NAME: get_tilted_radiation_pattern
% INPUT:
%   > radiation_pattern_vector: normalized intensity distribution table
%   with on column 1 the zenith angle (theta) in degrees [0°, 90°]
%   and on column 2 the corresponding normalized intensity values
%   [0, 1]
%   > tilt_angle:                zenith angle with which the radiation
%   pattern should be tilted [°]
% OUTPUT:
%   > tilted_radiation_pattern: normalized intensity distribution
%   table I(phi, theta); the rows correspond to the azimuth angle
%   (phi) with range [0°, 359°] and steps of 1°; the columns
%   correspond to the zenith angle (theta) with range [0°, 180°]
%   and steps of 1°; the range of the normalized intensity values
%   is [0, 1]
% FUNCTION:
%   This function first transforms the normalized intensity
%   distribution I(theta) corresponding to only the zenith angle
%   (theta), to a symmetrical normalized intensity distribution
%   I(phi, theta) corresponding to both the azimuth (phi) and zenith
%   angle (theta). It then transforms this table so that the normalized
%   intensity distribution is shifted along the zenith angle at
%   phi = 0 rad by an angle of tilt_angle. Finally it returns this
%   transformed normalized intensity distribution table.
%   Note: radiation pattern = normalized intensity distribution.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:     17-5-2020
% -----
function tilted_radiation_pattern = ...
    get_tilted_radiation_pattern( radiation_pattern_vector, ...
                                tilt_angle)

% declare and define variables
og_radiation_pattern_matrix = zeros(360, 181);
tilted_radiation_pattern = zeros(360, 181);

% extend the radiation pattern I(theta) to a domain of [0°, 180°]
extended_radiation_pattern = zeros(181, 1);
extended_radiation_pattern(1:size(radiation_pattern_vector, 1)) = ...
    radiation_pattern_vector(:, 2);

% transform the radiation pattern I(theta) into a symmetrical radiation
% pattern dependent on both azimuth and zenith angle I(phi, theta)
for i = 1:360

```

```

og_radiation_pattern_matrix(i, :) = extended_radiation_pattern;
end

% go through every element in the radiation pattern I(phi, theta)
for phi = 0:359
    for theta = 0:180

        % transform every element on the unit circle indicated with
        % (phi, theta) to cartesian coordinates (x, y, z)
        x = sin(theta*pi/180) * cos(phi*pi/180);
        y = sin(theta*pi/180) * sin(phi*pi/180);
        z = cos(theta*pi/180);

        % compute the radius of the circle obtained by making a cross
        % section of the unit sphere at y along the xz-plane
        r = sqrt(x^2 + z^2);
        % compute the zenith angle of the current point (x, y, z) [rad]
        psi = atan2(x, z);
        % compute the new zenith angle when tilting the point along the
        % previously described circle by angle tilt_angle
        psi_new = psi + tilt_angle*pi/180;
        % compute the new x and z values after the tilt
        x = r * sin(psi_new);
        z = r * cos(psi_new);

        % compute the azimuth angle (phi) [rad] and map it to [0, 2pi]
        phi_new = atan2(y, x);
        if round(phi_new*180/pi) < 0
            phi_new = 2*pi + phi_new;
        end
        % compute the zenith angle (theta) [rad] with range [0, pi/2]
        theta_new = atan2(sqrt(x^2 + y^2), z);

        % put the original radiation pattern value at I(phi, theta) in the
        % new radiation pattern at I(phi_new, theta_new)
        tilted_radiation_pattern(round(phi_new*180/pi)+1, ...
            round(theta_new*180/pi)+1) = ...
            og_radiation_pattern_matrix(phi+1, theta+1);
    end
end

% smooth out the radiation pattern, so that most missing values are given
% an appropriate value
tilted_radiation_pattern(tilted_radiation_pattern == 0) = NaN;
tilted_radiation_pattern = ...
    fillmissing(tilted_radiation_pattern, 'pchip', 1);

% the first column somehow always has a value of NaN; these are given thus
% given an more appropriate value by interpolation of the first ~20
% elements of each row
tilted_radiation_pattern(:, 1:20) = ...
    fillmissing(tilted_radiation_pattern(:, 1:20), 'lin', 2);
tilted_radiation_pattern(tilted_radiation_pattern < 0) = 0;

% set all missing number to zero
tilted_radiation_pattern(isnan(tilted_radiation_pattern)) = 0;

```

end

## D.6. Normalized intensity distribution: cross section

