DETERMINATION OF THE SUITABILITY OF MT GAMSBERG IN NAMIBIA FOR MILLIMETRE WAVE ASTRONOMY BY MEASUREMENTS OF THE PRECIPITABLE WATER VAPOUR

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN PHYSICS

OF

THE UNIVERSITY OF NAMIBIA

BY

LOTT NDEYANALE FRANS

(201208271)

April 2020

Supervisor: Dr Rhodri Evans

(Department of Physics, University of Namibia)

Co-Supervisor: Dr Michael Backes

(Department of Physics, University of Namibia)

Abstract

Precipitable Water Vapour (PWV) is the amount of water vapour in the atmospheric column above a location equivalent to the amount of liquid precipitation that would result if all the water vapour in the column was condensed. Water vapour is the main source of opacity in the Earth's atmosphere at infrared and millimetre to sub-millimetre wavelengths. The Event Horizon Telescope (EHT) is a large network of millimetre to sub-millimetre telescopes across the globe that is used to image supermassive black holes. The Africa Millimetre Telescope (AMT) is planned to be built on Mt Gamsberg in Namibia and aims to complement the EHT. In this study, PWV at Mt Gamsberg and at the H.E.S.S. site was determined in order to asses the two sites regarding their suitability to conduct millimetre wave astronomy and for the AMT to be built on Mt Gamsberg. PWV at the H.E.S.S. site was indirectly determined from sky temperature data which was taken from 2004 to 2019 by radiometers on four of the H.E.S.S. Cherenkov Telescopes (CT). The PWV at H.E.S.S. as a function of sky temperature as given by the CT radiometers was determined by relationships given by data from two other equipment present at the H.E.S.S. site, namely the NASA AERONET station and an ATMOSCOPE with the equipment recording PWV and sky temperature, respectively. The PWV at Mt Gamsberg was determined by scaling the PWV from the H.E.S.S. site to what it would be at the height of Mt Gamsberg. The relative frequency distribution of PWV, monthly mean PWV along with its seasonal variations were com-

puted for both sites. The presented results indicate Mt Gamsberg is a suitable site for millimetre wave astronomy and to host the AMT. However it was recommended that direct PWV measurements should be taken at Mt Gamsberg to confirm the result of this study, as these were not based on direct in-situ measurements.

All diagrams, graphs and tables presented in this thesis were produced by the author, unless otherwise stated.

Contents

List of Figures

List of Acronyms and abbreviations

Abbreviation	Meaning
PWV	Precipitable Water Vapour
mm	millimetre
IR	Infrared
UV	Ultra-Violet
AERONET	Aerosol Robotic Network
Atmoscope	Autonomous Tool for Measuring Observatory Site Conditions Precisely
EHT	Event Horizon Telescope
AMT	Africa Millimetre Telescope
H.E.S.S.	High Energy Stereoscopic System
CT	Cherenkov Telescope
VLBI	Very Long Baseline Interferometry
ADC	Analog-to-digital converter
UNAM	University of Namibia
ESO	European Southern Observatory

JCMT	James Clerk Maxwell telescope
SMA	Submillimeter Array
ALMA	Antacama Large Millimeter Array
LMT	Large Millimeter Telescope
SPT	South Pole Telescope
IRAM	Institut de Radioastronomie Millimtrique
SMT	Submillimeter Telescope
SEST	Swedish-ESO Submillimter Telescope
M86	Messier 86
MPIA	Max Planck Institute of Astronomy
AGN	Active Galactic Nuclear
GBT	Green Bank Telescope

Prefixes: $n = 10^{-9}$, $\mu = 10^{-6}$, m=10^{-3}, k=10^{3}, G=10^{9}

Constants: $h = 6.63 \times 10^{-34}$ J.s, $c = 2.998 \times 10^{8}$ m/s

Acknowledgments

I would like to express a deep sense of gratitude and sincere thanks to my supervisors Dr. Rhodri Evans and Dr. Michael Backes for their continuous motivation and guidance throughout this project. Without their wise council and guidance it would have been impossible to complete the present work.

I would also like to thank Mr. Jimmy Shapopi for helping me in writing the data extraction script in C++ and for working closely with me in preparation of research proposals and making sure we were progressing well before due dates.

Lastly, I thank the PI(s) for their effort in establishing and maintaining H.E.S.S. sites AERONET station.

Dedication

This thesis is dedicated to my mother, Hendrina Shishiveni, and my aunt, Rauna Hangula. Without their endless support, love and encouragement I would never have been able to complete my studies. I appreciate everything that you have done for me.

This thesis is also dedicated to my grandma, Tuyimo Rainhold and my siblings, Elkana Frans and Kronelia Kalute. I love you all.

Declarations

I, Lott Ndeyanale Frans, declare hereby that this study is a true reflection of my own research, and that this work, or part thereof has not been submitted for a degree in any other institution of higher education.

No part of this thesis may be reproduced, stored in any retrieval system, or transmitted in any form, or by means (e.g. electronic, mechanical, photocopying, recording or otherwise) without the prior permission of the author, or the University of Namibia in that behalf.

I, Lott Ndeyanale Frans, grant the University of Namibia the right to reproduce this thesis in whole or in part, in any manner or format, which the University of Namibia may deem fit, for any person or institution requiring it for study and research; providing that the University of Namibia shall waive this right if the whole thesis has been or is being published in a manner satisfactory to the University.

Lott Ndeyanale Frans   Date: 20 October 2019

Chapter 1

Introduction

The Event Horizon Telescope (EHT) is a large network of millimetre to sub-millimetre telescopes across the world that is used to image supermassive black holes. The Africa Millimetre Telescope (AMT) will be a 15 m single dish telescope that [1] shall compliment the EHT. The AMT is planned to be built on Mt Gamsberg in the Khomas Highlands of Namibia [1]. The AMT will significantly increase the resolution and sensitivity of the EHT [2].

Precipitable Water Vapour (PWV) is the amount of water vapour in the atmospheric column above a location equivalent to the amount of liquid precipitation that would result if all the water vapour in the column is condensed [3]. Water vapour is the main source of opacity in the Earth's atmosphere at infrared (IR) and millimetre to sub-millimetre wavelengths. That opacity is the primary reason that ground-based astronomical observations are only conducted at certain wavelength regions, called windows. These windows are relatively well-defined and stable for most observations at temperate latitudes and moderate elevations above sea level because of the relative lack of water absorption lines and other telluric absorption features [4].

This study underlines effects of PWV on millimeter wave radio astronomy and uses mathematical and statistical means to determine the PWV at Mt Gamsberg. In order to determine the PWV at Mt Gamsberg and its suitability for millimetre (mm) wave astronomy the following objectives were set for this study:

(a) Determine a mathematical relationship between PWV and sky temperature by means of data calibration of the AERONET PWV data, Atmoscope IR sky temperature data and the Cherenkov Telescope (CT) IR sky temperature data.

(b) Use the relationship in (a) to determine the PWV at the H.E.S.S. site as given by the CT IR sky temperature data,

(c) Determine the PWV at Mt Gamsberg by scaling the PWV obtained at the H.E.S.S. site in (b) for Mt Gamsberg,

(d) Determine the relative frequency distribution of PWV, mean monthly PWV and the PWV seasonal variations at both the H.E.S.S. site and Mt Gamsberg,

(e) Analyze and compare PWV results of both sites in (d) to PWV results from other sites that host mm wave observatories and determine if Mt Gamsberg is a suitable site to conduct millimeter wave astronomy.

The following is an outline of how this thesis is structured:

Chapter 2: Gives an overview of the electromagnetic spectrum, the radio window, mm wave astronomy and the effects of PWV on mm wave astronomy. Moreover PWV conditions from some sites that host mm wave astronomy observatories are discussed along with previous measured PWV conditions at Mt Gamsberg.

Chapter 3: A brief description is given of the instruments used in this research, namely the Atmoscope, AERONET and CT radiometers.

Chapter 4: A description of the data recorded by the Atmoscope, AERONET and CT radiometers is given. Furthermore, the processes involved in obtaining the PWV, its relative frequency distribution, seasonal variations and monthly mean at both Mt Gamsberg and the H.E.S.S. site are outlined.

Chapter 5: The results obtained based on chapter 4 are presented and discussed, moreover these results of the computed PWV are compared to the previous PWV measurements taken at Mt Gamsberg and those of other sites where mm wave astronomy is conducted.

Chapter 6: Conclusion to the discussions on the suitability of Mt Gamsberg for mm wave astronomy as presented by the results (Chapter 5) and further recommendations are given.

Chapter 2

Literature Review

2.1 Electromagnetic Spectrum

Although all electromagnetic waves follow the same concept in their formation, they are classified into different groups depending on their property, namely wavelength (λ), frequency (ν) or energy [5]. Figure 2.1 shows the classification of electromagnetic waves, with the waves having wavelength of 400 nm to 700 nm classified to be visible light [6]. The visible spectrum is the only range which the human eye is sensitive to, however astronomical objects mainly radiates over a range of wavelength which are not visible to the human eye [5].

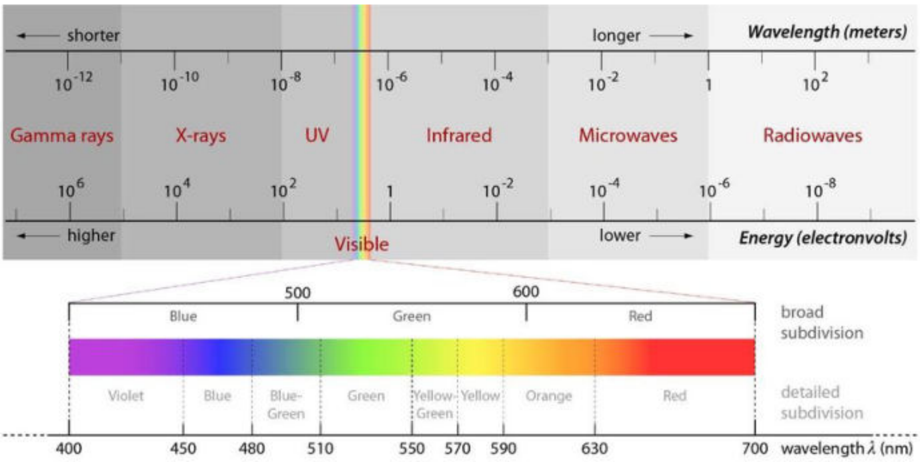


Figure 2.1: The classification of electromagnetic waves into groups depending on wavelength and/or energy. Altogether these groups make up the electromagnetic spectrum [6]. The electromagnetic spectrum ranges from radio waves which have the longest wavelength and lowest energy to the gamma rays which have the shortest wavelength and highest energy.

In 1905, Albert Einstein showed that despite light being a wave it can sometimes act as a particle (photon). This Einstein proposed in-order to explain the results of the photoelectric effect phenomena. Furthermore, Einstein showed that energy carried by a photon is proportional to the frequency of radiation and is related by,

$$E = h\nu \tag{2.1}$$

where h is the Planck's constant and is equal to 6.63×10^{-34} joule seconds (Js), which therefore quantified the double nature of light [5]. This led to the definition of quantum mechanics, which shows that atoms can absorb or emit radiation of only certain fixed wavelengths and hence energies, which are related to fixed state transitions [5].

The understanding of the dual nature of electromagnetic radiation allowed astronomers to decipher information of celestial bodies that is carried by electromagnetic waves on to Earth [5] by a method of spectroscopy. Spectroscopy is the study of incoming radiation by splitting it into its component wavelengths. This process has allowed astronomers to not only deduce certain properties of celestial bodies but also their distance and relative motion just by studying the radiation emitted by them [5].

Despite electromagnetic waves having a wide variety of wavelengths as in figure 2.1, only the visible and radio waves at frequencies greater than 10 MHz reach the ground [7] as can be seen in figure 2.2. This is because the earths atmosphere absorbs electromagnetic radiation at other wavelengths, making ground base astronomy only accessible over the radio and visible spectrum [7].

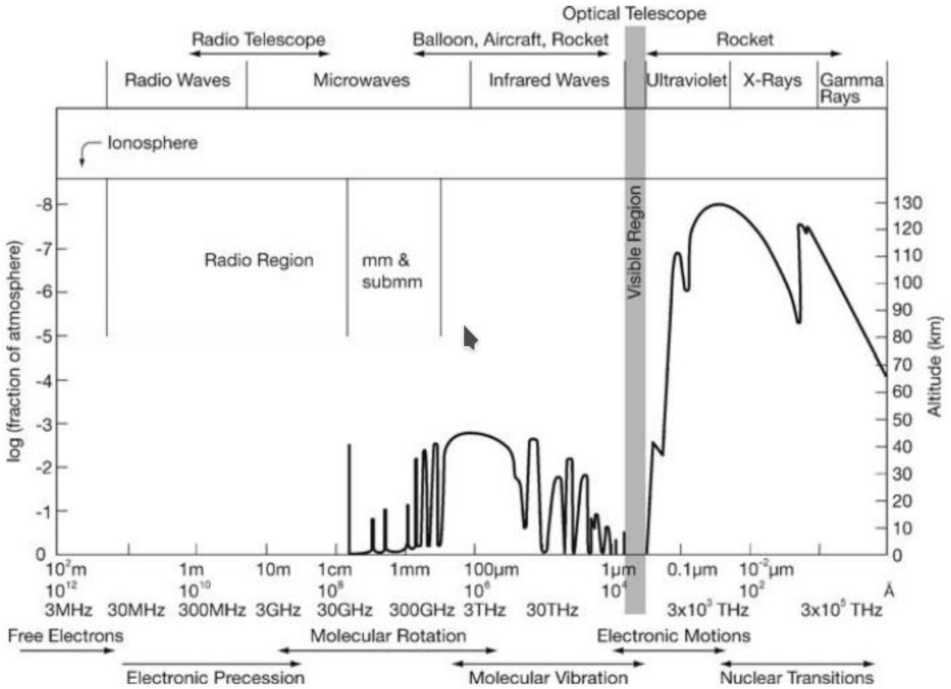


Figure 2.2: The transmission of electromagnetic radiation (waves) in the Earth's atmosphere [8].

2.2 Types of radiations

Accelerated charged particle emits electromagnetic radiation with power specified by Larmors equation,

$$P = \frac{2}{3}\frac{q^2\dot{v}^2}{c^3} \tag{2.2}$$

where P is the power of the electromagnetic, q the electrical charge, $\dot{v}$ the acceleration vector and c the speed of light [7]. Equation 2.2 implies that a charged particle radiates when accelerated and that the total radiated power is proportional to the square of the acceleration [7]. Electromagnetic forces are in astrophysical known to produce the strongest acceleration of charged particles [7]. Acceleration by an electric field accounts for free-free radiation whilst that by the magnetic field produces magneto-bremsstrahlung [7].

2.2.1 Magnetobremsstrahlung

The character of magnetobremsstrahlung depends on the speeds of the electrons [7], with:

Gyro radiation - Produced when the velocities are much smaller than the speed of light ($v << c$) [7].
Cyclotron radiation - Produced when the kinetic energy of the electrons are comparable with their rest mass ($m_e c$) [7].
Synchrotron radiation - Produced when the kinetic energy of the electrons is much larger than the rest mass ($m_e c <<$ kinetic energy) [7].

Synchrotron radiation is common in astronomy and accounts for most of the radio emission from active galactic nuclei (AGN) which is powered by supermassive black holes in galaxies and quasars, and it dominates the radio continuum emission from star-forming galaxies like our own at frequencies below ≈ 30 GHz [7].

2.2.2 Electrostatic bremsstrahlung

Most astronomical sources of electrostatic bremsstrahlung are thermal because the radiating electrons have the Maxwellian velocity distribution of particles in LTE [7].Thermal emission is produced by a source whose emitting particles are in local thermodynamic equilibrium (LTE), otherwise non-thermal emission is produced [7]. LTE is a termed referred to, when an the emitting/absorbing material is in thermal equilibrium at a well-defined temperature even if it is not in equilibrium with the radiation field [7].

2.3 Radio window

As can be seen in Figure 2.2 the radio band ranges roughly from lower frequency limit of 15 MHz (λ= 20 m) to a high frequency cut-off at 1.5 THz (λ=0.2 mm). The high frequency cut-off occurs because of the resonant absorption rotation bands of molecules in the troposphere in this frequency range. This is mainly due to two molecules namely Water Vapor (H_2O) and O_2 with Water Vapor having an absorption band at 22 GHz (λ=1.35 cm) and 183 GHz (λ=1.63 mm) whilst O_2 has a strong absorption band at 60 GHz (λ=5 mm) [8].

2.4 Millimeter radio astronomy

Radio astronomy is the study of natural radio emission from celestial sources [7]. The millimetre wave band is part of the radio band and as can be seen from Figure 2.2, the millimetre wave band ranges from 1 cm to 1 mm and beyond to submillimetre wavelengths up until 200 μm. Although the angular resolution of telescopes in radio astronomy is generally poor compared to other telescopes such as optical telescopes [5], radio astronomy generally has a lot of advantages.

Firstly, because the radio window is broad, almost all astronomical sources, thermal and non thermal radiation mechanism, and propagation phenomena can be observed. Secondly, dust scattering is negligible because interstellar dust is much smaller than radio wavelengths thus making the dusty interstellar medium transparent. Thirdly, nearly every astronomical object is a thermal radio source. Lastly, radio synchrotron sources live long after their emitting electrons were accelerated to relativistic energies thus they can provide long lasting information of a past energetic phenomena [7].

2.5 Single dish radio astronomy

A radio telescope is an astronomical instrument consisting of a radio receiver and an antenna system that is used to detect radio-frequency radiation emitted by extraterrestrial sources. Radio telescopes are built with a large mount which supports a huge curved dish. The dish with diameter D is designed to capture incoming radiation (radio waves) with wavelength λ and reflect it onto the focus, where the receiver detects it [5]. The angular resolution of a telescope is the smallest angle on the sky between two sources of radiation that can be discerned as separate sources with that telescope [9]. The angular resolution θ in radians of a single dish telescope is given by

$$\theta = \frac{\lambda}{D}. \tag{2.3}$$

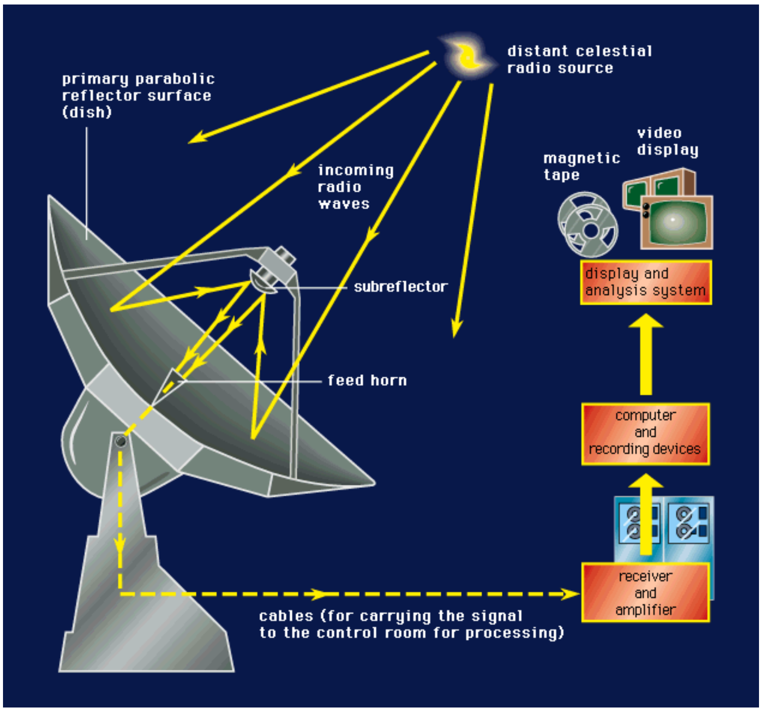


Figure 2.3: Schematic diagram of the telescope shows the path taken by an incoming beam of radio radiation [10]. A distant object emits radio signals which are then collected by the dish and reflected onto the subreflector. The subreflector then reflects the waves into the feed horn where the electromagnetic waves are converted into a radio frequency voltage (small voltage) which is then received and amplified by the amplifier. The amplified signal is then stored for analysis with computer.

