A CUSTOM-DESIGN ATMOSPHERIC CHANNEL EMULATOR FOR THE PERFORMANCE EVALUATION OF FREE SPACE OPTICAL COMMUNICATION SYSTEMS

A Thesis

by

Burak KEBAPCI

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master in Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
August 2017

A CUSTOM-DESIGN ATMOSPHERIC CHANNEL EMULATOR FOR THE PERFORMANCE EVALUATION OF FREE SPACE OPTICAL COMMUNICATION SYSTEMS

Approved by:

Professor Murat Uysal, Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Assistant Professor Tuncer Baykas
Department of Electrical and Electronics Engineering
Özyeğin University

Assistant Professor Kadir Durak
Department of Electrical and Electronics Engineering
Özyeğin University

Date Approved: 2 August 2017

To my Mother & Father

ABSTRACT

In this thesis, we present our own custom design atmospheric channel emulator for free space optic (FSO) system evaluations and experimentally investigate the performance of FSO links in different weather conditions. Our emulator is in the form of an atmospheric chamber with dimensions of 60 cm $\times$ 40 cm $\times$ 300 cm . It is made from plexiglas with 20 mm thickness. It has two main compartments. The upper compartment includes all electronics and control units while the optical propagation takes place in the lower compartment. On the left and right hand sides of lower compartment, we have 5 holes to launch/collect the optical signal. The emulator is equipped with adjustable heaters, coolers and fans to create the turbulence. It also houses a fog generator and droplet watering system to generate different weather conditions. The conditions within the emulator are monitored through temperature, humidity, visibility, altitude, pressure, dust and wind speed sensors. The functioning of the emulator is controlled by a touch screen controller where heat level/uniformity, wind speed/uniformity, rain and fog levels can be adjusted.Using this custom emulator, we present a link budget and quantify the geometrical loss. Then, we emulate foggy,rainy and turbulent weather conditions inside the chamber and quantify the absorption loss for different wavelengths. Based on the experimental data, we also propose a new and accurate model for the prediction of fog attenuation coefficient.

ÖZETÇE

Bu tezde, kendi tasarımımız olan atmosferik kanal emülatörümüzü serbest uzay optik (free space optics,FSO) sistemini ölçümlemek ve deneysel olarak FSO linklerinin farklı hava koşullarında test etmek için kullanıyor ve sonuçlarını detaylı analiz ediyoruz. Emülatörümüz atmosferik çember formunda 60cmx40xcmx300cm boyutlarına sahiptir. 20mm kalınlığında pleksiglass maddesinden yapılmış olup iki ana bölüme sahiptir. Üst bölüm tüm elektroniklere ve kontrol ünitelerine ev sahipliği yapan kısımdır. Alt bölüm ise optik yayılımın yapıldığı yerdir. Alt kompartımanın sağ ve sol tarafında sinyali göndermek ve toplamak için yerleştirilmiş 5 adet delik bulunmaktadır. Emülatör ısıtıcılar soğutucular ve fanlar kullanarak türbülans yaratır. Ayrıca farklı hava koşulları yaratabilmek için sis üreteci ve bir yağmur sistemi barındırır. Emülatörün içerisindeki ortam sıcaklık, nem, görüş, yükseklik, basınç, toz ve anemometre sensörleriyle sürekli takip edilir. Emülatör sıcaklık seviyesinin ve dağılımının ,rüzgar hızının ve dağılımının, yağmur ve sis seviyelerinin ayarlanabildiği bir dokunmatik arayüzle kontrol edilir. Tasarladığımız atmosferik kanal emülatörüyle önce, link maliyetini sunuyor ve geometrik kaybı ölçümlüyoruz. Daha sonra sisli yağmurlu ve türbülanslı hava durumlarını yaratıyor, farklı dalgaboylarındaki sinyal gücü kaybını inceliyoruz. Elde edilen sonuçlardan sis nedeniyle oluşan sönümleme katsayısı için daha isabetli yeni bir model öne sürüyoruz.

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Professor Murat Uysal for his guidance to make me able to complete this thesis.

Also I want to thank my colleges Hatef Nouri and Farshad Miramirkhani for their their assistance. They helped me a lot during experiments.

Finally, I must thank to my beloved Father and Mother who believed me all the time and supported not just this thesis but for whole of my life.

TABLE OF CONTENTS

DEDICATION . . . . **iii**

ABSTRACT . . . . **iv**

ÖZETÇE . . . . **v**

ACKNOWLEDGEMENTS . . . . **vi**

LIST OF TABLES . . . . **ix**

LIST OF FIGURES . . . . **x**

I INTRODUCTION **1**

1.1 Historical Overview of FSO Communication 1

1.2 Fundamentals of FSO Communication 3

1.3 Atmospheric Attenuation 5

1.3.1 Geometric Loss 5

1.3.2 Fog Attenuation 5

1.3.3 Rain Attenuation 6

1.4 Atmospheric Turbulence 6

1.5 Related works on Atmospheric Emulation 9

II DESIGN OF ATMOSPHERIC EMULATOR **17**

2.1 Mechanical Design 17

2.2 Electronic Design 19

2.3 Software Architecture 22

III MEASUREMENT CAMPAIGN **24**

3.1 Experimental Set-up 25

3.2 FSO Link Budget 26

3.2.1 Link Margin Analysis 30

3.2.2 Pointing Loss 31

3.2.3 Optical Loss 32

3.2.4 Geometrical Loss 32
3.3 Characterization of Rain Attenuation 34
3.4 Characterization of Fog Attenuation 39
3.5 Characterization of Turbulence 47
3.6 Noise of the FSO System 49
IV CONCLUSION 51
APPENDIX A — ATMOSPHERIC EMULATOR SOURCE CODES 52
REFERENCES 91
VITA 97

LIST OF TABLES

1 Literature search of atmospheric chamber articles. 10
2 Literature search of atmospheric chamber articles. 11
3 Literature search of atmospheric chamber articles. 12
4 Literature search of atmospheric chamber articles. 13
5 Literature search of atmospheric chamber articles. 14
6 Literature search of atmospheric chamber articles. 15
7 Literature search of atmospheric chamber articles. 16
8 Parameters of FSO Link 29
9 Received power after discrimination of Pointing, Geometrical and Optical Losses. 34
10 The table of link margin for the system before the rain function starts 34
11 Average of measured temperatures and refractive indices for different cases 49

LIST OF FIGURES

1 Side perspective of Atmospheric Chamber. Dimensions: 3 meter long 60 cm tall. 18

2 Side perspective of Atmospheric Chamber. Dimensions: 3 meter long 60 cm tall. 18

3 Top perspective of Atmospheric Chamber. Dimensions: 3 meter long 60cm width 18

4 3D design of Atmospheric Chamber. 19

5 User interface of Atmospheric Chamber. 21

6 Transmitted laser beam in atmospheric chamber. 24

7 Three different photodiodes two of them covers 1550nm and 1310nm one of them is for 850nm 26

8 Three different laser diodes with different wavelengths 1550nm, 1310nm, 850nm. 27

9 Traveled path of laser beam inside of Atmospheric Chamber. 28

10 Current and temperature controller for Laser diode. 28

11 The aspheric lens at front of the laser diode for collimating laser beam. 28

12 The Plano-Convex lens at front of the photo diode for focusing light to active area of photodiode. 29

13 Power meter that used for measuring received signal power. 29

14 Demonstration of measurement equipment. 29

15 Demonstration of sending collimated laser beam. 31

16 Calculation of beam size after 9mrad divergence. 33

17 Demonstration of increased divergence angle of laser beam 33

18 Geometrical loss for different divergence angles. 35

19 Received power using 1550nm laser beam, before and after rainy weather. 36

20 Received power using 1310nm laser beam, before and after rainy weather. 36

21 Attenuation of rainy weather using 1310nm laser beam. 37

22 Attenuation of rainy weather using 1550nm laser beam. 37

23 Comparison of different wavelengths received power in time during fog events. 40

24 Received power in time during foggy events using 1550nm laser beam. 41

25 Received power in time during foggy events using 1550nm laser beam. 41

26 Received power in time during foggy events using 1310nm laser beam. 42

27 Received power in time during foggy events using 1310nm laser beam. 42

28 Attenuation coefficient with respect to visibility for 850 nm 44

29 Attenuation coefficient with respect to visibility for 1310 nm 44

30 Attenuation coefficient with respect to visibility for 1550 nm 45

31 Attenuation coefficient with respect to visibility for comparison of fitted curves for three wavelengths. 45

32 Comparison of the proposed model for q parameter with experimental data 46

33 Effect of turbulence on channel fading coefficient 47

34 Normalized received power in time during heat events. 48

35 Gaussian fit of received noise samples. 50

CHAPTER I

INTRODUCTION

1.1 Historical Overview of FSO Communication

Communication is most essential thing for people.Establishing a wireless communication link between people was one the most important challenge in history of technology. People always dreamed about communication from far distance and they tried many things to do that from birds to smoke signals. In 1684 first optical telegraph invented by Robert Hooke and it implemented by Sir Richard Lovell Edgeworth in 1767. They were trying to communicate with visual position of some mechanical objects. In 1791 first wireless communication link established between Brulon and Parce a distance of 16 kilometres. There were many towers between those cities and they were simply redirecting messages to next towers. This technology is used until 19th century. Then electrical telegraph invented and many of old fashioned communication technologies became history. Since wired communication were more simpler then wireless communication , after invention of telegraph wired communication technologies became too popular and many researchers started working on. As we have understand from history of human being able to establish a wireless communication link was a instinctive choice of human rather than wired communication link.

In early 20th Century after Heinrich Hertz proved existence of radio waves first long range radio system invented by Guglielmo Marconi. He succeeded to send message ”s” through Atlantic. That boosted motivations of researches on wireless communication technology development. He won 1909 nobel physics prize for his works done on area of wireless telegraph.

WWII had big impact on rapid development of radio technology. All countries were asking for wireless communication links to communicate their troops. They were asking for reliable, secure and long distance wireless communication links. Decoding German Enigma Codes led allies to won war 2 years early some historicians said. Enigma codes were using to secure wireless transmitted signals. Enigma were small encyrpytion machine which decrypts and encrypts messages.

1G which means First Generation of cellular communication technology first used in Tokyo 1979. It rapidly spread across the whole of japan then Nordic countries in 1981. 2G used 900MHz for communication and it widely spread to whole world, because people moved from brick like phones to 200 gram lightweight mobile phones. In last 30 years near to all emerging wireless communication technologies used radio waves for communication. That can be seen in figure XX. Allocation of frequency for a technology became very problematic. For example there are many technologies in 2.4Ghz from sensor networks to bluetooth. Those crowded spectrum creates huge interference to all radio wave based wireless technologies. Due to crowd of technology in spectrum emerging technologies can not find enough bandwitdh for their applications. Those problems creating limitations for recent emerging wireless technology.

The term optical wireless communication covers Ultra-Violet light, Visible Light and Infrared communication. First example of optical wireless communication technology is invented by Alexander Graham Bell. He used some mirrors and lenses for voice transmission using sunlight. He was able to receive optically converted voice signal through 213 meters away.

Using optical wireless communication rather than radio waves for communication is bypassing all crowdence of radio spectrum. That enables having extremely high bandwidth interference free communication links.

Point to Point wireless communication technologies has different requirements for different applications such as delay, data rate, range ,reliability and so on. There are various types of point to point communications such as RF, Microwave or FSO (Free

Space Optics). Those point to point technologies demands different service quality, service life, cost (Terminal costs and regulations) and also data rate.

Free space optical (FSO) systems offers an extensive modulation bandwidth and higher data rates then many others, because it uses fully available optical spectrum. It provides low power consumption, enhanced security, a reliable solution to the last mile problem and does not require administrative licensing. Therefore, FSO can be implemented as an good option in urban and local areas. FSO communication systems can be used in a wide range of application areas including enterprise/campus connectivity, fiber back-up, backhauling for cellular networks, and airborne communications among others [1]. Atmospheric attenuation of FSO links is dependent on weather conditions such as fog, rain, dust, and various combinations of each. Furthermore, FSO links experience atmospheric turbulence induced fading as a result of the variations in the refractive index.

Due to the difficulty to predict the occurrence of specific atmospheric effects and to repeat the measurements under the same conditions, it is highly desirable to evaluate the FSO system performance in a controlled laboratory environment. Along this direction, several works have been recently reported in the literature [2-9] where various environmental effects (turbulence, fog, smoke, dust, ash, and sand) are investigated in custom design atmospheric chambers. Particularly, the effect of foggy weather conditions is studied in [7-9] based on measurements in an atmospheric chamber. The degrading effect of fog on the bit error rate (BER) performance of the FSO link is quantified in [7]. In [9], a wavelength-dependent model is proposed for fog and smoke channels valid within the visibility ranges of less than 1 km.

1.2 Fundamentals of FSO Communication

Free-Space optical communication (FSO) is a wireless communication technology that uses light as carrier for communication through free space. The sentence Free-Space includes air, outer space and so on. FSO technology uses lasers and different types of photo-diodes for data transmission. Generally 2 types of photodetectors are used for FSO systems, PIN (Undoped intrinsic semiconductor between P-type and N-type semiconductor.) and APD (Avalanche Photodiode). Both has different characteristics, while PIN is suitable for low bandwidth low data rate communications, APD is more suitable for high bandwidth and high data rate communications. APD has more layers which makes depletion region level lower, so that makes APD more sensitive for same amount of photons over PIN. The main reason that electronic companies still want to use PIN is very cheap and easy to fabricate technology compared to APD [1]. At the transmitter side mostly used ones are continuous wave laser diodes which is cheap and common technology. Bandwidth of lasers are very dependent to working temperature so monitoring laser is very important for stable laser driving operation. To have a stable laser driven laser a monitoring photodiode is been used. Most of the lasers has that photodiode cell next to laser diode cell to monitor level of dissipated photons by laser. Automatic Power Control (APC) is an integrated circuit technology that has been developed for limiting driving current of laser diode which uses that internal photodiode as reference. APC technology is crucial for FSO terminals, because FSO terminals generally deployed on top of buildings. Since it is an outdoor environment terminal temperature changes every time. There is also some researches on usage of femtosecond lasers on FSO. The main difference is femtosecond lasers has very short light pulses meanwhile continuous wave laser diodes has constant amplitude constant phase angle. Femtosecond lasers are generally used in high power operations, because those lasers focuses all of its power to single very short pulse. Recent researches show that low power femtosecond lasers can be used as transmitter in FSO terminals. Because of their very short duration pulses they have much more bandwidth then conventional continuous wave laser diodes, but in terms of fabrication they are harder to produce and more expensive. That's why current

FSO terminals using continuous wave laser diodes.

The optical communication technology behind of FSO comes from fiber based optical communications. Fiber optic communication has time invariant channel. On the other hand, FSO channel varies too fast and that makes FSO more dependent to channel. Because of that analysis of channel affects on FSO is vital as other wireless communication technologies.

1.3 Atmospheric Attenuation

1.3.1 Geometric Loss

An unguided electromagnetic wave propagating in free space experiences power loss at the receiver end due to the geometry of diffraction combined with the limited effective area of the receiver. The reduction in optical power at the receiver due to the geometrical loss h_g can be estimated by [1]

$$h_g \approx \frac{D_{rx}^2}{(D_{tx} + \theta_{tx} L)^2} \approx \frac{D_{rx}^2}{(\theta_{tx} L)^2} \qquad D_{tx} << \theta_{tx} L \tag{1}$$

where D_{tx} is the transmitting lens diameter, θ_{tx} is the transmitted beam divergence angle, L is the propagation distance, and D_{rx} is the receiving lens diameter. Geometric loss arises due to divergence of the transmitted optical beam with respect to the line-of-sight (LOS) and increases with propagation distance.The geometric loss is a fixed loss for a given WOC link independent of weather conditions.

1.3.2 Fog Attenuation

According to Beer's law, the attenuation of light traveling through the atmosphere due to both absorption and scattering is given as $h_a = \exp\left(\alpha_{\mathrm{fog}} L\right)$ where α_{fog} is the fog attenuation coefficient. It is defined as [11]

$$\alpha_{\mathrm{fog}} = (3.91/V)\,(\lambda/\lambda_0)^{-q} \tag{2}$$

where V is the visibility in km, λ_0 =550 nm is the maximum spectrum of solar band (i.e., green light) and q is a parameter that can be determined from experimental data. According to the Kruse model [11], we have

$$q = \begin{cases} 0.585V^{1/3} & V < 6\text{km} \\ 1.3 & 6\text{km} < V < 50\text{km} \end{cases} \quad (3)$$

According to Kim's model [12], q is given by

$$q = \begin{cases} 1.6 & \text{for} \quad V > 50 \ \text{km} \\ 1.3 & \text{for} \quad 6 < V < 50 \ \text{km} \\ 0.16V + 0.34 & \text{for} \quad 1 < V < 6 \ \text{km} \\ V - 0.5 & \text{for} \quad 0.5 < V < 1 \ \text{km} \\ 0 & \text{for} \quad V < 0.5 \ \text{km} \end{cases} \quad (4)$$

In a recent work [9], Ijaz *et al.* argue that the Kim model overestimates the fog attenuation for low visibility and propose

$$q = 0.1428\lambda - 0.0947 \ , \quad V < 1 \ \text{km}. \quad (5)$$

1.3.3 Rain Attenuation

Although the main attenuation factor for an optical wireless link is fog, the rain can also impose certain attenuation levels for such links. For the rain attenuation coefficient of WOC channel, the following model can be used [4]

$$\sigma_{rain} = 1.076R_{rain}^{0.67} \quad (6)$$

where R_{rain} is the rain rate in mm/h. The attenuation coefficient caused by rain can be calculated as

$$h_{rain} = e^{-\sigma_{rain} L}$$

1.4 Atmospheric Turbulence

The main limiting factor of FSO is the signal power noise that caused by scintillation. Since FSO is an point to point signal and uses highly directed beams on air the signal passes through many turbulent areas. Those turbulent areas creates scintillation.

Scintillation happens ,because turbulent areas have slightly different heat levels and air pressure. While just a turbulent area distorts propagated laser beam slightly, many of them causes huge distortion in signal. This distortion affects signal intensity and amplitude at receiver side. This difference varies in time. For all optical wireless communication systems investigating performance of the system in a turbulent channel starts with characterization of the scintillation effects. This characterization is a big challenge, because testing those systems requires too much time and effort.[2] Generally, those tests requires a whole year to see all affects of weather on system. Hence, there are many analytic and numerical methodologies have been developed to estimated affect of channel on transmitted signal.

The scintillation index σ_I is the measurement type of the atmospheric scintillation and it is defined as normalized variance of the signal intensity fluctuations.[2]

$$\sigma_I = \frac{< I^2 >}{< I >^2} - 1 \tag{7}$$

In (7) I denotes optical intensity at the receiver aperture and ¡¿ refers to average of inner values. In this equation medium is assumed as an homogeneous channel. For the following numerical calculation fluctuations at the received optical signal is just considered by refractive-index of the atmosphere.

$$\mathfrak{D}_n\left(P\right) =< \mathfrak{n}\left[r,t\right], \mathfrak{n}[r+\rho,t] > \tag{8}$$

For (8) refractive index is denoted by $\mathfrak{D}_n$ and r is location in space, ρ is the distance between tx and rx and t is time. Since atmospheric turbulence varies by time refractive index also varies by time. There is also an formula for refractive-index structure and it is approximated by the following equation:

$$\mathfrak{D}_n\left(\rho\right) = C_n^2 \left(\sqrt[3]{|\rho|}\right)^2 \tag{9}$$

C_n^2 refers to refractive-index structure parameter. When $\sigma_I << 1$ atmospheric condition has very weak turbulence. In that kind of condition scintillation index is proportional to Rytov variance β_0 which is defined by this equation:

$$\beta_0 = \alpha C_n^2 k^{7/6} L^{11/6} \tag{10}$$

where L is length of propagation path. In short distances σ can be considered as $\sigma = 1.23$ in long distances $\sigma = 0.5$ which is related to approximation of the propagated wave. Distance σ relation is comes from that approximation; it can be considered as plane wave in short distances while spherical wave for long distances. The Rytov variance is applicable for weak turbulence conditions. Once atmospheric turbulence increases, scintillation index also starts increasing. We can call C_n^2 and L as optical turbulence parameters and both increasing or decreasing C_n^2 and L have same effect on Rytov variance.

Refractive index (or atmospheric) structure parameter C_n^2 is highly dependent on the small scale temperature fluctuations, the temperature structure constant, and the atmospheric pressure P is given by

$$C_n^2 = \left(79 \times 10^{-6} \frac{P}{T^2}\right)^2 C_T^2$$

where $P = 1002.8$ millibar and T is the absolute temperature in Kelvin. The temperature structure constant C_T^2 is related to the universal 2/3 power law of temperature variation as given in

$$D_T = \left\langle (T_1 - T_2)^2 \right\rangle = \begin{cases} C_T^2 l_0^{-4/3} R^2 & \text{for} \quad 0 \ll R \ll l_0 \\ C_T^2 R^{2/3} & \text{for} \quad l_0 \ll R \ll L_0 \end{cases}$$

where T_1 and T_2 are the temperatures at two points separated by the propagation distance R. The parameters l_0 and L_0 are inner and outer scales of the small-temperature fluctuations.

The Rytov variance of the channel is related to the refractive-index structure constant C_n^2, and is given as $\sigma_R^2 = 0.124\,(2\pi/\lambda)^{7/6}\, C_n^2 d^{11/6}$.

1.5 Related works on Atmospheric Emulation

Optical channel is highly dependent to weather condition and distance between terminals. The best approach for testing an optical wireless communication system is testing in different weather conditions. Generally researchers wait a year to test their own optical wireless link in different weather conditions . To perform too many tests in short time building an atmospheric emulator become compulsory for testing our optical communication systems. We have started to design our own atmospheric emulator to satisfy our needs. We have listed our emulators weather controlling abilities as; heat,visibility range, fog, wind, humidity, rain.