```

% -----
% NAME: get_irradiance_grid_single
% INPUT:
%   > radiant_intensity:    the absolute radiant intensity with a
%                           particular weighted normalized intensity distribution [mW/sr]
%   > led_position:         3-dimensional vector indicating the
%                           position of the UV LED (x, y, z) [mm]
%   > led_direction:        angle that the UV LED is directed at
%                           (referenced to the horizontal x-axis) [rad]
%   > perimeter_radius:     radius of the perimeter lining out the
%                           surface on which the UV LEDs provide radiation [mm]
%   > grid_dim:             size of the grid
%   > grid_length:          width and height of the grid [mm]
%   > is_reflected:        logical value (true/false) indicating if
%                           reflections from the walls surrounding the irradiated surface
%                           should be accounted for
%   > reflection_coefficient:
%                           coefficient indicating the fraction of
%                           reflected UV light intensity
%   > reflection_resolution:
%                           resolution with which the position of
%                           reflection is computed [°]
% OUTPUT:
%   > irradiance_grid:      2-dimensional vector indicating the
%                           irradiance of each grid element [mW/cm^2]
% FUNCTION:
%   This function computes the irradiance of a single UV led with
%   defined radiation_pattern, radiant_flux, and three dimensional
%   position. The irradiance is determined on a two dimensional grid
%   for each grid element.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:     7-5-2020
% -----
function irradiance_grid = ...
    get_irradiance_grid_single( radiant_intensity, ...
                               led_position, ...
                               led_direction, ...
                               perimeter_radius, ...
                               grid_dim, ...
                               grid_length, ...
                               is_reflected, ...
                               reflection_coefficient, ...
                               reflection_resolution)

% declare and define variables:
% grid with dimensions of grid_dim
irradiance_grid = zeros(grid_dim, grid_dim);

% compute the irradiance on the grid [mW/cm^2]
for y = 1:grid_dim
    for x = 1:grid_dim

```

```

% only compute the irradiance on a grid element within surface
% outlined by perimeter with radius perimeter_radius: if the
% current grid element is not within this perimeter, skip it
if sqrt((grid_length * x/grid_dim - grid_length/2)^2 + ...
        (grid_length * y/grid_dim - grid_length/2)^2) > ...
    perimeter_radius
    continue;
end

% compute the distance from grid element to projected led position
% on the grid [mm]
el_to_led_proj = sqrt(...
    (grid_length * y/grid_dim - grid_length/2 ...
    - led_position(2))^2 + ...
    (grid_length * x/grid_dim - grid_length/2 ...
    - led_position(1))^2);

% compute the distance from grid element to led position [mm]
el_to_led = sqrt(el_to_led_proj^2 + led_position(3)^2);
% compute the zenith angle of the grid element to the led position
% with respect to the surface [°]
theta = round(atan2(el_to_led_proj/led_position(3)));

% compute the azimuth angle of the grid element to the led position
% with respect to the surface [rad]
phi = atan2((grid_length * y/grid_dim - ...
    grid_length/2 - led_position(2)) / ...
    (grid_length * x/grid_dim - ...
    grid_length/2 - led_position(1)));
% map the resulting azimuth angle to [-pi, pi]
if (grid_length * x/grid_dim - grid_length/2 - ...
    led_position(1)) < 0
    if (grid_length * y/grid_dim - grid_length/2 - ...
        led_position(2)) < 0
        phi = phi - pi;
    else
        phi = phi + pi;
    end
end

% compute the difference between the above obtained azimuth angle
% and the angle of the UV LED radiation direction [rad]
phi = led_direction - phi;
% check and correct for abnormal values
if isnan(phi)
    phi = 0;
elseif round(phi*180/pi) < 0
    phi = 2*pi + phi;
end
% convert the angle into degrees and map it to [0, 359]
phi_d = round(phi*180/pi);
if phi_d > 359
    phi_d = phi_d - 360;
elseif phi_d < 0
    phi_d = 360 + phi_d;
end

```

```

% compute the irradiance on the grid element [mW/cm^2]
irradiance_grid(y, x) = cos(pi*theta/180) * ...
    radiant_intensity(phi_d+1, theta+1) / eL_to_led^2 * 100;

% if reflections should be accounted for (is_reflection is true),
% compute the reflected irradiance at the current grid element
% position [mW/cm^2]
if is_reflected == true
    irradiance_grid(y, x) = irradiance_grid(y, x) + ...
        get_reflected_irradiance(    radiant_intensity, ...
            reflection_coefficient, ...
            reflection_resolution, ...
            led_position, ...
            led_direction, ...
            [grid_length * ...
                x/grid_dim - ...
                grid_length/2, ...
                grid_length * ...
                y/grid_dim - ...
                grid_length/2], ...
            perimeter_radius, ...
            false);
end
end
end
end

```

## D.7. Normalized intensity distribution: 3D image

```

% -----
% NAME: plot_radiation_pattern_3d
% INPUT:
%     > radiation_pattern:    normalized intensity distribution
%     table I(phi, theta); the rows correspond to the azimuth angle
%     (phi) with range [0°, 359°] and steps of 1°; the columns
%     correspond to the zenith angle (theta) with range [0°, 180°]
%     and steps of 1°; the range of the normalized intensity values
%     is [0, 1]
% OUTPUT:
%     -
% FUNCTION:
%     This function plots a 3-dimensional version of the normalized
%     intensity distribution I(phi, theta) mapped to cartesian
%     coordinates (x, y, z).
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:    17-5-2020
% -----
function plot_radiation_pattern_3d(radiation_pattern)