Figure 2.3 shows the typical sequence of events carried out from observations to imaging of a radio source if a radio wave from the source with energy such as that in equation 2.1, travels through space and reaches the telescope, where it is reflected onto the receiver by the radio dish. The electromagnetic energy is then converted into electrical energy by the feed horn and fed with cables to the analog-to-digital converters (ADC) which amplifies and digitized the signal. The digitized signal is fed to be processed, analyzed and stored by a computer [5].

2.5.1 Africa Millimetre Telescope

The Africa Millimetre Telescope (AMT) will be made up of a 15 m diameter Swedish-ESO Submillimetre Telescope (SEST) that will be built on Mt Gamsberg. SEST has a surface accuracy of 70 μm (rms) and has a pointing accuracy of 3 arcsecond in azimuth and elevation [11]. The AMT will host a 1.3 mm (230 GHz) and 0.8 mm 345 GHz band receiver. The AMT will also host the receivers that observe in the centimeter wave regime of radio emissions [1].

As a single dish telescope the AMT will be used in single dish science cases such as mm wave observation for flux density monitoring of Active Galactic Nuclei (AGN) and studies of molecular emission lines [1]. Figure 2.4, shows a SEST which is planned to be adopted for the AMT.

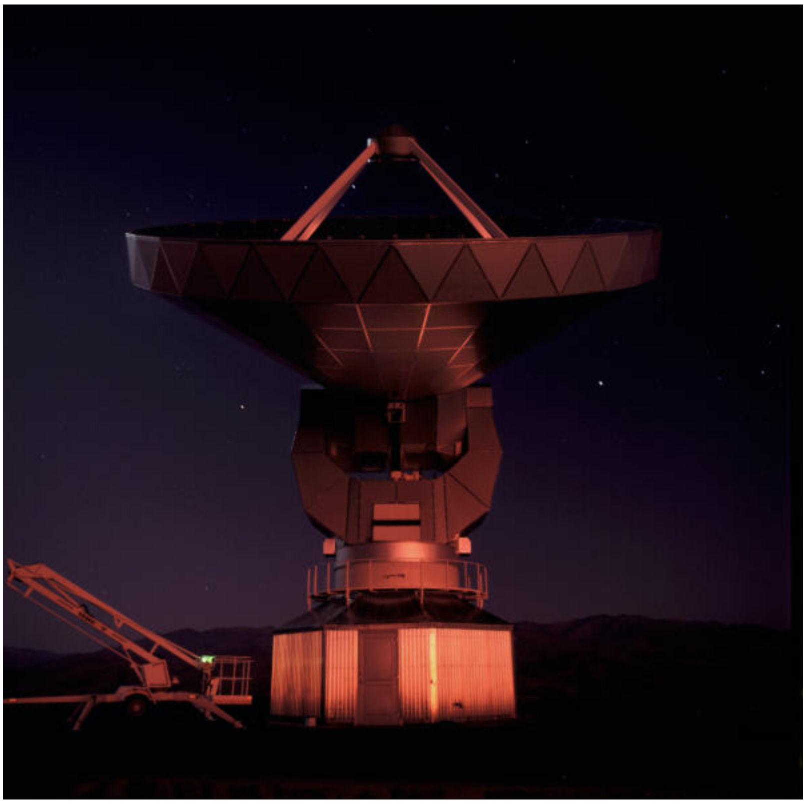


Figure 2.4: The 15 m diameter SEST, credits: ESO/H.Zodet [12]. The AMT will consist of the decommissioned SEST telescope which is currently placed at the La Silla Observatory in Chile.

2.6 Radio interferometry

Given equation 2.3, in order to have high angular resolution when observing at longer wavelengths (λ) in the radio spectrum, the diameter (D) of the telescope has to be very large. The largest single steerable radio telescope the Green Bank Telescope (GBT) is 100 m in diameter, though impressive it is dwarfed by the size of the diameter of a telescope needed to archive sub-arcsecond resolution. For example using the relation in equation 2.3, if the GBT is to reach 1 arcsecond angular resolution θ when observing at 21 cm then,

$$D = \frac{\lambda}{\theta} \tag{2.4}$$

since 1 arcsecond= 0.0000048 rad and 21 cm = 0.21 m, then

$$D = \frac{0.21\ \text{m}}{0.0000048} = 43750\ \text{m} = 43.75\ \text{km} \tag{2.5}$$

hence, the GBT will need to be 43.75 km in diameter to reach angular resolution of 1 arcsecond. Building a single dish telescope with a diameter of that size is difficult and nearly impossible. Therefore single dish radio telescopes have relatively low angular resolution [7].

Astronomical radio interferometry is the practice of correlating the outputs from multiple antenna pairs to synthesize an aperture equivalent to the distances between the antennas. This allows for higher resolution such as arcsecond or better to be achieved in comparison to those that can be achieved through a single dish antenna [13]. The angular resolution of the interferometer then becomes

$$\theta = \frac{\lambda}{B}\,, \tag{2.6}$$

were B is the longest distance between any two antennas (baseline) in the interferometer, and λ the observing wavelength [14]. The baseline can extend thousands of kilometers therefore giving high angular resolutions.

Figure 2.5 shows a setup of a two antenna interferometer, with b the separation distance (baseline) between the two antennas, with τ_g the geometric delay, τ_0 the instrumental delay, V_1 and V_2 the output voltages from antenna 1 and 2, respectively.
The relative phase ϕ between the signals received at telescopes 1 and 2 as shown in figure 2.6, dependent on the geometric delay τ_g and the observing wavelength λ, with the geometric delay τ_g given by equation 2.7

$$\tau_g = \frac{\text{distance of projection } \vec{b} \text{ in the diection of } \hat{\boldsymbol{s}}_o}{\text{speed of light}} = \frac{\vec{b}\cdot\hat{\boldsymbol{s}}_o}{c}\,. \tag{2.7}$$

The phase difference can be eliminated in any one direction $\hat{\boldsymbol{s}}_o$ called the delay center by introducing a compensating delay $\tau_0 \approx \tau_g$ in the signal path of the leading antenna as shown in Figure 2.5.

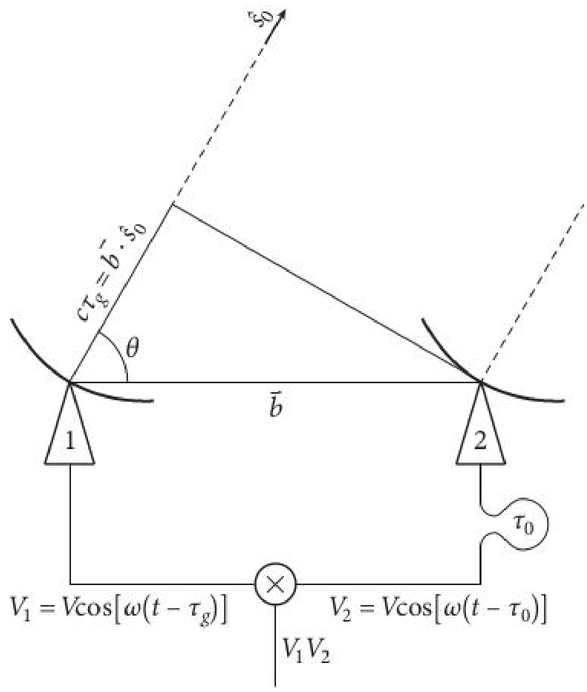


Figure 2.5: A schematic diagram of a two-element correlation interferometer [7]. A compensating delay (τ_0) is introduced in the signal by electronics of the leading antenna (antenna 2) to compensate for the geometric delay (τ_g) of the signal to antenna 1.

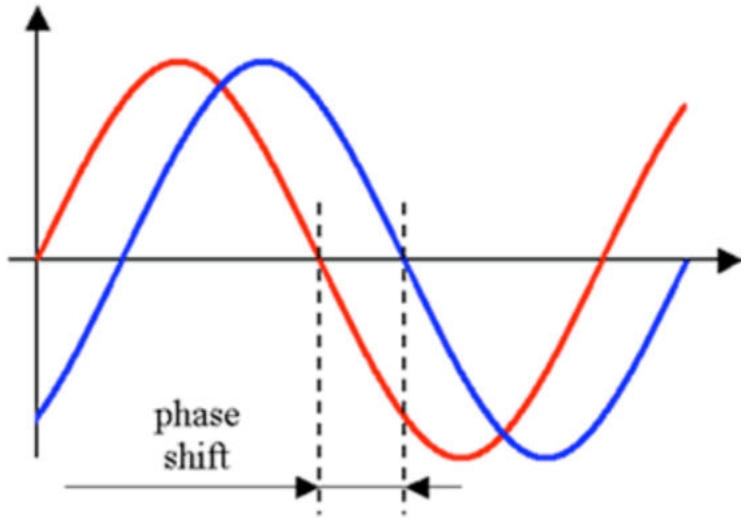


Figure 2.6: Phase difference due to τ_g which is compensated by the electronics of the leading antenna with τ_0 [7].

These voltages (V_1 & V_2) are amplified, multiplied, and time averaged by the correlator to yield an output response whose amplitude (R) is proportional to the flux density of the point source [7].

2.6.1 The Event Horizon Telescope

The Event Horizon Telescope (EHT) is a very long baseline interferometry (VLBI) network of telescopes around the globe that observe at millimetre wavelengths [1]. The EHT is capable of achieving 12-20 μarcsecond resolution and is used to image supermassive black holes. Table 2.1 shows the 7 telescopes around the globe that makes up the EHT.

Table 2.1: Characetristic of the Telescopes that make up the EHT [1].

Telescope name	Acronym	Location (Country)	Number of antennas	Diameter [m]
James Clerk Maxwell telescope	JCMT	Hawaii	1	15
Submillimeter Array	SMA	Hawaii	8	8
Large Millimeter Telescope	LMT	Mexico	1	50
South Pole Telescope	SPT	South Pole	1	10
Submilimeter Telescope	SMT	Arizona	1	10
Institut de Radioastronomie Millimtrique	IRAM	Spain	1	30
Antacama Large Millimeter Array	ALMA	Chile	66	12 & 7

Figure 2.7 shows the location of the telescopes that makes up the EHT around the globe and the baselines between this telescopes.

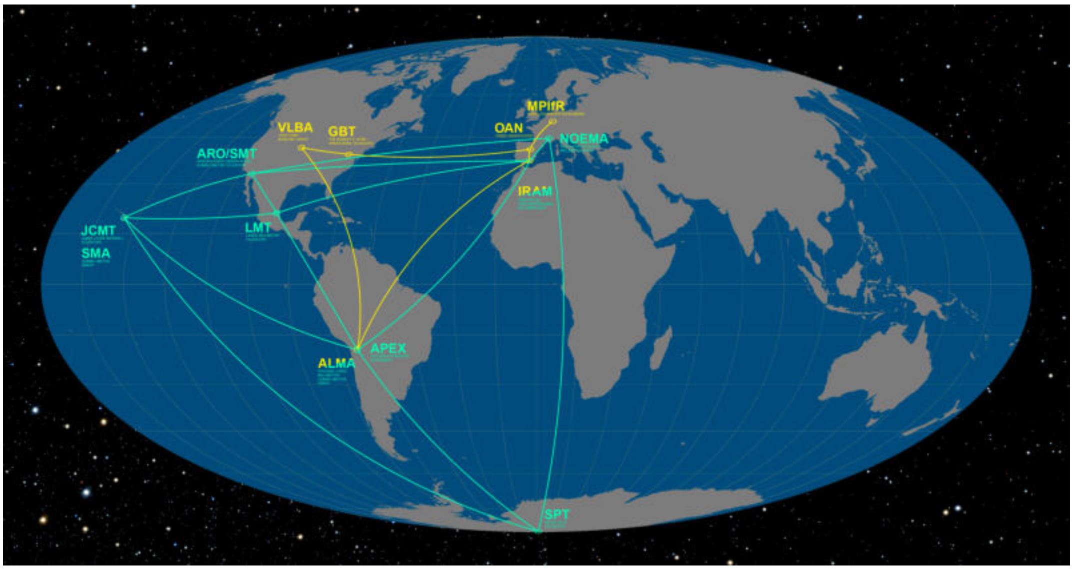


Figure 2.7: The EHT array, credits: ESO/O. Furtak [15]. The EHT is a very long baseline interferometry array that is made up of telescopes around the globe. These telescopes include the SPT, ALMA, LMT, JCMT, SMT, SMA and IRAM

In 2017, the EHT observed M87 at 1.3 mm (230 GHz) wavelength on the 5^{th}, 6^{th}, 7^{th} and 11^{th} of April with 7 of its station from 5 geographical locations. In 2019, the first images of the black hole was revealed, and showed a prominent ring with a diameter of $\approx 40\ \mu$as, consistent with the size and shape of the lensed photon orbit encircling the shadow of a supermassive black hole. The ring was found to be consistent across the four nights of observation and shows enhanced brightness in the south [16] as can be seen in figure 2.8.

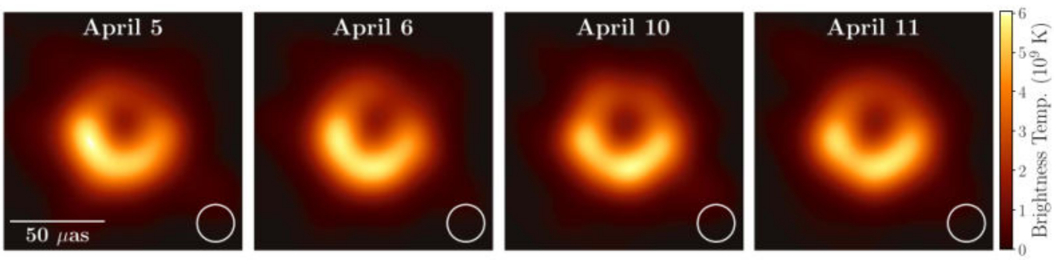


Figure 2.8: Image of M87 across 4 days of observations. A ring of diameter $\approx 40\ \mu$as and a shape of the lensed photon orbit encircling the shadow of a supermassive black hole was observed [16].

2.6.2 EHT + AMT

Apart from operating as a single dish telescope, the AMT will also be an addition to the EHT and therefore it will significantly increase the *u*-*v* coverage of the EHT [1] as can be seen in figure 2.9.

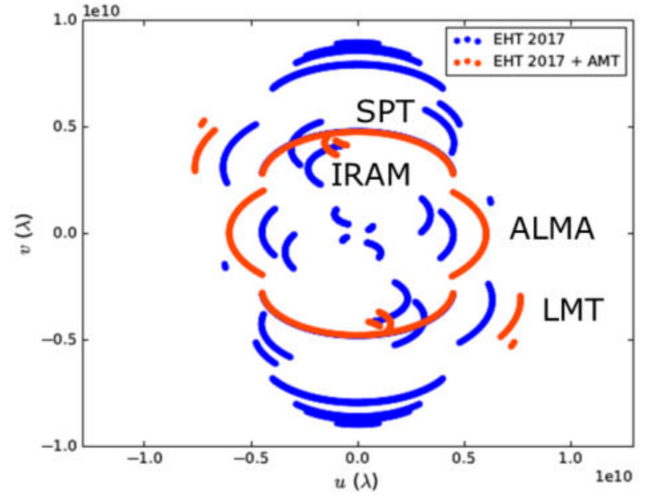


Figure 2.9: The UV coverage of the EHT and that of the EHT+AMT [1].

As can be seen in Figure 2.10, the addition of the AMT to the EHT will also add essential baselines to all other facilities (telescopes) allowing continuous observations as the Earth rotates and the telescope that can see the Black Hole changes. In addition, every addition to the network increases its sensitivity and the AMT, being a dedicated telescope, provides for ample redundancy of the network.

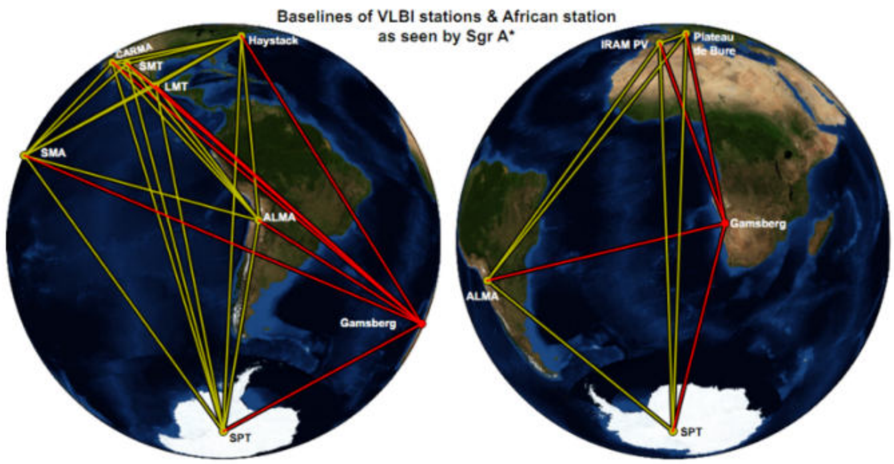


Figure 2.10: The existing baselines of the EHT (yellow) and that added when the AMT (red) is added to the EHT array [1].

2.7 The effect of Precipitable Water Vapour (PWV) on observations

Phase fluctuations caused by water vapour (discussed earlier in section 2.3) are generally the limiting factor at obtaining high angular resolution at millimetre to submillimetre interferometers such as the EHT. Moreover, water vapour also attenuates the astronomical signal and increases the system temperatures. Therefore the main challenge for millimetre and submillimetre interferometers would be to correct for phase fluctuations caused by water vapour [17].

Figure 2.11 shows a signal from an astronomical radio source being received by two observing antennas in an interferometer. Since the refractive index of air rises with the amount of water vapour it contains, the electromagnetic waves propagate more slowly through moist than dry air. Therefore the signal from antenna 1 will arrive later than that of antenna 2, with the interferometer measuring the difference in arrival time Δt as a phase given by,

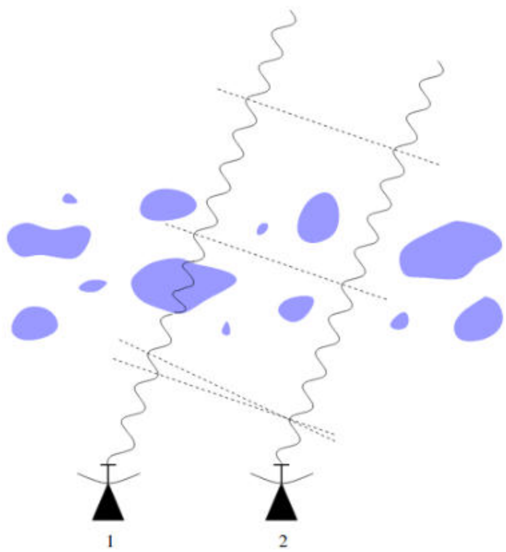


Figure 2.11: A cartoon on effect of water on an astronomical signal [13]. Whilst there is no delay of the signal at antenna 2, the EM waves propagate slowly due to water vapour at antenna 1 causing a delay in the signal received at antenna 1.

$$\phi = \nu \Delta t \tag{2.8}$$

with ν the observing frequency. Additionally, phase fluctuations also decreases the signal strength from the source [17]. Moreover, finding the phase corrections due to the large separation is already difficult without the PWV effect and thus the PWV effect therefore makes finding the correct phase correction even more complex.

2.7.1 Atmospheric transmission

Apart from water vapour causing phase fluctuations, it also attenuates electromagnetic radiation. Figure 2.12 shows an example of the transmission of electromagnetic radiation at different frequency at a site of an elevation of 2400 m above sea level when water vapour is 1 mm and 4 mm [18].