There are several research groups exists which uses an atmospheric emulator for analyzing atmospheric effects on FSO systems. In the following tables a detailed list of the works that has been done on this field can be found. All of the existing atmospheric chambers are built based on manual control. If they want to increase heat a person opens a heater and watches heat sensors manually until desired point. It is same for fog, rain and many other works. This situation can create possibility of erroneous measurements, due to manually controlled equipment's. In our design we have proposed fully autonomous atmospheric chamber details can be found in next chapter.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[22]	-Closed glass chamber -550 × 30 × 30 cm^3 -Seven compartment -Cold air (20°-25°) -Hot air (20°-80°) -Wind (4-5 m/s)	-Weak and medium	-830 nm	-Narrow divergence beam laser (TX) -Concentration lens and PIN PD (RX)	-Clear weather	-NRZ / RZ	-Eye diagram -BER vs. Rytov variance -Q factor vs. Rytov variance
[23]	-Closed glass chamber -550 × 30 × 30 cm^3 -Cold air -Hot air -Fog generator	-Weak	-830 nm	-Narrow divergence beam laser (TX) -Concentration lens and PIN PD (RX)	-Clear weather -Foggy weather	-OOK-RZ -4-PAM -SIM (subcarrier intensity modulation)-BPSK	-Q factor vs. Rytov variance for different modulation scheme
[24]	-Closed glass chamber -600 × 30 × 30 cm^3 -Circulating system -Fog generator	-Weak	-830 nm	-Narrow divergence beam laser (TX) -Concentration lens and PIN PD (RX)	-Foggy weather	-OOK-NRZ	-Q-factro and BER vs. transmittance (different fog conditions) -Eye diagram
[25]	-Closed glass chamber -550 × 30 × 30 cm^3 -Cold air -Hot air -Fog generator	-Weak	-830 nm	-Narrow divergence beam laser and lens (TX) -Concentration lens and PIN PD (RX)	-Foggy weather	-OOK-NRZ -OOK-RZ -4-PPM	-Q-factro vs. transmittance (different fog conditions)

Table 1: Literature search of atmospheric chamber articles.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[26]	-Closed glass chamber -550 × 30 × 30 cm^3 -Cold air -Hot air -Fog generator	-Weak	-830 nm	-Narrow divergence beam laser (TX) -Concentration lens and PIN PD (RX)	-Foggy weather	-OOK-NRZ -OOK-RZ -4-PPM	-Q-factro vs. transmittance (different fog conditions) -Eye diagram
[27]	-Closed glass chamber -550 × 30 × 30 cm^3 -Seven compartment -Cold air (20°-25°) -Hot air (20°-65°) -Wind (<10 m/s)	-Weak	-830 nm	-Narrow divergence beam laser and lens (TX) -PIN PD (RX)	-Clear weather	-NRZ	-Measured Rytov variance vs. temperature -Eye diagram -BER and Q-factro vs. scintillation index
[28]	-Closed glass chamber -550 × 30 × 30 cm^3 -Fog generator	-No turbulence	-830 nm -1550 nm	-Narrow divergence beam laser (TX) -Power meter (RX)	-Foggy weather	-	-Attenuation vs. visibility
[29]	-Closed glass chamber -550 × 30 × 30 cm^3 -Fog generator	-No turbulence	-1550 nm	-Narrow divergence beam laser (TX) -Power meter (RX)	-Foggy weather	-	-Visibility -Attenuation -Extinction coefficient vs. visibility

Table 2: Literature search of atmospheric chamber articles.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[30]	-Closed glass chamber -550 × 30 × 30 cm^3 -Fog generator -Smoke generator	-No turbulence	-830 nm -1310 nm -1550 nm	-Tungsten halogen and lens (TX) -Optical spectrum analyzer and lens (RX)	-Foggy weather -Smoky weather	-	-Attenuation vs. visibility -Received power and loss vs. wavelength
[31]	-Closed glass chamber -550 × 30 × 30 cm^3 -Fog generator -Smoke generator	-No turbulence	-550 nm -670 nm -830 nm -1310 nm -1550 nm	-Tungsten halogen and lens (TX) -Optical spectrum analyzer and lens (RX)	-Foggy weather -Smoky weather	-	-Attenuation vs. visibility -Attenuation vs. wavelength
[32]	-Closed glass chamber -550 × 30 × 30 cm^3 -Fog generator	-No turbulence	-830 nm -1310 nm -1550 nm	Narrow divergence beam laser and lens (TX) -Power meter and lens (RX)	-Foggy weather	-	-Attenuation vs. visibility -Attenuation vs. wavelength
[33]	-Closed glass chamber -100 × 15 × 15 cm^3 -Circulating system	-Exist	-785 nm	Narrow divergence beam laser (TX) -PD (RX)	-Sandstorm	-OOK-NRZ -OOK-RZ -4-PPM	-Eye diagram -Q-factor -BER

Table 3: Literature search of atmospheric chamber articles.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[34]	-Closed glass chamber -300 × 30 × 30 cm^3 -Cold air (20°-25°) -Hot air (20°-70°) -Wind (4-5 m/s)	-Weak	-830 nm	-Narrow divergence beam laser (TX) -Concentration lens and PIN PD (RX)	-Clear weather	-NRZ -RZ -SIM-BPSK	-Histogram of received signal -Eye diagram
[35]	-Closed glass chamber -300 × 30 × 30 cm^3	-No turbulence	-	-Narrow divergence beam laser and lens (TX) -Concentration lens and PIN PD (RX)	-Foggy weather	-BPSK	-Eye diagram Q-factro vs. transmittance (different fog conditions)
[36]	-Closed glass chamber -550 × 30 × 30 cm^3 -Air temperature (20°-35°) -Fog generator	-Weak	-785 nm	-Narrow divergence beam laser (TX) -Concentration lens and PD (RX)	-Foggy weather	-NRZ-OOK	-Output voltage and BER vs. visibility
[37]	-Closed glass chamber -550 × 30 × 30 cm^3 -Circulating system	-No turbulence	-550 nm -830 nm -1550 nm	-Tungsten halogen and lens (TX) -Optical spectrum analyzer and lens (RX)	-Smoky weather	-	-Loss vs. visibility -Attenuation vs. wavelength

Table 4: Literature search of atmospheric chamber articles.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[38]	-Closed glass chamber -550 × 30 × 30 cm^3 -Cold and hot weather	-Weak	-830 nm	-Narrow divergence beam laser (TX) -PD (RX)	-Clear weather	-NRZ -PPM -BPSK -PPM-BPSK	-Q-factor and SER vs. scintillation index
[39]	-Closed glass chamber -550 × 30 × 30 cm^3 -Hot air (50°-60°) -Circulating system	-Weak	-830 nm	-Narrow divergence beam laser and lens (TX) -PIN PD (RX)	-Clear weather	-NRZ-OOK	-Eye diagram -Q-factor vs. scintillation index -Packet error rate vs. receiver area
[40]	-Heat chamber -7 m link range -Air temperature (25°-110°) -Wind speed (< 15 m/s)	-Moderate to strong	-830 nm	-Narrow divergence beam laser (TX) -PIN PD (RX)	-Clear weather	-NRZ-OOK	-Eye diagram -Histogram of received signal
[41]	-Closed glass chamber -550 × 30 × 30 cm^3 -Air temperature (20°-65°) -Wind speed (< 10 m/s)	-Weak	-830 nm	-Narrow divergence beam laser and lens (TX) -PIN PD (RX)	-Clear weather	-NRZ	-Average refractive index structure and scintillation index vs. temperature gradient -Eye diagram -Received signal distribution -Q-factor and BER vs. scintillation index
	-Closed glass chamber -550 × 30 × 30 cm^3 -Fog and smoke generator	-No turbulence	-670 nm -830 nm -1550 nm	-Tungsten halogen and lens (TX) -Optical spectrum analyzer and lens (RX)	-Foggy and smoky weather	-	-Attenuation vs. visibility and wavelength

Table 5: Literature search of atmospheric chamber articles.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[42]	-Closed glass chamber -550 × 30 × 30 cm^3 -Air temperature (20°-60°) -Wind speed (4 m/s-10 m/s) 6 m & 12 m link ranges	-Weak	-830 nm	-Narrow divergence beam laser (TX) -PIN PD and lens (RX)	-Clear weather	-IM/DD -NRZ-OOK	-Received signal distribution -Q-factor vs. scintillation index
[43]	-Experimental setup with mirror -12 m link range	-No turbulence	-670 nm -850 nm	-Narrow divergence beam laser (TX) -PD (RX)	-Clear weather	-IM/DD -NRZ-OOK -Differential signaling scheme (DS)	-Effect of pointing error -BER vs. SNR
[44]	-Closed glass chamber -650 × 10 × 10 cm^3 -Air temperature (20°-60°) -Wind speed (3 m/s)	-Weak	-650 nm -830 nm	-Narrow divergence beam laser (TX) -PD (RX)	-Smoggy weather (Ash & Sand)	-NRZ-OOK	-Eye diagram -Attenuation vs. air flow velocity -Q-factor
[45]	-Three closed glass chamber -Air temperature (25°-80°) -Wind speed (< 10 m/s) -6.6 m link range	-Weak to strong	-1550 nm	-Narrow divergence beam laser and lens (TX) -PIN and lens (RX) -Optical spectrum analyzer	-Clear weather	-NRZ-OOK -AF relaying scheme -One, two and three hopes	-BER vs. SNR

Table 6: Literature search of atmospheric chamber articles.

	Configuration	Turbulence	Wavelength	Prototype Details	Weather Conditions	PHY Layer Techniques	Performance Metrics
[25]	-Glass chamber -Up to 4 m link range	-Weak and medium	-850 nm	-	-	-	-Spatial coherence distance
[46]	-Three glass chamber -2.2 m link range for each chamber -Cold and hot air	-Weak	-1550.12 nm	-Narrow divergence beam laser and lens (TX) -PIN PD and lens (RX)	-Clear weather	-OOK-NRZ -OOK-RZ	-Eye diagram -BER
[47]	-Free space - 2 m link range	-Weak	-1550 nm	-Narrow divergence beam laser (TX) -PIN PD and lens (RX)	-Clear weather	-16 QAM -OFDM	-Transmission vs. Frequency -EVM vs. SNR
[28]	-Glass chamber -6 m link range	-Weak	-830 nm	-Narrow divergence beam laser (TX) -PIN PD (RX)	-Clear weather	-NRZ-OOK	-Variance of detection threshold
[48]	-Glass chamber -90 × 40 × 40 cm^3 -Circulating system	-No turbulence	-1550 nm	-Narrow divergence beam laser and lens (TX) -Power sensor (RX)	-Dusty weather	-	-Attenuation vs. visibility

Table 7: Literature search of atmospheric chamber articles.

CHAPTER II

DESIGN OF ATMOSPHERIC EMULATOR

Design of atmospheric emulator requires knowledge of different disclipines as mechanics,electronics and software. In this chapter we explain those in detail.

2.1 Mechanical Design

The list of requirements conduct us to create our own chassis design. During our searches we have realized that creating that system on plexy glass material is more cost effective and flexible then other options such as aluminum, plastic and so on. After some tests and measurements on optical communication link that we have decided to set dimensions of our design to 3 meters long 60 cm width 60 cm tall. We have started to design with plotting 2D design on Autocad Mechanical software. Those 2D plots can bee seen in Fig 1, 2 and Fig 3. As seen in Fig 2 Design has 3 floors for different purposes. System creates emulated atmosphere at bottom floor. Second floor for just generating rain effect and top floor of the system has all electronics. The reason we put all electronics to top floor is preventing possible short circuits that may occur, because of the existence of generated water droplets from rain effect. In fig 1,2 and 3 yellow lines denotes floor separations, green lines denotes sensor plates. Well made air isolation between floors also creates more safe emulation environment. There are two wind tunnels exists in this design. With help of those tunnels any form of air can be created. For generation of a simplistic log normal turbulence ,it is important to keep temperature level stable while generating different air forms for small air gaps. That has been done with connecting a air pipe between wind tunnels. That helps us to create a perpendicular stable air flow to signal.

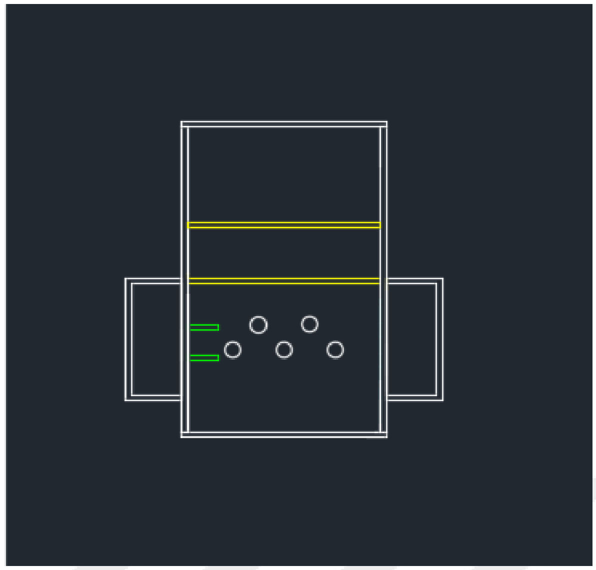

Figure 1: Side perspective of Atmospheric Chamber. Dimensions: 3 meter long 60 cm tall.

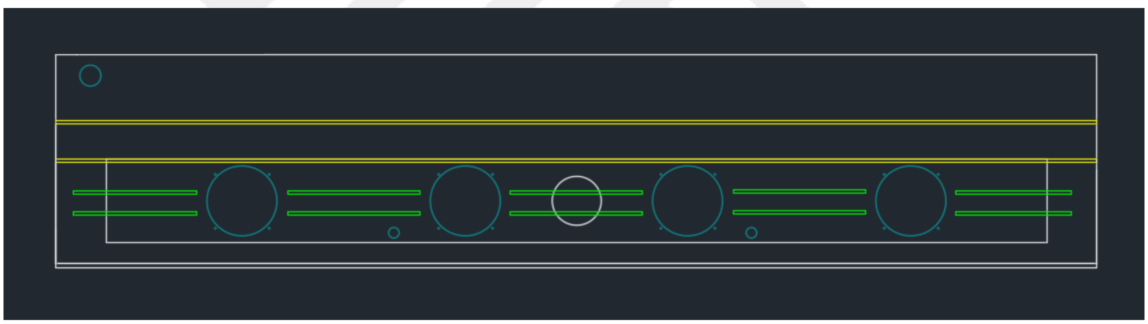

Figure 2: Side perspective of Atmospheric Chamber. Dimensions: 3 meter long 60 cm tall.

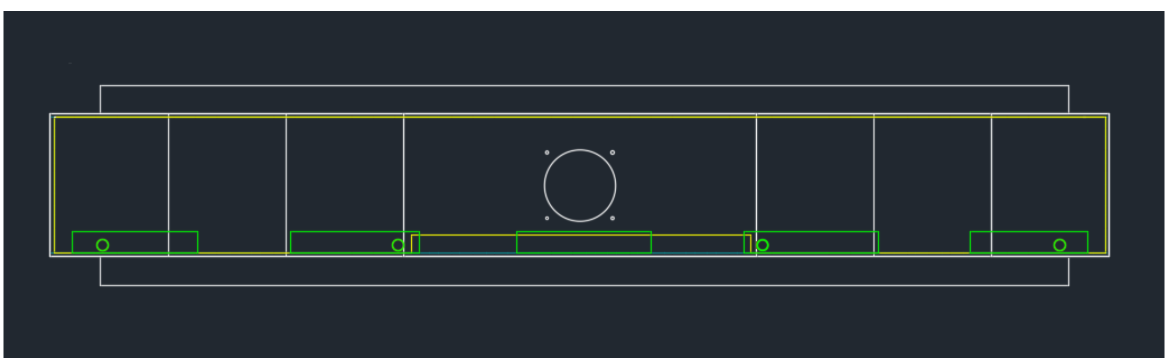

Figure 3: Top perspective of Atmospheric Chamber. Dimensions: 3 meter long 60cm width

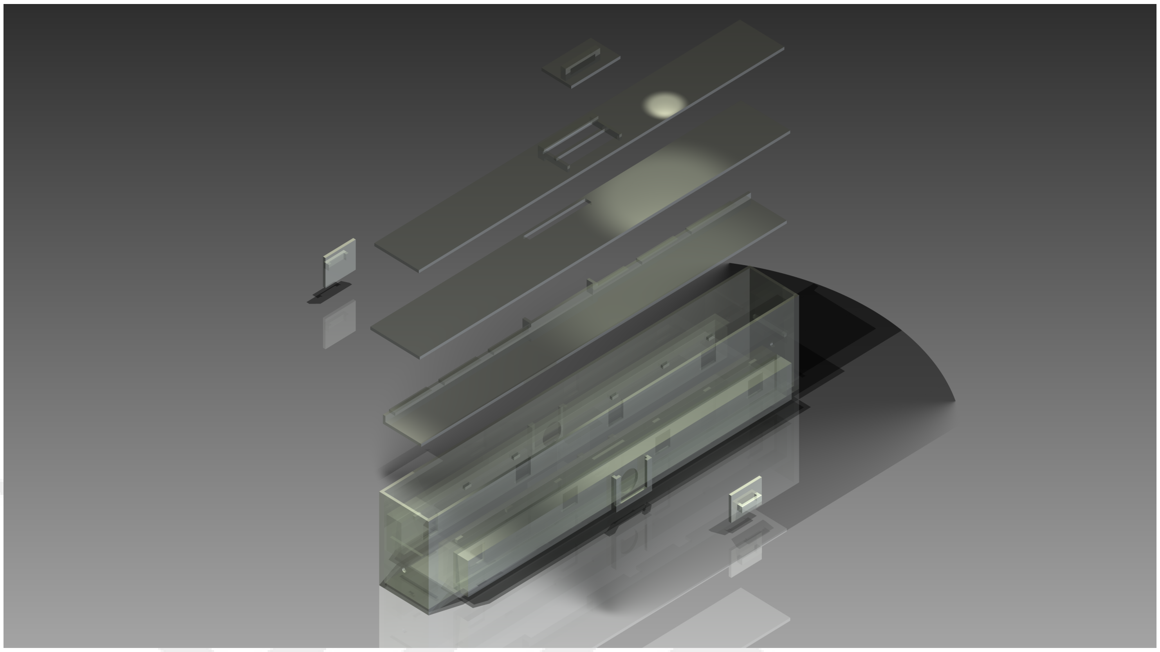

Figure 4: 3D design of Atmospheric Chamber.

2.2 Electronic Design

To provide ease of use and improve efficiency and accurecy of the system we decided to use an centralized controller with touch screen. Parts:

- **Main Controller: Raspberry Pi 2**
  - CPU: 900MHz quad-core ARM Cortex-A7
  - RAM: 1GB LPDDR2 SDRAM
  - USB Slots: 4x USB2.0
  - GPIO: 40 pins
  - Supply: 1A - 5V
- **Slave Micro Controllers: Arduino Mega 2560**
  - Digital Pins: 54
  - Analog Pins: 16

  - Clock Speed: 16 MHz
  - Flash Memory: 256KB

- **Touch Screen:**
  - 7 inches
  - RGB 800x480 display @60fps
  - 24-bit colour
  - 70 degree viewing angle
  - FT5406 10 point capacitive touchscreen

We use two slave micro controllers one for receiving sensor datas and one for controlling relays, motors etc. Separating control and sensing to two different systems enabled more flexibility for future upgrades. When we have wanted to add new sensors or new devices to control we can easily integrate them to related subsystem.

For generating wind effect we use eight 20cm, 130CFM fans. We control those fans with L298 motor drivers. With controlling those fans precisely we can emulate wind speeds from 0 to 20m/s. In normal air conditions laser signal passes through different wind speeds and wind direction also changes directly ability to create that various environments also important for accurate channel emulation. System that we build has ability to generate storm affect , time variant wind speed and different levels of uniformity of wind. We have also used an anemometer for giving feedback to our controlling system to increase accuracy of generated wind form. To generate different wind forms we have designed side tunnels to guide air flow. Those tunnels can be seen in figure 1.

To generate fog we use a fog generator that has specifications of throwing 2500 cubic feet fog per 60 minute. Fog is the one of the most important problem in free space optical communication. Every time fog increases it directly reduces arrival photon

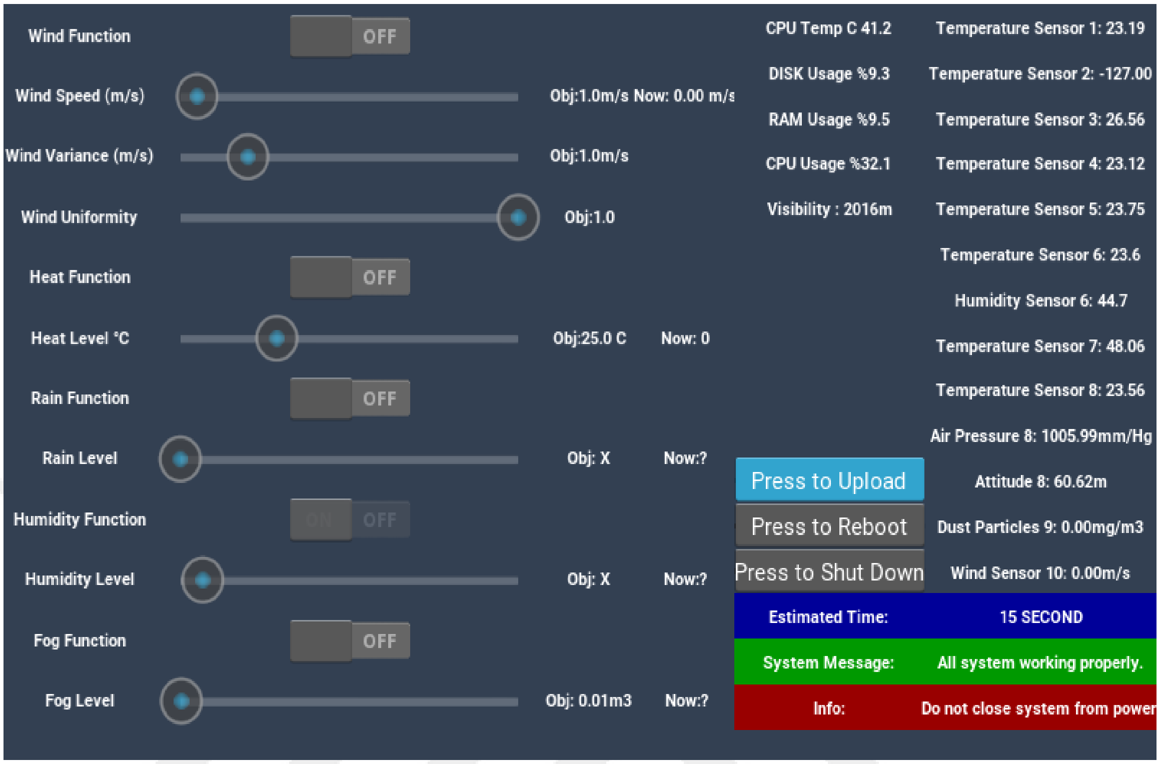

Figure 5: User interface of Atmospheric Chamber.

number, in other words it creates path loss effect. Generation fog is most sensitive part in our design, because in overall fog is the one which affects optical wireless channel at most. So we have started developing our sensor and field of view measurement methodology. We have used one laser and a LDR (Light Dependent Resistor) with appropriate circuit design to measure intensity of the fog in atmospheric chamber. As a result we can directly calculate field of view. Dust particles also has same effect on optical channel as fog effect. To increase correctness of results we measure dust particle numbers in air with accurate optical sensors.