% declare and define variables:
n_lines = 60;
% create an array filled with 361 rows of with the same values in each
% column: a linear count from -pi/2 to pi/2 with steps of pi/180 (181
% columns)
theta_array = repmat(pi/2 - (0:pi/180:pi), 361, 1);

```

```

% create an array filled with 181 column of with the same values in each
% row: a linear count from 0 to pi with steps of pi/180 (361 rows)
phi_array = repmat((0:pi/180:2*pi)', 1, 181);

% map the spherical coordinates (phi, theta, radiation_pattern) to their
% corresponding cartesian coordinates
[X, Y, Z] = sph2cart(phi_array, theta_array, ...
    cat(1, radiation_pattern, radiation_pattern(1, :)));

% create a new figure
figure();

% plot all obtained cartesian coordinates (X, Y, Z) as a surface
surf(X, Y, Z, 'EdgeColor', 'None');
hold on;

% draw n_lines black lines on the plotted surface in two directions for
% better depth visualization
for i = 1: round(size(X, 2)/n_lines): size(X, 2)
    plot3(X(:, i), Y(:, i), Z(:, i), '-k');
end
for i = 1: round(size(X, 1)/n_lines): size(X, 1)
    plot3(X(i, :), Y(i, :), Z(i, :), '-k');
end

% add a title to the figure
title("Normalized Intensity Distribution");
% add appropriate labels to each axis
xlabel('X');
ylabel('Y');
zlabel('Z');
% set the length of each axis equal
axis equal;
end

```

## D.8. Irradiance reflections

```

% -----
% NAME: get_reflected_irradiance
% INPUT:
%     > radiant_intensity:    the absolute radiant intensity with a
%                             particular weighted normalized intensity distribution [mW/sr]
%     > reflection_coefficient:
%                             coefficient indicating the fraction of
%                             reflected UV light intensity
%     > reflection_resolution:
%                             resolution with which the position of
%                             reflection is computed [°]
%     > led_position:         3-dimensional vector indicating the
%                             position of the UV LED (x, y, z) [mm]
%     > led_direction:       angle that the UV LED is directed at
%                             (referenced to the horizontal x-axis) [rad]
%     > grid_element_position:
%                             2-dimensional vector indicating the
%                             position of the grid element under investigation (x, y, z) [mm]
%     > perimeter_radius:    radius of the perimeter lining out the

```



```

%         surface on which the UV LEDs provide radiation [mm]
%     > create_figure:         logical value (true/false) indicating if a
%         plot is to be created of the reflection points on the surface
%         perimeter with traces of the reflected UV light
%         the rings shifted by the optimal angle
% OUTPUT:
%     > reflected_irradiance: irradiance measured at the indicated grid
%         element position due to reflection from the wall surrounding
%         the surface with radius perimeter_radius [mW/cm^2]
% FUNCTION:
%     This function computes the irradiance at the grid element indicated
%     by the grid_element_position parameter, but only due to reflection
%     from the wall surrounding the irradiated surface. It does this by
%     walking along the surface perimeter and computing if a reflection
%     is possible from the UV LED position and grid element position via
%     that particular position along the perimeter. Note, it is assumed
%     that the wall surrounding the surface supports only specular
%     reflection and has a reflection coefficient value of
%     reflection_coefficient.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:     24-5-2020
% -----
function reflected_irradiance = ...
    get_reflected_irradiance(    radiant_intensity, ...
                                reflection_coefficient, ...
                                reflection_resolution, ...
                                led_position, ...
                                led_direction, ...
                                grid_element_position, ...
                                perimeter_radius, ...
                                create_figure)

% declare and define variables
wall_position = [0, 0];
reflected_irradiance = 0;
distance_from_reflection_prev = 0;
psi_prev = 0;
is_decrease_distance_from_reflection_prev_flag = 0;

% if create_figure is true, generate an image describing the position of
% the grid element and UV LED under investigation within the perimeter of
% the irradiated surface
if create_figure == true

    % create a new figure
    figure();

    % draw the perimeter of the surface
    plot(perimeter_radius*cos(0:pi/180:2*pi), ...
          perimeter_radius*sin(0:pi/180:2*pi), ...
          'Color', 'Black', 'LineWidth', 1);
    hold on;
    % indicate the position of the grid element
    el_plot = plot(grid_element_position(1), grid_element_position(2), ...
                   'o', 'Color', 'Magenta', 'LineWidth', 2);
    % indicate the position of the UV LED

```

```

    led_plot = plot(led_position(1), led_position(2), ...
        'o', 'Color', 'Red', 'LineWidth', 2);
end