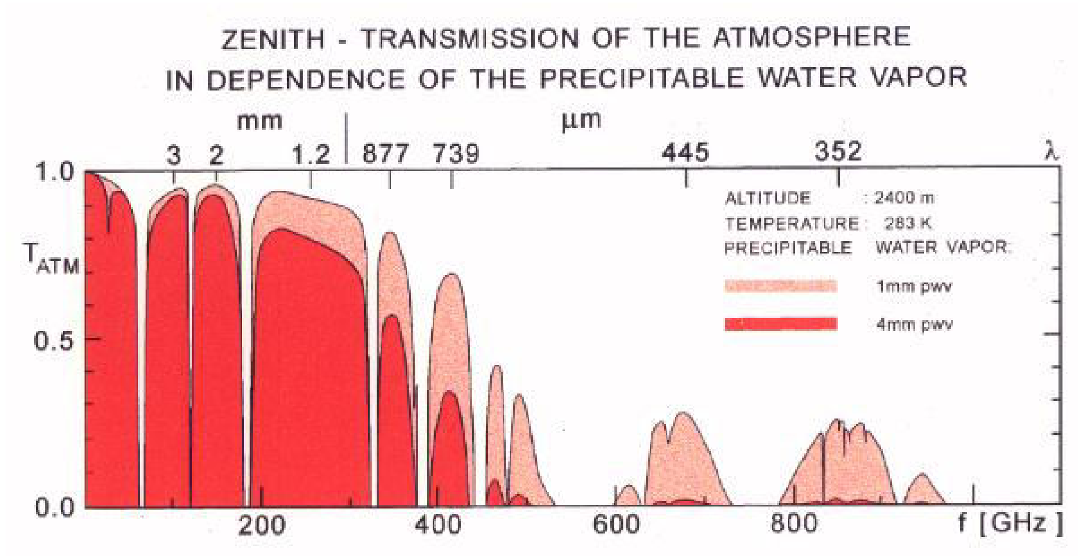


Figure 2.12: An example of transmission of EM waves in the atmosphere depending on the amount of PWV and height [18].

Figure 2.12 shows when observing at an altitude of 2400 m above sea level, at 800 GHz only 2% of the electromagnetic radiation reaches the telescope when the PWV is 4 mm but if the PWV is 1 mm (better PWV conditions) then 25% of the electromagnetic radiation reaches the telescope [18]. The EHT observes at 230 GHz, at this frequency about 80% of the electromagnetic radiation will be observed even when the PWV is 4 mm and even higher at 90% at 1 mm. When observing at 200 GHz or lower, PWV has minimum effect and most observations can be conducted under PWV conditions that would not be suitable at sub-millimetre observations [18].

2.8 PWV at millimeter astronomy sites

A study was carried out to determine the best site for millimetre wave astronomy across 11 different sites by determining PWV [19]. Data recorded by the Infrared Atmospheric Sounding Interferometer (IASI) on a Metop-A satellite was used to retrieve PWV from all 11 sites for three years namely 2008, 2009 and 2010. The sites included, Dome C, Dome A, Cerro Chajnantor, Cerro Macon, Summit (Greenland), Yangbajing (Tibet), Palmdale, Christchurch, South Pole, Mauna Kea and Chajnantor Plateau. Table 2.2 shows the profiles of the latter three sites [19].

Table 2.2: Profile of some sites where mm wave astronomy is conducted.

site	latitude	longitude	altitude [m]	telescope
Chajnantor Plateau	23°00′s	67°45′w	5100	ALMA
South Pole	90°s	0°e	2800	SPT
Mauna Kea	19°45′n	155°27′w	4207	JCMT

These three sites host telescopes that partake in the EHT and thus are of particular interest for comparison with the results of this study. Figure 2.13 shows the PWV results in days over Chajnantor Plateau which houses ALMA.

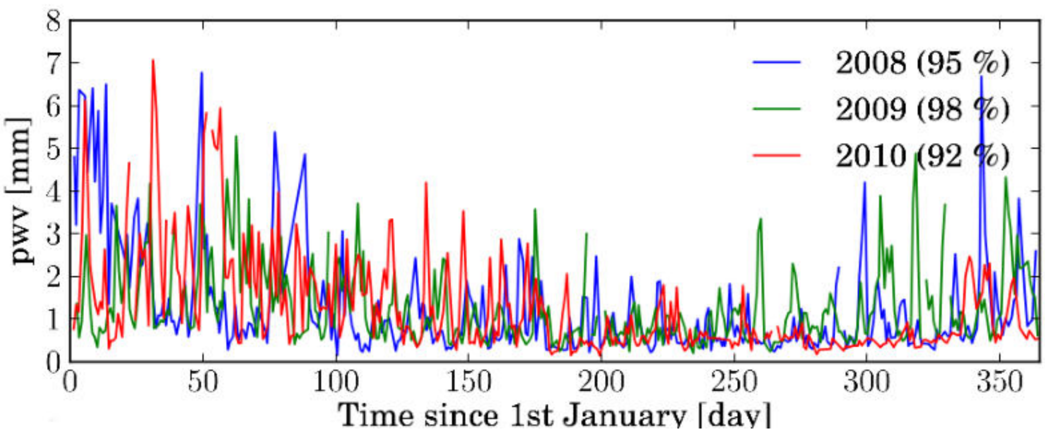


Figure 2.13: PWV variations from 2008 – 2010 at Chajnantor Plateau in Chile where ALMA is located [19]. The percentage indicate fraction of days of the year of which satellite measurements were taken. The PWV for all three years show a similar trend of PWV rising in the summer period and dropping during the winter period.

The percentage in Figure 2.13 indicates the fraction of days of the year on which data from satellite measurements could be extracted. As can be seen there is a clear seasonal variation with the PWV being as high as 7 mm for the first 100 days and then dropping below 4 mm from 150-300 days and again rising towards the year end, since Chajnantor Plateau is in the southern hemisphere this can simply be interpreted as the PWV rising in the summer periods and dropping in the winter period.

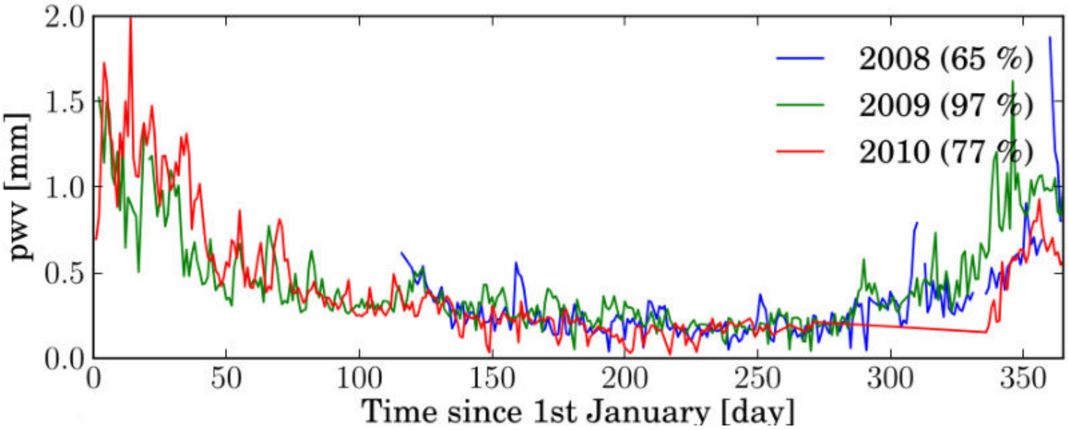


Figure 2.14: PWV variations from 2008 – 2010 at the South Pole site, where the SPT is located. The PWV for all three years show a similar trend of PWV rising in the summer months and dropping in the winter months, also the PWV at the South Pole is lower then the PWV of the other sites in winter. The south pole is among the sites with lowest PWV with it frequently droping below 0.5 mm in the winter period [19].

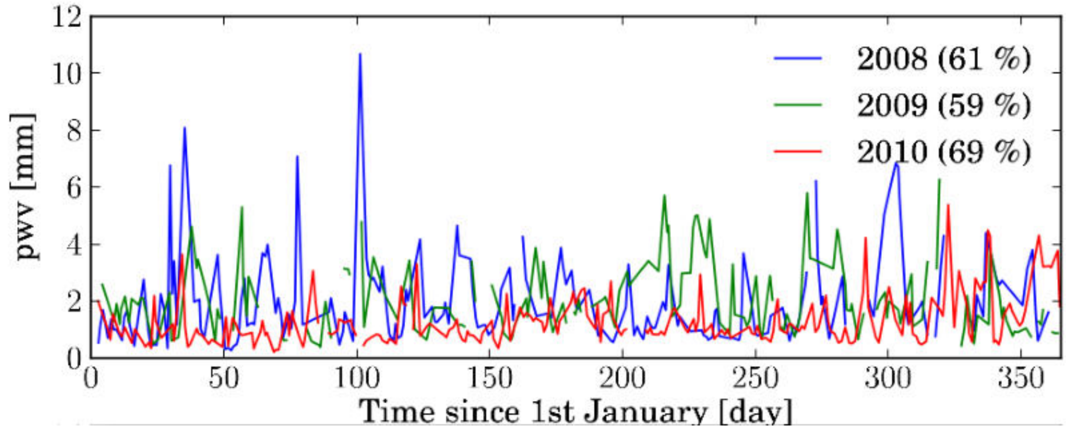


Figure 2.15: PWV variations from 2008 – 2010 at Mauna Kea, where the JCMT is located [19]. Considering Mauna Kea is in the Northern Hemisphere, the PWV for all three years show a common trent of PWV rising in the summer period (mid-year) and dropping during the winter period (start year-end year). Mauna Kea generally has higher PWV then Chajnantor and the South Pole.

The same trend can be seen in figure 2.14 for the South Pole site, with the PWV being extremely low (less than 1.5 mm) throughout the year and even lower than 0.5 mm in the winter period (150 days to 300 days). Similarly, the same trend can be observed in figure 2.15 when considering Mauna Kea is in the Northern hemisphere. The PWV drops below 4 mm in the winter months (begining of year and towards end year), and rises as high as 10 mm during the year.

Ultimately the study concluded on all the 11 sites and in doing so ranked the site located at the South Pole first with PWV lower than 0.5 mm for 75% of the time for the three years (2008 – 2010), the Chile sites which has the Chajnantor Plateau second with 1.5 mm for 75% of the time for the three years and in accordance with this study the Mauna Kea last with 2.6 mm 75% of the time for the three years [19].

2.9 Mt Gamsberg and the H.E.S.S. site

2.9.1 Site Testing at Gamsberg Mountain

Figure 2.16: Mt Gamsberg in the Khomas Highlands of Namibia [20]. The mountain is generally considered to be dry with its condition previosly compared to that of La Silla in [21].

Standing at 2347 m above sea level and located 23°3′S and 16°2′E in the Khomas Highlands of Namibia, Mt Gamsberg is considered to be a great site for astronomy mostly due to its number of photometric nights and seeing condition which is comparable to ESO site at La Silla in Chile [21].

A study was carried out at Mt Gamsberg by the European Southern Observatory (ESO) and the Max Planck Institute of Astronomy (MPIA) in 1994 − 1995, which measured the yearly average PWV column on photometric nights to be 5.2 mm, with the maximum monthly average of 7.5 mm in December 1994 and the minimum monthly average of 2.5 mm in August 1994. The results are presented in Figure 2.17 and showed favorable PWV conditions at Mt Gamsberg for mm wave observation throughout the year. The PWV results also indicate the mountain is capable of hosting a mm telescope that can seasonally participate in EHT observations in the months of June, July and August [1,22].

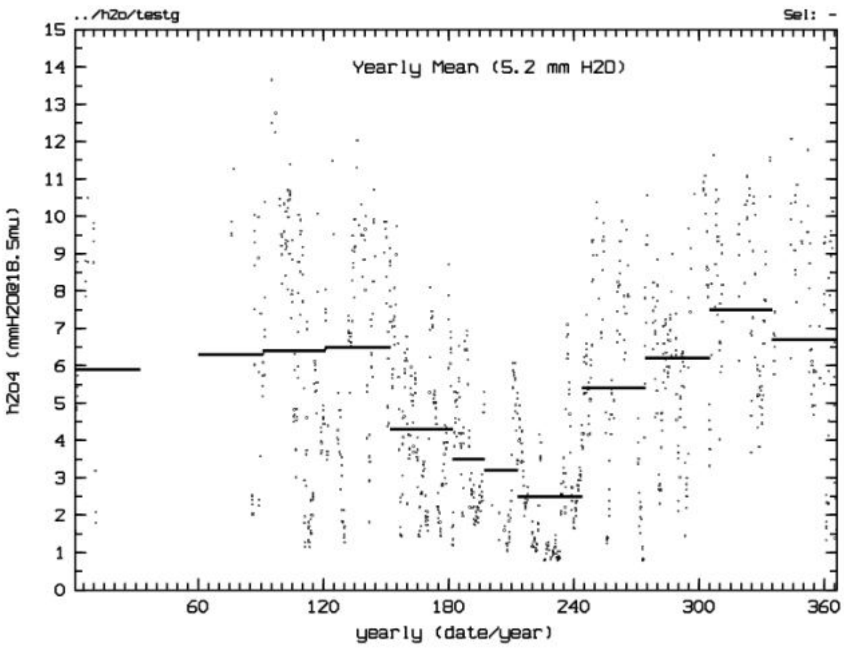


Figure 2.17: The mean PWV recorded at Mt Gamsberg from mid July 1994 to mid July 1995 in photometric nights [22]. These results show favorable conditions to conduct mm wave observations.

2.9.2 The H.E.S.S. site

The High Energy Stereoscopic System (H.E.S.S.) is a system of imaging atmospheric Cherenkov Telescopes (CT) that is located in the Khomas Highlands 23°16′18"s and 16°30′00", approximately 100 km south west of Windhoek in Namibia. The observatory stands at 1800 m above sea level [23].

The H.E.S.S. observatory as can be seen in figure 2.18, consist of 5 telescopes that investigate cosmic gamma rays in the energy range from 10s of GeV to 10s of TeV. The first four telescopes of phase 1 all started operating in December 2003 [23].

Figure 2.18: The H.E.S.S. observatory. The High Energy Stereoscopic System (H.E.S.S.) is a system of 5 imaging atmospheric Cherenkov Telescopes (CT) that is located in the Khomas Highlands [23].

The H.E.S.S. observatory and Mt Gamsberg are separated by a mere distance of 30 km in Khomas highlands as can be seen in figure 2.19. Mt Gamsberg can well be seen in the background from the H.E.S.S. observatory as can be seen in figure 2.20.

Figure 2.19: The distance Between H.E.S.S. site and Mt Gamsberg is ≈ 30 km and the height difference between the two sites is 547 m. The flat top of Mt Gamsberg is clearly visible in the bottom left of the aerial map. credits: Google maps

Figure 2.20: The flat top Mt Gamsberg viewed from the H.E.S.S. site [23]. Both the H.E.S.S. site and Mt Gamsberg are located in the Khomas Highlands [23] and as can be seen from figure 2.19 they are seperated by a mere ≈ 30 km.

2.9.3 Scaling PWV with height

In Hydrostatic equalibruim, the variation of density with height follows an exponential law. The $1/e$ scale height of water vapour H is taken as 2 km [24] so that the PWV P_{scaled_H} at the scaled height is given by,

$$P_{\text{scaled}_H} = \alpha P \tag{2.9}$$

where P is the initial PWV to be scaled and α is the scaling function [24, 25]. The scaling function is given by,

$$\alpha = e^{-(\mu/H)} \tag{2.10}$$

with μ the elevation difference between the two sites [24].

Given the weather conditions are the same in the vicinity of the H.E.S.S. site including Mt Gamsberg and by also considering the elevation difference of 547 m between the two sites, then the PWV at Mt Gamsberg can be scaled from the PWV of the H.E.S.S. site using equation 2.10 and 2.9.

Chapter 3

Instruments

In this research, data from three instruments that are located at the H.E.S.S. site was used. In this section a brief description of the instruments is given. The instruments are namely the IR radiometers, Atmoscope and the NASA AERONET station.

3.1 Heitronics KT19.82 IR radiometers

The Heitronics KT19.82 IR radiometer as seen in Figure 3.1 has a 2° field of view and operates in the 8-14 μm IR band and is sensitive to a temperature range of -100°C to 3000°C. Four IR radiometers are attached on the H.E.S.S. phase I Cherenkov telescopes (CT), with each of the four telescope having one IR radiometer pointing in that telescope's observing direction [26].

Figure 3.1: Heitronics KT19.82 IR radiometer [26]. The IR radiometer measures sky temperature in the range of -100°C to 3000°C.

These IR radiometers each take a sky radiance, interpreted as a sky temperature using the Stefan-Boltzmann law [26]. The IR radiometers take one reading per single observation (run) in the direction which the telescope is observing. A single H.E.S.S. observation run is on average 28 minutes long and thus all four IR radiometers record a single sky temperature value for that observation at the same time.

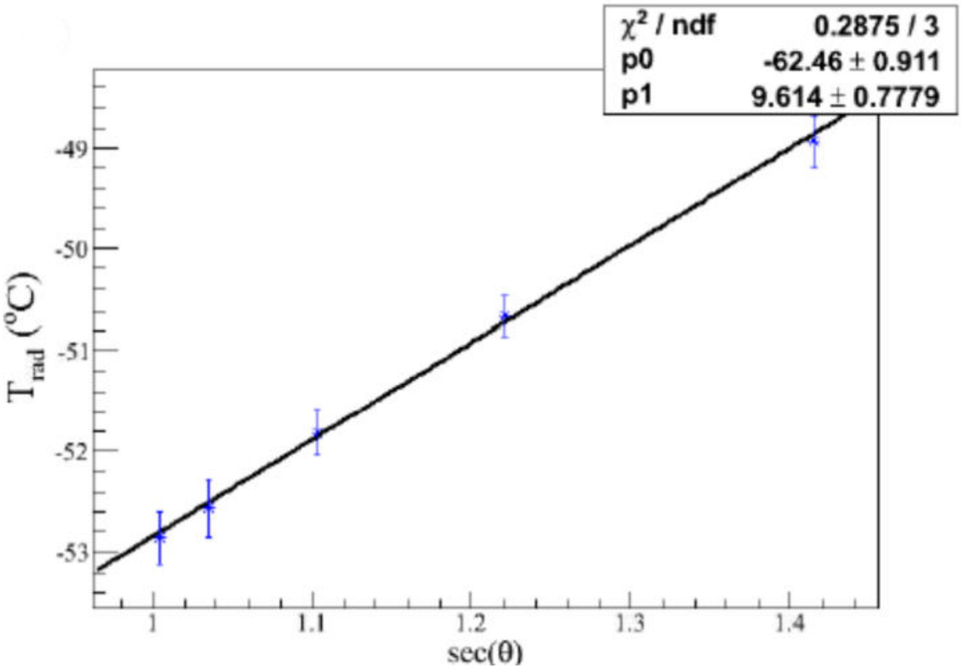


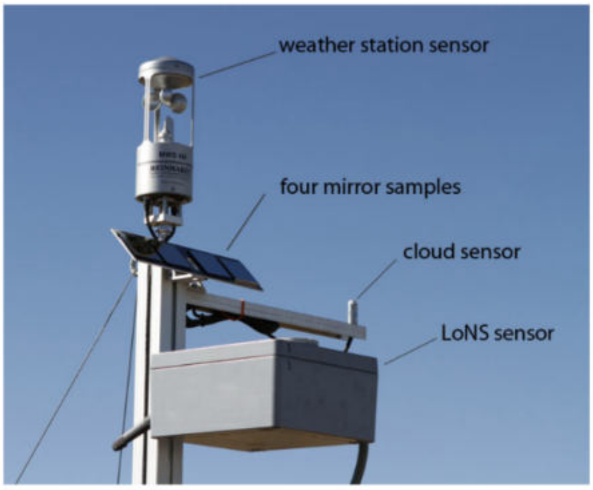
Figure 3.2: The sky temperature as a function of zenith angle for a clear sky (cloudless sky) as given by the IR radiometers [27].

A study was carried out [27] to investigate how sky temperature varies with elevation. This is because the telescopes would not be observing in the same direction and angle, thus zenith considerations would be needed in determining the sky temperature at the H.E.S.S. site. Figure 3.2 shows the relationship between sky temperature and the zenith angle for the IR radiometers on the CT [27].

3.2 Autonomous Tool for Measuring Observatory Site COnditions PrEcisely (Atmoscope)

The Atmoscope, as seen in Figure 3.3a is a weather station that was designed to find the best site for the Cherenkov Telescope Array (CTA). The Atmoscope records all information from observations to climate conditions.

(a) Atmoscope station.



(b) Atmoscope sensors.

Figure 3.3: The Atmoscope weather station [28]. The Atmoscope includes a host of sensors as can be seen in figure 3.3b and a solar panel as can be seen in figure 3.3a with two rechargeable batteries for power storage.

The Atmoscope station contains a host of "485 sensors" that detects all sort of weather elements as can be seen in figure 3.3b. The cloud sensor, uses a thermopile to measure the difference between ambient temperature and sky temperature readings in °C [28]. The WKS 485 internally calculates the cloud base from the sky temperature readings and ambient temperature readings using the following relationship,

$$T(h) = T_0 - hy \tag{3.1}$$

where $T(h)$ is the sky temperature, T_0 is the ambient temperature, h in meters the altitude and y is the temperature gradient given as 6.5 Kelvin per kilometer [28, 29]. The temperature sensor is sensitive to temperature between $-40°$C to $60°$C and is accurate to $\pm 0.3°$C. The cloud sensor has a $10°$ full field of view for an inhomogeneous cloud coverage with only a mean value measured [28]. The Atmoscope was set to take one reading per minute.