Since generating rain naturally is too hard in a small environment we have decided to use water pumps and accurate water pipes. To increase uniformity we have created a rain floor design which has uniform holes in all over the floor. At the bottom of the bottom floor there is 4 cm tall water level and pump is also placed at that level so we directly push that water to upper rain floor. After that water droplets drops to bottom level this circulation of water runs continuously to generate rain effect.

Generating heat and making accurate measurements all over the channel is also too important. In our design we use 8 DSB1820 temperature sensors to measure heat level at every part of the system. At the end it provides sytem to decide wheter system should open cooler or heater. System follows all sensor readings and makes calculations to estimate overall air temperature. Humidity in air is generally uniform, because of that we use just two humidity sensors. For generating humidity we plan to use humidity generator. Currently it is not in our main interest, because humidity doesn't have too much impact on optical channel compared to heat and visibility.

System has many sensors to measure all properties of air condition. All controlling system is working as a closed loop SCADA ssytem. User dont knows internal processes, but system automatically measures air and decides what next step should be. This increases realism and usability of the system. System has touch screen user interface that provided in figure 5. From that user interface user can control every single aspect of the emulated atmosphere and also user can read sensor values from the touch screen.

2.3 Software Architecture

There are 3 different languages working simultaneously together in atmospheric chamber. At microcontroller side C++ language is used. Kivy language is used for creating user interface and python2 is used as main controlling code and some bash scripts are running for fault detection.

Raspberry pi drives two slave arduinos one of them for measuring sensor values one them for controlling equipment's. This approach is very simplifies the problem and reduces coding requirement significantly. Both of them connected to raspberry pi by USB 2.0. Sensor Slave Arduino sends sensor values to main controller 1 second at time. There is dedicated listener for this and it updates sensor values in RAM

(Random Access Memory). If record button pressed by user a new thread starts for collecting those information and pushing all of them in a .xls file in real time. Meanwhile user clicks stop button .xls file is saved to temporary folder and sensor log file queued for uploading to dropbox file.

Controller Slave Arduino has an active interrupt always waits for a message and executes given task. There is a library for directly sending those tasks from python called nanpy. All of the source code can be found in appendix A.

CHAPTER III

MEASUREMENT CAMPAIGN

Our emulator is in the form of an atmospheric chamber with dimensions of 60 cm $\times$ 40 cm $\times$ 300 (see Fig. 1.a). It is made from plexiglas with 20 mm thickness. It has two main compartments. The upper compartment includes all electronics and control units while the optical propagation takes place in the lower compartment. On the left and right hand sides of lower compartment, we have used 5 holes to launch/collect the optical signal. The emulator is equipped with adjustable heaters, coolers and fans to create the turbulence. It also houses a fog generator and droplet watering system to generate different weather conditions. The conditions within the emulator are monitored through temperature, humidity, visibility, altitude, pressure, dust and wind speed sensors. The functioning of the emulator is controlled by a touch screen controller (see Fig. 1.b) where heat level/uniformity, wind speed/uniformity, rain and fog levels can be adjusted.

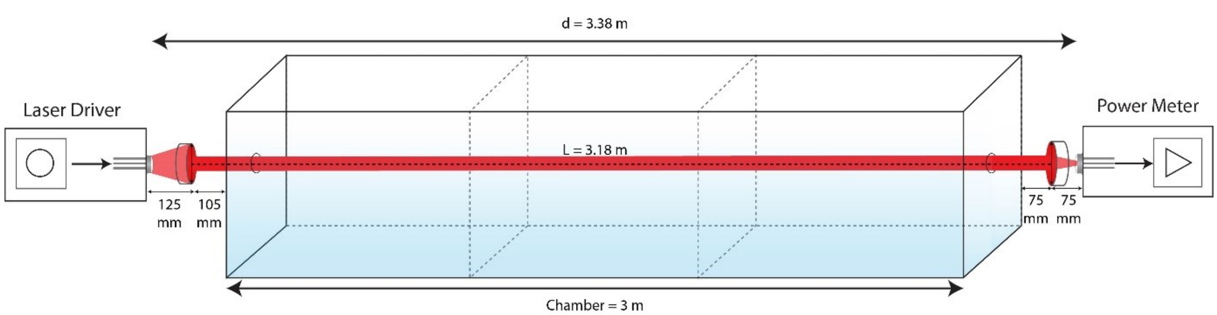

Figure 6: Transmitted laser beam in atmospheric chamber.

Our measurement campaign on Atmospheric Chamber focused on measurement of different weather conditions such as rain, fog and heat we also added measurements of geometric loss.

3.1 Experimental Set-up

At the transmitter side, there is a 850 nm laser beam using the laser driver set LTC100-B and laser diode VCSEL-850. The divergence angle of laser diode itself is about 0.5 rad which is large for transmission purposes. A collimating lens (C230TMD-B) is used after LD to create a narrow beam of light. The diameter of this lens is equal to 5.5 mm. A plano-convex lens (LA1384-B) is used to parallelize the transmitted beam and to achieve a very small divergence angle. The diameter of this lens is equal to 50.8 mm. This lens is placed after the laser diode in such a way that the laser is located at the focal point of the lens which is equal to 125 mm.

The actual link in this experiment is shown in the following picture which is taken from transmitter side view. At the receiver side, a plano-convex lens (LA1145-B) is used to collect and focus the received beam on to the photodetector. The diameter of this lens is equal to 50.8 mm. A power meter (PM-100) is placed right behind the lens in such a way that the photodetector area is located at the focal point of the lens which is equal to 75 mm.

The atmospheric chamber is 3m long our transmitter and receiver stays outside of the chamber. Link distance is equal to $L = 3.1$m. For the experimental measurements perfect alignment is very important for accurate results.

The schematics of the custom laser link is given in figure below which consists of a laser diode working at 850 nm wavelength and a collimating lens for creating a narrower beam light. At the other side a Plano-convex lens will focus the beam on to the photodetector area of power meter. In this experiment we have a clear line of sight to minimize the atmospheric loss which is mostly significant for foggy weather or when the water-vapor density is considerably high. It can be ignored in this experiment where we have a clear weather conditions.

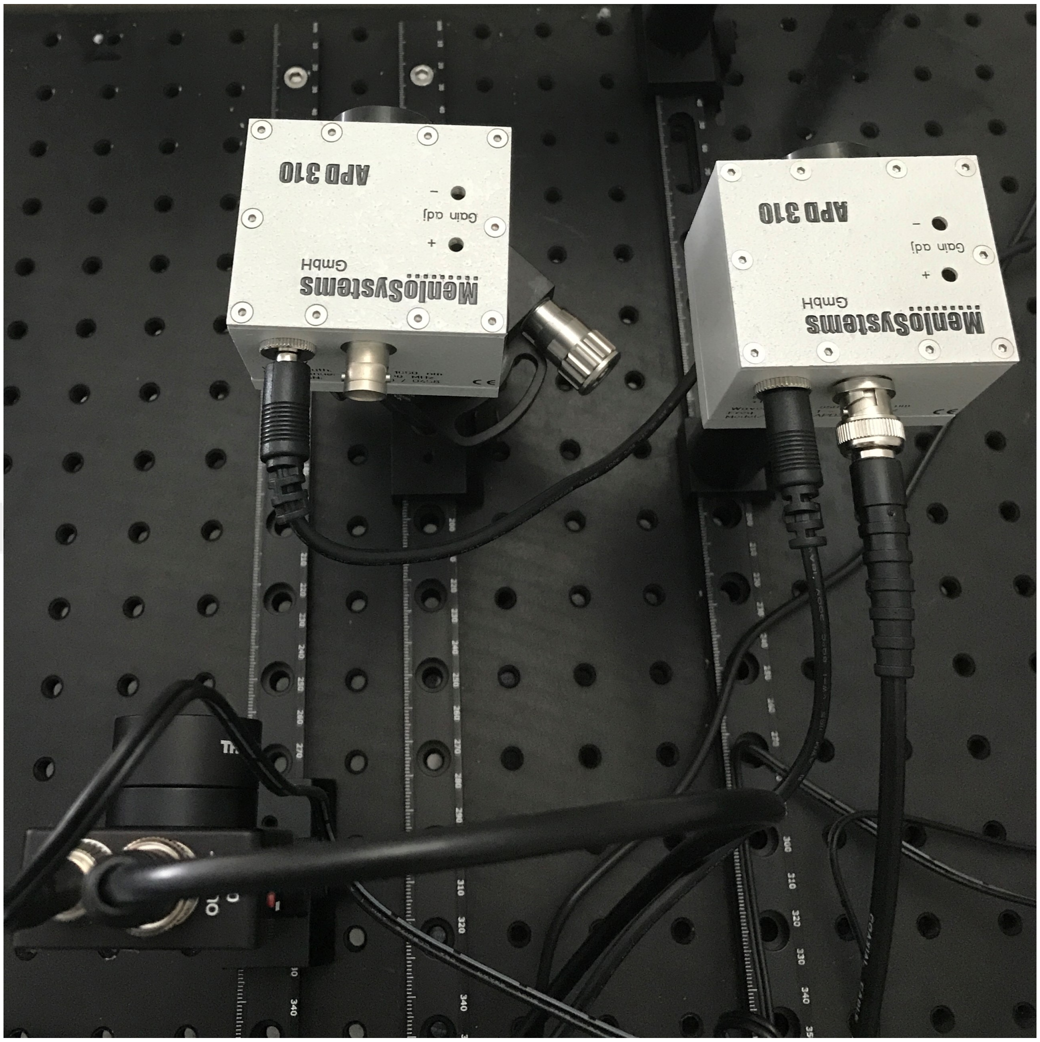

Figure 7: Three different photodiodes two of them covers 1550nm and 1310nm one of them is for 850nm

3.2 FSO Link Budget

In a point-to-point FSO link, the received power P_{rx} can be expressed as

$$P_{rx} = P_{tx} - h_g - h_a[dB] \tag{11}$$

where P_{tx} is the transmit power, h_g is the geometrical loss and h_a is the atmospheric loss.

In our experimental set-up, we use a VCSEL infrared laser with maximum optical power of 10 mW. At the transmitter side, a collimating lens of $D_{tx} = 5.5$ mm and focal length of 4.51 mm is used. At the receiver side, a plano-convex lens of $D_{rx} = 30$

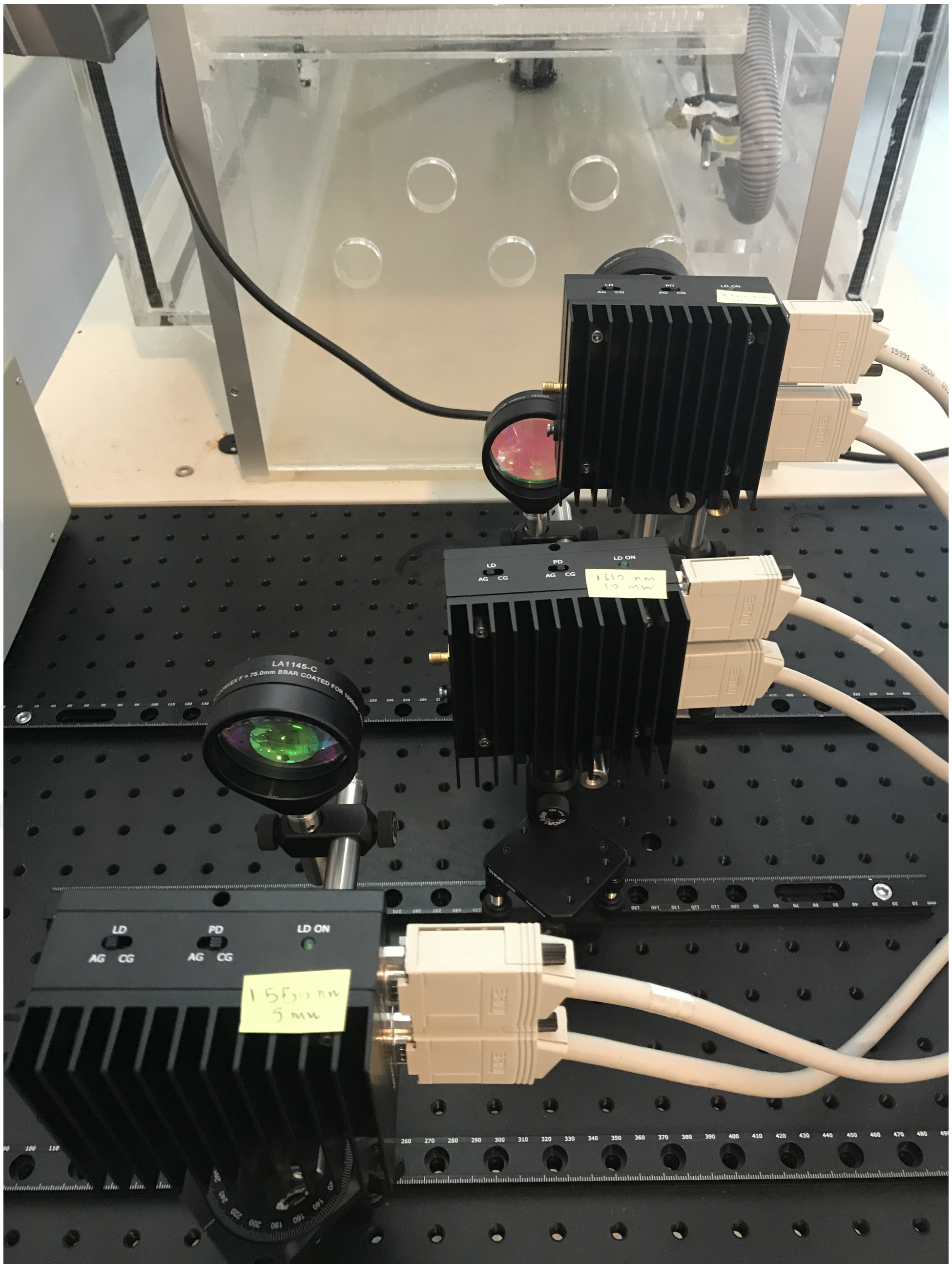

Figure 8: Three different laser diodes with different wavelengths 1550nm, 1310nm, 850nm.

mm and focal length of 75 mm is used to collect the power. We set beam divergence angles θ_{tx} =14, 17 and 20 mrad. The link distance (i.e., length of chamber) is $L = 3$ m. The transmit power of laser (P_{tx}) is changed from 1 to 10 mW. The average received power (P_{rx}) is measured and plotted in Fig. 2. Noting that atmospheric

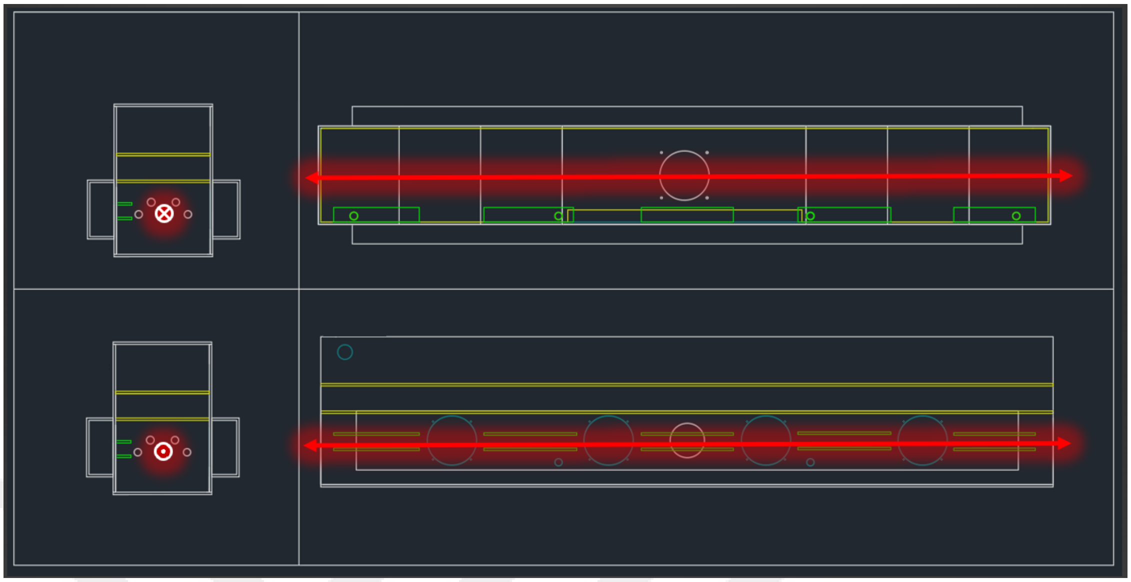

Figure 9: Traveled path of laser beam inside of Atmospheric Chamber.

Figure 10: Current and temperature controller for Laser diode.

Figure 11: The aspheric lens at front of the laser diode for collimating laser beam.

loss is zero (i.e., $h_a = 0$) under clear weather conditions, the analytical results based on (11) and (**??**) are also included in Fig. 2. It can be observed that experimental and analytical results match each other well with a small difference about 0.34 dB. This difference comes from optical loss of the system and power detector's efficiency

Figure 12: The Plano-Convex lens at front of the photo diode for focusing light to active area of photodiode.

Figure 13: Power meter that used for measuring received signal power.

Figure 14: Demonstration of measurement equipment.

Parameters		Values
Transmitter	Wavelength	850 nm
	Maximum optical power	10 mW
	Beam diameter at aperture	30.8 mm
	Minimum beam divergence	1 mrad
	Collimating lens	$f_1 = 125$ mm
Channel	Dimension	$3 \times 0.4 \times 0.4$ m^3
	Link distance	3.18 m
Receiver	Detector type	Si Photodiode
	PD absorption range	350-1100 nm
	Integrating sphere material (size)	50.8 mm
	PD responsivity R	0.65 mA/W at 850 nm
	Efficiency	90% - 95%
	Plano-convex lens	$f_2 = 75$ mm

Table 8: Parameters of FSO Link

(i.e., 97% efficiency) as well as the fact that our laser diode produces a top-hat beam shape.

3.2.1 Link Margin Analysis

A detailed link margin analysis is an important engineering task for any communication system. For example, for optical systems, the engineer looks at the amount of power that is launched at the transmitter side and then determines all potential losses until the signal arrives at the receiver side. The receiver typically has a specific minimum sensitivity at a given data rate, and the task is to make sure that the launch power minus the loss factor stays above the minimum sensitivity to guarantee reliable operation of the system.

In FSO communication systems, the fact is that the loss of the media (air) between the transmitter and the receiver can vary in time due to the impact of weather. Therefore, it is important for FSO systems to take weather conditions into consideration. The following sections explain the link margin analysis for FSO systems in more detail.

To quantify these factors, we build a link budget for our communications system, totaling the expected gains and losses, and then comparing the received signal power to the level required by the signal-detecting circuitry. Any excess of power is dubbed the link margin; it is a measure of how much leeway, or buffer, we have in the event that undesirable time-varying channel effects (fading) cause a deterioration in the power at the receive side.

To make the link more reliable, we would want to leave a larger link margin than the minimum nominal received power level so that a larger buffer (fade margin) is available. However, allocating signal power to the link margin means taking away power that would otherwise go to increasing the distance at which we could separate the transmitter and receiver. This simple logic shows a trade-off between distance and reliability.

In our system we have a power output from the transmitter of 10.5 mW, which corresponds to 10.2 decibels per milliwatt (dBm) on the frequently used decibel scale. This value will be our starting point for the link budget calculation. To figure out how much of this power arrives at the receiver, we need to look at the losses in the communications channel.

3.2.2 Pointing Loss

In this scenario, we first create a perfect parallel beam of laser light with a very small divergence angle (on the order of 1 mrad). In this way, we ideally transfer whole amount of transmitted energy to the receiver side. The practical configuration of this scenario is shown in the following figure.

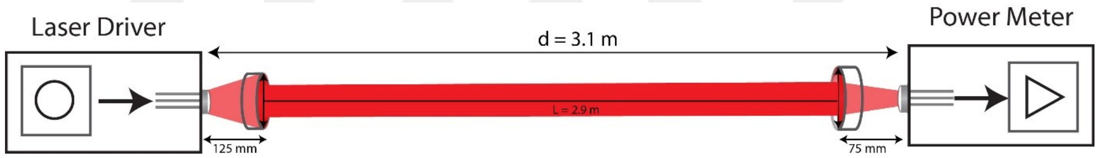

Figure 15: Demonstration of sending collimated laser beam.

If distance between transmitter and receiver is very large and there is no automatic tracking system capability in FSO link some pointing error can appear due to installation error. Another option is if there is a tracking system in FSO link which has residual steady-state errors that can also create some pointing error. If either of these conditions exists, additional loss can be incurred because the transmitter is not pointed accurately enough at the receiver. Typically, these effects are seen for distances in excess of 3 km, at which point we might subtract an additional dB of power from the link budget. In our system, we have a short link distance of 3.1 meter since we are emulating an FSO system in lab under atmospheric chamber emulator. Therefore, we will not consider the pointing loss to be significant, and we can easily say that pointing error is 0dB. In this scenario the received power is measured to be

9.7 mW. Considering the transmit power of 10.5 mW, there is 0.8 mW loss which is negligible.

3.2.3 Optical Loss

The first source of loss in a free-space optical system is due to imperfect lenses and other optical elements (such as couplers). For example, a lens might transmit 96% of the light, but 4% gets reflected or absorbed and is no longer available. To take this into account, we will put a line called “optical loss” in the link budget calculation. The amount of loss depends on the characteristics of the equipment and quality of the lenses. This value needs to be measured or derived from the manufacturer of the optical components. In our system, we use three lenses which all reflect 0.35% of light at 850 nm according to datasheets. Therefore all has a loss of -0.01 dB. The optical loss in our system has been measured as a 0.34 dB reduction in signal power. If we reduce the -0.01 db loss due to lens reflection we have -0.33 dB loss due to efficiency of photodetector of power meter which is between 90% to 95% efficiency.