% increase the perimeter radius ever so slightly, such that reflections on
% grid elements on the actual perimeter are still computed correctly [mm]
perimeter_radius = perimeter_radius + 0.01;

% walk along the perimeter of the irradiated surface with steps of size
% reflection resolution [rad]
for psi = -reflection_resolution:reflection_resolution:360

    % compute the angle of the tangent line (aka the reflective wall) to
    % the circle at angle psi [rad]
    wall_angle = psi*pi/180 + pi/2;
    % compute the position of the reflective wall in cartesian coordinates
    % (x, y) [mm]
    wall_position(1) = perimeter_radius * cos(psi*pi/180);
    wall_position(2) = perimeter_radius * sin(psi*pi/180);

    % compute the angle of the grid element with the location of the wall
    % taken as center [rad]
    grid_element_angle = atan2(...
        grid_element_position(2) - wall_position(2), ...
        grid_element_position(1) - wall_position(1));
    % compute the angle of the UV LED with the location of the wall taken
    % as center [rad]
    led_angle = atan2(led_position(2) - wall_position(2), ...
        led_position(1) - wall_position(1));

    % compute the difference between the angle of the grid element and the
    % reflective wall and map it to [0, 2pi] [rad]
    theta_wall_grid = mod(grid_element_angle - wall_angle, 2*pi);
    if (theta_wall_grid == 0) && (grid_element_angle - wall_angle > 0)
        theta_wall_grid = 2*pi;
    end

    % compute the difference between the angle of the UV LED and the
    % reflective wall and map it to [0, 2pi] [rad]
    theta_wall_led = mod(led_angle - wall_angle, 2*pi);
    if (theta_wall_led == 0) && (led_angle - wall_angle > 0)
        theta_wall_led = 2*pi;
    end

    % if the difference between pi and the sum of the wall to grid angle
    % plus the wall to UV LED angle is less than the previous step set a
    % flag indicating such an event
    if abs(180 - (theta_wall_grid + theta_wall_led)*180/pi) <= ...
        distance_from_reflection_prev
        is_decrease_distance_from_reflection_prev_flag = 1;
    % if the difference between pi and the sum of the wall to grid angle
    % plus the wall to UV LED angle is more than the previous step...
    else
        % if the previous step a decrease was detected and now an increase
        % is detected, a minimum of the difference between pi and the sum
        % of the wall to grid angle plus the wall to UV LED angle is

```

```

% assumed, indicating UV light reflection at the previous step
% position
if is_decrease_distance_from_reflection_prev_flag == 1

    % compute the distance between the reflective wall position and
    % the projected UV LED position [mm]
    led_to_wall = sqrt(...
        (wall_position(1) - led_position(1))^2 + ...
        (wall_position(2) - led_position(2))^2);
    % compute the distance between the reflective wall position and
    % the current grid element position [mm]
    el_to_wall = sqrt(...
        (wall_position(1) - grid_element_position(1))^2 + ...
        (wall_position(2) - grid_element_position(2))^2);

    % compute the height between the UV LED and the point of
    % reflection on the reflective wall [mm]
    height = led_position(3) * led_to_wall / ...
        (led_to_wall + el_to_wall);

    % compute the distance the UV light travels from the UV LED to
    % the current grid element when reflected at the reflective
    % wall position [mm]
    el_to_led = sqrt(el_to_wall^2 + (led_position(3) - height)^2) + ...
        sqrt(led_to_wall^2 + height^2);

    % compute the zenith angle of the reflected UV light beam with
    % respect to the UV LED position [°]
    theta = round(atan(led_to_wall/height));

    % compute the azimuth angle of the reflective wall to the LED
    % position with respect to the surface [rad]
    phi = atan((wall_position(2) - led_position(2)) / ...
        (wall_position(1) - led_position(1)));
    % map the resulting azimuth angle to [-pi, pi]
    if (wall_position(1) - led_position(1)) < 0
        if (wall_position(2) - led_position(2)) < 0
            phi = phi - pi;
        else
            phi = phi + pi;
        end
    end

    % compute the difference between the above obtained azimuth
    % angle and the angle of the UV LED radiation direction [rad]
    phi = led_direction - phi;
    % check and correct for abnormal values
    if isnan(phi)
        phi = 0;
    elseif round(phi*180/pi) < 0
        phi = 2*pi + phi;
    end

    % convert the angle into degrees and map it to [0, 359]
    phi_d = round(phi*180/pi);
    if phi_d > 359
        phi_d = phi_d - 360;

```

```

elseif phi_d < 0
    phi_d = 360 + phi_d;
end

% compute the irradiance on the grid element due to UV light
% reflection at the reflective wall position [mW/cm^2]
reflected_irradiance = reflected_irradiance + ...
    cos(pi*theta/180) * ...
    radiant_intensity(phi_d+1, theta+1) / ...
    el_to_led^2 * 100;