3.3 The National Aeronautics and Space Administration (NASA) AErosol RObotic NETwork (AERONET) station

The AERONET is a ground based aerosol monitering system. The AERONET contains a CIMEL Electronique 318A spectral radiometer which is solar powered and is a robotically pointed sun and sky radiometer. The instrument has an approximate $1.2°$ full angle field of view and has two detectors for measurements of direct sun, aureole and sky radiance. The instrument has a sun/aureole collimator that is protected by a quartz window which then allows for observation with a UV enhenced silicon detector with sufficient signal to noise for spectral observations between 300 nm to 1020 nm [30].

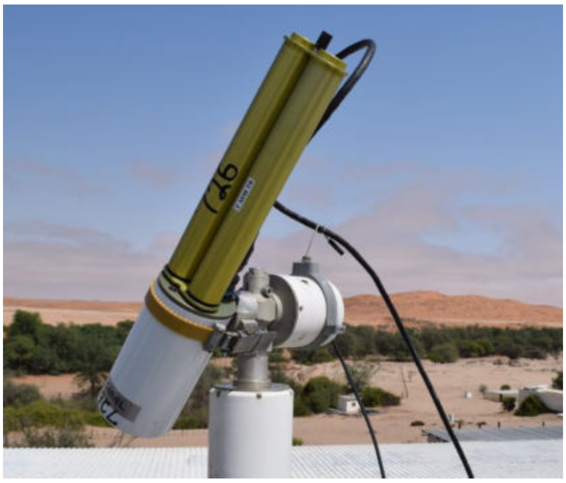


Figure 3.4: The AERONET station at Gobabeb, Namibia. The AERONET sun photometer measures the light energy that reaches it. Since the energy the sun produces at the top of the atmosphere is known, any difference measured by the instrument at the ground is caused by Aerosols such as smoke, dust and sea salt between the top of the atmosphere and the ground. From this the aerosol concentrations in the atmosphere can be calculated [31]. Credit: NASA/Jane Peterson

The column water vapour is determined by the AERONET using three channels of 675 nm, 870 nm and 940 nm. The total transmission (T) is computed for 675 nm and 870 nm using the Rayleigh and aerosol optical depths whilst the total transmission for 940 nm (T_{940}) is determined through extrapolation. The extrapolated transmission at 940 nm is subtracted from a measured transmission at 940 nm (T_{940}) resulting only in the transmission due to water vapour (T_w). The precipitable water vapour in cm is therefore determined by the AERONET using the following equations [32],

$$\ln[T_w] = \ln\left[T_{940[measured]}\right] - \ln\left[T_{940[extrapolated]}\right] \tag{3.2}$$

$$-\ln[T_w] = \ln\left[V_{0_{940}} d^{-2}\right] - \ln[T_{940}] - m\left(\tau_{940_{AOT}} + \tau_{940_{Rayleigh}}\right) \tag{3.3}$$

$$-\ln[T_w] = a\,(m_w u)^b \tag{3.4}$$

$$u = \frac{[-\ln T_w/a]^{1/b}}{m_w} \tag{3.5}$$

where u is the precipitable water in centimetres (cm), T_w is the transmission due to water vapour, a and b are filter dependent constants and m_w the water vapour optical air mass, d is the ratio of the average to the actual earth-sun distance, m the optical air mass, V_0 the extraterrestrial voltage, V the digital voltage and $\tau_{940_{AOT}}$ and $\tau_{940_{Rayleigh}}$ the aerosol and Rayleigh optical depths [32], respectively.

Chapter 4

Data and Methods

Data sets from the three instruments described in section 3 were used in determining the PWV at the H.E.S.S. site and Mt Gamsberg. A brief description of the instrument data is given in this section and how it was processed with two python scripts, one to read in data, calibrate the data and to obtain CT PWV whilst the other was to process the data to obtain monthly mean, seasonal variations and frequency distribution of the CT PWV for H.E.S.S. site. Moreover the second script was used to scale H.E.S.S. site PWV for Mt Gamsberg and compute the monthly mean, seasonal variations and frequency distribution for Mt Gamsberg. Both these scripts are attached at the appendix section together with the C++ script that was used to extract CT data from the H.E.S.S. database.

4.1 Data from Heitronics KT19.82 IR radiometers

H.E.S.S. observations take place from sunset to sunrise (astronomical dark time) with each H.E.S.S. observation run lasting 28 minutes on average and thus the IR radiometers record over night with each measurement being taken every 28 minutes on average. The H.E.S.S. telescopes have been observing since 2004, with the IR radiometers on the telescopes also operating for the same amount of time and has thus far recorded over a period of 16 years.

A script was written in C++ to access and extract data from the H.E.S.S. database. The script extracted and wrote into a file the sky temperature readings given by the IR radiometers on CT 1, CT 2, CT 3 and CT 4 with their corresponding elevation ($\theta_{CT1}, \theta_{CT2}, \theta_{CT3}, \theta_{CT4}$) and the recording time. In this thesis, the IR radiometer on CT 1 will be referred to as CT1, that on CT 2 as CT2, that of CT 3 as CT3, that on CT 4 as CT4 and all the four radiometers together as CT.

4.1.1 CT's IR data processing

A python script was written to visualize and analyze the data from CT1, CT2, CT3 and CT4. Firstly the zenith angle (θ_Z) corrections where applied on the sky temperature readings of all four CT's as follows,

$$\theta_Z = 90 - \theta_e \qquad\qquad \theta_{CT1}, \theta_{CT2}, \theta_{CT3}, \theta_{CT4} \in \theta_e \tag{4.1}$$

also from the relationship in figure 3.2,

$$T(\theta_Z) = P_1 \sec(\theta_Z) + P_0 \tag{4.2}$$

such that

$$T(\theta_e) = P_1 \sec(\theta_e) + P_0 \tag{4.3}$$

at zenith, $\theta_e = 90°$ then according to equation 4.1, $\theta_Z = 0°$ such that equation 4.2 becomes,

$$T(0) = P_1 \sec(0) + P_0 = P_1 + P_0 \tag{4.4}$$

subtract equation 4.3 from equation 4.4,

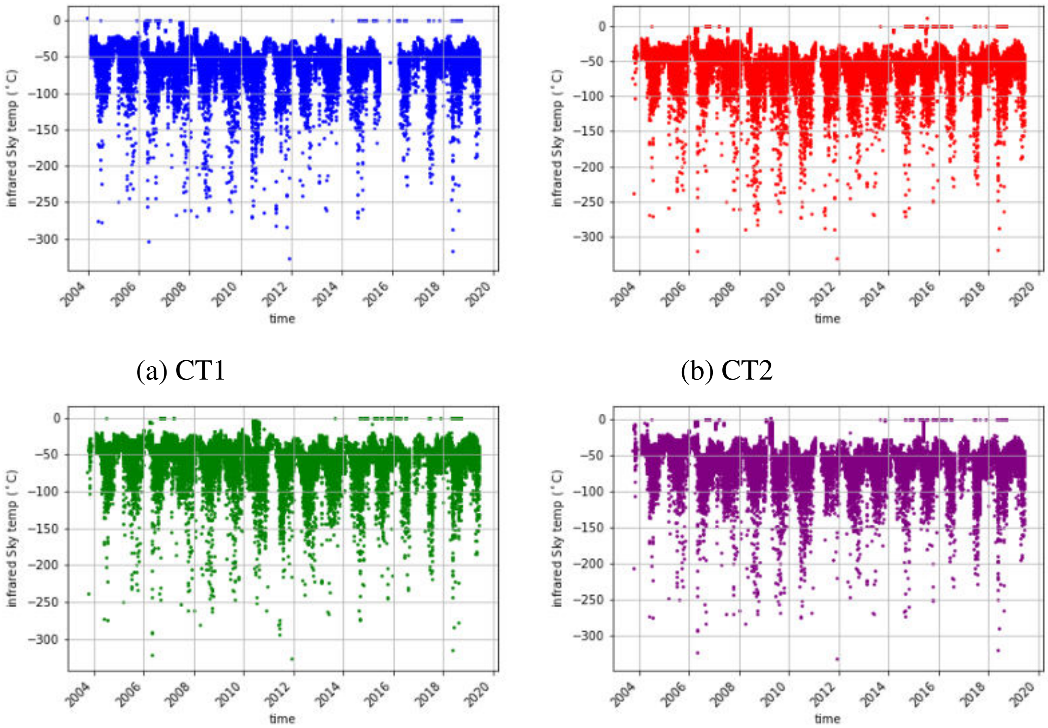
$$T(\theta_e) - T(0) = P_1 \sec(\theta_e) - P_1 \tag{4.5}$$

so that,

$$T(0) = P_1(1 - \sec(\theta_e)) + T(\theta_e) \tag{4.6}$$

where T(0) is the Sky temperature in the Zenith direction.

The graphs of sky temperature against time at zenith were then plotted for the four CT's as in figure 4.1.

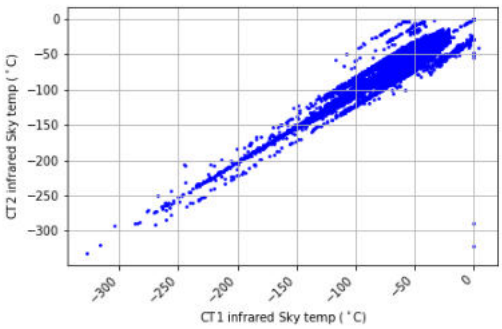


(a) CT1 (b) CT2

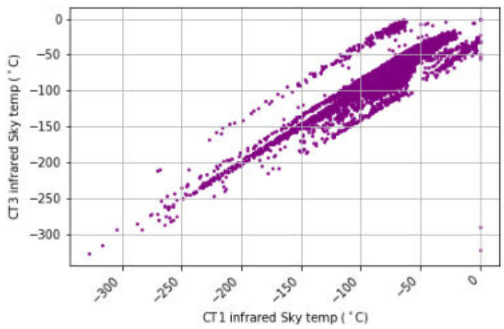
(c) CT3 (d) CT4

Figure 4.1: Infrared sky temperature as measured by all four CT radiometers. The sky temperatures measured by all CTs ranges from $\approx -300°C$ to around $0°C$. The period during which data was recorded across all the CT radiometers is also consistent, ranging from 2004 to 2019.

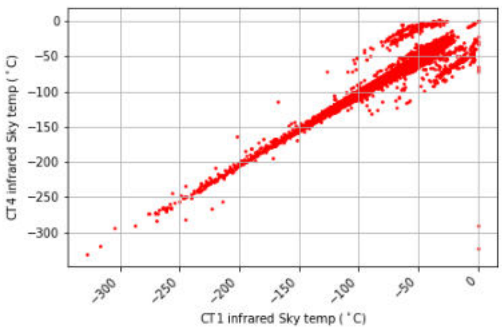
As can be seen from figure 4.1, data from all four CT's have a maximum threshold of around $0°C$ and a minimum of $-300°C$, thus indicating a consistency between the four plots already. In order to further determine if all the CTs measure the same sky temperature and measure the same point in the sky, they were plotted against each other as follows, CT1 against CT2, CT1 against CT3, CT1 against CT4, CT2 against CT3, CT2 against CT4 and CT3 against CT4, as shown in figure 4.2.



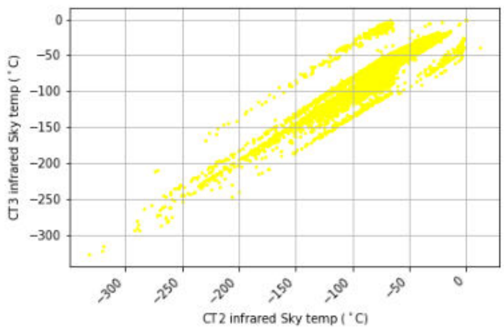
(a) CT1 against CT2



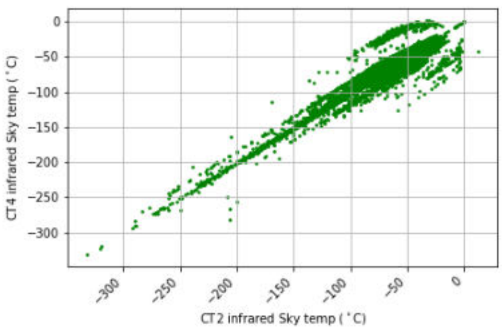
(b) CT1 against CT3



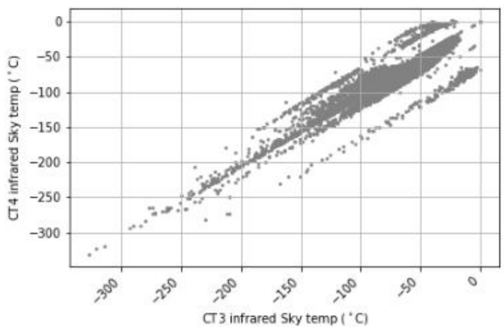
(c) CT1 against CT4



(d) CT2 against CT3



(e) CT2 against CT4

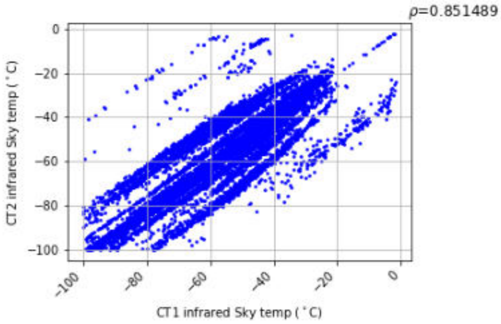


(f) CT3 against CT4

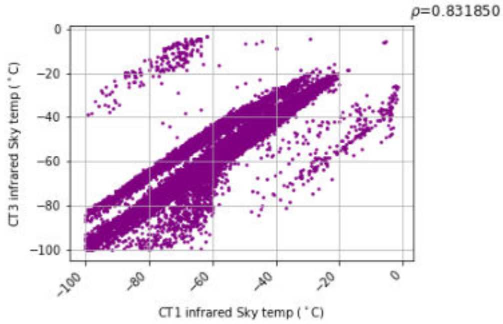
Figure 4.2: Sky temperatures from all four CT radiometers plotted against each other.

As can be seen in figure 4.2, a consistent linear relationship is common in all graphs, with few outliers that are also common. However figure 4.2 also show a few inconsistent outliers at 0°C or greater, these points are not not consistent throughout all the four CT IR radiometers and would thus be disregarded. There is also data points less then -273.15°C, these points would be disregarded as they fall below absolute zero and thus would be considered unphysical. Furthermore as discussed under the instrument description, the CT radiometers all have a temperature range between -100°C to 3000°C, thus the radiometers would not accurately record temperatures below -100°C and thus they were also disregarded. For those reasons altogether, the points below

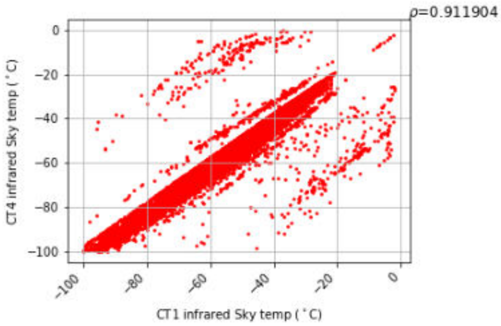
$-100°C$ and those equal or above $0°C$ were flagged as shown in Figure 4.3. The CT



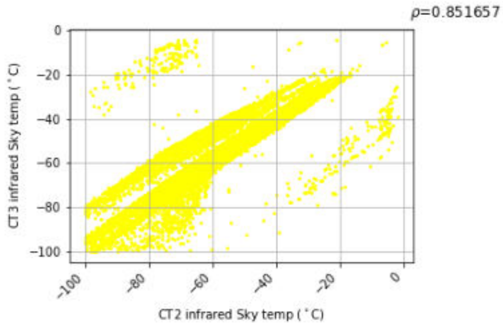
(a) CT1 against CT2



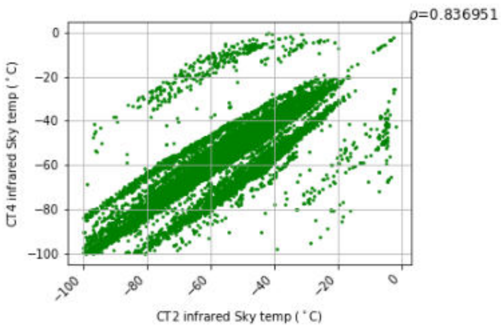
(b) CT1 against CT3



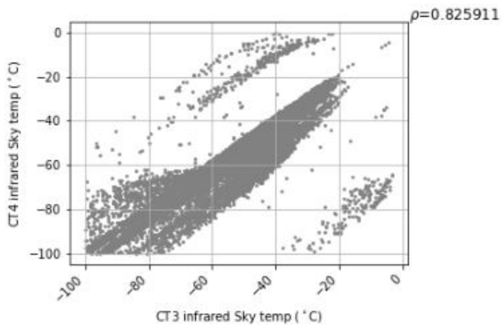
(c) CT1 against CT4



(d) CT2 against CT3



(e) CT2 against CT4



(f) CT3 against CT4

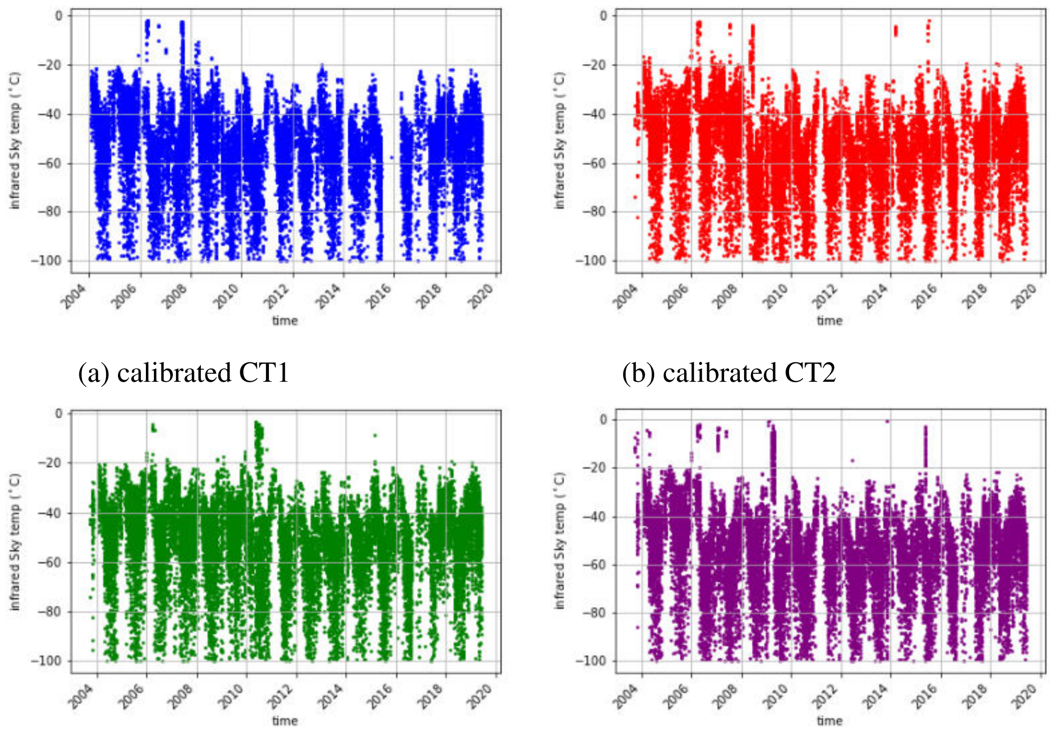
Figure 4.3: Infrared sky temperature as measured by CT radiometers after flagging of data. The Pearson coefficient is also displayed for all plots showing a high positive correlation of atleast a minimum 0.82.

graphs in Figure 4.3 all still show a consistent linear relationship with a few data points that are offset from the main linear trend of points. These points would not be flagged as they are also linearly correlated and consistent through the four CT IR radiometers. The correlation was then calculated using,

$$\rho = \frac{cov(x,y)}{\sigma_x \sigma_y} = \frac{\overline{xy} - \bar{x}\bar{y}}{\sigma_x \sigma_y} \tag{4.7}$$

where ρ is the Pearson's correlation coefficient [33] and x and y are the CT's sky temperature data with σ_x and σ_y their respective standard deviations. The correlation coefficient would indicate if the measurements are the same and if the CT see and measure the same sky. All the CT permutations gave a positive correlation and all show a high correlation with a minimum of 0.82. This indicates the CT's do measure the same sky and have and almost have same data points.