3.2.4 Geometrical Loss

The term geometrical loss refers to the losses that occur due to the divergence of the optical beam. Typically, a free-space optical system is engineered such that the beam diverges by some amount over the path from the transmitter to the receiver. In some systems that use active tracking, this divergence can be quite small. In systems that do not use active tracking or in which the tracking system is in the several-hertz range, the divergence of the beam is engineered so that when the beam wobbles, some part of it will always hit the receiver, and the link will be maintained. The result of divergence is that much, or even most, of the light is never collected by the receiver.

To measure geometrical loss we need to increase the divergence angle in our setup in such a way that a part of the optical beam is collected by the receiver. In this scenario, we manipulate the setup to increase the divergence angle. The practical

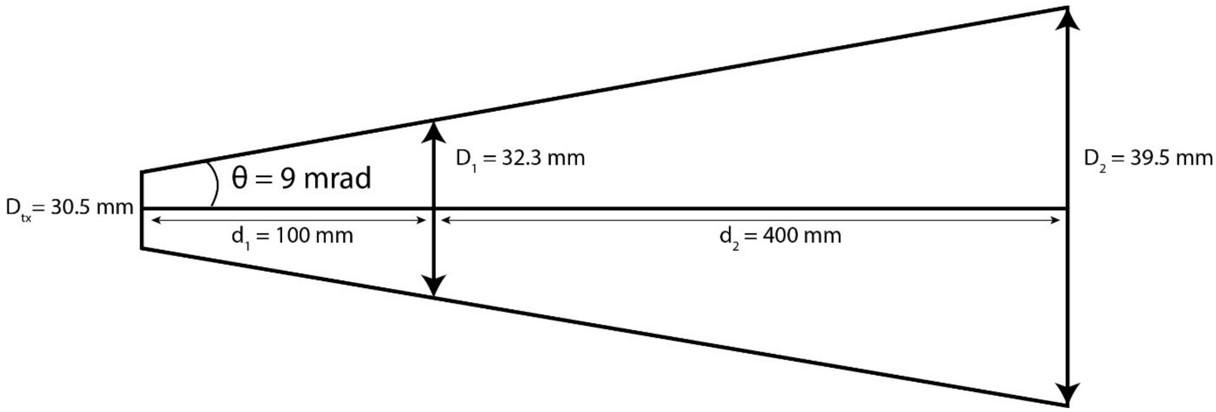

Figure 16: Calculation of beam size after 9mrad divergence.

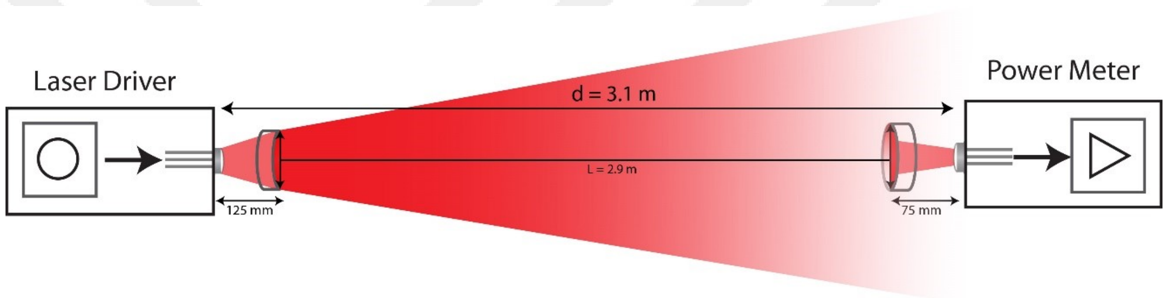

Figure 17: Demonstration of increased divergence angle of laser beam

configuration of this scenario is shown in the figure 17.

Average of the received power for finding the geometric loss as [in dB]:

$$h_g = P_{TXA} - \bar{P}_{RXA} - \alpha_{atm}$$

Usually the atmospheric loss is significant for foggy weather or when the water-vapor density is considerably high, but it can be ignored for clear weather conditions compared to the geometrical loss.

$$h_g = P_{TXA} - \bar{P}_{RXA}$$

We can extract the received power data at the receiver and average the data. Then according to the above formulas Geometric loss can be calculated.

At the transmitter we use a remote optical power meter (PM200) right after the collimating lens and we measure the transmitted power as $P_{TXA} = 10.45$ mW.

At the receiver side, we average the received optical power using PM100 which is connected to a PC for further processes. The averaged received optical power is obtained as $\bar{P}_{RXA} = 3.61$mW.

Therefore the geometrical loss can be obtained as

10.2 [dBm] - 0.34 [dB] + Geometrical Loss [dB] = 5.57 [dBm]

h_g = Geometrical Loss = - 4.29 [dB]

Description	*Value (mW)*	*Value (dB)*
Transmit Power	10.45 mW	10.2 dBm
Pointing Loss	-	0 dB
Optical Loss	-	-0.34 dB
Geometrical Loss	-	-4.29 dB
Received Power	3.61 mW	5.57 dBm

Table 9: Received power after discrimination of Pointing, Geometrical and Optical Losses.

3.3 Characterization of Rain Attenuation

The table of link margin for the system before the rain function starts is given as We

Description	*Value (mW)*	*Value (dB)*
Transmit Power	10.5 mW	10.2 dBm
Pointing Loss	-	0 dB
Optical Loss	-	-0.34 dB
Received Power	9.7 mW	9.86 dBm

Table 10: The table of link margin for the system before the rain function starts

have tested rain generation system by two different lasers sources which has 1550nm and 1310nm wavelengths and same amount of output power. Following figures shows effect of rain on those lasers.It can be easily seen that avarage power of rain is lower then clear weather. Our aim is matching theoretical formulas and measurements.

In the next figure, we calculate the attenuation caused by rainy weather. In this figure we have constant power for clear atmosphere. It can be observed that there is

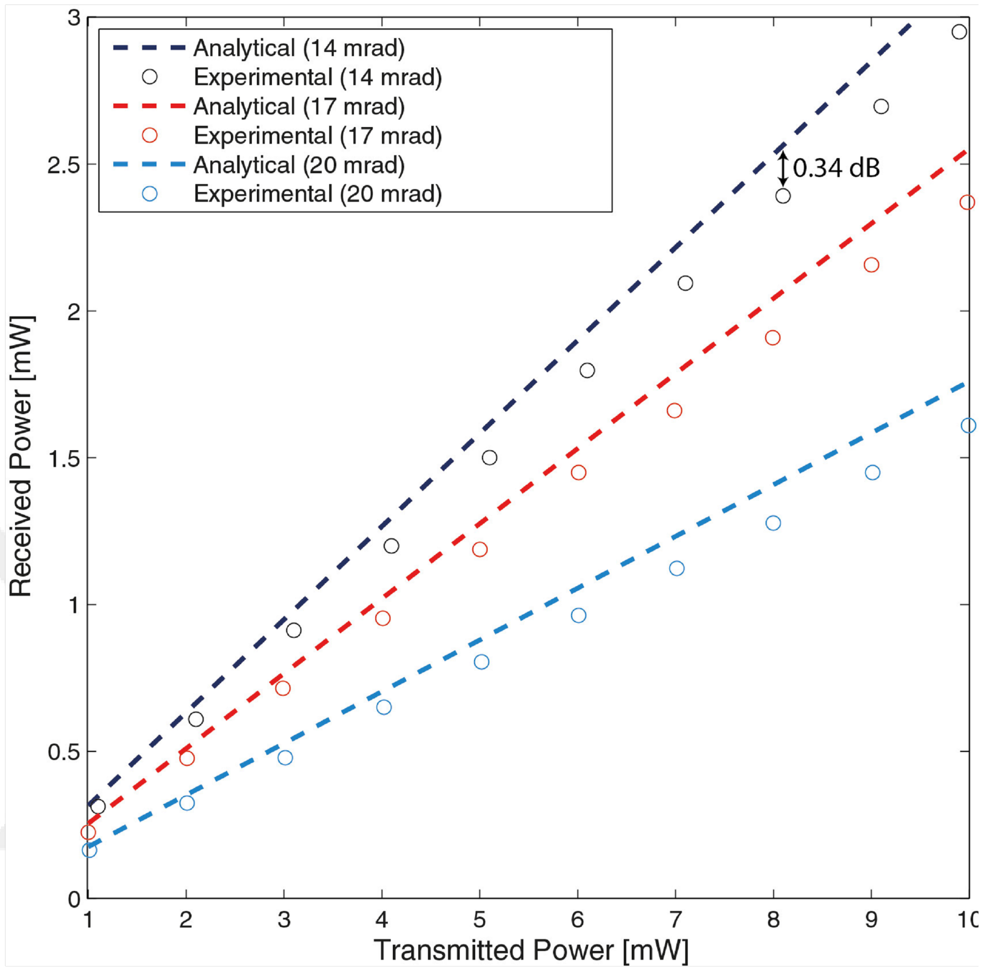

Figure 18: Geometrical loss for different divergence angles.

constant amount of loss in average of power due to rain attenuation. And also because of randomness of rainfalls the power has fluctuations around this average power.

The average of power during rainy weather is measured as $\bar{P}_{RX}^{rain} = 9.48$ mW. Hence we can easily find h_{rain} as;

$$h_{rain} = \frac{\bar{P}_{RX}^{rain}}{\bar{P}_{RX}^{clear}} = \frac{9.48}{9.7} = 0.9783 = -0.095 \text{ dB}$$

Received signal is almost not attenuated by rain.

This value can be checked from theoretic formula, i.e., $h_{rain} = e^{-\sigma_{rain} L}$ and $\sigma_{rain} = 1.076 R_{rain}^{0.67}$. Having $h_{rain} = 0.9783$ and $L = 2.9$m, we can find the rain rate, R_{rain} in

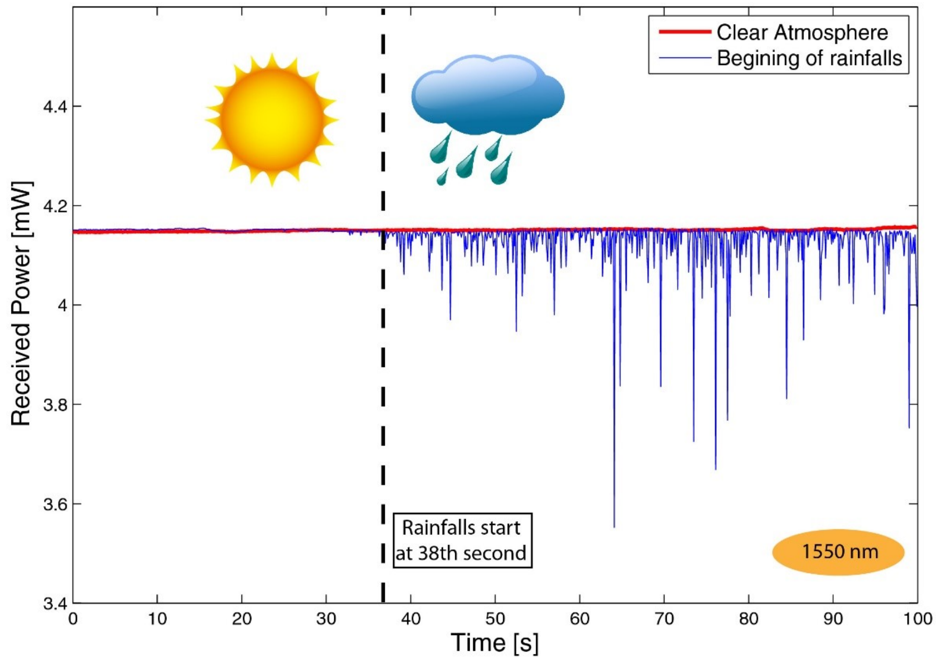

Figure 19: Received power using 1550nm laser beam, before and after rainy weather.

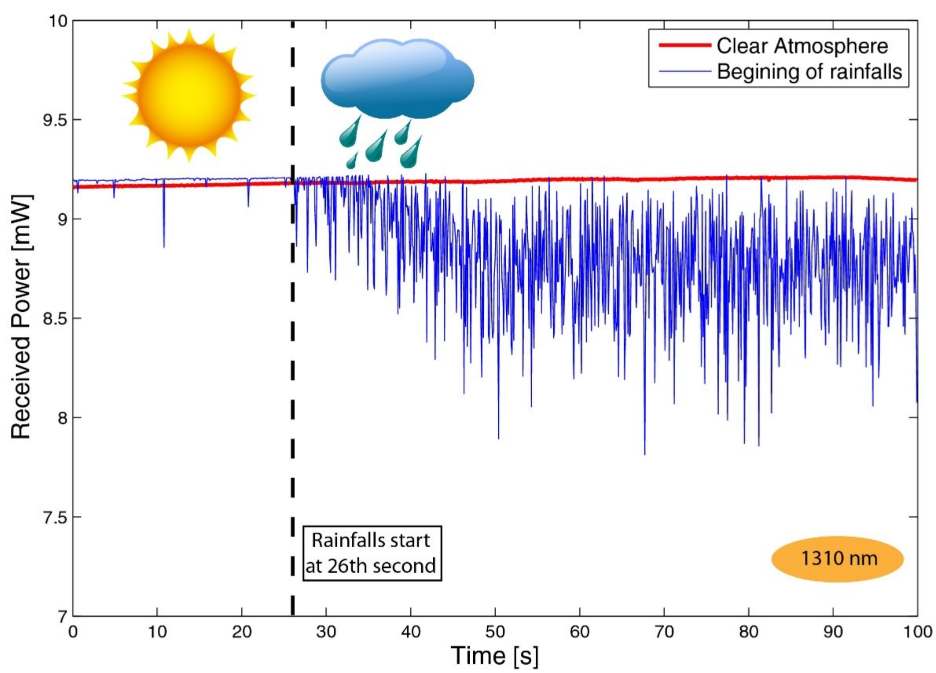

Figure 20: Received power using 1310nm laser beam, before and after rainy weather.

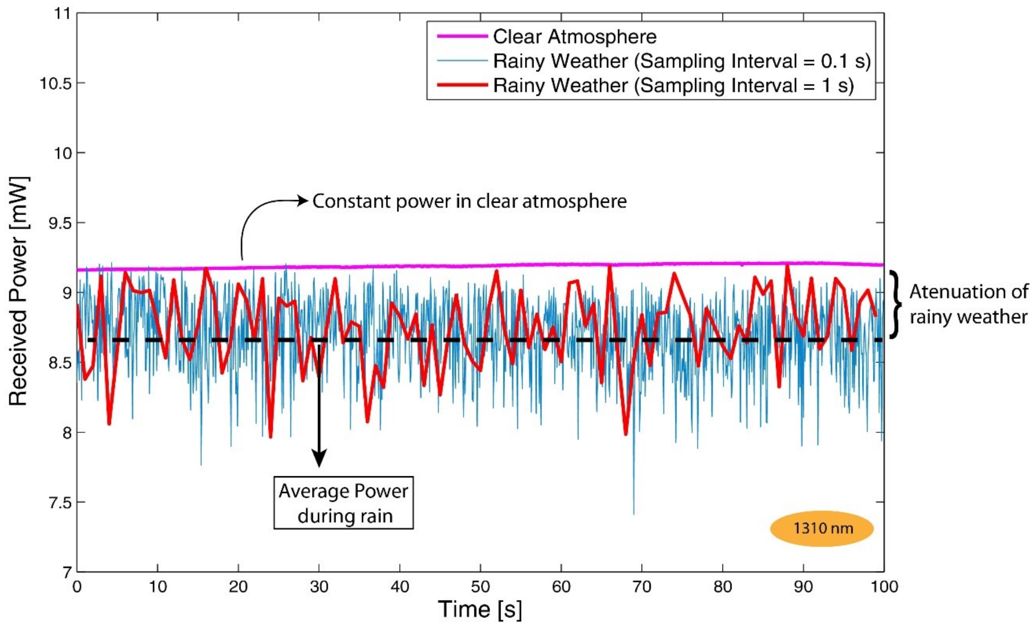

Figure 21: Attenuation of rainy weather using 1310nm laser beam.

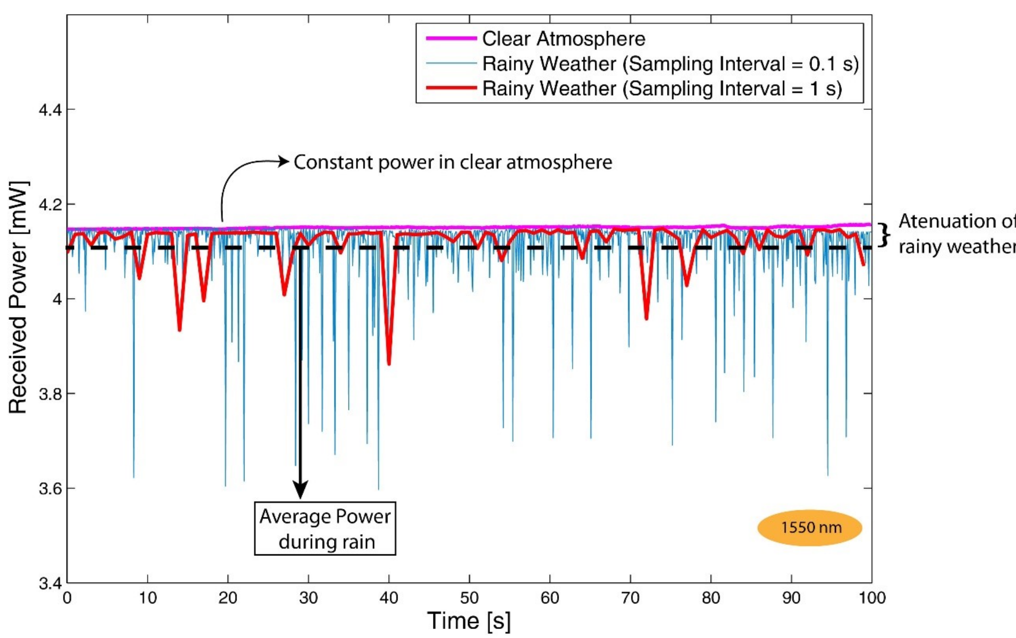

Figure 22: Attenuation of rainy weather using 1550nm laser beam.

mm/h, as

$$R_{rain} = 18.37mm/h$$

Received signal effected by medium rain.

In the following we repeat the experiment for 1310 nm and 1550 nm laser diode wavelengths. For 1310 nm wavelength laser diode we transmit a beam with power of 10 mW and we have $\bar{P}_{RX}^{rain} = 8.69$ mW. Hence we can easily find h_{rain} as;

$$h_{rain} = \frac{\bar{P}_{RX}^{rain}}{\bar{P}_{RX}^{clear}} = \frac{8.69}{9.18} = 0.9467 = -0.2377 \text{ dB}$$

Received signal is almost not attenuated by rain.

This value can be checked from theoretic formula, i.e., $h_{rain} = e^{-\sigma_{rain} L}$ and $\sigma_{rain} = 1.076 R_{rain}^{0.67}$. Having $h_{rain} = 0.9467$ and $L = 2.9$m, we can find the rain rate, R_{rain} in mm/h, as $R_{rain} = 71.98$ mm/h (heavy rain). For 1550 nm wavelength laser diode we transmit a beam with power of 5 mW and we have $\bar{P}_{RX}^{rain} = 4.11$ mW. Hence we can easily find h_{rain} as

$$h_{rain} = \frac{\bar{P}_{RX}^{rain}}{\bar{P}_{RX}^{clear}} = \frac{4.11}{4.15} = 0.99 = -0.042 \text{ dB}$$

Received signal is almost not attenuated by rain.

This value can be checked from theoretic formula, i.e., $h_{rain} = e^{-\sigma_{rain} L}$ and $\sigma_{rain} = 1.076 R_{rain}^{0.67}$. Having $h_{rain} = 0.99$ and $L = 2.9$m, we can find the rain rate, R_{rain} in mm/h, as

$$R_{rain} = 5.73mm/h$$

Received signal effected by light rain.

3.4 Characterization of Fog Attenuation

The ACE is designed to emulate different weather conditions, where here we use fog generation functionality to create different amount of fog in the chamber to cover all the fog regions including dense, moderate, light fog and haze. It is important to note that the setup should be under the same foggy weather condition for all of the wavelengths under the test to have an accurate measurement for observing the effect of different wavelengths. In order to do so, during the experiment, we launch three different wavelengths of 850, 1310, and 1550 nm at the same time to emulate the same foggy condition for all. At the same time a green laser is also launched to measure the visibility. Here, the challenge is to synchronize the received power data samples at each time instance. Having separate instruments for each wavelength, synchronization methods are used to synchronize all data according to certain points such as clear weather, minimum visibility, start and end of each fog event. On the other hand to mimic the outdoor fog circumstances, the challenge is to manage almost a uniform density fog event inside the chamber. This is necessary for chamber measurements to assess the empirical expressions which is performed outdoor. To handle this issue the strategy is to insert a constant amount of fog into the top compartment of chamber where fog is stored and move it fast into the main compartment of the chamber where links are established. This has been possible by the property of the chamber which allows fine control of the existing fans along the chamber. While all of the amount of fog is inserted uniformly into the chamber the fans are turned off to leave the fog event fade away as slowly as possible. The slowness of the event helps to acquire many samples for each value of visibility which leads to an accurate estimate of exact attenuation. We repeat this event several times and whole period of experiment last 4000 second (1 hour and 6 minutes). In Fig. 7, normalized received power is shown with respect to time for all of the lasers (Green, 850, 1310, 1550 nm) during the fog events. The fog events are generated repeatedly and stronger each time. Different regions of fog including dense, medium, and light fog is indicated in the figure. It can be observed that at the beginning of each fog event, the received powers for all wavelengths drop down very fast. However for higher wavelength we have less power reduction. This is in agreement with theoretical models above as bigger wavelengths are less affected by fog attenuations.