% if create_figure is true, draw a dashed line from the UV LED
% to the reflective wall and from the current grid
% element to the reflective wall along with an indication of
% the position of the reflective wall
if create_figure == true
    % indicate the location of the reflective wall
    wall_plot = plot(perimeter_radius*cos(psi_prev), ...
        perimeter_radius*sin(psi_prev), ...
        'o', 'Color', 'Blue', 'LineWidth', 2);

    % draw a line from current grid element to reflective wall
    line([grid_element_position(1) ...
        perimeter_radius*cos(psi_prev)], ...
        [grid_element_position(2) ...
        perimeter_radius*sin(psi_prev)], ...
        'LineStyle', '--', 'Color', 'Black', 'LineWidth', 1);

    % draw a line from UV LED to reflective wall
    line([led_position(1), perimeter_radius*cos(psi_prev)], ...
        [led_position(2), perimeter_radius*sin(psi_prev)], ...
        'LineStyle', '--', 'Color', 'Black', 'LineWidth', 1);

    % generate a legend
    legend([el_plot, led_plot, wall_plot], ...
        {'Grid element', 'UV LED', 'Reflection locations'});
    % set limits for the axes
    axis([-1.2 * perimeter_radius, 1.2 * perimeter_radius, ...
        -1.2 * perimeter_radius, 1.2 * perimeter_radius]);
    % set the position, width, and height of the figure
    set(gcf, 'position', [200, 200, 500, 450]);
end

% clear the flag
is_decrease_distance_from_reflection_prev_flag = 0;
end
end

% store the current important variables for use during next iteration
distance_from_reflection_prev = ...
    abs(180 - (theta_wall_grid + theta_wall_led)*180/pi);
psi_prev = psi*pi/180;
end

% account for loss in intensity due to reflection and transmission of
% incident UV light on the reflective wall with the reflection

```

```
% coefficient
reflected_irradiance = reflection_coefficient * reflected_irradiance;
end
```

## D.9. Irradiance uniformity

```
% -----
% NAME: compute_irradiance_uniformity
% INPUT:
%   > irradiance_grid:    2-dimensional vector indicating the
%                         irradiance of each grid element [mW/cm^2]
%   > perimeter_radius:   radius of the perimeter of the filter and
%                         thus the outer boundary between which the uniformity is
%                         computed
%   > grid_dim:           size of the grid
%   > grid_length:        width and height of the grid [mm]
% OUTPUT:
%   > irradiance_min:     minimal irradiance found in the grid
%                         [mW/cm^2]
%   > irradiance_max:     maximal irradiance found in the grid
%                         [mW/cm^2]
%   > position_min:       position of the minimal radiance in the
%                         grid (x, y) [mm]
%   > position_max:       position of the maximal radiance in the
%                         grid (x, y) [mm]
% FUNCTION:
%   This function computes the uniformity of the irradiance on the grid
%   inside the perimeter indicated with the perimeter radius. The
%   function returns the minimal and maximal irradiance values along
%   with the positions of these two irradiance values.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:     10-5-2020
% -----
function [irradiance_min, irradiance_max, position_min, position_max] = ...
    compute_irradiance_uniformity( irradiance_grid, ...
                                   perimeter_radius, ...
                                   grid_dim, ...
                                   grid_length)

% declare and define variables
irradiance_max = irradiance_grid(grid_dim/2, grid_dim/2);
irradiance_min = irradiance_grid(grid_dim/2, grid_dim/2);
position_max = [0, 0];
position_min = [0, 0];

% go through every grid element
for row = 1:grid_dim
    for column = 1:grid_dim

        % only check the irradiance values of grid elements inside the
        % filter perimeter with radius perimeter_radius
        if sqrt((grid_length * row/grid_dim - grid_length/2)^2 + ...
                (grid_length * column/grid_dim - grid_length/2)^2) <= ...
            perimeter_radius

            % if the irradiance on the current grid element is larger than
```

```

% the until now maximum, store the location and irradiance
% value of the current grid element
if irradiance_grid(row, column) > irradiance_max
    irradiance_max = irradiance_grid(row, column);
    position_max = ...
        [grid_length * row/grid_dim - grid_length/2, ...
         grid_length * column/grid_dim - grid_length/2];

% if the irradiance on the current grid element is smaller than
% the until now minimum, store the location and irradiance
% value of the current grid element
elseif irradiance_grid(row, column) < irradiance_min
    irradiance_min = irradiance_grid(row, column);
    position_min = ...
        [grid_length * row/grid_dim - grid_length/2, ...
         grid_length * column/grid_dim - grid_length/2];
end
end
end
end

```

## D.10. Irradiance optimization