The graphs of the CTs sky temperature was then plotted to see if they varied by much, and as can be seen in Figure 4.4 all CT graphs are similar.



(a) calibrated CT1

(b) calibrated CT2

(c) calibrated CT3

(d) calibrated CT4

Figure 4.4: calibrated Infrared sky temperature as measured by CT radiometers. After flagging of unwanted data, this is how the sky temperature data from all CT's look like.

Finally the average sky temperature for the CT's was computed with its standard deviation by considering the following:

Since the CTs all record a sky temperature at a certain time then each CT can be represented in terms of its corresponding sky temperature such that CTn =$T_{skyt}n$ where $n = 1,2,3$ or 4. Then the mean [33] of the sky temperature can be calculated for all CTs at that certain time as,

$$\bar{T}_{Skyt} = \frac{1}{n}\sum_{1}^{4} T_{skyt}n \tag{4.8}$$

and the standard deviation [33] can be calculated as,

$$\sigma_{T_{skyt}} = \sqrt{\frac{1}{n-1}\sum_{1}^{4}(T_{skyt}n - \bar{T}_{skyt})^2} \tag{4.9}$$

this yields the mean sky temperature (4.8) and its standard deviation (4.9) $(\bar{T}_{sky}, \sigma_{T_{skyt}})$, which was plotted with respect to time to give figure 4.5.

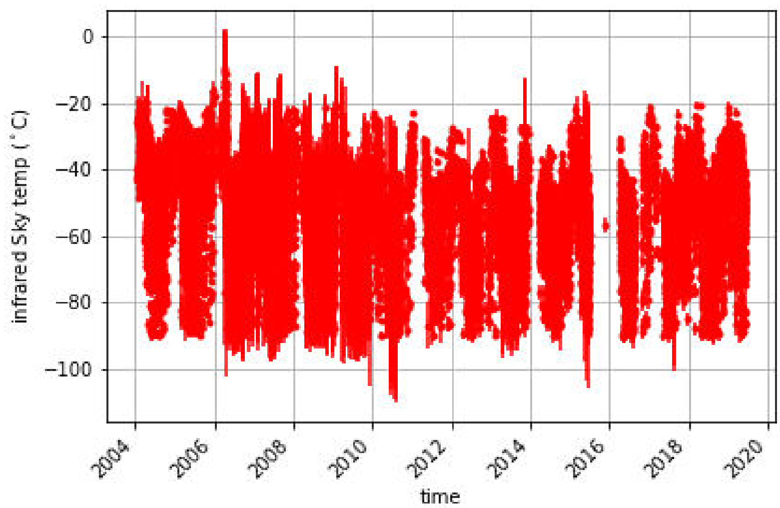


Figure 4.5: The mean sky temperature as measured by the CTs. The scatter plot also has the standard deviation in sky temperature (error bars) as calculated using equation 4.9.

4.2 Data from the Autonomous Tool for Measuring Observatory Site COnditions PrEcisely (Atmoscope)

The Atmoscope at the H.E.S.S. site has been operational between 2011 and 2018, recording at least 6 years of data. The Atmoscope takes a reading on average of 1 minute and records throughout the day as well as the night.

As discussed in the instrument section 3.2, the Atmoscope measures the ambient temperature, sky temperature and then calculates cloud height at the specific time the measurement was taken. Although the Atmoscope does not record the sky temperature, it records the ambient temperature and the cloud height which was converted to the sky temperature using equation 3.1. The sky temperature was then plotted with respect to time from 2011 to 2017 as in figure 4.6.

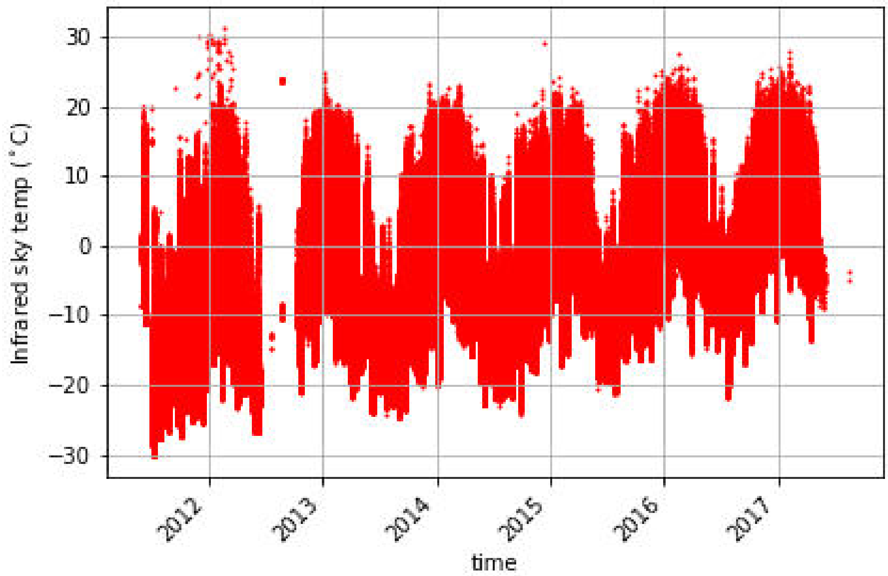


Figure 4.6: Sky temperature as given by the Atmoscope. An upward trend from year to year is clearly visible in the data.

The sky temperature clearly lies between $-30°C$ and $30°$ which is well in the sensitive range of Atmoscope's temperature sensor. There is however a visible upward trend in the sky temperature data that suggest an increase in sky temperature from year to year. It is not yet known if this yearly upward trend was a true natural phenomenon which is truly reflected by the data in figure 4.6. Explaining this trend is beyond the scope of this study and for that reason the data was kept and used as it is in figure 4.6.

4.3 Data from the National Aeronautics and Space Administration (NASA) AErosol RObotic NETwork (AERONET) station

The AERONET directly calculates and records the PWV as given by the equation 3.5 in section 3.3. The AERONET recorded a single PWV value randomly within a time difference ranging from 1 minute to 5 minutes.

The PWV and its corresponding time was then extracted from its data file and plotted as in Figure 4.7.

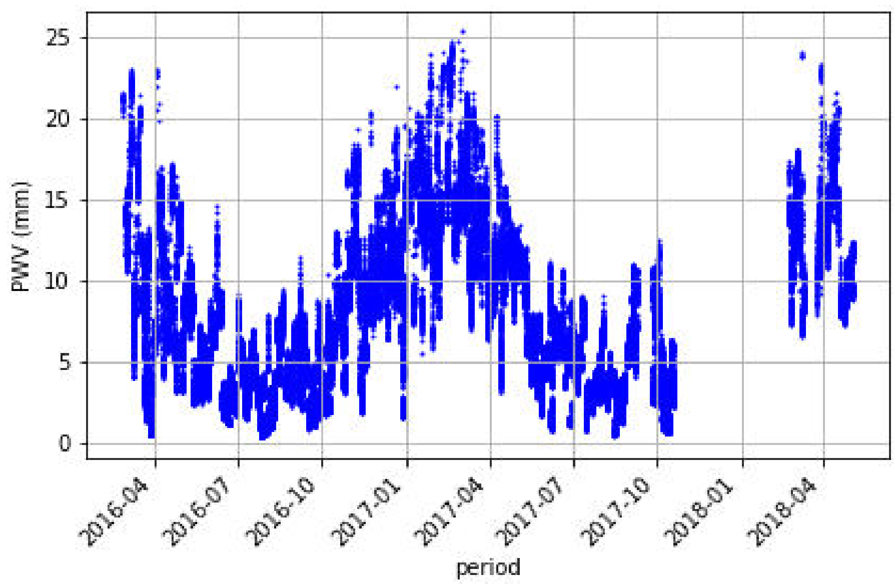


Figure 4.7: The PWV as measured by the AERONET station. A clear visible seasonal trend in the PWV data can be seen, with the PWV dropping in the winter months and peaking in the summer months. The figure also shows a period in 2018 where PWV was not recorded.

The AERONET data plot shows that the PWV was taken between 2016 and 2018, and has a few months in 2018 without any PWV measurements. Figure 4.7 shows a clear seasonal variation trend as the daily PWV drops in the winter periods from April to October and rises from October to April.

4.4 DATA callibration

The sky temperature of both CT and the Atmoscope was converted into PWV through the means of calibration and finding the relationships between the data from both instruments. The PWV given by the CT is what we desire to get and use in determining the PWV at the H.E.S.S. site and Mt Gamsberg for the following reasons. Firstly, CT data was taken for longer periods and will show in depth seasonal variations of PWV at the H.E.S.S. site. Secondly the results of CT will also make for good comparison with the results of 1994-1995, since those values were taken during the night and the CT radiometers only measures the sky temperature through the night as H.E.S.S. observes. Thirdly, from the three equipment, only the CT data was more consistent, with the AERONET not recording for a few months and with the Atmoscope sky temperature having an un-explainable seasonal upward trend. Lastly, the CT sky temperature data is given as mean of the four CT sky temperature which would make it more accurate then the data from the other two equipment.

In order to get the CT PWV from CT sky temperature $(\bar{T}_{skyt}, \sigma_{T_{skyt}})$, two calibration processes were conducted. Firstly, taking advantage of the fact the AERONET only takes PWV measurements during the day time and the Atmoscope takes sky temperature measurements during both day and night, an overlap of data recorded at the same time during the day by the AERONET and the Atmoscope was found. By means of cross calibration and finding the relationship between the two equipment, all the sky temperature of the Atmoscope was converted into PWV, which is PWV for both day and nigh.

Secondly, the converted PWV of the Atmoscope was then used in getting the CT PWV by finding the data recorded at the same time during the night time by CT (sky temperature) and the Atmoscope (converted PWV). This data was then cross calibrated to get the relationship between the instruments which was then used to convert the CT sky temperature into PWV.

4.4.1 Atmoscope (sky temperature) and AERONET (PWV)

In the first step of the cross calibration process, the day time sky temperature from the Atmoscope was plotted against the corresponding day time PWV from the AERONET. Figure 4.8 shows a plot of Aeronet PWV against Atmoscope sky temperature.

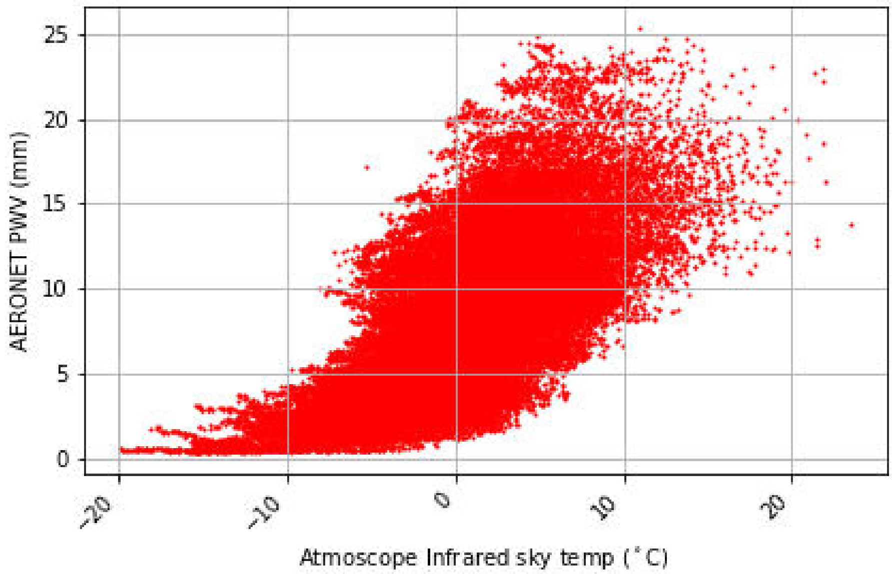


Figure 4.8: Day time AERONET PWV against corresponding day time Atmoscope sky temperature. The data from the two instruments has an exponential fit.

As can be seen from Figure 4.8 there is an exponential relationship between the data given by the two instruments, thus an exponential fit was applied to the graph as in figure 4.9.

The exponential fit line is described by the relation,

$$y = 8.074815\, e^{0.071097x} \tag{4.10}$$

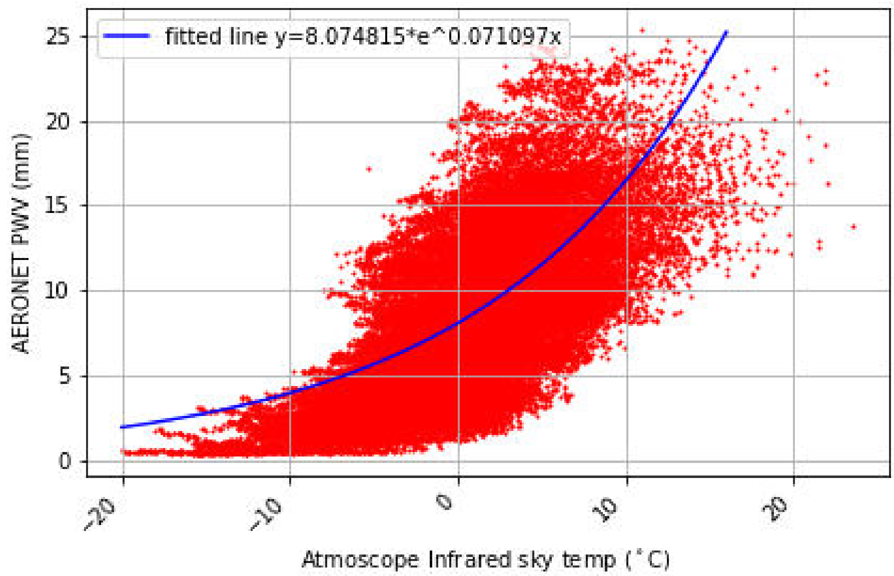


Figure 4.9: The exponential fit to the AERONET and Atmoscope data. The fit in blue is given by the relation of equation 4.10.

where y is the PWV in millimeters given by the AERONET and x is the sky temperature in degree Celsius given by the Atmoscope. Hence all sky temperature from an Atmoscope can now be converted into PWV as it would be given by the AERONET using the relationship in equation 4.10.

The sky temperature of the Atmoscope including that of the night was then converted to PWV, giving day and night PWV for the Atmoscope as in Figure 4.10.

4.4.2 Atmoscope (PWV) and CT (sky temperature)

In the second step of calibration, the night time PWV from the Atmoscope data was used to find the relation between converted Atmoscope PWV and CT temperature. This relations was then used to convert the CT sky temperature into CT PWV data, therefore giving the night time CT data at the H.E.S.S site. As done with the Atmoscope and the Aeronet data, the Atmoscope night time PWV was also plotted against the the corresponding CT sky temperature and its standard deviation data as in Figure 4.11.

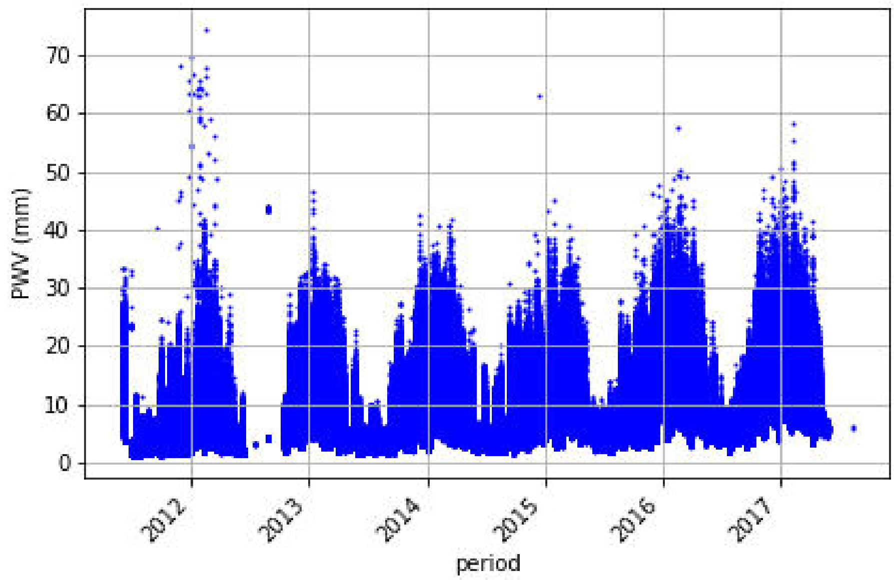


Figure 4.10: The Atmoscope PWV as given by the relation in equation 4.10.

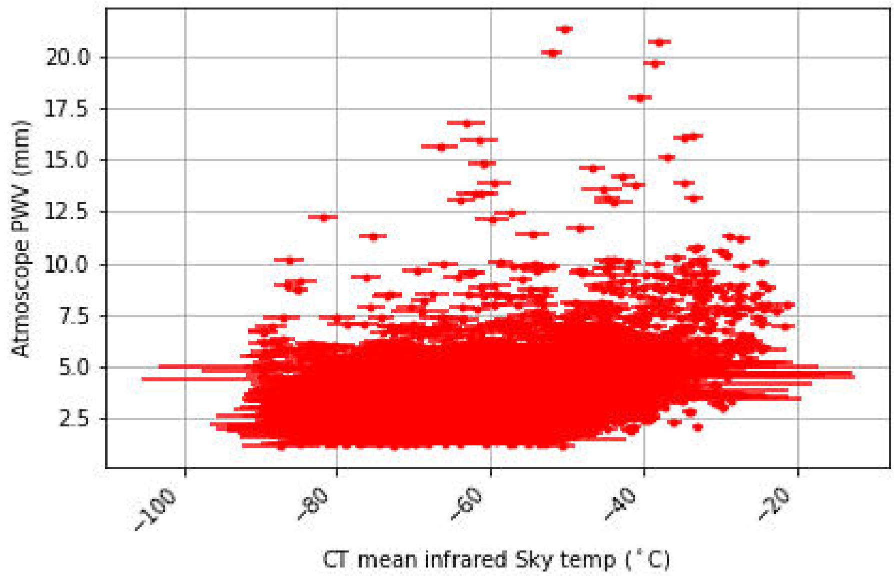


Figure 4.11: Night time CT sky temperature against night time Atmoscope PWV.

As can be seen from Figure 4.11, there is a few outliers but there is a definite exponential trend in the data with a slight rise starting around the $-40°$ mark of the sky temperature. An exponential fit was applied to the data as in figure 4.12 with the fit governed by the relationship,

$$y = 8.394457\, e^{0.010886x} \tag{4.11}$$

where y is the PWV given by the Atmoscope and x the sky temperature given by the CTs.

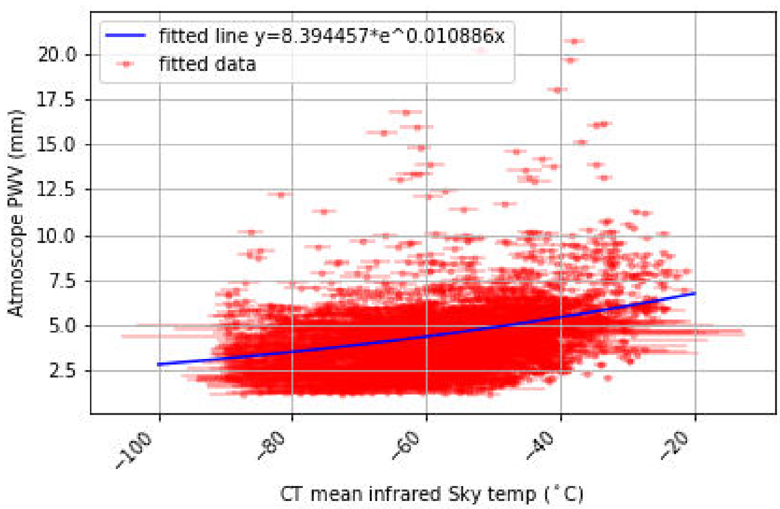


Figure 4.12: Atmoscope PWV against CT sky temperature fit. The fit is given by relation in equation 4.11.