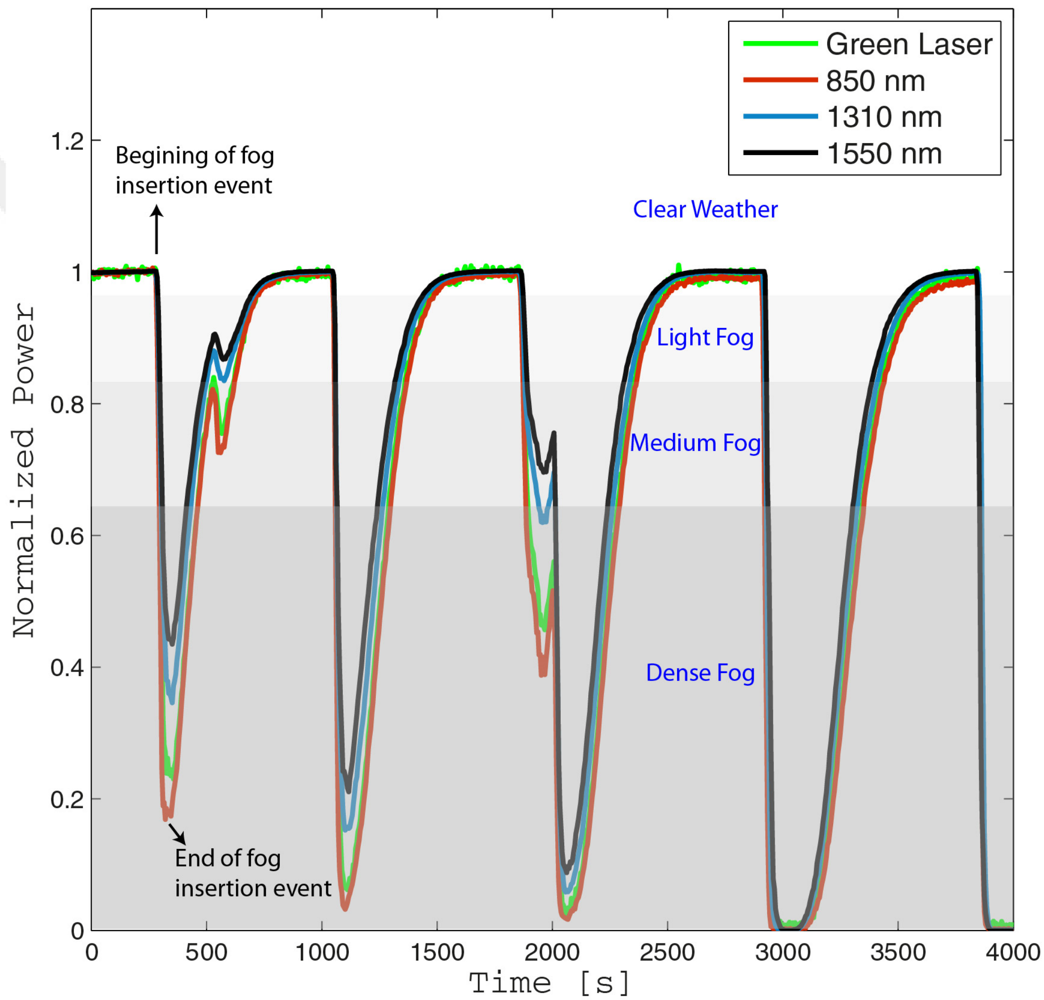

Figure 23: Comparison of different wavelengths received power in time during fog events.

During our measurement campaign using our own atmospheric emulator in an effort to validate the existing models in the literature. We use three laser diodes operating at λ =850, 1310, and 1550 nm. A green laser (λ_0 =550 nm) is simultaneously launched to measure the visibility. We repeat fog injection 5 times during the

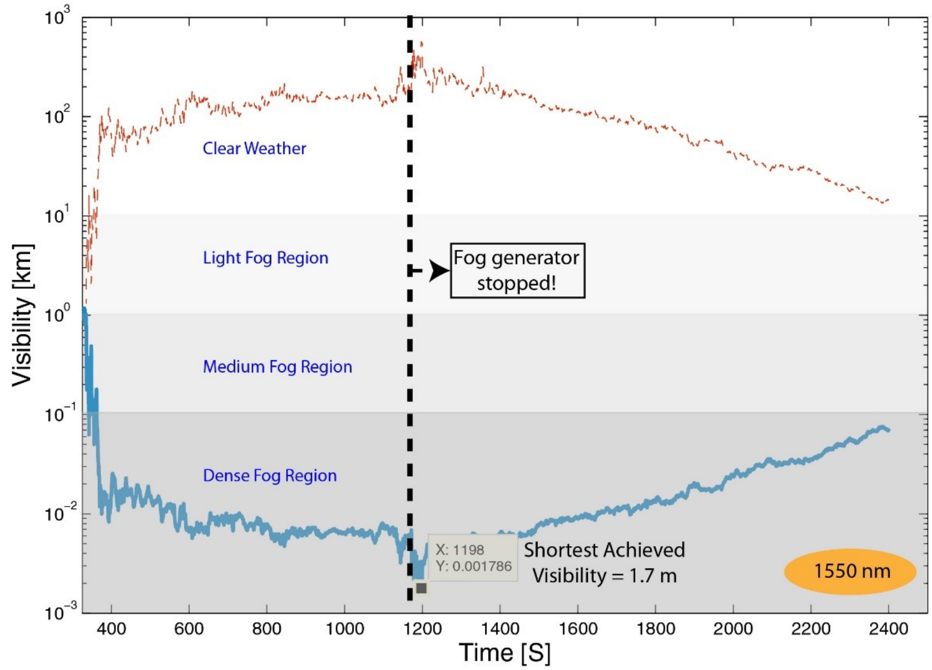

Figure 24: Received power in time during foggy events using 1550nm laser beam.

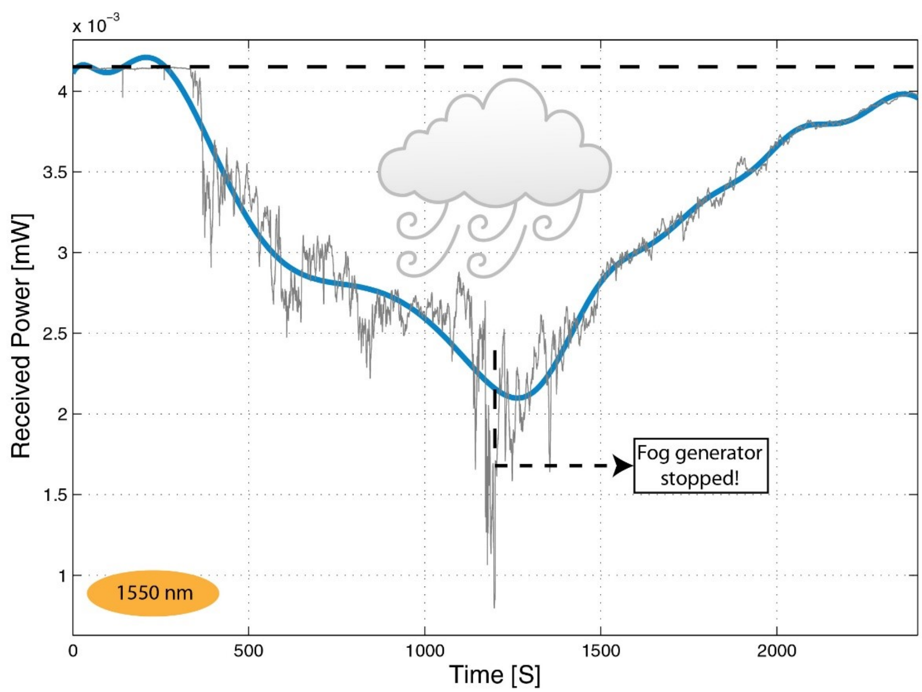

Figure 25: Received power in time during foggy events using 1550nm laser beam.

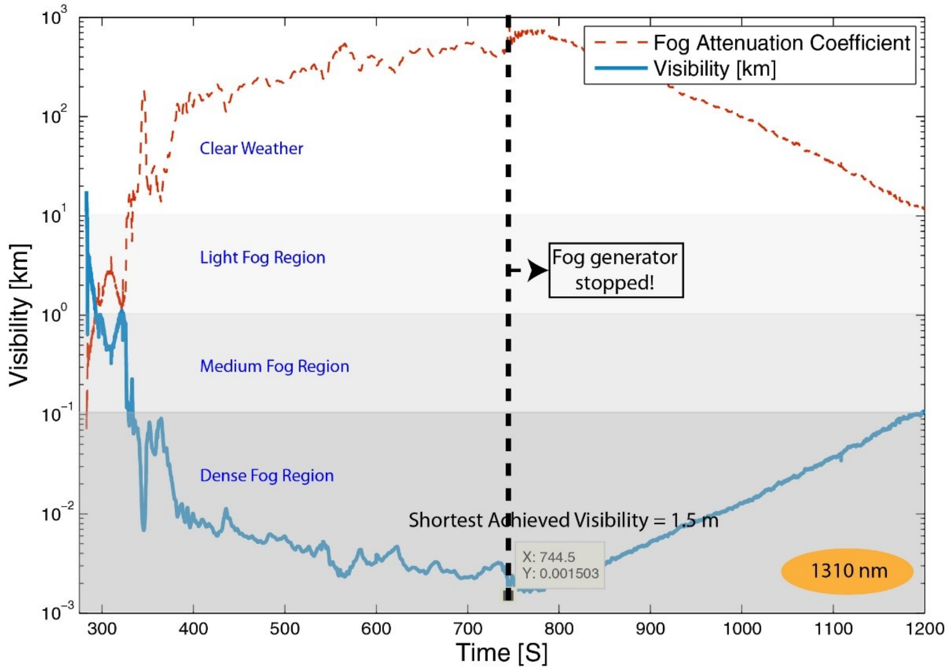

Figure 26: Received power in time during foggy events using 1310nm laser beam.

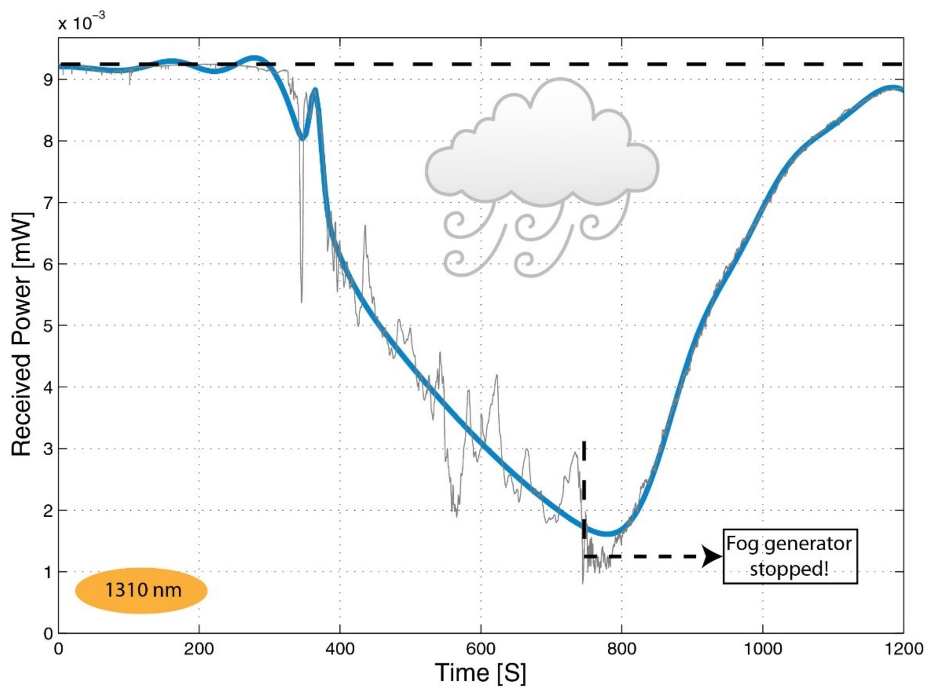

Figure 27: Received power in time during foggy events using 1310nm laser beam.

experiment with a duration of 4000 seconds. In Fig. 3, we plot normalized received power, i.e., $h_a = P_{rx}^{\text{fog}} / P_{rx}^{\text{clear}}$ where P_{rx}^{clear} and P_{rx}^{fog} respectively denote average received power in clear and foggy conditions. Based on the received power associated with green laser, the visibility values are estimated and thereby different regimes of fog [11] can be defined, i.e., dense fog ($V < 0.5$ km), medium fog ($0.5 < V < 1$ km), light fog ($1 < V < 2$ km), and haze ($2 < V < 4$ km). It is observed that at the beginning of each injection, the received powers for all wavelengths drop down very fast. The power reduction is relatively less for higher wavelengths. This could also be seen from (2) where larger wavelengths result in smaller fog attenuation coefficients.

In figures 24-27, the fog attenuation coefficient is shown in dB/km with respect to visibility for 850, 1310, and 1550 nm wavelengths, respectively, on a log-log scale. Here, we consider visibility ranges up to 1 km which includes dense and medium fog regimes. The attenuation coefficients are calculated from $\alpha_{\text{fog}} = -\log (h_a) / L$. Furthermore, we include a fitted curve in the form of $\alpha_{\text{fog}} = aV^{-b}$ where the coefficients a and b are chosen based on our experimental data. The existing models (Kim, Kruse, and Ijaz models) are further plotted as benchmarks.

It is observed from Fig. 28 that our fitted curve is in a good agreement with Ijaz model for λ =850 nm. However, this is not the same for 1310 nm and 1550 nm results (see Figs. 28 and 29) which show better agreement with Kruse model. For all three wavelengths under consideration, Kim's model fails to provide a good match. Furthermore, both Ijaz and Kim models overestimate the fog attenuation at λ =1310 and 1550 nm. In an attempt to compare the three wavelengths under consideration, we plot our fitted curves together in Fig. 30. It is observed that the higher the wavelength, the less is its fog attenuation coefficient and more robust to foggy weather conditions. For example, at a visibility of $V = 0.5$km, the fog attenuation coefficients are calculated as α_{fog} =31.5, 23.5, and 20.3 dB for λ =850, 1310 and 1550 nm respectively.

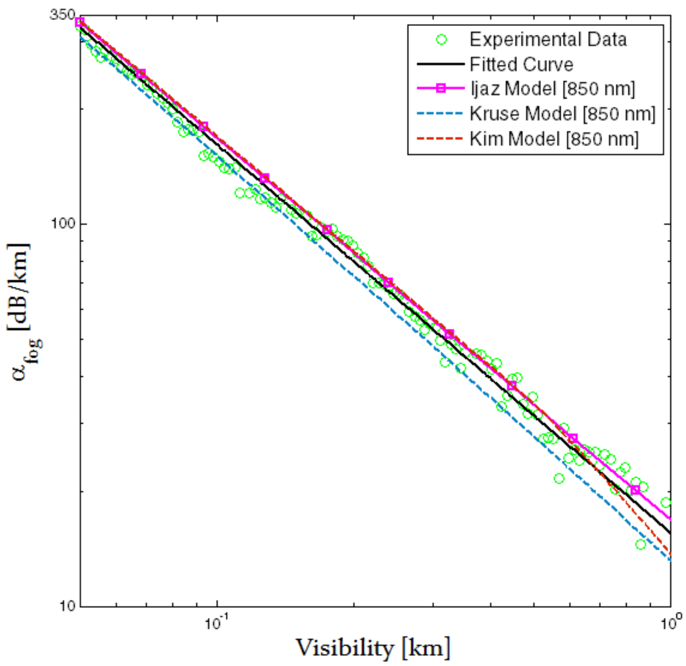

Figure 28: Attenuation coefficient with respect to visibility for 850 nm

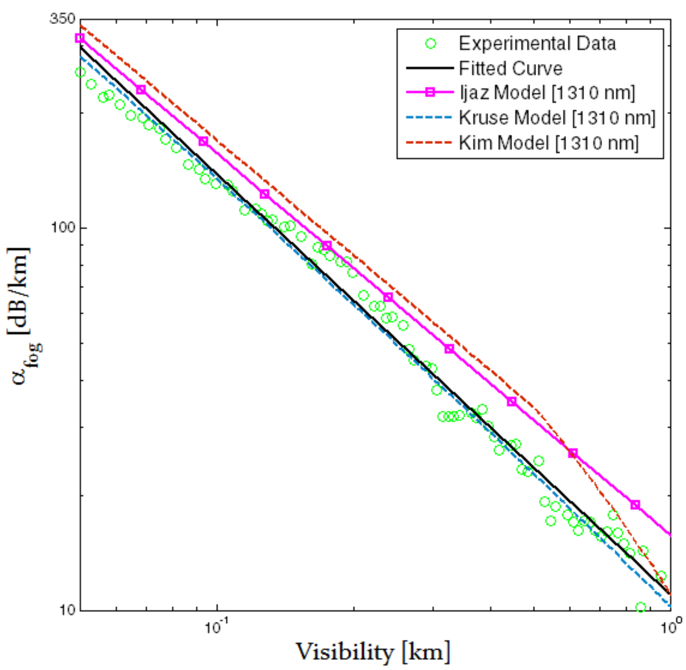

Figure 29: Attenuation coefficient with respect to visibility for 1310 nm

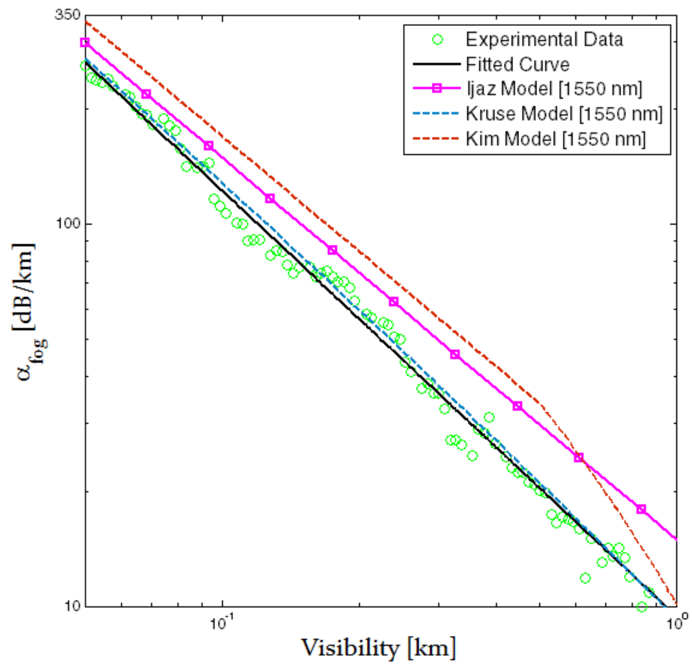

Figure 30: Attenuation coefficient with respect to visibility for 1550 nm

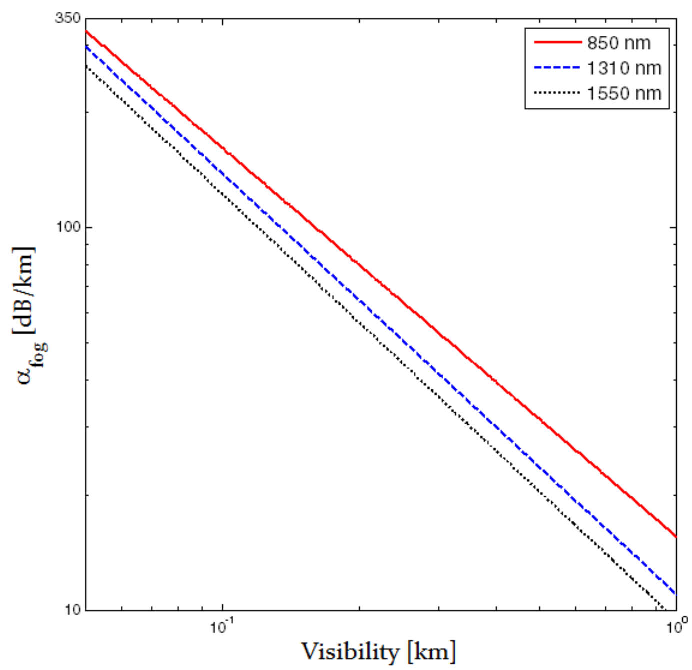

Figure 31: Attenuation coefficient with respect to visibility for comparison of fitted curves for three wavelengths.

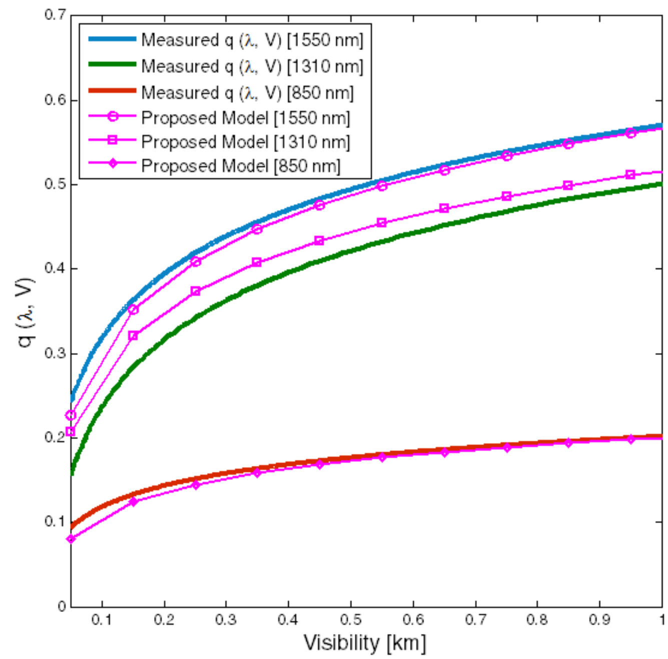

Figure 32: Comparison of the proposed model for q parameter with experimental data

Having observed that none of the existing models is universal in the sense that they are sensitive to deployed wavelength, we propose a new model for the parameter q. First note that based on (2), we have

$$q = \left[\log(3.91) - \log(V) - \log(\alpha_{\text{fog}})\right] / \log(\lambda/\lambda_0) \tag{12}$$

Replacing measured α_{fog} in (2) and using fitting methods in the resulting expression, we obtain

$$q(\lambda, V) = \left(-1.36 \times 10^{-7}\lambda^2 + 4.31 \times 10^{-4}\lambda - 0.228\right)(\ln(V) + 5) \tag{13}$$

where λ is in nm and V is in km. In Fig.5, for the three wavelengths under consideration, we plot the proposed model in (13) that demonstrates a very good match with the experimental data.