```

% -----
% NAME: irradiance_optimization
% INPUT:
% > n_rings_boundaries: boundaries outlining the range of numbers
% to try for the number of UV LED rings (min, max)
% > leds_per_ring_boundaries:
% boundaries outlining the range of numbers
% to try for the number of UV LEDs in each ring (min, max)
% > distance_rings_min: minimum distance between: each UV LED ring;
% the inner UV LED ring and center UV LED; outer UV LED ring and
% perimeter of the surface to be irradiated [mm]
% > tilt_angle_boundaries:
% boundaries outlining the range of angles to
% try for the tilt angle of each UV LED ring (min, max) [°]
% > led_distance_to_grid_boundaries:
% boundaries outlining the range of distances
% to try for the distance between all UV LED rings and the
% surface to be irradiated (min, max) [mm]
% > overdose_factor_max: maximum value allowed for optimization
% variable  $2 = I_{max} / I_{min}$ ; in words, this factor describes the
% ratio of maximum measured irradiance on the surface over the
% minimum measured irradiance on the surface
% > radiation_pattern: normalized intensity distribution table
% corresponding to degrees [0°-90°]
% > perimeter_radius: radius of the perimeter lining out the
% surface on which the UV LEDs provide radiation [mm]
% > grid_dim: size of the grid
% > grid_length: width and height of the grid [mm]
% > is_reflected: logical value (true/false) indicating if
% reflections from the walls surrounding the irradiated surface
% should be accounted for
% > reflection_coefficient:

```

```

%               coefficient indicating the fraction of
%               reflected UV light intensity
%   > reflection_resolution:
%               resolution with which the position of
%               reflection is computed [°]
% OUTPUT:
%   > n_rings_opt:      number of UV LED rings for optimal power
%                       usage
%   > leds_per_ring_opt: vector of size n_rings indicating the
%                       number of UV LEDs in each ring (in to out) for optimal power
%                       usage
%   > center_distance_opt: vector of size n_rings indicating the
%                       radius of each ring (in to out) for optimal power usage [mm]
%   > tilt_angle_opt:   vector of size n_rings indicating the
%                       zenith angle of each UV LED in each ring (in to out) for
%                       optimal power usage [°]
%   > led_distance_to_grid_opt:
%                       distance from LEDs to the grid for optimal
%                       power usage [mm]
% FUNCTION:
%   This function goes through all possible permutations in layout of
%   the UV LEDs within the bounds for each variable indicated by the
%   parameters of this function.
%   The function optimizes for the so-called optimization_variable1:
%   N/I_min. Here, N is the total number of UV LEDs used in the layout
%   and I_min is the minimal irradiance found on the irradiated
%   surface. Optimizing for this variable means finding the UV LED
%   layout which gives the lowest possible value for this variable. The
%   corresponding layout provides the least amount of power consumption
%   for the highest irradiance on the surface.
%   The second value which is used to optimize the UV LED layout is
%   called optimization_variable2: I_max / I_min. A more descriptive
%   name would be the overdose factor: the ratio of maximum measured
%   irradiance on the surface over the minimum measured irradiance on
%   the surface. The position on the grid with the lowest measured
%   irradiance (I_min) has to be given a dose D, meaning that the
%   position with the highest measured irradiance (I_max) is given an
%   overdose of D*(I_max/I_min). From this description it should be
%   obvious that this overdose factor should be kept as small as
%   possible, therefore the optimization_variable2 is only used to
%   check if it is below the overdose_factor_max parameter.
% AUTHOR:   Roy Bakker & Marcel Brouwers
% DATE:    19-5-2020
% -----
function [n_rings_opt, ...
         leds_per_ring_opt, ...
         center_distance_opt, ...
         tilt_angle_opt, ...
         led_distance_to_grid_opt] = ...
         irradiance_optimization(
         n_rings_boundaries, ...
         leds_per_ring_boundaries, ...
         distance_rings_min, ...
         tilt_angle_boundaries, ...
         led_distance_to_grid_boundaries, ...
         overdose_factor_max, ...
         radiation_pattern, ...

```

```

        perimeter_radius, ...
        grid_dim, ...
        grid_length, ...
        is_reflected, ...
        reflection_coefficient, ...
        reflection_resolution)

% declare and define variables
break_flag = 0;
center_distance_resolution = 1;
optimization_variable1 = inf;
optimization_variable2 = inf;
irradiance_min_opt = -1;
irradiance_max_opt = -1;
n_rings_opt = -1;
leds_per_ring_opt = -1;
center_distance_opt = -1;
tilt_angle_opt = -1;
led_distance_to_grid_opt = -1;

% compute the maximum number of iterations needed to have tried every
% possible permutation of distance between UV LED rings
center_distance_iterations = 0;
max_center_distance_iterations = 0;
for n = n_rings_boundaries(1):n_rings_boundaries(2)
    if perimeter_radius < (n+1) * distance_rings_min
        continue
    else
        % the formula for calculating the number of iterations needed to
        % have tried every permutation of distance between UV LED rings
        % for n number of UV LED rings
        max_center_distance_iterations = ...
            max_center_distance_iterations + ...
            nchoosek((perimeter_radius - ...
                ((n+1)*distance_rings_min) + 1) + n - 1, n);
    end
end