The relation in equation 4.11 was then used to give the night time PWV as given by the night time CT sky temperature. Error propagation was then conducted to convert the sky temperature standard deviation to the PWV standard deviation as follows. Equation 4.11 gives the following,

$$PWV = 8.394457e^{0.010886\bar{T}_{skyt}} \tag{4.12}$$

Then PWV is function of the mean sky temperature from the CTs with its uncertainty,

$$PWV = f(\bar{T}_{skyt}) = f(\bar{T}_{skyt} \pm \sigma_{T_{skyt}}) \tag{4.13}$$

where $\sigma_{T_{skyt}}$ is the error on the sky temperature $\bar{T}_{skyt}$ as given by equation 4.9 and 4.8, respectively. Then from error propagation [34],

$$\sigma_{PWV} = \sqrt{\left(\frac{\partial PWV}{\partial \bar{T}_{skyt}}\right)^2 \sigma^2_{T_{skyt}}} \tag{4.14}$$

taking the partial derivative of equation 4.12 with respect to $\bar{T}_{skyt}$,

$$\begin{aligned}\frac{\partial PWV}{\partial \bar{T}_{skyt}} &= \frac{\partial(8.394457e^{0.010886\bar{T}_{skyt}})}{\partial \bar{T}_{skyt}} \\ &= (8.394457)(0.010886)\,e^{0.010886\bar{T}_{skyt}} \\ &= 0.091382e^{0.010886\bar{T}_{skyt}}\end{aligned} \tag{4.15}$$

replacing it in equation 4.14,

$$\begin{aligned}\sigma_{PWV} &= \sqrt{\left(0.091382e^{0.010886\bar{T}_{skyt}}\right)^2 \sigma^2_{T_{skyt}}} \\ &= \left(0.091382e^{0.010886\bar{T}_{skyt}}\right)\sigma_{T_{skyt}}\end{aligned} \tag{4.16}$$

Giving the PWV and its error ($PWV \pm \sigma_{PWV}$). The desired graph containing the night time PWV with its error from 2004 to 2019 was then plotted as in figure 4.13.

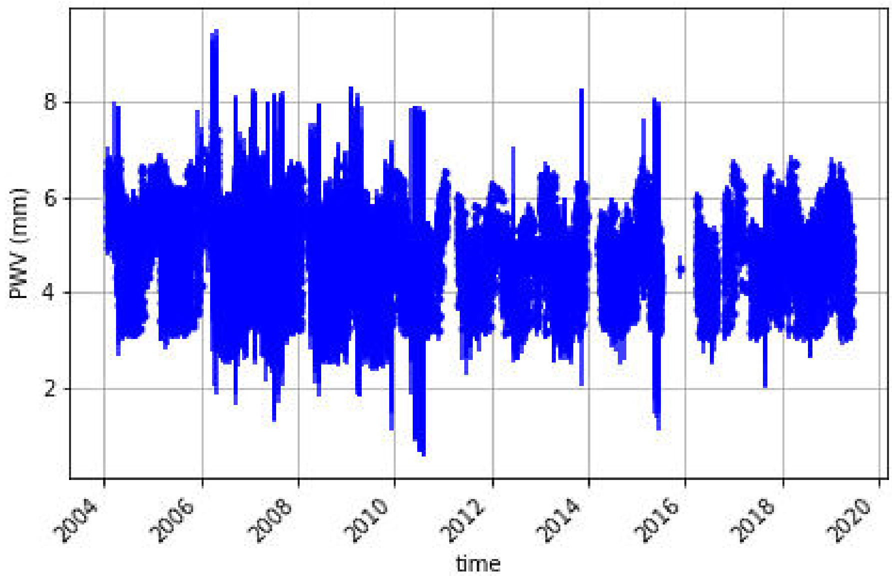


Figure 4.13: The PWV as given by CT sky temperature at the H.E.S.S. site. The relation in equation 4.11 to convert the sky temperature with the error propagation in equation 4.16 to give PWV error.

As can be seen from figure 4.13, the night time PWV lies between 2 mm and 8 mm which is more or less consistent with data from the CTs. In order to further evaluate this data for further information, monthly means would be computed.

This two step calibration process yielded two important interchangeable relationships between sky temperature and PWV among the three different equipment.

4.5 CT PWV Processing

Since the CTs yielded PWV data from the H.E.S.S. site, another python script was written to scale the PWV at the H.E.S.S. site for what it would be at Mt Gamsberg and compute the frequency distribution, evaluate the monthly mean PWV and compute the seasonal variations over the 16 year period for both H.E.S.S. site and Mt Gamsberg.

Using the scaling function in equation 2.10 and the H.E.S.S. PWV as P in equation 2.9, the scaled PWV for Mt Gamsberg was found as P_G and yielded the plot in Figure 4.14.

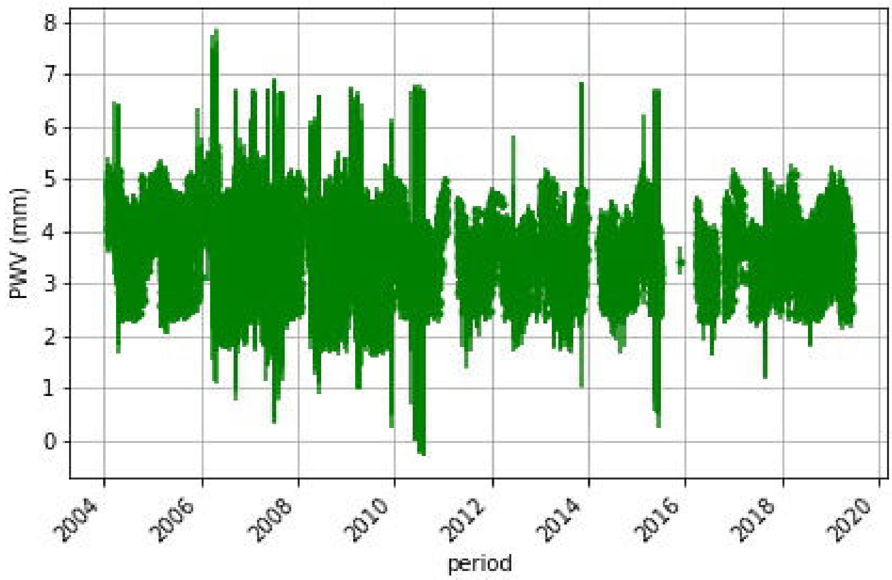


Figure 4.14: The PWV at Mt Gamsberg as scaled from the H.E.S.S. site PWV by using the scaling function.

Given that the PWV data for both sites was obtained, the relative frequency distribution of all the PWV at both sites was then computed in bins of 0.5 mm in order to see the distribution of PWV on a smaller scale, this can be observed in Figure 5.1 and Figure 5.4.

Since H.E.S.S. only operates from sunset to sunrise (night time) and under cloudless skies, its amount of observations per month varies by season. Figure 4.15 shows the amount of monthly recording taken in 2004 by the CT radiometers.

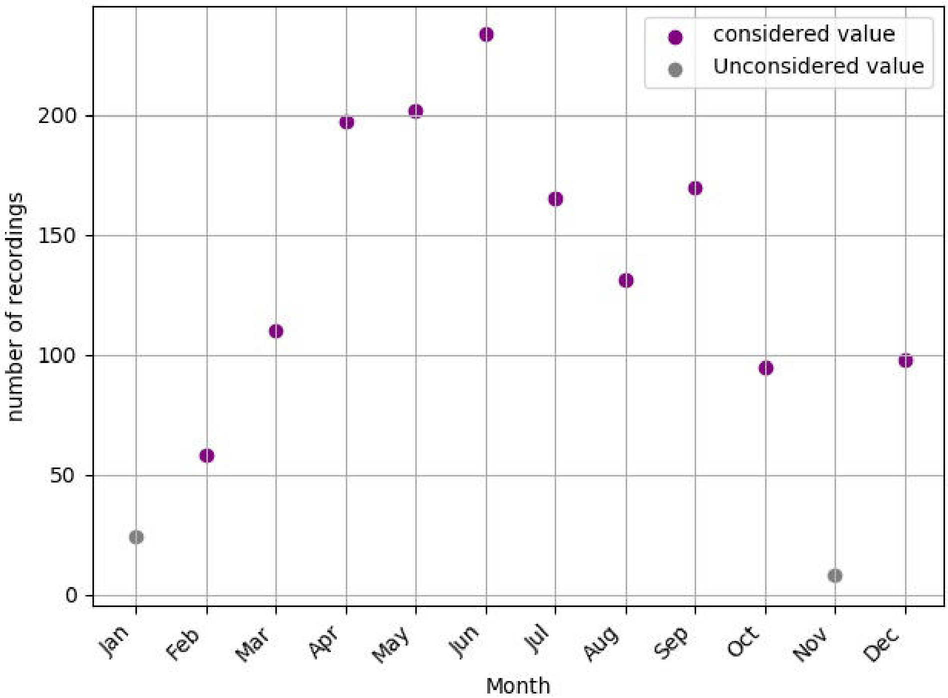


Figure 4.15: Number of considered points for each month and discarded points. Only means of months having at least $\frac{1}{5}$ of possible observations were calculated.

As can be seen, more observations took place in the winter months then in the summer period, this was because more cloudless skies occurred in the winter period.

Weighted mean PWV values were considered and calculated for only months having 50 or more observations. The reason being that a maximum of approximately 250 observations was recorded in the winter months of each year when the weather conditions were more favorable. The 50 observations account for at least $\frac{1}{5}$th of the maximum amount of observation that can be done in a single month. If the number of data recordings in a month was less than 50, the data points were disregarded and the weighted mean for the specific month was not computed.

The weighted mean PWV ($\overline{P_H}$) at the H.E.S.S. site was then computed for each month having 50 or more recordings for each year from 2004 to 2019 with the following formula [33] in equation 4.17. Let $PWV = P$ and $\sigma_{PWV} = \sigma_P$ then,

$$\overline{P} = \frac{\sum_i P_i/\sigma^2_{P_i}}{\sum_i 1/\sigma^2_{P_i}} \tag{4.17}$$

This was then plotted on the same figure as in figure 5.2, in order to visualize and interpret how the PWV varies with seasons at the H.E.S.S. site. A weighted mean $\overline{Z_H}$ as in equation 4.17 for each specific month was then computed with the weights provided by the standard deviation σ_{P_H} of the weighted mean points $\overline{P_H}$, for each specific month. This was done in order to give each specific month a mean value regardless of year as in Figure 5.3.

The weighted mean PWV ($\overline{P_G}$) for each month having 50 or more data points was computed for the 16 year period for Mt Gamsberg using equation 4.17 to yield the results as in Figure 5.5. The weighted mean $\overline{Z_G}$ for each specific month was then computed using equation 4.17 with weights provided by the standard deviation σ_{P_G} of the weighted mean $\overline{P_G}$ of each specific month to give each specific month a mean value regardless of year as in figure 5.6.

The computed mean values at Mt Gamsberg in figure 5.5 were then plotted on the same axis (figure 5.8) with mean PWV values taken at Mt Gamsberg in 1994 and 1995 for comparison and to validate the computed values.

Chapter 5

Results and Discussions

5.1 H.E.S.S. site

As a result of the procedures in chapter 3, the following figures containing the results were obtained for the PWV at the H.E.S.S. site. Figure 5.1 shows the relative frequency distribution of PWV obtained at the H.E.S.S. site from 2004 to 2019 and figure 5.2 shows the seasonal variations of the PWV at the H.E.S.S. site over the same period, whilst figure 5.3 is showing the single monthly mean PWV for each specific month over the 16 year period.

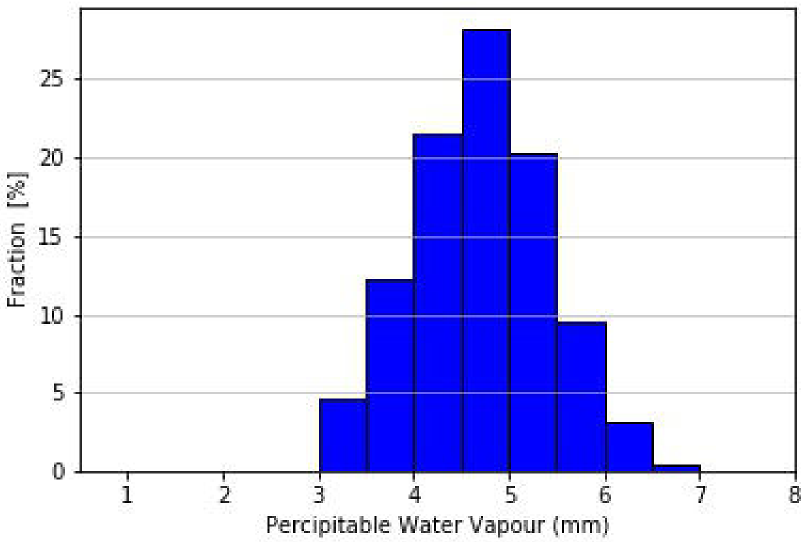


Figure 5.1: The relative frequency of PWV at the H.E.S.S. site. The *y*-axis shows the fraction of PWV out of 100% with the division of bins in steps 0.5 mm.

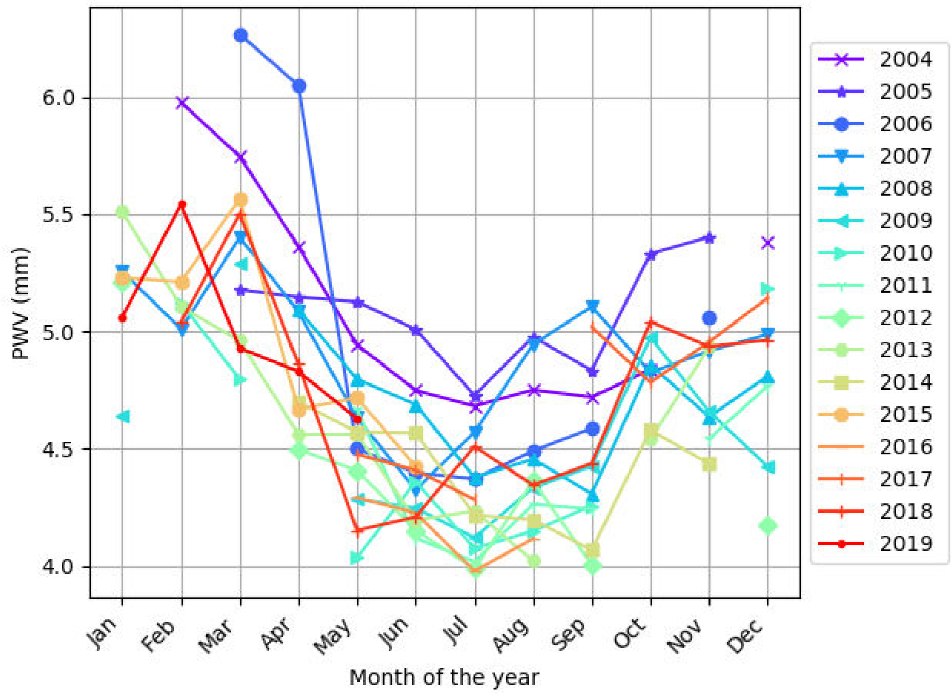


Figure 5.2: Monthly weighted mean PWV variations at the H.E.S.S. site from 2004 to 2019. The monthly PWV means have a similar trend were the PWV drops in the winter months and peaks in the summer months.

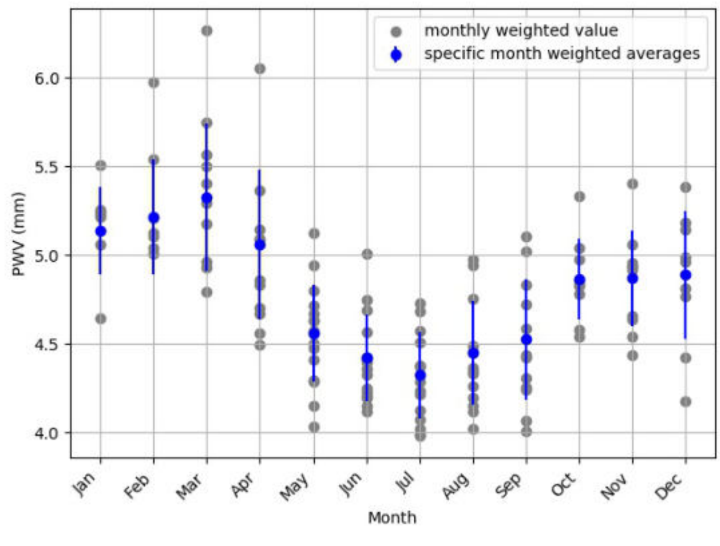


Figure 5.3: Single weighted mean PWV for each month at the H.E.S.S. site. The single weighted averages confirms the trend of the PWV dropping in the winter months and rising and peaking in the summer months.

The relative frequency distribution in figure 5.1indicates that a fraction of 28% of the PWV at the H.E.S.S. site is between 4.5 mm to 5 mm alone, and a fraction of 71% is between 4 mm and 5.5 mm. As can be seen from figure 5.2, for all 16 years the PWV is below 7 mm and well below 5 mm in the winter month. There is clear seasonal trend through the years in the data, with the PWV rising in the summer seasons and falling to its lowest in the winter months. These seasonal variation results are comparable to the seasonal variation results found at Chajnantor Plateau, with the difference being that the H.E.S.S. site PWV is a bit higher in winter and does not go below 3 mm as can be seen in figure 5.2 and is proven in Figure 5.1 and similarly this could also be seen in Figure 5.3. Furthermore, Figure 5.3 indicates that the PWV at the H.E.S.S. site fall below to the lowest of maximum of 4.5 mm in June, July and August which would make it the best period to observe if mm observations were to be conducted at the H.E.S.S. site.

5.2 Mt Gamsberg

The following results were obtained for Mt Gamsberg. Figure 5.4 is showing the relative frequency distribution of PWV at Mt Gamsberg from 2004 to 2019 and Figure 5.5 shows the seasonal variations of the PWV scaled for Mt Gamsberg whilst Figure 5.6 is showing the monthly mean PWV for each specific month over the 16 year period at Mt Gamsberg.

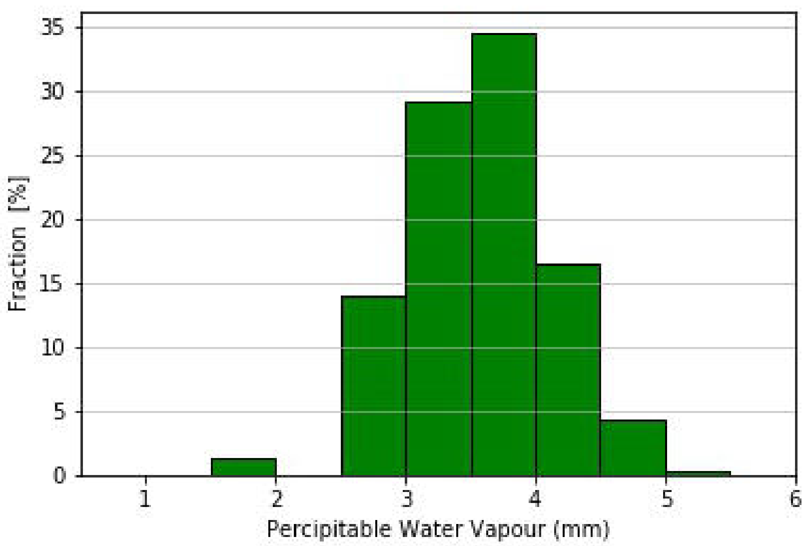


Figure 5.4: The PWV at Mt Gamsberg as scaled from the H.E.S.S. site PWV. The y-axis shows the fraction of PWV out of 100% with the division of bins in steps 0.5 mm.