3.5 Characterization of Turbulence

Atmospheric turbulence results from random fluctuation of the atmospheric refractive index n along the path of a wave traversing the atmosphere. This refractive index fluctuation is the direct product of random variations in atmospheric temperature along the wave's path. The randomly changing atmospheric temperature is a function of the altitude and wind speed. The scintillation effect due to atmospheric turbulence will be simulated by exploiting the dependence of the channel index of refraction on the temperature variations and wind speed. The temperature and the

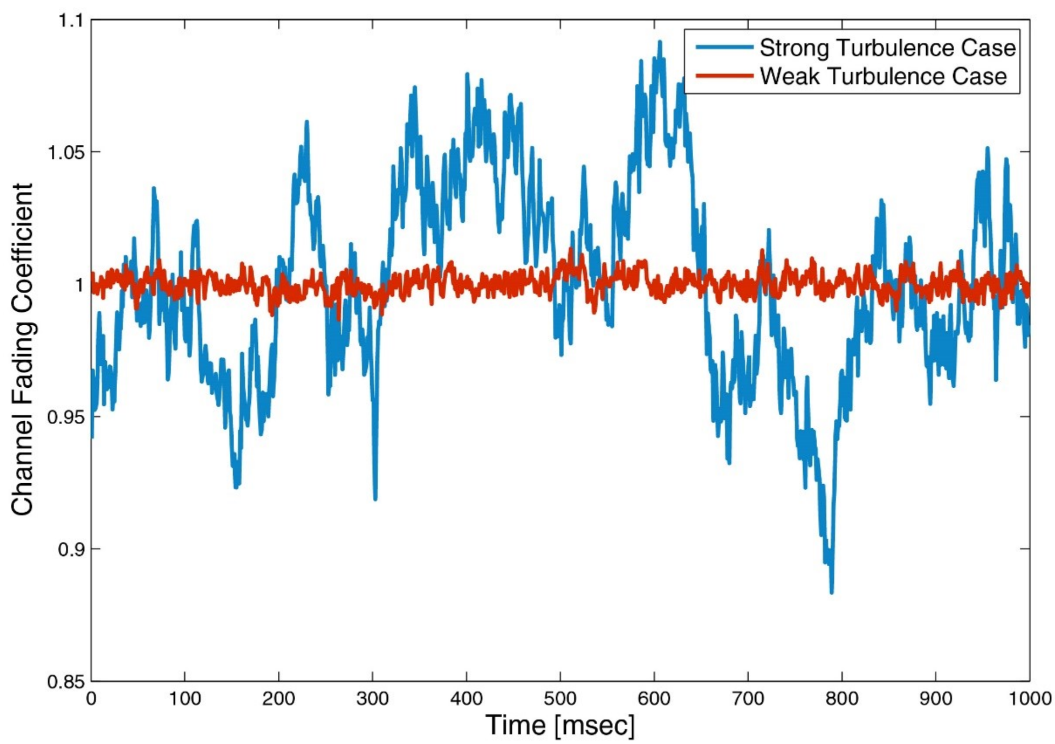

Figure 33: Effect of turbulence on channel fading coefficient .

wind velocity within the chamber can be readily controlled to mimic the outdoor atmospheric conditions as closely as possible. Heaters are used to blow hot air in the direction perpendicular to the optical beam propagating through the chamber to create variation in the temperature and the wind speed.

The hot air temperature is kept in the range of 20 to 65°C. Temperature sensors will be used to measure the instantaneous temperatures and are placed at different distances along the length of the chamber. In this way a temperature gradient ($\Delta T/\Delta L$) between the transmitting and the receiving ends within the chamber is produced and so we have a controlled level of turbulence. The strength of turbulence can also be controlled by placing a heating source near and/or away from the FSO transmitter. The whole experiment is conducted in a dark room to reduce the effect of ambient light to a barest minimum.

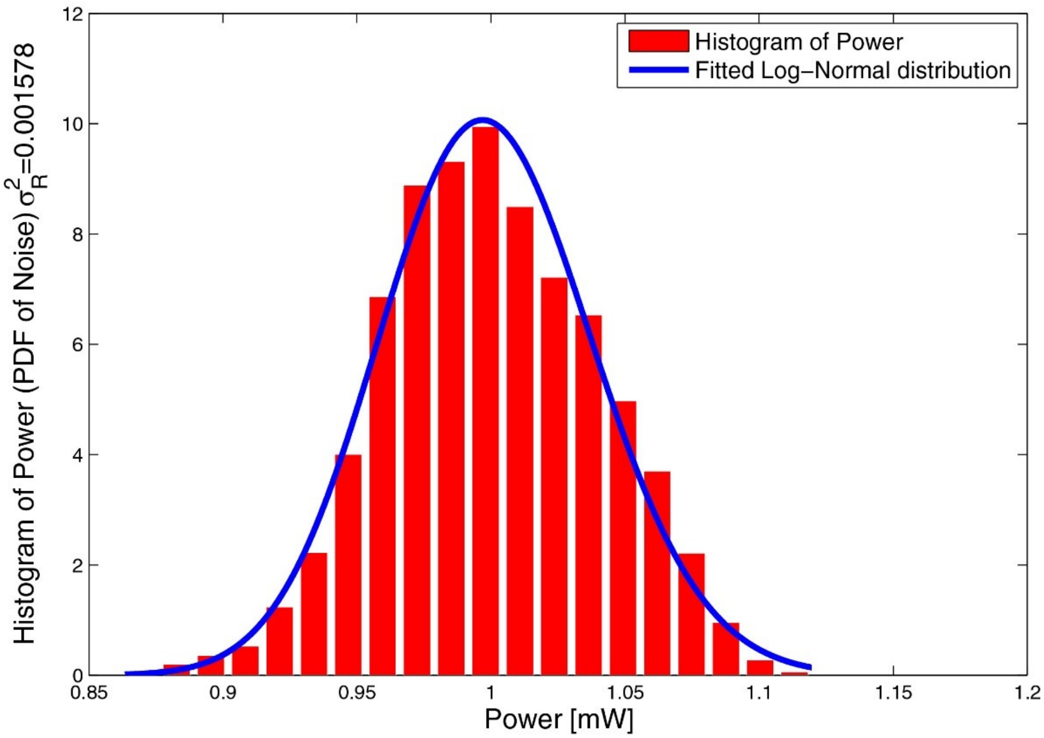

Figure 34: Normalized received power in time during heat events.

For channel measurements we need to capture the received signal samples. Here we have run the experiment for two case as following; First, we measure the received power where we do not apply any turbulence. This case can be categorized as weak turbulence case. In the following case, we measure the received power where we apply turbulence. This case can be categorized as strong turbulence case.

The average of measured temperatures during the experiments at five positions within the chamber and refractive indices are calculated and given in the following table.

Experiments	$\bar{T}_1$ (°C)	$\bar{T}_2$ (°C)	$\bar{T}_3$ (°C)	$\bar{T}_4$ (°C)	$\bar{T}_5$ (°C)	C_n^2 ($\mathrm{m}^{-2/3}$)	σ_R^2
Case 1	24.67	24.20	24.16	24.98	25.66	6.50×10^{-13}	5.85×10^{-5}
Case 2	50.64	45.45	40.31	38.04	36.75	1.77×10^{-11}	0.0016

Table 11: Average of measured temperatures and refractive indices for different cases

In the figure 34 we extract the histogram of channel fading coefficient and fit it with a lognormal distribution. It can be seen that in both cases of weak and strong turbulence it is fitted accurately with log-normal distribution and the value of rytrov variance match with the theoretical values in the table 11.

3.6 Noise of the FSO System

In FSO systems, the dominant noise sources are the background radiation and the thermal noise at the receiver. Background and thermal noise is usually modelled as an AWGN.

$$p\left(n\right) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left[-\frac{\left(n-\mu_w\right)^2}{2\sigma_w^2}\right]$$

where n is the noise sample.

We do this experiment under clear weather condition and let the laser on without any signal modulated to it. We store the samples of received signal power and it can be observed that there is noise on top of the DC signal. We sample the signal for 90 second and record 900 of samples with sampling interval of 0.1 second.

Next, the histogram of the noise samples will be obtained by uniformly dividing the noise samples range into M bins each with equal width.

Next, The Gaussian fit with noise samples can be obtained using the maximum likelihood method by maximizing the log-likelihood function given by

$$L\left(\mu_w, \sigma_w^2\right) = \sum_{i=1}^{N} \ln\left\{\frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left[-\frac{\left(n_i - \mu_w\right)^2}{2\sigma_w^2}\right]\right\}$$

where n_i is the ith received noise sample and $N$is the total number of samples. It can be shown that μ_w and σ_w^2 that maximize above equation are given by

$$\mu_w = \frac{1}{N}\sum_{i=1}^{N} r_i \qquad \text{and} \qquad \sigma_w^2 = \frac{1}{N}\sum_{i=1}^{N}\left(r_i - \mu_w\right)^2$$

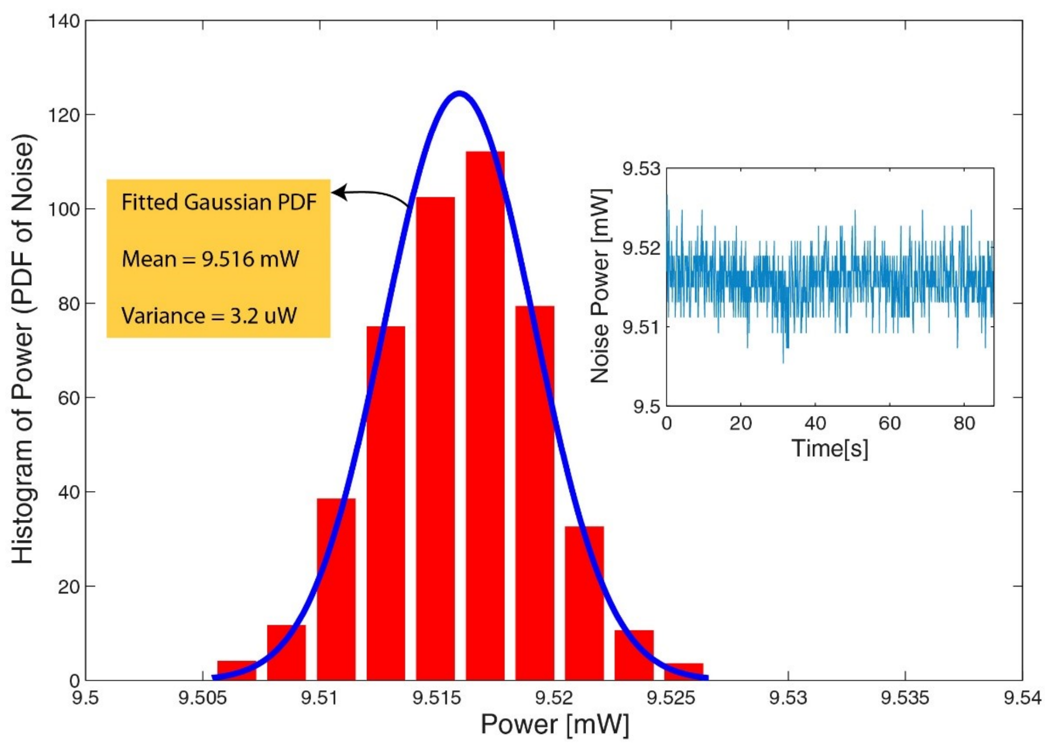

Figure 35: Gaussian fit of received noise samples.

CHAPTER IV

CONCLUSION

In this thesis, we introduced our custom design atmospheric channel emulator for FSO system evaluations and measurement results in foggy,rainy and turbulent weather conditions. The existing fog channel models were compared against the experimental results. It was observed that the Ijaz model [9] is accurate for lower wavelengths; however, for higher wavelengths, Kruse model [11] presents better accuracy. Having observed that the existing models are not universal, a new refined model is proposed that provides a good match for 850, 1310, and 1550 nm wavelengths under consideration.