% go through every possible value of number of UV LED rings bounded by the
% n_rings_boundaries parameter
for n_rings = n_rings_boundaries(1):n_rings_boundaries(2)

    % declare and initialize the vector of size n_rings indicating the
    % radius of each ring (in to out) [mm]
    center_distance = zeros(n_rings, 1);

    % go through every possible value of the sum of radii of each UV LED
    % ring
    for center_distance_sum = ...
        0: center_distance_resolution: perimeter_radius^n_rings

        % compute the vector of size n_rings indicating the radius of each
        % ring (in to out) using the current value of the sum of radii of
        % each UV LED ring [mm]
        for n = 1:n_rings

```



```

% compute the radius of the n'th UV LED ring (in to out) [mm]
center_distance(n) = rem(floor(...
    (center_distance_sum-center_distance_resolution) / ...
    perimeter_radius^(n-1)), perimeter_radius) + 1;

% if the computed radius is larger than the outer boundary of
% the surface to be irradiated minus some margin, skip the
% whole irradiance calculation as the current UV LED layout
% would not be valid
if center_distance(n) > perimeter_radius - distance_rings_min
    break_flag = 1;
    break;
end
% if the most inner UV LED ring has a radius smaller than the
% minimum radius, skip the whole irradiance calculation as
% the current UV LED layout would not be valid
if n == 1
    if center_distance(1) < distance_rings_min
        break_flag = 1;
        break;
    end
end
% if the computed radius of the current observed UV LED ring is
% smaller than the that of the previous ring plus some
% margin, skip the whole irradiance calculation as the
% current UV LED layout would not be valid
else
    if center_distance(n) < ...
        center_distance(n-1) + distance_rings_min
        break_flag = 1;
        break;
    end
end
end
end

% if the current iteration needs to be skipped, break from the
% current iteration and clear the flag indicating such action
if break_flag == 1
    break_flag = 0;
    continue;
end

% increment the iteration counter for the center distance loop
center_distance_iterations = center_distance_iterations + 1;

% compute the maximum number of iterations needed to have tried
% every possible permutation of number of UV LEDs in every UV LED
% ring
leds_per_ring_iterations = 0;
max_leds_per_ring_iterations = (leds_per_ring_boundaries(2) - ...
    leds_per_ring_boundaries(1) + 1)^n_rings;

% declare and initialize the vector of size n_rings indicating the
% number of UV LEDs in each ring (in to out)
leds_per_ring = zeros(n_rings, 1);

% go through every possible value of the sum of UV LEDs in all UV

```

```

% LED rings (excluding the center UV LED)
for n_total_leds = 1:((leds_per_ring_boundaries(2) - ...
    leds_per_ring_boundaries(1) + 1)^n_rings)

    % compute the vector of size n_rings indicating the number of
    % UV LEDs in each ring (in to out) using the current value of
    % the sum of UV LEDs in all UV LED rings
    for n = 1:n_rings

        % compute the number of UV LEDs in the n'th UV LED ring (in
        % to out)
        leds_per_ring(n) = rem(floor(...
            (n_total_leds-1) / ...
            (leds_per_ring_boundaries(2) - ...
            leds_per_ring_boundaries(1) + 1)^(n-1)), ...
            leds_per_ring_boundaries(2) - ...
            leds_per_ring_boundaries(1) + 1) + ...
            leds_per_ring_boundaries(1);

    end

    % increment the iteration counter for the UV LEDs per ring loop
    leds_per_ring_iterations = leds_per_ring_iterations + 1;

    % compute the maximum number of iteration needed to have tried
    % every possible permutation of tilt angle of UV LEDs in
    % every UV LED ring [°]
    tilt_angle_iterations = 0;
    max_tilt_angle_iterations = (tilt_angle_boundaries(2) - ...
        tilt_angle_boundaries(1) + 1)^n_rings;

    % declare and initialize the vector of size n_rings indicating
    % the zenith angle of each UV LED in each ring (in to out)
    % [°]
    tilt_angle = zeros(n_rings, 1);

    % go through every possible value of the sum of tilt angles of
    % all UV LED rings
    for tilt_angle_sum = 1:((tilt_angle_boundaries(2) - ...
        tilt_angle_boundaries(1) + 1)^n_rings)

        % compute the vector of size n_rings indicating the zenith
        % angle of each UV LED in each ring (in to out) using the
        % current value of the sum of tilt angles of all UV LED
        % rings [°]
        for n = 1:n_rings