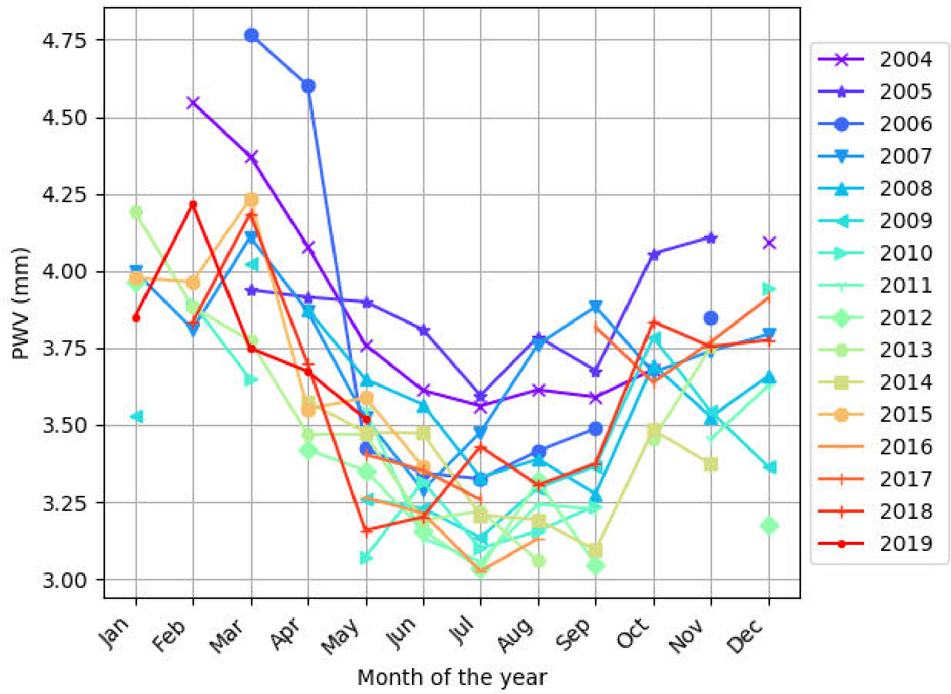


Figure 5.5: monthly weighted mean PWV variations at Mt Gamsberg from 2004 to 2019. The mean monthly at Mt Gamsberg show a maximum threshold of approximately 4.75 and a minimum of 3 mm. Clearly a trend can be seen in the data for all years with the PWV rising in the summer months and dropping in the winter months.

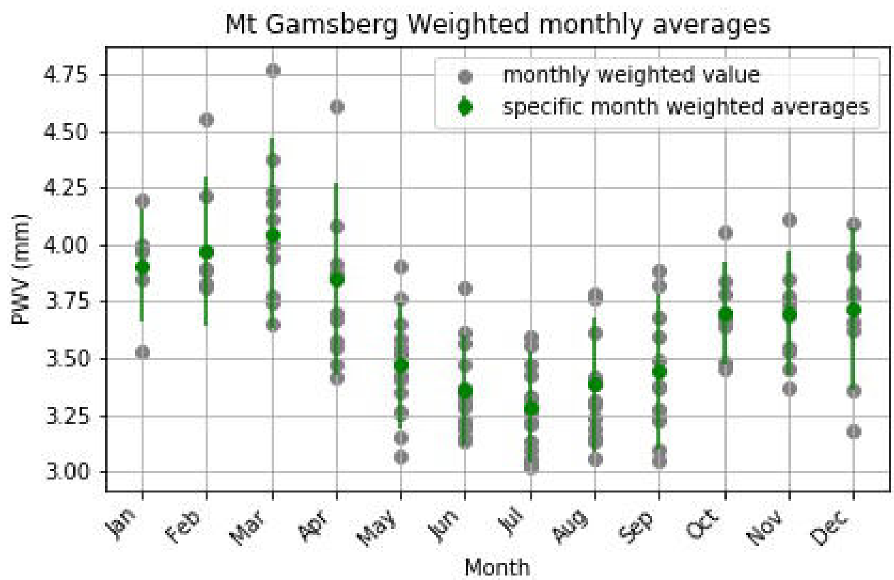
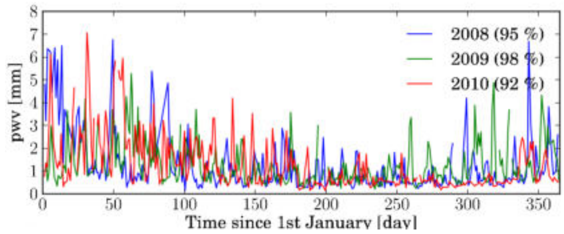


Figure 5.6: Single weighted mean PWV at the Mt Gamsberg site.

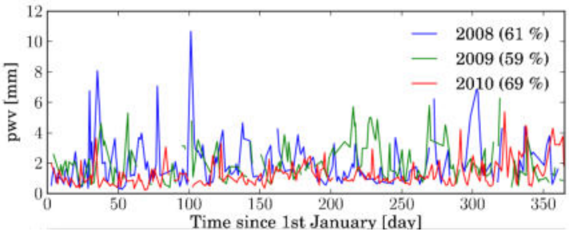
The relative frequency (Figure 5.4) indicate that a fraction 79% of the PWV lie between 1.5 mm - 4mm of which around 63% lie between 3 mm and 4 mm alone.

As is for the H.E.S.S. PWV, the PWV at Mt Gamsberg also shows a clear trend of seasonal variations, with the PWV occasionally going below 3.75 mm in the winter months, and as well rising in the summer season. The monthly mean PWV in figure 5.6 indicate that the best possible period for the AMT to conduct observations if it were to be build on Mt Gamsberg would be in May, June, July, August and September.

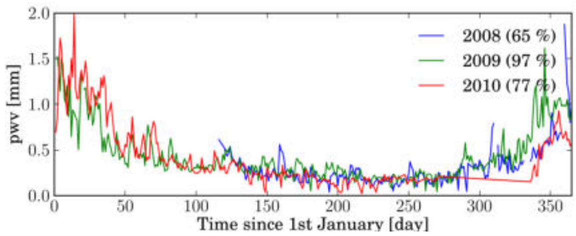
5.3 Comparison of PWV at Mt Gamsberg to some sites in the EHT



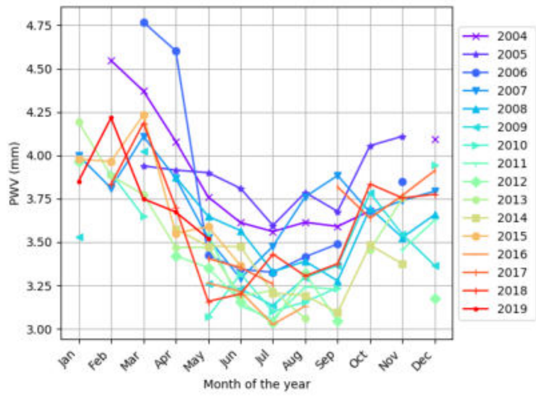
(a) Chajnantor Plateau



(b) Mauna Kea



(c) South Pole



(d) Mt Gamsberg

Figure 5.7: Seasonal variations in PWV at sites that conduct mm wave astronomy in comparison to Mt Gamsberg [19].

As can be seen in figure 5.7, the mean PWV results of Mt Gamsberg shows similarities in the seasonal variations when compared to the seasonal variations of the Chajnantor, Mauna Kea and South Pole with the PWV peaking in the summer period and dropping in the winter period across all four sites. The magnitude of the seasonal PWV at Mt Gamsberg is more comparable to that of the Chajnantor Plateau and Mauna Kea with Mauna Kea seemingly having higher PWV than that computed at Mt Gamsberg.

5.4 Mt Gamsberg PWV comparisons

Figure 5.8 shows the results of the mean PWV taken at Mt Gamsberg between 1994 and 1995 [22], and those that are computed in this thesis from 2004 to 2019.

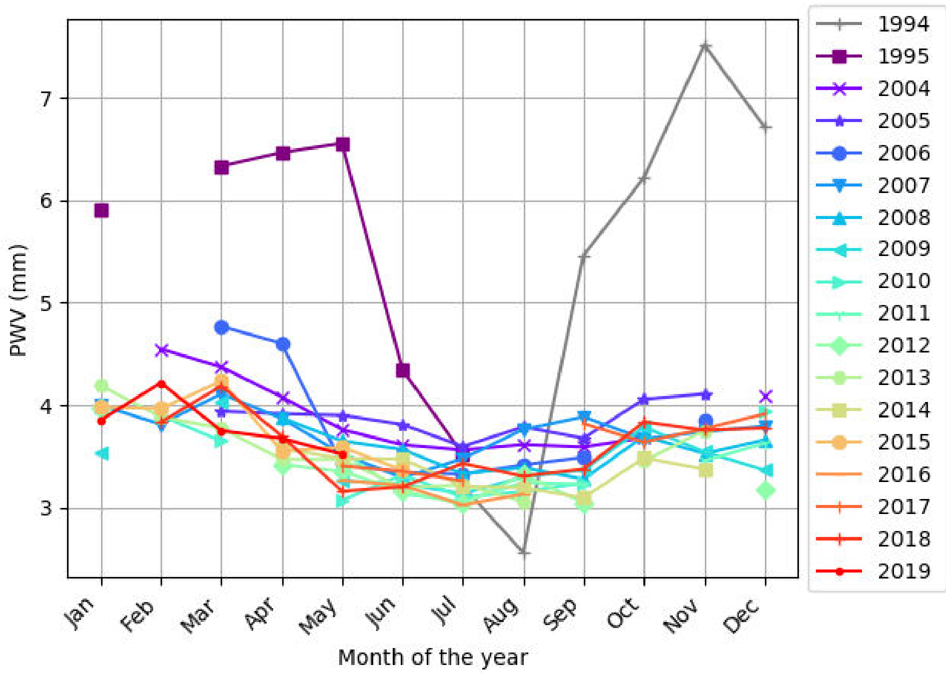


Figure 5.8: Mt Gamsberg computed monthly mean PWV results from 2004 to 2019 plus measured results of 1994 and 1995.

The computed PWV and the previously taken result all have the same trend, with the PWV rising in the summer period and dropping in the winter period of June, July and August. There is however a huge clear difference in the mean PWV taken between 1994 and 1995, with it being significantly higher above 5 mm in the summer period, whilst the PWV computed from 2004 to 2019 indicate PWV lower than 5 mm for the same period for all years.

The mean PWV results in this study were computed only by means of mathematical and statistical methods rather then directly measuring the PWV physically at Mt Gamsberg. Also, this results are based on an assumption made of the sky temperature being the same at both sites but in reality this may well vary at both sites. This can be seen in figure 5.8, which shows a difference in the actual data taken at Mt Gamsberg and those values calculated in this study. This could very well tell us that the sky temperature vary at the two sites. Thus these results cannot be considered as the true PWV conditions at Mt Gamsberg but rather as estimates of how the PWV conditions is likely to be at Mt Gamsberg given the weather conditions are the same as those at the H.E.S.S. site. Therefore, these results would have to be verified with actual PWV data readings with a radiometer at Mt Gamsberg.

Chapter 6

Conclusions and Recommendation

The results, indicate that both Mt Gamsberg and the H.E.S.S. site have PWV content suitable enough for mm wave observations. The mean PWV over the years show that multiple observations can be conducted in the mm wave regime at Mt Gamsberg, with PWV conditions more suitable for high frequency observation in the winter period as it occasionally drops below 3.5 mm and for lower frequency observation throughout the year as the mean PWV over Mt Gamsberg has a maximum PWV level of 5 mm. Moreover, the PWV result of Mt Gamsberg was found to be comparable to that of the Chajnantor Plateau and Mauna Kea which host telescopes that conduct mm wave astronomy and partake in EHT observations. This would indicate that Mt Gamsberg would make an ideal site to conduct mm wave astronomy. Furthermore, Mt Gamsberg is found to have the same seasonal variation trends in PWV as that of the other three sites (South Pole, Chajnantor Plateau and Mauna Kea) having telescopes which partake in EHT observations, which indicates that the AMT if build on Mt Gamsberg can as well observe with the EHT during this period.

These results and those obtained in 1994 and 1995 at Mt Gamsberg, all indicate Mt Gamsberg has the potential to host the AMT, and all indicate that if the AMT is to be hosted at Mt Gamsberg, it will also make it possible for it to observe in an interferometer with the EHT seasonally, in the winter periods. However, since the EHT observes specifically in April, the PWV at Mt Gamsberg PWV would not be suitable during this period.

The results however when compared with those of 1994 and 1995 show that the computed results may not accurately give the true reflection of PWV at Mt Gamsberg and thus the clear out-most recommended way to get accurate results would be for a radiometer to be set up on Mt Gamsberg and directly take PWV measurements.

Bibliography

[1] Backes M, Müller C, Conway J E, Deane R, Evans R, Falcke H, Fraga-Encinas R, Goddi C, Klein Wolt M and Krichbaum T P 2016 The Africa Millimetre Telescope *Proceedings of the 4th Annual Conference on High Energy Astrophysics in Southern Africa (HEASA 2016). 25-26 August* p 29

[2] Mizuno Y, Younsi Z, Fromm C M, Porth O, De Laurentis M, Olivares H, Falcke H, Kramer M and Rezzolla L 2018 *Nature Astronomy* **2** 585–590 (*Preprint* `1804.05812`)

[3] Smette A, Horst H and Navarrete J 2008 Measuring the amount of precipitable water vapour with visir *The 2007 ESO Instrument Calibration Workshop* (Springer) pp 433–438

[4] Querel R R and Kerber F 2014 All-sky homogeneity of precipitable water vapour over Paranal *Ground-based and Airborne Instrumentation for Astronomy V* (*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* vol 9147) p 914792 (*Preprint* `1408.1753`)

[5] Chaisson E and McMillan S 2014 *Astronomy Today*

[6] Verhoeven G 2017 *AARGNews* **55** 13–18

[7] Condon J J and Ransom S M 2016 *Essential Radio Astronomy*

[8] Wilson T L, Rohlfs K and Hüttemeister S 2013 *Tools of Radio Astronomy*

[9] Vogel M 2017 *Contemporary Physics* **58** 193–193

[10] Radio telescope accessed 2019-11-01 URL http://abyss.uoregon.edu/~js/glossary/radio_telescope.html

[11] Mac-Auliffe F 2001 *SEST Parameters* Accessed 2019-09-26 URL http://www.apex-telescope.org/sest/html/telescope-instruments/telescope/index.html

[12] Zodet H 2009 *SEST at night* Accessed 2019-09-26 URL https://www.eso.org/public/images/esopia00050teles/

[13] Sault R J, Carrad G J, Hall P J and Crofts J 2006 *arXiv e-prints* astro-ph/0701016 (*Preprint* astro-ph/0701016)

[14] Maoz D 2007 *Astrophysics in a Nutshell*

[15] Furtak O 2017 *The Event Horizon Telescope and Global mm-VLBI Array on the Earth* Accessed 2019-09-26 URL https://www.eso.org/public/images/ann17015a/

[16] Event Horizon Telescope Collaboration, Akiyama K, Alberdi A, Alef W, Asada K, Azulay R, Baczko A K, Ball D, Baloković M, Barrett J, Bintley D, Blackburn L, Boland W, Bouman K L, Bower G C, Bremer M, Brinkerink C D, Brissenden R, Britzen S, Broderick A E, Broguiere D, Bronzwaer T, Byun D Y, Carlstrom J E, Chael A, Chan C k, Chatterjee S, Chatterjee K, Chen M T, Chen Y, Cho I, Christian P, Conway J E, Cordes J M, Crew G B, Cui Y, Davelaar J, De Laurentis M, Deane R, Dempsey J, Desvignes G, Dexter J, Doeleman S S, Eatough R P, Falcke H, Fish V L, Fomalont E, Fraga-Encinas R, Freeman W T, Friberg P, Fromm C M, Gómez J L, Galison P, Gammie C F, García R, Gentaz O, Georgiev B, Goddi C, Gold R, Gu M, Gurwell M, Hada K, Hecht M H, Hesper R, Ho L C, Ho P, Honma M, Huang C W L, Huang L, Hughes D H, Ikeda S, Inoue M, Issaoun S, James D J, Jannuzi B T, Janssen M, Jeter B, Jiang W, Johnson M D, Jorstad S, Jung T, Karami M, Karuppusamy R, Kawashima T, Keating G K, Kettenis M, Kim J Y, Kim J, Kim J, Kino M, Koay J Y, Koch P M, Koyama S, Kramer M,

Kramer C, Krichbaum T P, Kuo C Y, Lauer T R, Lee S S, Li Y R, Li Z, Lindqvist M, Liu K, Liuzzo E, Lo W P, Lobanov A P, Loinard L, Lonsdale C, Lu R S, MacDonald N R, Mao J, Markoff S, Marrone D P, Marscher A P, Martí-Vidal I, Matsushita S, Matthews L D, Medeiros L, Menten K M, Mizuno Y, Mizuno I, Moran J M, Moriyama K, Moscibrodzka M, Müller C, Nagai H, Nagar N M, Nakamura M, Narayan R, Narayanan G, Natarajan I, Neri R, Ni C, Noutsos A, Okino H, Olivares H, Oyama T, Özel F, Palumbo D C M, Patel N, Pen U L, Pesce D W, Piétu V, Plambeck R, PopStefanija A a, Porth O, Prather B, Preciado-López J A, Psaltis D, Pu H Y, Ramakrishnan V, Rao R, Rawlings M G, Raymond A W, Rezzolla L, Ripperda B, Roelofs F, Rogers A, Ros E, Rose M, Roshanineshat A, Rottmann H, Roy A L, Ruszczyk C, Ryan B R, Rygl K L J, Sánchez S, Sánchez-Arguelles D, Sasada M, Savolainen T, Schloerb F P, Schuster K F, Shao L, Shen Z, Small D, Sohn B W, SooHoo J, Tazaki F, Tiede P, Tilanus R P J, Titus M, Toma K, Torne P, Trent T, Trippe S, Tsuda S, van Bemmel I, van Langevelde H J, van Rossum D R, Wagner J, Wardle J, Weintroub J, Wex N, Wharton R, Wielgus M, Wong G N, Wu Q, Young A, Young K, Younsi Z, Yuan F, Yuan Y F, Zensus J A, Zhao G, Zhao S S, Zhu Z, Farah J R, Meyer-Zhao Z, Michalik D, Nadolski A, Nishioka H, Pradel N, Primiani R A, Souccar K, Vertatschitsch L and Yamaguchi P 2019 **875** L4 (*Preprint* `1906.11241`)

[17] Wiedner M C 1998 *Atmospheric Water Vapour and Astronomical Millimetre Interferometry* Ph.D. thesis University of Cambridge

[18] Arizona Radio Observatory 2011 *what is submillimeter astronomy* Accessed 2019-09-26 URL `http://aro.as.arizona.edu/docs/what_is_submillimeter.htm`

[19] Tremblin P, Schneider N, Minier V, Durand G A and Urban J 2013 A worldwide comparison of the best sites for submillimetre astronomy *Astrophysics from Antarctica* (*IAU Symposium* vol 288) ed Burton M G, Cui X and Tothill N F H pp 29–33

[20] The Gamsberg Mountain accessed 2019-10-21 URL `https://www.namibiatourism.com.na/blog/Four-Scenic-4x4-Mountain-Passes-in-Namibia`

[21] Sarazin M 1994 *Part I: General Climatology, Technical notes on Gamsberg Namibian site* Accessed 2019-06-12 URL `http://www.eso.org/gen-fac/pubs/astclim/espas/gamsberg/`

[22] Sarazin M 1995 *Final Summary Report: Environmental conditions on Potential Observatories, Gamsberg Astroclimatological Summary Report VLT.TRE.ESO.17400* Accessed 2019-06-12 URL `http://www.eso.org/gen-fac/pubs/astclim/espas/gamsberg/`

[23] High Energy Stereoscopic System accessed 2019-10-07 URL `https://www.mpi-hd.mpg.de/hfm/HESS/pages/about/`

[24] Wilkinson P 2015 *Literature survey on atmospheric opacities/brightness temperatures at mm and infra-red wavelengths.* Private communication

[25] Otarola A C, Querel R and Kerber F 2011 *arXiv e-prints* arXiv:1103.3025 (*Preprint* `1103.3025`)

[26] Aye K M, Chadwick P M, Hadjichristidis C, Daniel M K, Latham I J, Le Gallou R, McComb T J L, McKenny J M, Noutsos A, Orford K J, Osborne J L, Rayner S M and H E S S Collaboration 2003 Atmospheric Monitoring for the H.E.S.S. Project *International Cosmic Ray Conference* (*International Cosmic Ray Conference* vol 5) p 2879

[27] Daniel M K, Lo Y T E and Chadwick P M 2014 *arXiv e-prints* arXiv:1402.4739 (*Preprint* `1402.4739`)

[28] Fruck C, Gaug M, Ernenwein J P, Mandát D, Schweizer T, Häfner D, Bulik T, Cieslar M, Costantini H, Dominik M, Ebr J, Garczarczyk M, Lorentz E, Pareschi G, Pech M, Puerto-Giménez I and Teshima M 2015 *Journal of Instrumentation* **10** P04012 (*Preprint* `1501.02156`)