APPENDIX A

ATMOSPHERIC EMULATOR SOURCE CODES

```
#!/usr/bin/env python
import kivy
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.widget import Widget
from kivy.uix.popup import Popup
from kivy.properties import NumericProperty, ObjectProperty
from kivy.lang import Builder
from kivy.properties import StringProperty
from kivy.uix.label import Label
import sys
import xlsxwriter
from datetime import datetime, timedelta

from kivy.uix.button import Button
from kivy.uix.gridlayout import GridLayout

from nanpy import (ArduinoApi, SerialManager)
from time import sleep
from kivy.clock import Clock
import thread
import time
import psutil
import serial
import os , sys
import subprocess
from functools import partial
```

```
import random

device_1 = '/dev/ttyACM0'
device_2 = '/dev/ttyACM1'
actualMotorSpeed =0
connection_1 = SerialManager(device=device_1)
a1 = ArduinoApi(connection=connection_1)
on1=on2=on3=on4=arka1=arka2=arka3=arka4=1
avarageTemp=0
V12isON=False
rainFunctionFlag=False
windFunctionFlag=False
estimatedTime=0
estimatedFogTime=0
estimatedHeatTime=0
infoCounter=0
xlsxfile = ''
workbook =''
worksheet=''
airQualitySensor='1'
windVariance=0
visibilityRange=0
windVarianceEnabled=False
T1,T2,T3,T4,T5,H6,T6,W10,T7,A8,P8,T8,D9,SWP,LDR=('0',)*15
lockFogFlag=False
fogThread=False
fogDensity=''

a1.pinMode(2, a1.OUTPUT)
a1.pinMode(3, a1.OUTPUT)
a1.pinMode(4, a1.OUTPUT)
a1.pinMode(5, a1.OUTPUT)
```

```
a1.pinMode(6, a1.OUTPUT)
a1.pinMode(7, a1.OUTPUT)
a1.pinMode(8, a1.OUTPUT)
a1.pinMode(9, a1.OUTPUT)
a1.pinMode(10, a1.OUTPUT)
a1.pinMode(11, a1.OUTPUT)
a1.pinMode(12, a1.OUTPUT)

a1.pinMode(14, a1.OUTPUT)
a1.pinMode(15, a1.OUTPUT)
a1.pinMode(16, a1.OUTPUT)
a1.pinMode(17, a1.OUTPUT)
a1.pinMode(18, a1.OUTPUT)
a1.pinMode(19, a1.OUTPUT)
a1.pinMode(20, a1.OUTPUT)
a1.pinMode(21, a1.OUTPUT)

a1.digitalWrite(14, a1.HIGH)#2 fog machine button
a1.digitalWrite(15, a1.HIGH)#6 peltier
a1.digitalWrite(16, a1.HIGH)#1 fog machine
a1.digitalWrite(17, a1.HIGH)#4
a1.digitalWrite(18, a1.HIGH)#7 heater
a1.digitalWrite(19, a1.HIGH)#5 rain
a1.digitalWrite(20, a1.HIGH) #8 #12V
a1.digitalWrite(21, a1.HIGH) #3

reload(sys)
sys.setdefaultencoding('utf8')

def createxls():
        global workbook
        global worksheet
```

```
        global xlsfile
        os.system('sudo rm /home/pi/Documents/atm/Atm_Main/XLSX/*.*')
        xlsxfile=datetime.now().isoformat() + '.xlsx'
        workbook = xlsxwriter.Workbook('/home/pi/Documents/atm/Atm_Main/
            XLSX/'+xlsxfile)
        worksheet = workbook.add_worksheet()
#Extend Columns to fit data
        worksheet.set_column('A:A', 10)
        worksheet.set_column('B:B', 7)

def closexls(self):
        self.close()
        os.system('/home/pi/Documents/atm/Atm_Main/Dropbox-Uploader/
            dropbox_uploader.sh upload ' +'/home/pi/Documents/atm/
            Atm_Main/XLSX/'+ xlsxfile + ' /AtmosphericChamber')

def pngController():
        global nameofPNG
        os.system('/home/pi/Documents/atm/Atm_Main/Dropbox-Uploader/
            dropbox_uploader.sh upload ' +nameofPNG+ ' /
            AtmosphericChamber/PNG')

def xlsxController():
                global T1,T2,T3,T4,T5,H6,T6,W10,T7,A8,P8,T8,D9
                global actualMotorSpeed
                global airQualitySensor
                global windVariance
                global windVarianceEnabled,V12isON,lockFogFlag
                global on1,on2,on3,on4,arka1,arka2,arka3,arka4
                global workbook
                global worksheet
                global xlsfile
```

```
createxls()
num=1
worksheet.write('A'+str(num), 'T1')
worksheet.write('A'+str(num+1), T1)
worksheet.write('B'+str(num), 'T2')
worksheet.write('B'+str(num+1), T2)
worksheet.write('C'+str(num), 'T3')
worksheet.write('C'+str(num+1), T3)
worksheet.write('D'+str(num), 'T4')
worksheet.write('D'+str(num+1), T4)
worksheet.write('E'+str(num), 'T5')
worksheet.write('E'+str(num+1), T5)
worksheet.write('F'+str(num), 'T6')
worksheet.write('F'+str(num+1), T6)
worksheet.write('G'+str(num), 'T7')
worksheet.write('G'+str(num+1), T7)
worksheet.write('H'+str(num), 'T8')
worksheet.write('H'+str(num+1), T8)
worksheet.write('I'+str(num), 'A8')
worksheet.write('I'+str(num+1), A8)
worksheet.write('J'+str(num), 'P8')
worksheet.write('J'+str(num+1), P8)
worksheet.write('K'+str(num), 'H6')
worksheet.write('K'+str(num+1), H6)
worksheet.write('L'+str(num), 'W10')
worksheet.write('L'+str(num+1), W10)
worksheet.write('M'+str(num), 'D9')
worksheet.write('M'+str(num+1), D9)
worksheet.write('N'+str(num), 'actualMotorSpeed')
worksheet.write('N'+str(num+1), actualMotorSpeed)
worksheet.write('O'+str(num), 'on1')
worksheet.write('O'+str(num+1), actualMotorSpeed*on1)
```

```
            worksheet.write('P'+str(num), 'on2')
            worksheet.write('P'+str(num+1), actualMotorSpeed*on2)
            worksheet.write('Q'+str(num), 'on3')
            worksheet.write('Q'+str(num+1), actualMotorSpeed*on3)
            worksheet.write('R'+str(num), 'on4')
            worksheet.write('R'+str(num+1), actualMotorSpeed*on4)
            worksheet.write('S'+str(num), 'arka1')
            worksheet.write('S'+str(num+1), actualMotorSpeed*arka1)
            worksheet.write('T'+str(num), 'arka2')
            worksheet.write('T'+str(num+1), actualMotorSpeed*arka2)
            worksheet.write('U'+str(num), 'arka3')
            worksheet.write('U'+str(num+1), actualMotorSpeed*arka3)
            worksheet.write('V'+str(num), 'arka4')
            worksheet.write('V'+str(num+1), actualMotorSpeed*arka4)
            closexls(workbook)

def xlsxRecorder():
            global T1,T2,T3,T4,T5,H6,T6,W10,T7,A8,P8,T8,D9
            global actualMotorSpeed
            global airQualitySensor
            global windVariance
            global windVarianceEnabled,V12isON,lockFogFlag
            global on1,on2,on3,on4,arka1,arka2,arka3,arka4
            global workbook
            global worksheet
            global xlsfile
            global recordingisActive
            recordSpeed=5 #Record all in X"5" seconds
            createxls()
            num=1
            worksheet.write('A'+str(num), 'T1')
            worksheet.write('A'+str(num+1), T1)
```

```
worksheet.write('B'+str(num), 'T2')
worksheet.write('B'+str(num+1), T2)
worksheet.write('C'+str(num), 'T3')
worksheet.write('C'+str(num+1), T3)
worksheet.write('D'+str(num), 'T4')
worksheet.write('D'+str(num+1), T4)
worksheet.write('E'+str(num), 'T5')
worksheet.write('E'+str(num+1), T5)
worksheet.write('F'+str(num), 'T6')
worksheet.write('F'+str(num+1), T6)
worksheet.write('G'+str(num), 'T7')
worksheet.write('G'+str(num+1), T7)
worksheet.write('H'+str(num), 'T8')
worksheet.write('H'+str(num+1), T8)
worksheet.write('I'+str(num), 'A8')
worksheet.write('I'+str(num+1), A8)
worksheet.write('J'+str(num), 'P8')
worksheet.write('J'+str(num+1), P8)
worksheet.write('K'+str(num), 'H6')
worksheet.write('K'+str(num+1), H6)
worksheet.write('L'+str(num), 'W10')
worksheet.write('L'+str(num+1), W10)
worksheet.write('M'+str(num), 'D9')
worksheet.write('M'+str(num+1), D9)
worksheet.write('N'+str(num), 'actualMotorSpeed')
worksheet.write('N'+str(num+1), actualMotorSpeed)
worksheet.write('O'+str(num), 'on1')
worksheet.write('O'+str(num+1), actualMotorSpeed*on1)
worksheet.write('P'+str(num), 'on2')
worksheet.write('P'+str(num+1), actualMotorSpeed*on2)
worksheet.write('Q'+str(num), 'on3')
worksheet.write('Q'+str(num+1), actualMotorSpeed*on3)
```

```
worksheet.write('R'+str(num), 'on4')
worksheet.write('R'+str(num+1), actualMotorSpeed*on4)
worksheet.write('S'+str(num), 'arka1')
worksheet.write('S'+str(num+1), actualMotorSpeed*arka1)
worksheet.write('T'+str(num), 'arka2')
worksheet.write('T'+str(num+1), actualMotorSpeed*arka2)
worksheet.write('U'+str(num), 'arka3')
worksheet.write('U'+str(num+1), actualMotorSpeed*arka3)
worksheet.write('V'+str(num), 'arka4')
worksheet.write('V'+str(num+1), actualMotorSpeed*arka4)
worksheet.write('W'+str(num), 'time')
worksheet.write('W'+str(num+1), 0)
time.sleep(recordSpeed)
while(recordingisActive):
        num=num+1
        worksheet.write('A'+str(num), T1)
        worksheet.write('B'+str(num), T2)
        worksheet.write('C'+str(num), T3)
        worksheet.write('D'+str(num), T4)
        worksheet.write('E'+str(num), T5)
        worksheet.write('F'+str(num), T6)
        worksheet.write('G'+str(num), T7)
        worksheet.write('H'+str(num), T8)
        worksheet.write('I'+str(num), A8)
        worksheet.write('J'+str(num), P8)
        worksheet.write('K'+str(num), H6)
        worksheet.write('L'+str(num), W10)
        worksheet.write('M'+str(num), D9)
        worksheet.write('N'+str(num), actualMotorSpeed)
        worksheet.write('O'+str(num), actualMotorSpeed*
           on1)
```

```
                    worksheet.write('P'+str(num), actualMotorSpeed*
                        on2)
                    worksheet.write('Q'+str(num), actualMotorSpeed*
                        on3)
                    worksheet.write('R'+str(num), actualMotorSpeed*
                        on4)
                    worksheet.write('S'+str(num), actualMotorSpeed*
                        arka1)
                    worksheet.write('T'+str(num), actualMotorSpeed*
                        arka2)
                    worksheet.write('U'+str(num), actualMotorSpeed*
                        arka3)
                    worksheet.write('V'+str(num), actualMotorSpeed*
                        arka4)
                    worksheet.write('W'+str(num+1), str(num*
                        recordSpeed))
                    time.sleep(recordSpeed)
                closexls(workbook)

def isNotRepresentsFloat(s):
    try:
        float(s)
        return False
    except ValueError:
        return True
    pass

def isNotRepresentsInt(s):
    try:
        int(s)
        return False
    except ValueError:
```

```
        return True
    pass

def lockFog(timeInfo):
        global lockFogFlag
        print 'lock fog initialized'
        lockFogFlag=True
        time.sleep(timeInfo)
        lockFogFlag=False
        pass

def fogController():
        global estimatedTime,estimatedFogTime
        global fogThread
        global fogDensity
        fogThread=True
        fogDensity=1
        fogDensity=fogDensity*0.86
        global visibilityRange
        global lockFogFlag
        global D9
        global T7
        if(isNotRepresentsFloat(T7)):
                print 'error fog follower temp sensor not initialized'
                contentx = GridLayout(cols=1)
                contentx_cancel = Button(text='Cancel', size_hint_y=None
                    , height=40)
                contentx.add_widget(Label(text='Error at T7 sensor.\n ')
                    )
                popup = Popup(title='Error!',
                size_hint=(None, None), size=(256, 256),
```

```
                    content=contentx, disabled=False,
                        auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
else:
        if(lockFogFlag==False):
                T7i=float(T7)
                a1.digitalWrite(16, a1.LOW)#1 fog machine
                if(T7i>30):
                        time.sleep(2)
                        a1.digitalWrite(14, a1.LOW)#2 fog
                            machine button
                        time.sleep(fogDensity)
                        estimatedFogTime=2
                elif(T7i<30):
                        a1.digitalWrite(16, a1.LOW)#1 fog
                            machine
                        for x in xrange(1,360):
                                time.sleep(1)
                                print 'fog machine'
                                print x
                                lockFogFlag=True
                                if(T7i>30):
                                        x=360
                                        estimatedFogTime=x
                                        lockFogflag=False
                                pass
                        a1.digitalWrite(14, a1.LOW)#2 fog
                            machine button
                        time.sleep(fogDensity)
                print 'fog machine is closed'
```

```
                a1.digitalWrite(14, a1.HIGH)#2 fog machine
                   button
                thread.start_new_thread(lockFog,(2,))
            else:
                print 'Fog is locked for security reasons.
                   Please wait for a while.'
                contentx = GridLayout(cols=1)
                contentx_cancel = Button(text='Cancel',
                   size_hint_y=None, height=40)
                contentx.add_widget(Label(text='Fog is locked
                   for security reasons.\n Please wait ~2sec'))
                estimatedFogTime=2
                popup = Popup(title='Warning',
                size_hint=(None, None), size=(256, 256),
                                content=contentx, disabled=False
                                   , auto_dismiss=True)
                contentx_cancel.bind(on_release=popup.dismiss)
                contentx.add_widget(contentx_cancel)
                thread.start_new_thread(popup.open,())
                pass
        print 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
        fogThread=False
        pass




def closeAllRelays():
        a1.digitalWrite(14, a1.HIGH)#2 fog machine button
        a1.digitalWrite(15, a1.HIGH)#6 peltier
        a1.digitalWrite(16, a1.HIGH)#1 fog machine
        a1.digitalWrite(17, a1.HIGH)#5
```

```
        a1.digitalWrite(18, a1.HIGH)#7 heater
        a1.digitalWrite(19, a1.HIGH)#4
        a1.digitalWrite(20, a1.HIGH) #8 #12V
        a1.digitalWrite(21, a1.HIGH) #3
        pass

def writeFirstMotor(value):
        global on1,on2,on3,on4,arka1,arka2,arka3,arka4
        value=value*12
        if (value>230):
                value =230
        elif (value<0):
                value=0
                pass

        a1.analogWrite(7,int(value*on1))        #on 1.
        a1.analogWrite(4,int(value*on2))#on 2.
        a1.analogWrite(8,int(value*on3))        #on 3.
        a1.analogWrite(5,int(value*on4))        #on 4.
        a1.analogWrite(6,int(value*arka1))      #arka 1.
        a1.analogWrite(3,int(value*arka2))      #arka 2.
        a1.analogWrite(11,int(value*arka3))     # arka 3.
        a1.analogWrite(2,int(value*arka4))      #arka 4.
        a1.analogWrite(9,int(value))    #null
        #a1.analogWrite(10,value) #pompa
        a1.analogWrite(12,int(value)) #null
        pass

def windUniformityLevel(value):
        global on1,on2,on3,on4,arka1,arka2,arka3,arka4
        on1=random.random()*value
        on2=random.random()*value
```

```
        on3=random.random()*value
        on4=random.random()*value
        arka1=random.random()*value
        arka2=random.random()*value
        arka3=random.random()*value
        arka4=random.random()*value
        writeFirstMotor(actualMotorSpeed)
        pass

def varianceMotor():
        print "Variance Motor Func. Running"
        global actualMotorSpeed
        global windVariance
        global windVarianceEnabled
        while True:
                if(windVarianceEnabled==True):
                        i=windVariance
                        while i>0:
                                writeFirstMotor(i+actualMotorSpeed)
                                print actualMotorSpeed+i
                                i=i-1
                                time.sleep(0.1)
                        pass
                        while i<windVariance:
                                writeFirstMotor(i+actualMotorSpeed)
                                i=i+1
                                print actualMotorSpeed+i
                                time.sleep(0.1)
                        pass
                        time.sleep(0.1)
                else:
                        time.sleep(1)
```

```
                    print "VARIANCE IDLE"
                    print actualMotorSpeed
                    print windVariance
                    print windVarianceEnabled
                pass
        pass

def heatController(instance1):
        global estimatedTime,estimatedHeatTime
        global T1,T3,T5,T6,T8,avarageTemp
        if(isNotRepresentsFloat(T1) or isNotRepresentsFloat(T3) or
            isNotRepresentsFloat(T5) or isNotRepresentsFloat(T6) or
            isNotRepresentsFloat(T8)):
                print 'error fog follower temp sensor not initialized'
                contentx = GridLayout(cols=1)
                contentx_cancel = Button(text='Cancel', size_hint_y=None
                    , height=40)
                contentx.add_widget(Label(text='Error at T7 sensor.\n ')
                    )
                popup = Popup(title='Error!',
                size_hint=(None, None), size=(256, 256),
                                content=contentx, disabled=False,
                                    auto_dismiss=True)
                contentx_cancel.bind(on_release=popup.dismiss)
                contentx.add_widget(contentx_cancel)
                thread.start_new_thread(popup.open,())
        else:
                T1f=float(T1)
                T3f=float(T3)
                T5f=float(T5)
                T6f=float(T6)
                T8f=float(T8)
```

```
            avarageTemp=(T1f+T3f+T5f+T6f+T8f)/5
            if(instance1<25):
                a1.digitalWrite(15, a1.LOW)#6 peltier
                a1.digitalWrite(18, a1.HIGH)#7 heater
                estimatedHeatTime=(25-instance1)*200
            elif(instance1>25):
                a1.digitalWrite(18, a1.LOW)#7 heater
                a1.digitalWrite(15, a1.HIGH)#6 peltier
                estimatedHeatTime=(instance1-25)*10
            else:
                pass
        pass

class RootWidget(BoxLayout):
        global T1,T2,T3,T4,T5,H6,T6,W10,T7,A8,P8,T8,D9
        global actualMotorSpeed
        global airQualitySensor
        global windVariance
        global windVarianceEnabled,V12isON,lockFogFlag,recordingisActive

        def commandPass(self,shellCommand,timeInfo):
            global commandPassFlag
            time.sleep(timeInfo)
            if(commandPassFlag==True):
                subprocess.Popen(shellCommand, shell = True)
            pass
        def canceling(self,opt,*args):
            global commandPassFlag
            commandPassFlag=False
            opt.dismiss()
            pass
```

```
def startRecording(self,instance1,*args):
        global recordingisActive
        contentx = GridLayout(cols=1)
        contentx_cancel = Button(text='Cancel', size_hint_y=None
            , height=40)
        contentx.add_widget(Label(text='Recording Started! '))
        popup = Popup(title='                    DONE!',
                                size_hint=(None, None), size
                                    =(256, 256),
                                content=contentx, disabled=False
                                    , auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
        recordingisActive=True
        thread.start_new_thread(xlsxRecorder,())
        instance1.disabled=True
        pass

def stopRecording(self,instance1,*args):
        global recordingisActive
        contentx = GridLayout(cols=1)
        instance1.disabled=False
        recordingisActive=False
        contentx_cancel = Button(text='Cancel', size_hint_y=None
            , height=40)
        contentx.add_widget(Label(text='Recording stopped &
            uploaded!'))
        popup = Popup(title='                    DONE!',
                                size_hint=(None, None), size
                                    =(256, 256),
```

```
                              content=contentx, disabled=False
                                , auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())

        pass
    def reboot(self,*args):
        global commandPassFlag
        shellCommand = 'sudo reboot'
        contentx = GridLayout(cols=1)
        contentx_cancel = Button(text='Cancel', size_hint_y=None
            , height=40)
        contentx.add_widget(Label(text='Rebooting in 5 seconds
            !'))
        commandPassFlag = True
        thread.start_new_thread(self.commandPass,(shellCommand
            ,5))
        popup = Popup(title='Rebooting!',
                size_hint=(None, None), size=(256, 256),
                content=contentx, disabled=False, auto_dismiss=
                    True)
        contentx_cancel.bind(on_release=partial(self.canceling,
            popup))
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
        pass

    def shutdown(self,*args):
        global commandPassFlag
        shellCommand = 'sudo halt'
        contentx = GridLayout(cols=1)
```

```
        contentx_cancel = Button(text='Cancel', size_hint_y=None
            , height=40)
        contentx.add_widget(Label(text='Shut down in 5 seconds
            !'))
        commandPassFlag = True
        thread.start_new_thread(self.commandPass,(shellCommand
            ,5))
        popup = Popup(title='Shut down!',
                size_hint=(None, None), size=(256, 256),
                content=contentx, disabled=False, auto_dismiss=
                    True)
        contentx_cancel.bind(on_release=partial(self.canceling,
            popup))
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
        pass

'''This the class representing your root widget.
         By default it is inherited from BoxLayout,
         you can use any other layout/widget depending on your
             usage.
'''

#       mylabelAIRQ = ObjectProperty(None)
#       mylabelHUMI =   ObjectProperty(None)
myswitch = ObjectProperty(None)
mysliderWS = ObjectProperty(None)
mysliderTL = ObjectProperty(None)
mysliderFL = ObjectProperty(None)
mysliderRL = ObjectProperty(None)
mylabelT1 =     ObjectProperty(None)
mylabelT2 =     ObjectProperty(None)
```

```
mylabelT3 =     ObjectProperty(None)
mylabelT4 =     ObjectProperty(None)
mylabelT5 =     ObjectProperty(None)
mylabelT6 =     ObjectProperty(None)
mylabelT6 =     ObjectProperty(None)
mylabelT7 =     ObjectProperty(None)
mylabelA8 =     ObjectProperty(None)
mylabelP8 =     ObjectProperty(None)
mylabelT8 =     ObjectProperty(None)
mylabelD9 =     ObjectProperty(None)
mylabelW10 = ObjectProperty(None)
mylabelCPU = ObjectProperty(None)
mylabelRAM = ObjectProperty(None)
mylabelDISK = ObjectProperty(None)
mylabelTEMP = ObjectProperty(None)
mylabelFOG = ObjectProperty(None)
mylabelVISI = ObjectProperty(None)
mylabelInfo = ObjectProperty(None)
mylabelReady = ObjectProperty(None)
mylabelTimer = ObjectProperty(None)
mybuttonStartR = ObjectProperty(None)
mybuttonStopR = ObjectProperty(None)


def functionx(True,value):
        print value
        pass

def exitProgram(self):
        contentx = GridLayout(cols=1)
        contentx_cancel = Button(text='Exit', size_hint_y=None,
           height=40)
```

```
        contentx.add_widget(Label(text='Closing Program Safely
           !'))
        popup = Popup(title='                       DONE!',
                              size_hint=(None, None), size
                                 =(256, 256),
                              content=contentx, disabled=False
                                 , auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
        thread.interrupt_main()
        pass

    def about(self):
        contentx = GridLayout(cols=1)
        contentx_cancel = Button(text='Done', size_hint_y=None,
           height=40)
        contentx.add_widget(Label(text='This device is created
           by Burak KEBAPCI\n\n\nTo access uploaded data go /
           AtmosphericChamber folder.\n\nDropbox account
           information:\nE-mail: okatemrobot@gmail.com\
           nPassword: okatemrobot2015'))
        popup = Popup(title='                       About!',
                              size_hint=(None, None), size
                                 =(400, 400),
                              content=contentx, disabled=False
                                 , auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
        pass
```

```
def uploadtoDropboxPNG(self):
        global nameofPNG
        nameofPNG='/home/pi/Documents/atm/Atm_Main/PNG/'+
           datetime.now().isoformat()+'.png'
        contentx = GridLayout(cols=1)
        contentx_cancel = Button(text='Cancel', size_hint_y=None
           , height=40)
        contentx.add_widget(Label(text='Screenshot has been
           uploaded!'))
        popup = Popup(title='                    DONE!',
                                size_hint=(None, None), size
                                   =(256, 256),
                                content=contentx, disabled=False
                                   , auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
        contentx.add_widget(contentx_cancel)
        thread.start_new_thread(popup.open,())
        self.export_to_png(nameofPNG)
        thread.start_new_thread(pngController,())
        pass
def uploadtoDropboxXLSX(self):
        contentx = GridLayout(cols=1)
        contentx_cancel = Button(text='Cancel', size_hint_y=None
           , height=40)
        contentx.add_widget(Label(text='Screenshot has been
           uploaded!'))
        popup = Popup(title='                    DONE!',
                                size_hint=(None, None), size
                                   =(256, 256),
                                content=contentx, disabled=False
                                   , auto_dismiss=True)
        contentx_cancel.bind(on_release=popup.dismiss)
```

```
            contentx.add_widget(contentx_cancel)
            thread.start_new_thread(popup.open,())
            thread.start_new_thread(xlsxController,())
            pass

    def setWindSpeed(self,value,*args):
            global actualMotorSpeed
            actualMotorSpeed= value
            writeFirstMotor(value)
#           a2.analogWrite(2,value)
            pass

    def windFunction(self,instance1,instance2,instance3,*args):
            global windFunctionFlag
            actualMotorSpeed = instance1.value
            print "THIS IS ACTUAL :" + str(actualMotorSpeed)
            if(instance1.disabled==True):
                    a1.digitalWrite(20, a1.LOW) #8 #12V
                    instance1.disabled=False
                    instance2.disabled=False
                    instance3.disabled=False
                    windVarianceEnabled=True
                    writeFirstMotor(actualMotorSpeed)
            elif(instance1.disabled==False):
                    instance1.disabled=True
                    instance2.disabled=True
                    instance3.disabled=True
                    writeFirstMotor(0)
                    a1.digitalWrite(20, a1.HIGH) #8 #12V
                    windVarianceEnabled=False
                    #         a2.analogWrite(2,'0')
            pass
```

```
def setHeatLevel(self,instance1,*args):
        #thread.start_new_thread(heatController,(instance1))
        heatController(instance1)
        pass

def heatFunction(self,instance1,*args):
        global estimatedHeatTime
        if(instance1.disabled==True):
                instance1.disabled=False
                heatController(instance1.value)
        elif(instance1.disabled==False):
                instance1.disabled=True
                estimatedHeatTime=0
                a1.digitalWrite(15, a1.HIGH)#6 peltier
                a1.digitalWrite(18, a1.HIGH)#6 heater
        pass

def setWindUniformity(self,value,*args):
        windUniformityLevel(value)
        pass

def setWindVariance(self,instance1,*args):
        global windVariance
        global windVarianceEnabled
        windVariance=instance1
        if(instance1<1):
                windVarianceEnabled=False
        else:
                windVarianceEnabled=True
        pass
```

```
def setRainLevel(self,value,*args):
        pass

def setHumidityLevel(self,instance1,*args):
        pass

def setFoglevel(self,value,*args):
        global fogDensity
        global lockFogFlag
        global fogThread
        if(lockFogFlag==True):
                print 'Fog is locked for security reasons.
                   Please wait for a while.'
                contentx = GridLayout(cols=1)
                contentx_cancel = Button(text='Cancel',
                   size_hint_y=None, height=40)
                contentx.add_widget(Label(text='Fog is locked
                   for security reasons.\n Please wait ~10sec')
                   )
                popup = Popup(title='Warning',
                size_hint=(None, None), size=(256, 256),
                                content=contentx, disabled=False
                                   , auto_dismiss=True)
                contentx_cancel.bind(on_release=popup.dismiss)
                contentx.add_widget(contentx_cancel)
                thread.start_new_thread(popup.open,())
        elif(fogThread==True):
                fogDensity=value
        else:
                fogDensity=value
                try:
```

```
                    thread.start_new_thread(fogController,()
                        )
                except:
                    contentx = GridLayout(cols=1)
                    contentx_cancel = Button(text='Cancel',
                        size_hint_y=None, height=40)
                    contentx.add_widget(Label(text='
                        Something went wrong. :(\n Try again
                        .'))
                    popup = Popup(title=':(',
                    size_hint=(None, None), size=(256, 256),
                                    content=contentx,
                                        disabled=False,
                                        auto_dismiss=True)
                    contentx_cancel.bind(on_release=popup.
                        dismiss)
                    contentx.add_widget(contentx_cancel)
                    thread.start_new_thread(popup.open,())
            pass

    def rainFunction(self,*args):
            global rainFunctionFlag,windFunctionFlag
            if(rainFunctionFlag==False):
#                       instance1.disabled=False
                    a1.digitalWrite(19, a1.LOW) #8 #12V
                    rainFunctionFlag=True
                    print 'enabled rain function
                        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                        '
            elif(rainFunctionFlag==True):
#                       instance1.disabled=True
```

```
                print 'disabled rain function
                    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                    '
                a1.digitalWrite(19, a1.HIGH) #8 #12V
                rainFunctionFlag=False
        pass

def humidityFunction(self,instance1,*args):
        if(instance1.disabled==True):
                instance1.disabled=False
        elif(instance1.disabled==False):
                instance1.disabled=True
        pass
def fogFunction(self,instance1,*args):
        global estimatedFogTime
        if(instance1.disabled==True):
                instance1.disabled=False
                if(isNotRepresentsFloat(T7)):
                        print 'error fog follower temp sensor
                            not initialized'
                        contentx = GridLayout(cols=1)
                        contentx_cancel = Button(text='Cancel',
                            size_hint_y=None, height=40)
                        contentx.add_widget(Label(text='Error at
                             T7 sensor.\n '))
                        popup = Popup(title='Error!',
                        size_hint=(None, None), size=(256, 256),
                                        content=contentx,
                                            disabled=False,
                                            auto_dismiss=True)
                        contentx_cancel.bind(on_release=popup.
                            dismiss)
```

```
            contentx.add_widget(contentx_cancel)
            thread.start_new_thread(popup.open,())
    else:
            thread.start_new_thread(fogController,()
                )
            if(float(T7)<33):
                    contentx = GridLayout(cols=1)
                    contentx_cancel = Button(text='
                        Cancel', size_hint_y=None,
                        height=40)
                    contentx.add_widget(Label(text='
                        Enabling Fog Function takes
                        time.\n Please wait ~3min'))
                    popup = Popup(title='Warning',
                    size_hint=(None, None), size
                        =(256, 256),
                                    content=contentx
                                        , disabled=
                                        False,
                                        auto_dismiss
                                        =True)
                    contentx_cancel.bind(on_release=
                        popup.dismiss)
                    contentx.add_widget(
                        contentx_cancel)
                    thread.start_new_thread(popup.
                        open,())
            elif(float(T7)>33):
                    contentx = GridLayout(cols=1)
                    contentx_cancel = Button(text='
                        Cancel', size_hint_y=None,
                        height=40)
```

```
                                contentx.add_widget(Label(text='
                                   Fog Machine Ready to go!'))
                                popup = Popup(title='Warning',
                                              size_hint=(None,
                                                  None), size
                                                 =(256, 256),
                                              content=contentx
                                                 , disabled=
                                                 False,
                                                 auto_dismiss
                                                 =True)
                                contentx_cancel.bind(on_release=
                                   popup.dismiss)
                                contentx.add_widget(
                                   contentx_cancel)
                                thread.start_new_thread(popup.
                                   open,())
                elif(instance1.disabled==False):
                        instance1.disabled=True
                        estimatedFogTime=0
                        a1.digitalWrite(16, a1.HIGH)
                        a1.digitalWrite(14, a1.HIGH)
                pass

        def updateCpu(self, dt):
                global visibilityRange,avarageTemp,readLabel,infoCounter
                   ,estimatedTime,estimatedHeatTime,estimatedFogTime
                self.mylabelCPU.text='CPU Usage %'+str(psutil.disk_usage
                   ('/').percent)
                self.mylabelRAM.text='RAM Usage %'+str(psutil.
                   virtual_memory().percent)
```

```
self.mylabelDISK.text='DISK Usage %'+str(psutil.
    cpu_percent(interval=None))
res = os.popen('vcgencmd measure_temp').readline()
self.mylabelCPUTemp.text='CPU Temp C '+(res.replace("
    temp=","").replace("'C\n",""))
#ana panel
self.mylabelTEMP.text = 'Now: '+ str(avarageTemp)
self.mylabelWIND.text = 'Now: '+ str(W10)+' m/s'
#self.mylabelHUMI.text = 'Obj: '+str(self.mysliderTL.
    value)+'Current: '+ str(H6)
#tum sensorlerin dokumu
infoCounter==infoCounter+1
if(infoCounter==100):
        infoCounter=0
self.mylabelT1.text='Temperature Sensor 1: '+ str(T1)
self.mylabelT2.text='Temperature Sensor 2: '+ str(T2)
self.mylabelT3.text='Temperature Sensor 3: '+ str(T3)
self.mylabelT4.text='Temperature Sensor 4: '+ str(T4)
self.mylabelT5.text='Temperature Sensor 5: '+ str(T5)
self.mylabelT6.text='Temperature Sensor 6: '+ str(T6)
self.mylabelH6.text='Humidity Sensor 6: '+ str(H6)
self.mylabelHUMI.text='Now: '+ str(H6)

self.mylabelT7.text='Temperature Sensor 7: '+ str(T7)
self.mylabelT8.text='Temperature Sensor 8: '+ str(T8)
self.mylabelA8.text='Attitude 8: '+ str(A8) +'m'
self.mylabelP8.text='Air Pressure 8: '+ str(P8) + 'mm/Hg
    '
self.mylabelD9.text='Dust Particles 9: '+ str(D9) + 'mg/
    m3'
self.mylabelW10.text='Wind Sensor 10: '+ str(W10)+ 'm/s'
```

```
self.mylabelVISI.text='Visibility : '+ str(
   visibilityRange)+ 'm'
estimatedTime=estimatedFogTime+estimatedHeatTime
if(estimatedTime>0):
        estimatedFogTime=estimatedFogTime-0.5
        estimatedHeatTime=estimatedHeatTime-0.5

if(infoCounter==10):
        self.mylabelInfo.text='Dont forget closing power
           '
elif(infoCounter==20):
        self.mylabelInfo.text='Dont forget to check fog
           liquid'
elif(infoCounter==30):
        self.mylabelInfo.text='Dont forget to check
           water level'
elif(infoCounter==40):
        self.mylabelInfo.text='Close top plates before
           using fog'
elif(infoCounter==50):
        self.mylabelInfo.text='System is working
           properly'
if(estimatedTime<60):
        self.mylabelReady.text= 'System is Ready for
           Measurements'
elif(estimatedTime>60):
        self.mylabelReady.text= 'Please wait'
self.mylabelTimer.text=str(estimatedTime) + ' SECOND'

"""
if(int(airQualitySensor)<100):
self.mylabelAIRQ.text = 'Air Quality: Fresh'
```

```
                if(100<int(airQualitySensor)<300):
                self.mylabelAIRQ.text = 'Air Quality: Avarage'
                if(300<int(airQualitySensor)<600):
                self.mylabelAIRQ.text = 'Air Quality: Bad'
                if(600<int(airQualitySensor)):
                self.mylabelAIRQ.text = 'Air Quality: Danger'
                print 'Interrupting All Operations\
                System Will Halt in 10 Seconds\
                All Relays are Closed'
                """
                pass


class MainApp(App):
        global visibilityRange
        '''This is the main class of your app.
                Define any app wide entities here.
                This class can be accessed anywhere inside the kivy app
                    as,
                in python::

                app = App.get_running_app()
                print (app.title)

                in kv language::

                on_release: print(app.title)
                Name of the .kv file that is auto-loaded is derived
                   from the name
                of this class::

                MainApp = main.kv
```

```
            MainClass = mainclass.kv

            The App part is auto removed and the whole name is
                lowercased.
        '''



        def build(self):
                '''Your app will be build from here.
                        Return your root widget here.
                '''

                tb = RootWidget()
                print "value =",tb.mysliderWS.value # <---- value here
                print "min =",tb.mysliderWS.min # <--- min here
                Clock.schedule_interval(tb.updateCpu, 1.0/2.0)

                return tb

        def on_pause(self):
                '''This is necessary to allow your app to be paused on
                   mobile os.
                        refer http://kivy.org/docs/api-kivy.app.html#
                           pause-mode .
                '''
                return True

def decode(RAW):
        global airQualitySensor,visibilityRange
        global T1,T2,T3,T4,T5,H6,T6,W10,T7,A8,P8,T8,D9,SWP,LDR
        T1=''
        value1=RAW.find("T1-")
```

```
value2=RAW.find("-T2-")
if (value1&value2>-1):
        for num in range(value1+3,value2):
                T1=T1+RAW[num]
        pass
value1=RAW.find("-T2-")
value2=RAW.find("-T3-")
print T1
T2=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                T2=T2+RAW[num]
        pass
value1=RAW.find("-T3-")
value2=RAW.find("-T4-")
print T2
T3=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                T3=T3+RAW[num]
        pass
print T3
T4=''
value1=RAW.find("-T4-")
value2=RAW.find("-T5-")
if (value1&value2>-1):
        for num in range(value1+4,value2):
                T4=T4+RAW[num]
        pass
print T4
T5=''
value1=RAW.find("-T5-")
```

```
value2=RAW.find("-H6-")
if (value1&value2>-1):
        for num in range(value1+4,value2):
                T5=T5+RAW[num]
        pass
print T5
value1=RAW.find("-H6-")
value2=RAW.find("-T6-")
H6=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                H6=H6+RAW[num]
        pass
print H6
value1=RAW.find("-T6-")
value2=RAW.find("-W10-")
T6=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                T6=T6+RAW[num]
        pass
print T6
value1=RAW.find("-W10-")
value2=RAW.find("-T7-")
W10=''
if (value1&value2>-1):
        for num in range(value1+5,value2):
                W10=W10+RAW[num]
        pass
print W10
value1=RAW.find("-T7-")
value2=RAW.find("-D9-")
```

```
T7=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                T7=T7+RAW[num]
        pass
print T7
value1=RAW.find("-D9-")
value2=RAW.find("-P8-")
D9=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                D9=D9+RAW[num]
        pass
print D9
value1=RAW.find("-P8-")
value2=RAW.find("-A8-")
P8=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                P8=P8+RAW[num]
        pass
value1=RAW.find("-A8-")
value2=RAW.find("-T8-")
print P8
A8=''
if (value1&value2>-1):
        for num in range(value1+4,value2):
                A8=A8+RAW[num]
        pass
print A8
T8=''
value1=RAW.find("-T8-")
```

```
value2=RAW.find("-LDR-")

if (value1&value2>-1):
        for num in range(value1+4,value2):
                T8=T8+RAW[num]
                pass
        print T8
pass
value1=RAW.find("-LDR-")
value2=RAW.find("-SWP-")
LDR=’’
if (value1&value2>-1):
        for num in range(value1+5,value2):
                LDR=LDR+RAW[num]
                pass
        print LDR
pass
value1=RAW.find("-SWP-")
value2=RAW.find("-END-")
SWP=’’

if (value1&value2>-1):
        for num in range(value1+5,value2):
                SWP=SWP+RAW[num]
                pass
        print SWP
pass
LDRi=int(LDR)
if(LDRi<300):
        LDRi=300
elif(LDRi>499):
        LDRi=499
```

```
        if(LDRi<330):
                visibilityRange=((500-LDRi)*1500)
        else:
                visibilityRange=((500-LDRi)*56)
        print "visibilityRange"
        print visibilityRange



"""
        value=RAW.find("Air Quality: ")
        if (value>-1):
        airQualitySensor=RAW[value+12]+RAW[value+13]+RAW[value+14]
        print airQualitySensor
        pass
        pass
"""

def startUI():
        MainApp().run()
        while True:
                time.sleep(0.1)

def sensorReadFunction():
        try:
                global sensorReadFunctionFlag
                time.sleep(0.1)
                ser = serial.Serial(device_2, 9600)
                time.sleep(0.1)
                sensorReadFunctionFlag=True
                while True:
                        RAW= ser.readline()
```

```
                decode(RAW)
                print "OK! "
            pass
    except:
            print 'Sensor thread error!'
            sensorReadFunction()
    pass

def closeAll():
    writeFirstMotor(0)
    closeAllRelays()
    pass

if __name__ == '__main__':
    try:
            thread.start_new_thread(sensorReadFunction,())
            time.sleep(1)
            thread.start_new_thread(startUI,())
            thread.start_new_thread(varianceMotor,())
            while True:
                    time.sleep(5)
                    print "Running.."
            pass
    except (KeyboardInterrupt, SystemExit):
            closeAll()
            print "Exiting..."
    except:
            closeAll()
            print "Error: Terminating Thread"
    print "DONE!"
```