            % compute the tilt angle of UV LEDs in the n'th UV LED
            % ring (in to out) [°]
            tilt_angle(n) = rem(floor(...
                (tilt_angle_sum-1) / ...
                (tilt_angle_boundaries(2) - ...
                tilt_angle_boundaries(1) + 1)^(n-1)), ...
                tilt_angle_boundaries(2) - ...
                tilt_angle_boundaries(1) + 1) + ...
                tilt_angle_boundaries(1);

        end
    end

```

```

% increment the iteration counter for the tilt angle per
% ring loop
tilt_angle_iterations = tilt_angle_iterations + 1;

% go through every possible value for the distance of the
% UV LED rings from the surface to be irradiated bounded
% by the led_distance_to_grid_boundaries parameter [mm]
for led_distance_to_grid = ...
    led_distance_to_grid_boundaries(1): ...
    led_distance_to_grid_boundaries(2)

    % compute the irradiance on the surface with the
    % current LED layout permutation
    [irradiance_grid, ~] = ...
        get_irradiance_grid_sum_ring(...
            n_rings, ...
            leds_per_ring, ...
            center_distance, ...
            tilt_angle, ...
            led_distance_to_grid, ...
            1, ... % led size
            radiation_pattern, ...
            1, ... % led radiant flux
            perimeter_radius, ...
            grid_dim, ...
            grid_length, ...
            is_reflected, ...
            reflection_coefficient, ...
            reflection_resolution, ...
            false);

    % compute the minimum and maximum weighted irradiance
    % values on the surface [1/cm^2]
    [irradiance_min, irradiance_max, ~, ~] = ...
        compute_irradiance_uniformity(...
            irradiance_grid, ...
            perimeter_radius, ...
            grid_dim, ...
            grid_length);

    % check if the overdose value, optimization_variable2,
    % is lower than the maximum allowed overdose value,
    % and check if optimization_variable1, N/I_min, is
    % lower than any previously found value
    if irradiance_max / irradiance_min <= ...
        overdose_factor_max && ...
        (sum(leds_per_ring)+1) / irradiance_min < ...
        optimization_variable1

        % compute the new "best" optimization_variable1
        % and optimization_variable2 values
        optimization_variable1 = ...
            (sum(leds_per_ring)+1) / irradiance_min;
        optimization_variable2 = ...
            irradiance_max / irradiance_min;
    end
end

```

```

% store the minimum and maximum irradiance values
% for the current "best" UV LED layout
irradiance_min_opt = irradiance_min;
irradiance_max_opt = irradiance_max;

% store the UV LED layout parameters which provide
% the current "best" UV LED layout
n_rings_opt = n_rings;
leds_per_ring_opt = leds_per_ring;
center_distance_opt = center_distance;
tilt_angle_opt = tilt_angle;
led_distance_to_grid_opt = led_distance_to_grid;
end

% print various UV LED layout characteristics of the
% current iteration
fprintf("distance iteration = %d/%d, ", ...
        center_distance_iterations, ...
        max_center_distance_iterations);
fprintf("n_led iteration = %d/%d, ", ...
        leds_per_ring_iterations, ...
        max_leds_per_ring_iterations);
fprintf("tilt_angle iteration = %d/%d, ", ...
        tilt_angle_iterations, max_tilt_angle_iterations);
fprintf("dist_to_grid = %d, ", led_distance_to_grid);
fprintf("n_rings = %d, {"", n_rings);
for n = 1:n_rings
    fprintf(" n_leds(%d) = %d; ", n, leds_per_ring(n));
end
fprintf("}, {"");
for n = 1:n_rings
    fprintf(" r(%d) = %d; ", n, center_distance(n));
end
fprintf("}, {"");
for n = 1:n_rings
    fprintf(" a(%d) = %d; ", n, tilt_angle(n));
end
fprintf("}, opt. variable = %.3f, ", ...
        (sum(leds_per_ring)+1) / irradiance_min);
fprintf(" I_min = %.3f, I_max = %.3f", ...
        irradiance_min, irradiance_max);

% print various UV LED layout characteristics of the
% current "best" implementation
fprintf("\n\tcurrent optimum: dist_to_grid = %d, ", ...
        led_distance_to_grid_opt);
fprintf("n_rings = %d, {"", n_rings_opt);
for n = 1:n_rings_opt
    fprintf(" n_leds(%d) = %d; ", ...
            n, leds_per_ring_opt(n));
end
fprintf("}, {"");
for n = 1:n_rings_opt
    fprintf(" r(%d) = %d; ", ...
            n, center_distance_opt(n));
end
end

```

```
fprintf("}, {"");
for n = 1:n_rings
    fprintf(" a(%d) = %d; ", n, tilt_angle_opt(n));
end
fprintf("}, optimum (N/I_min) = %.3f, ", ...
    optimization_variable1);
fprintf("optimum (I_max/I_min) = %.3f, ", ...
    optimization_variable2);
fprintf(" I_min = %.3f, I_max = %.3f\n", ...
    irradiance_min_opt, irradiance_max_opt);
end
end
end
end
end
end
```