Bibliography

[1] F. Xu, M. A. Khalighi, and S. Bourennane, "Impact of different noise sources on the performance of pin- and apd-based fso receivers," pp. 211–218, June 2011.

[2] K. A. Mudge, K. K. M. B. D. Silva, B. A. Clare, K. J. Grant, and B. D. Nener, "Scintillation index of the free space optical channel: Phase screen modelling and experimental results," in *Space Optical Systems and Applications (ICSOS), 2011 International Conference on*, pp. 403–409, May 2011.

[3] M. Uysal, "et al., optical wireless communications: an emerging technology, springer," vol. 2016.

[4] N. A. M. Nor, "et al., performance analysis of all-optical amplify-and-forward fso relaying over atmospheric turbulence, ieee student conf," *Research and Development (SCOReD), pp*, pp. 289–293, 2015.

[5] P. Chvojka, "et al., laboratory and outdoor availability and spatial coherence tests of wireless optical links, ieee 2nd int," *Workshop Optical Wireless Communications (IWOW), pp*, pp. 109–112, 2013.

[6] J. Perez, "et al., ethernet fso communications link performance study under a controlled fog environment, ieee commun," *Lett.*, vol. 16, no. 3, pp. 408–410, 2012.

[7] M. A. Esmail, "et al., an experimental study of fso link performance in desert environment, ieee commun," *Lett.*, vol. 20, no. 9, pp. 1888–1891, 2016.

[8] J. Libich, "et al., combined effect of turbulence and aerosol on free space optical links, appl," *Opt*, vol. 56, no. 2, pp. 336–341, 2017.

[9] M. Ijaz, "et al., bit error rate measurement of free space optical communication links under laboratory-controlled fog conditions, ieee 16th european conf," *Networks and Optical Communications (NOC), pp*, pp. 52–55, 2011.

[10] M. Ijaz, "et al., analysis of fog and smoke attenuation in a free space optical communication link under controlled laboratory conditions, ieee int," *Workshop Optical Wireless Communications (IWOW), pp*, pp. 1–3, 2012.

[11] M. Ijaz, "et al., modeling of fog and smoke attenuation in free space optical communications link under controlled laboratory conditions, j. lightw," *Technol.*, vol. 31, no. 11, pp. 1720–1726, 2013.

[12] H. Willebrand and B. S. G. F. space optics:, "enabling optical connectivity in todays networks, sams publishing," vol. 2002.

[13] P. W. Kruse, "Elements of infrared technology: Generation, transmission, and detection, wiley," vol. 1962.

[14] I. Kim, "et.al," in *Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications*, (vol. 4214), pp. 26–37, Proc. SPIE, 2001.

[15] L. C. Andrews and R. L. Phillips, "Laser beam propagation through random media, spie optical engineering press," vol. 1998.

[16] A. A. Farid and S. Hranilovic, "outage capacity optimization for free-space optical links with pointing errors, ieee j," *Lightwave Tech*, vol. 25, July 2007.

[17] S. Bloom, E. Korevaar, J. Schuster, and H. Willebrand, "Understanding the performance of free-space optics [invited] journal of optical networking, vol. 2, no. 6, pp. 178-200, 2003.".

[18] I. Kim, B. McArthur, and E. Korevaar, "comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications," in *Proc. SPIE*, (pp. 2637), vol. 4214, February 2001.

[19] M. Ijaz, Z. Ghassemlooy, J. Pesek, O. Fiser, H. L. Minh, and E. Bentley, "modeling of fog and smoke attenuation in free space optical communications link under controlled laboratory conditions, journal of lightwave technology, vol. 31, no. 11," *p*, pp. 1720–1726, 2013.

[20] F. S. Vetelino, C. Young, L. Andrews, and J. Recolons, "aperture averaging effects on the probability density of irradiance fluctuations in moderate-to-strong turbulence applied optics, vol. 46, no. 11," *p*, pp. 2099–2108, 2007.

[21] L. C. Andrews, R. L. Phillips, and C. Y. Hopen, "Laser beam scintillation with applications (spie optical engineering press," vol. 2001.

[22] M. A. Al-Habash, L. C. Andrews, and R. L. Phillips, "mathematical model for the irradiance pdf of a laser beam propagating through turbulent media, opt," *Eng.*, vol. 40, p. 1554, 2001.

[23] F. S. Vetelino, B. Clare, K. Corbett, C. Young, K. Grant, and L. Andrews, "characterizing the propagation path in moderate to strong optical turbulence, appl," *Opt*, vol. 45, p. 3534, 2006.

[24] H. Le-Minh, "et al., experimental study of bit error rate of free space optics communications in laboratory controlled turbulence,ieee globecom workshops, pp.1072-1076," vol. 2010.

[25] S. Rajbhandari, "et al," in *On the study of the FSO link performance under controlled turbulence and fog atmospheric conditions*, pp. 223–226, IEEEProceedings of the 11th International Conference on Telecommunications (ConTEL, 2011.

[26] S. Rajbhandari, J. Perez, H. L. Minh, and . Z. Ghassemlooy, "fast ethernet fso link performance under the fog controlled environment, european conference and exposition on optical communications, pp," *We-*, vol. 10, 2011.

[27] J. Perez, Z. Ghassemlooy, S. Rajbhandari, M. Ijaz, and H. L. Minh, "ethernet fso communications link performance study under a controlled fog environment,ieee communications letters, vol.16, no. 3," *p*, pp. 408–410, 2012.

[28] Z. Ghassemlooy, H. L. Minh, S. Rajbhandari, J. Perez, and M. Ijaz, "performance analysis of ethernet/fast-ethernet free space optical communications in a controlled weak turbulence condition,journal of lightwave technology, vol.30, no. 13," *p*, pp. 2188–2194, 2012.

[29] M. Ijaz, Z. Ghassemlooy, S. Rajbhandari, H. L. Minh, J. Perez, and A. Gholami, "comparison of 830 nm and 1550 nm based free space optical communications link under controlled fog conditions," in *IEEE 8th International Symposium on Communication Systems*, pp. 1–5, Networks Digital Signal Processing (CSNDSP, 2012.

[30] Z. Ghassemlooy and S. Rajbhandari, "measuring the fog attenuation in an indoor free space optical laboratory chamber, measurement 206," *p*, pp. 5–7, 2012.

[31] M. Ijaz, Z. Ghassemlooy, H. L. Minh, S. Rajbhandari, and J. Perez, "analysis of fog and smoke attenuation in a free space optical communication link under controlled laboratory conditions, ieee international workshop on optical wireless communications (iwow)," *p*, pp. 1–3, 2012.

[32] M. Ijaz, Z. Ghassemlooy, J. Perez, V. Brazda, and O. Fiser, "enhancing the atmospheric visibility and fog attenuation using a controlled fso channel,ieee photonics technology letters,vol. 25, no. 13," *p*, pp. 1262–1265, 2013.

[33] Z. Ghassemlooy, J. Perez, and E. Leitgeb, "on the performance of fso communications links under sandstorm conditions, ieee 12th international conference ontelecommunications (contel)," *p*, pp. 53–58, 2013.

[34] A. Gatri, Z. Ghassemlooy, A. Valenzuela, O. Strobel, R. Rejeb, and E. Leitgeb, "experimental study of the performance for bpsk subcarrier intensity modulation free space optics communications in a laboratory controlled turbulence channel," in *IEEE18th European Conference on Network and Optical Communications (NOC) and 8th Conference on Optical Cabling and Infrastructure (OCI*, pp. 287–292, 2013.

[35] A. Gatri, Z. Ghassemlooy, A. Valenzuela, O. Strobel, and R. Rejeb, "experimental study of coherent detection bpsk modulated communication link under controlled fog atmospheric conditions, ieee 9th international symposium oncommunication systems, networks digital signal processing (csndsp)," *p*, pp. 357–361, 2014.

[36] I. E. Lee, Z. Ghassemlooy, W. P. Ng, V. Gourdel, M. A. Khalighi, S. Zvanovec, and M. Uysal, "practical implementation and performance study of a hard-switched hybrid fso/rf link under controlled fog environment, ieee 9th international symposium oncommunication systems, networks digital signal processing (csndsp)," *p*, pp. 368–373, 2014.

[37] M. Ijaz, Z. Ghassemlooy, A. Gholami, and X. Tang, "smoke attenuation in free space optical communication under laboratory controlled conditions, ieee 7th international symposium ontelecommunications (ist)," *p*, pp. 758–762, 2014.

[38] S. Rajbhandari, Z. Ghassemlooy, P. A. Haigh, T. Kanesan, and X. Tang, "experimental error performance of modulation schemes under a controlled laboratory turbulence fso channel,journal of lightwave technology, vol.33, no. 1," *p*, pp. 244–250, 2015.

[39] R. Pernice, A. Ando, M. Cardinale, L. Curcio, S. Stivala, A. Parisi, A. C. Busacca, Z. Ghassemlooy, and J. Perez, "indoor free space optics link under the weak turbulence regime: measurements and model validation,iet communications, vol.9, no. 1," *p*, pp. 62–70, 2015.

[40] N. A. M. Nor, E. Fabiyi, M. M. Abadi, X. Tang, Z. Ghassemlooy, and A. Burton, "investigation of moderate-to-strong turbulence effects on free space opticsa laboratory demonstration, ieee 13th international conference ontelecommunications (contel)," *p*, pp. 1–5, 2015.

[41] Z. Ghassemlooy, H. L. Minh, and M. Ijaz, *Free space optical communications, Optical and Microwave Technologies for Telecommunication Networks, Ch. 10.4*. J. Wiley, UK: pp. 349-377, 2016.

[42] M. Ijaz, R. Kharel, S. Ekpo, Z. Ghassemlooy, H. L. Minh, and X. Tang, "performance evaluation of free space optical communication under the weak turbulence regime, ieee 10th international symposium oncommunication systems, networks and digital signal processing (csndsp)," *p*, pp. 1–5, 2016.

[43] M. M. Abadi, Z. Ghassemlooy, M. R. Bhatnagar, S. Zvanovec, M. A. Khalighi, and A. Maheri, "using differential signalling to mitigate pointing errors effect in fso communication link," in *IEEE International Conference on Communications (ICC*, 2016.

[44] J. Libich, J. Perez, S. Zvanovec, Z. Ghassemlooy, R. Nebuloni, and C. Capsoni, "combined effect of turbulence and aerosol on free space optical links, applied optics," vol. 2016.

[45] N. A. M. Nor, Z. Ghassemlooy, J. Bohata, P. Saxena, M. Komanec, S. Zvanovec, M. Bhatnagar, and M. A. Khalighi, "experimental investigation of all-optical relay-assisted 10 gbps fso link over the atmospheric turbulence channel," *Journal of Lightwave Technology,*, vol. 2016.

[46] P. Chvojka, J. Bohata, J. Libich, S. Zvanovec, and J. Perez, "laboratory and outdoor availability and spatial coherence tests of wireless optical links, ieee 2nd international workshop on optical wireless communications (iwow)," *p*, pp. 109–112, 2013.

[47] N. A. M. Nor, Z. Ghassemlooy, S. Zvanovec, and M. A. Khalighi, "performance analysis of all-optical amplify-and-forward fso relaying over atmospheric turbulence," in *IEEE Student Conference on Research and Development (SCOReD*, pp. 289–293, 2015.

[48] H. K., T. C. Al-Musawi, J. Bohata, P. Pesek, W. P. Ng, Z. Ghassemlooy, E. Udvary, T. Berceli, and S. Zvanovec, "experimental optimization of the hybrid rommf-fso system using mode filtering techniques, ieee international conference oncommunications workshops (icc)," *p*, pp. 405–410, 2016.

[49] M. M. Abadi, Z. Ghassemlooy, M. A. Khalighi, S. Zvanovec, and . M. R. Bhatnagar, "detection using differential signaling in outdoor correlated-channels condition,ieee photonics technology letters, vol.28, no. 1," *p*, pp. 55–58, 2016.

[50] M. A. Esmail, H. Fathallah, and M. S. Alouini, "an experimental study of fso link performance in desert environment,ieee communications letters, vol.20, no. 9," *p*, pp. 1888–1891, 2016.

[51] M. Ijaz, Z. Ghassemlooy, J. Perez, V. Brazda, and O. Fiser, "Enhancing the atmospheric visibility and fog attenuation using a controlled fso channel," *Photonics Technology Letters, IEEE*, vol. 25, no. 13, pp. 1262–1265, 2013.

[52] C. R. W. Group *et al.*, "Sdrf cognitive radio definitions (sdrf-06-r-0011-v1. 0.0)," Technical report, The Wireless Innovation Forum, Retrieved on September 24 2012, from URL http:// groups. winnforum. org/ d/ do/ 1585, retrieved on September 24, 2012, from http:// groups. winnforum. org/ d/ do/ 1585, 2007.

[53] N. Instruments, "What is pxi?," 2016.

[54] S. Karp, R. M. Gagliardi, S. E. Moran, and L. B. Stotts, *Optical Channels: Fibers, Clouds, Water and the Atmosphere*. Plenum, 1988.

[55] I. Kim, J. Koontz, H. Hakakha, P. Adhikari, R. Stieger, C. Moursund, M. Barclay, A. Stanford, R. Ruigrok, and J. Schuster, "et al," in *Measurement of scintillation and link margin for the Terra Link laser communication system*, (vol. 3232, pp. 100118), in Proc. SPIE, January 1998.

[56] H. Willebrand and B. S. Ghuman, "Free space optics: Enabling optical connectivity in todays networks, sams publishing," vol. 2002.

[57] D. Sadot, A. Melamed, and N. S. Kopeika, "effects of aerosol forward scatter on the long- and short exposure atmospheric coherence diameter, waves in random media, vol. 4, no. 4," *p*, vol. 487, October 1994.

[58] S. Arnon and N. S. Kopeika, "effect of particulates on performance of optical communication in space and an adaptive method to minimize such effects, appl. opt., vol. 33, no. 21," *p*, vol. 4930, July 1994.

[59] X. Zhu and J. M. Kahn, "free-space optical communication through atmospheric turbulence channels, ieee trans. commun., vol. 50, no. 8," *p*, pp. 1293–1300, August 2002.

VITA

Burak Kebapci was born in Siirt, Turkey, on January 7, 1993. Between 2012 and 2015 he studied Electrical & Electronics Engineering Major and Computer Networks Minor at the Bahcesehir University in Istanbul, Turkey. He is accepted for M.Sc degree in Electrical Electronics Engineering from Ozyegin University in 2015. He joined Center Of Excellence in Optical Wireless Communication Technologies (OKATEM) in 2015 as project engineer. In 2016 he became Engineering Team Lead of OKATEM.