[29] Reinhardt System- und Messelectronic GmbH *MWS 3 and 485 Sensors with Microprocessor without datalogger* available at `https://www.reinhardt-testsystem.de/english/climate_sensors/weatherstations/weather_station_mws_3.php`

[30] Holben B N, Eck T F, Slutsker I, Tanré D, Buis J P, Setzer A, Vermote E, Reagan J A, Kaufman Y J, Nakajima T, Lavenu F, Jankowiak I and Smirnov A 1998 *Remote Sensing of Environment* **66** 1–16

[31] Into Africa Seeking the Desert Sun accessed 2019-11-28 URL `https://blogs.nasa.gov/earthexpeditions/2016/09/08/seeking-the-desert-sun/`

[32] Schmid B, Michalsky J J, Slater D W, Barnard J C, Halthore R N, Liljegren J C, Holben B N, Eck T F, Livingston J M, Russell P B, Ingold T and Slutsker I 2001 **40** 1886–1896

[33] Barlow R 1989 *Statistics. A guide to the use of statistical methods in the physical sciences*

[34] Palmer M 2003 *Massachusetts Institute of Technology*

Appendices

All the codes used in this thesis and presented in the appendix are altogether also available on GitHub at the following link: Github Appendix codes here or https://github.com/franslott/Lott-Ndeyanale-Frans-MSc-Thesis-Physics-codes-

Appendix A

CT data extraction script in C++

```
void data_extraction(string filename, string out){

TFile* f = new TFile(filename.c_str());    //load in file
ofstream myfile;
myfile.open(out.c_str(), ios::app);        //open data file for radio data storage

//Create Data structures for all parts needed:
Sash::DataSet* events = (Sash::DataSet*)(f->Get("dstevents")); // event data structure
Sash::DataSet* run = (Sash::DataSet*)(f->Get("run"));    //Run information such as run length etc.
run->GetEntry();

Sash::DataSet* runquality = (Sash::DataSet*)(f->Get("runquality"));

Sash::HESSArray* hess = run->GetHESSArray();       //Create Hess array
Sash::RunHeader* runhead = hess->Get((Sash::RunHeader*)0);  // runheader with telescope involvement information

myfile << runhead->GetRunNum() << "\t";

//the loop over all the telescopes
```

```
int ct;                                              //telescope number
for(int i = 1; i <= 4; i++){
ct = i;
run->GetEntry();                                     //get run information
Sash::Pointer<Sash::Telescope> tel(hess, ct);      //select telescope i
if(runhead->CheckTelInRun(tel)){                     //check if telescope i is involved in run
runquality->GetEntry();                            //get runquality info for meteo
ParisRunQuality::TelescopeMeteoInformation* meteo = hess->Get(ct,(ParisRunQuality::TelescopeMeteoInformation*)0); //get meteo information for ct = i
ParisRunQuality::TelescopeTrackingInformation* teltrack = hess->Get(ct,(ParisRunQuality::TelescopeTrackingInformation*)0);  //This is for Zenith and Azimuth
Double_t radioT = meteo->GetMeanRadiometerTemperature();  //get radio temperature
Double_t zenith = teltrack->GetZenithAngleMean();
Double_t azimuth = teltrack->GetAzimuthAngleMean();
myfile << radioT<< "\t" << zenith << "\t" << azimuth << "\t";
} else {
myfile << "nan\tnan\tnan\t";
}
}
runquality->GetEntry();
Sash::HESSArray* hess = runquality->GetHESSArray();
ParisRunQuality::MeteoInformation* Meteo = hess->Get((ParisRunQuality::MeteoInformation*)0); //
Double_t ambTemp = Meteo->GetMeanTemperature();
Double_t RelHum = Meteo->GetMeanRelativeHumidity();

events->GetEntry();
```

```
Sash::HESSArray* hess = events->GetHESSArray(); //rerun hess array creation
Sash::EventHeader* head = hess->Get((Sash::EventHeader*)0);   //get event header for time info
Sash::Time time1 = head->GetTimeStamp();                        // get time stamp from event header
Stash::Coordinate position = runhead->GetTargetPosition();        // Target Position
double time = time1.GetTimeDouble();                            // convert from Sash::Time to double value
myfile << Form("%f", time) << "\t" << Form("%f\t", (ambTemp)) << Form("%f", (RelHum)) << endl;                           //write the time value
myfile.close();
}
//_______________________________File format output______________________________
//run number  CT1_radioT CT1_zenith CT1_azimuth CT2_radioT CT2_zenith CT2_azimuth CT3_radioT CT3_zenith CT3_azimuth CT4_radioT CT4_zenith CT4_azimuth TimeStamp ambientTemp RelHum
```

Appendix B

Calibration script in python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 12 03:34:32 2019

@author: thecuriosvambo
"""

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Aug 19 03:55:15 2019

@author: thecurioswambo
"""
from datetime import datetime
import numpy as np
import matplotlib.pyplot as plt
from itertools import islice
import glob
from pathlib import Path
import math
import matplotlib.cm as cm
import csv
```

```
import itertools
from astropy.stats import median_absolute_deviation
#import scipy

from scipy.optimize import curve_fit
import numpy.polynomial.polynomial as poly
from scipy.interpolate import *
from scipy.stats import pearsonr

# ####################################################################
####################################################################


# Opening Aeronet data and processing

print("press 1 for Aeronet , Atmoscope calibration")
print("press 2 for Atmoscope , CT calibration")

select=int(input('select:'))  # Input of option 1 or 2

if select == 1:

    # Opens Aeronet datafile for processing
    aeronet=open("20160101_20191231_HESS.lev20",'r')

    Aerodate=[]
    AeronetPWV=[]
    fullAerodate=[]

    AeronetPWVdata=open('AeronetPWV.csv','w') # opens new file to write Aeronet PWV to

    for line in islice(aeronet, 7, None):
```

```
    pass
    aeroPWV=float(line.split(",")[26])

    if aeroPWV <= -50.0:
        continue

    date=line.split(",")[0]
    day=int(date.strip("")[0:2])
    month=int(date.strip(":")[3:5])
    year=int(date.strip(':')[6:])

    time=line.split(",")[1]
    hour=int(time.strip("")[0:2])
    minute=int(time.strip(":")[3:5])
    sec=int(time.strip(':')[6:])

    datetim=datetime(year,month,day,hour,minute)
    fulldatetim=datetime(year,month,day,hour,minute,sec)
    Aerodate.append(datetim)
    fullAerodate.append(fulldatetim)
    PWV=aeroPWV
    AeronetPWV.append(PWV*10.00)
aeronet.close()

Aeronetdata=[i for i in zip(Aerodate,AeronetPWV)]

for i,j in zip(fullAerodate,AeronetPWV):
    AeronetPWVdata.write('%s,%5.2f\n' %(i,j)) #writes PWV onto file
# #######################################################################


## Plots PWV from the Aeronet

```

```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(fullAerodate,AeronetPWV,color='blue')
ax.set_title('Aeronet PW vs Time at H.E.S.S')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('period')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("completeAeronet")
fig.show()



#=========== opens and reads in data from atmoscope ==================#

AtmoscopePWVdata=open('AtmosPWV.csv','w')
y=6.5*(10**-3)
Atmoscopetime=[]
skytemp_atmoscope=[]
fullAtmoscopetime=[]
for file in sorted(glob.glob('/home/thecuriosvambo/Documents/codes/hesspart/*.dat')):
    with open(file) as f:
        # skipping and passing the Header titles and not reading them
        for line in islice(f, 1, None):
            pass

            # Extracting the temperature and cloud height and converting it into a useful number
            temperature=float(line.split()[2])
            cloud_altitude=float(line.split()[7])
```

```
                # skipping and not reading inthe data when the instruments where at the inital point and not recording useful data
                if  temperature <= -50.0:
                    continue

                # extracts the date and time from the file
                Datetime=line.split()[0]
                # extract date and converst it to a float (useful number)
                date=Datetime.split('_')[0]
                year=int(date.split('-')[0])
                month=int(date.split('-')[1])
                day=int(date.split('-')[2])
                # extracts the time and converts it into a useful number
                time=Datetime.split('_')[1]
                hour=int(time.split(':')[0])
                minute=int(time.split(':')[1])
                sec=int(time.split(':')[2])

                # Conversion to julian dates
                dateHuman = datetime(year, month, day, hour, minute)
                Atmoscopetime.append(dateHuman)

                fullhumandate = datetime(year, month, day, hour, minute,sec)
                fullAtmoscopetime.append(fullhumandate)

                # Calculating the PWV from the ambient temp,cloud altitude altitude and gradient temperature.
                T0=temperature  # ( degrees Celscius)
                h=cloud_altitude  #(m)

```

```
                # MWS 3 & 485-Sensors with Microprocessor without datalogger manual

                # Formula to calculate the cloud temperature, page 9 section 3.2.6 The clouds Sensor WKS 485
                T=T0-h*y    # (K)   T=cloud temperature temperature, T0=ambient temp, y=gradient temperature
                #T= T - 273.15  # (Degrees celcius)

                skytemp_atmoscope.append(T)

    f.close()

    Atmoscopedata=[i for i in zip(Atmoscopetime,skytemp_atmoscope)]

    ##### plotting Atmoscope sky temparture at H.E.S.S.

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(fullAtmoscopetime,skytemp_atmoscope,color='red')
    ax.set_title('Atmoscope PW vs Time at H.E.S.S')
    ax.set_ylabel('Infrared sky temp ($^\circ$C)')
    ax.set_xlabel('time')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("Atmosrawdata")
    fig.show()

# #############################################################################
############ CALIBRATION of Atmoscope vs Aeronet ##############################
```

```
#
##############################################################################


#Atmos=sorted(list(set(i for i,j in Atmoscopedata) & set(x for x,y in Aeronetdata)))
AtmosAero=list(set(i for i,j in Atmoscopedata).intersection(x for x,y in Aeronetdata))


Atmoscopemapping = dict((a, b) for a, b in Atmoscopedata)
Atmoscoperesult = [Atmoscopemapping[x] for x in sorted(AtmosAero)]


Aeronetmapping = dict((a, b) for a, b in Aeronetdata)
Aeronetresult = [Aeronetmapping[i] for i in sorted(AtmosAero)]

print("The number of points with same time is %i"  %(len(AtmosAero)))

### plotted Aeronet PWV vs Atmoscope sky temperature

fig = plt.figure()
ax = fig.add_subplot(111)
ax.scatter(Atmoscoperesult,Aeronetresult,color='red')
ax.set_title('Aeronet PWV vs Atmoscope IR sky temp')
ax.set_ylabel('AERONET PWV (mm)')
ax.set_xlabel('Atmoscope Infrared sky temp ($^\circ$C)')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("Aeronet-Atmos")
```

```
    fig.show()

    # converted the data into x and y numpy arrays

    x = np.array(Atmosceperesult, dtype=float)
    y = np.array(Aeronetresult, dtype=float)


    # fitted the data and found the relationship between Aeronet PWV
     and Atmoscope IR temperature

    fitting=np.polyfit(x, np.log(y), 1, w=np.sqrt(y))
    xx = np.linspace(-20, 16, 500)
    yy=[np.exp(fitting[1])*np.exp(fitting[0]*i) for i in xx]

    #found PWV as given by model

    AtmoscopePWV=[ np.exp(fitting[1])*np.exp(fitting[0]*i)  for i in
     skytemp_atmoscope]

    for i,j in zip(fullAtmoscopetime,AtmoscopePWV):
        AtmoscopePWVdata.write('%s,%5.2f\n' %(i,j)) #writing
   Atmoscope data to file



    # plots the fit between Aeronet PWV vs Atmoscope sky temp

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(x,y,color='red',zorder=1)
    ax.plot(xx,yy,'blue',label='fitted line y=%f*e^%fx' %(np.exp(
   fitting[1]),fitting[0]))
    ax.set_title('Aeronet PWV vs Atmoscope IR')
    ax.set_ylabel('AERONET PWV (mm)')
```

```
    ax.set_xlabel('Atmoscope Infrared sky temp ($^\circ$C)')
    ax.grid(True)
    ax.legend()
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("Aeronet-Atmosfit")
    fig.show()

    ########## plots callibrated Atmoscope data as govern by fit

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(fullAtmoscopetime,AtmoscopePWV,color='blue')
    ax.set_title('callibrated Atmoscope PW vs Time at H.E.S.S')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('period')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("Atmoscalibrateddata")
    fig.show()



# #####################################################################
    print ("Callibration completed")
# #####################################################################

# Opening H.E.S.S data for caliberation
if select == 2 :    # Part TWO
```

```


fin=open("radio.dat",'r')

def zero_to_nan(values):
        return [float('nan') if x >= 0 else x for x in values]

def one_to_nan(values):
        return [float('nan') if x < -100 else x for x in values]

def delete_to_nan(values):
        return [float('nan') if x < -100 else x for x in values]

def two_to_nan(values):
        return [float('nan') if x < -100  else x for x in values]

def remove3_to_nan(values):
        return [float('nan') if x < -100 else x for x in values]


CT1skyt=[]
CT2skyt=[]
CT3skyt=[]
CT4skyt=[]

CTtime=[]
CTUTC=[]


P1=9.614
P0=-62.46

for line in fin:
    run_no=float(line.split()[0])
```

```
        skyt_ct1=float(line.split()[1])
        CT1_zenith=float(line.split()[2])
        CT1_azimuth=float(line.split()[3])

        #print(skyt_ct1,Tz1)

        skyt_ct2=float(line.split()[4])
        CT2_zenith=float(line.split()[5])
        CT2_azimuth=float(line.split()[6])

       # print(skyt_ct1,Tz2)

        skyt_ct3=float(line.split()[7])
        CT3_zenith=float(line.split()[8])
        CT3_azimuth=float(line.split()[9])

        skyt_ct4=float(line.split()[10])
        CT4_zenith=float(line.split()[11])
        CT4_azimuth=float(line.split()[12])
        unix=float(line.split()[13])
```

```
        amb_T=float(line.split()[8])
        humid=float(line.split()[9])



        UTC=datetime.fromtimestamp(unix)
        UTCtime=UTC.replace(microsecond=0)
        UTCdate=UTC.replace(second=0, microsecond=0)

        CTtime.append(UTCtime)
        CTUTC.append(UTCdate)

        if CT1_zenith != 0.0:

            z_theta1=90-CT1_zenith
            x_z1=(1/np.cos(np.deg2rad(z_theta1)))
            Tz1=(P1*(1-x_z1)) + skyt_ct1
            CT1skyt.append(Tz1)


        if CT1_zenith == 0.0:
            CT1skyt.append(0.0)


        if CT2_zenith != 0.0:

            z_theta2=90.0-CT2_zenith
            x_z2=(1/np.cos(np.deg2rad(z_theta2)))
            Tz2=(P1*(1-x_z2)) + skyt_ct2
            CT2skyt.append(Tz2)

        if CT2_zenith == 0.0:
            CT2skyt.append(0.0)
```

```


        if CT3_zenith != 0.0:

            z_theta3=90.0-CT3_zenith
            x_z3=(1/np.cos(np.deg2rad(z_theta3)))
            Tz3=(P1*(1-x_z3)) + skyt_ct3
            CT3skyt.append(Tz3)


        if CT3_zenith == 0.0:
            CT3skyt.append(0.0)

        if CT4_zenith != 0.0:

            z_theta4=90.0-CT4_zenith
            x_z4=(1/np.cos(np.deg2rad(z_theta4)))
            Tz4=(P1*(1-x_z4)) + skyt_ct4
            CT4skyt.append(Tz4)

        if CT4_zenith == 0.0:
            CT4skyt.append(0.0)


    fin.close()

    ############################# plotting to data
   ##########################


    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CTtime,CT1skyt,color='blue')
    ax.set_title('CT1 infrared Sky temp')
```

```
    ax.set_ylabel('infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('time')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("rawskyCT1")
    fig.show()

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CTtime,CT2skyt,color='red')
    ax.set_title('CT2 infrared Sky temp')
    ax.set_ylabel('infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('time')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("rawskyCT2")
    fig.show()


    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CTtime,CT3skyt,color='green')
    ax.set_title('CT3 infrared Sky temp')
    ax.set_ylabel('infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('time')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("rawskyCT3")
    fig.show()

```

```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CTtime,CT4skyt,color='purple')
ax.set_title('CT4 infrared Sky temp')
ax.set_ylabel('infrared Sky temp ($^\circ$C)')
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("rawskyCT4")
fig.show()



###################### inspection of data #################################

##### To see if the CT telescopes are recording the same temperature ######
#### ploting against each other
# ##########################################################################




print("--unflagged raw data--")

# raw plot CT1 vs CT2 plot

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
```

```
    ax.scatter(CT1skyt,CT2skyt,color='blue')
    ax.set_title('CT1 vs CT2')
    ax.set_ylabel('CT2 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT1 infrared Sky temp ($^\circ$C)')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("unskytCT1CT2")
    fig.show()

    # raw plot CT1 vs CT4 plots

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CT1skyt,CT4skyt,color='red')
    ax.set_title('CT1 vs CT4')
    ax.set_ylabel('CT4 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT1 infrared Sky temp ($^\circ$C)')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("unskytCT1CT4")
    fig.show()

    # raw plot CT2 vs CT4 plot

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CT2skyt,CT4skyt,color='green')
    ax.set_title('CT2 vs CT4')
    ax.set_ylabel('CT4 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT2 infrared Sky temp ($^\circ$C)')
    ax.grid(True)
```

```
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("unskytCT2CT4")
    fig.show()

    # raw plot CT1 vs CT3 plot

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CT1skyt,CT3skyt,color='purple')
    ax.set_title('CT1 vs CT3')
    ax.set_ylabel('CT3 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT1 infrared Sky temp ($^\circ$C)')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("unskytCT1CT3")
    fig.show()

    # raw plot CT2 vs CT3 plot

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CT2skyt,CT3skyt,color='yellow')
    ax.set_title('CT2 vs CT3')
    ax.set_ylabel('CT3 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT2 infrared Sky temp ($^\circ$C)')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("unskytCT2CT3")
    fig.show()

```

```
# raw plot CT3 vs CT4 plot

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CT3skyt,CT4skyt,color='grey')
ax.set_title('CT3 vs CT4')
ax.set_ylabel('CT4 infrared Sky temp ($^\circ$C)')
ax.set_xlabel('CT3 infrared Sky temp ($^\circ$C)')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("unskytCT3CT4")
fig.show()


############### removing outliers and flagging of data
###################
print("------unwanted data flagged------")

CT1sky=zero_to_nan(CT1skyt)
CT1sky=one_to_nan(CT1sky)

CT2sky=zero_to_nan(CT2skyt)
CT2sky=two_to_nan(CT2sky)

CT3sky=zero_to_nan(CT3skyt)
CT3sky=remove3_to_nan(CT3sky)

CT4sky=zero_to_nan(CT4skyt)
CT4sky=delete_to_nan(CT4sky)

# removing nan values from  CT 1 and CT 2 data
CTs12=[(i,j) for i,j in zip(CT1sky,CT2sky) if str(i) != 'nan'
and str(j) != 'nan']
```

```python
    # splitting CTs12 in CT1 AND CT2
    x11=[i[0] for i in CTs12]
    y11=[i[1] for i in CTs12]

    # calculating the Pearson Coefficient of CT1 and CT2
    corr12, p12 = pearsonr(x11, y11)
    print(corr12,p12)

    # plotting CT1 vs CT3

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CT1sky,CT2sky,color='blue')
    ax.set_title('CT1 vs CT2 after flagging')
    ax.set_ylabel('CT2 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT1 infrared Sky temp ($^\circ$C)')
    ax.text(2, 6, r'$\rho$=%f' %(corr12), fontsize=10)
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("flagskyCT1CT2")
    fig.show()

    # removing nan values from CT1 and CT3 data
    CTs13=[(i,j) for i,j in zip(CT1sky,CT3sky) if str(i) != 'nan'
   and str(j) != 'nan']

    # splitting CTs13 to CT1 and CT3 clean list
    x13=[i[0] for i in CTs13]
    y13=[i[1] for i in CTs13]

    # calculating the Pearson Coefficient of CT1 and CT3
    corr13, p13 = pearsonr(x13, y13)
    print(corr13,p13)
```

```python

# Plotting CT1 VS CT3

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CT1sky,CT3sky,color='purple')
ax.set_title('CT1 vs CT3 after flagging')
ax.set_ylabel('CT3 infrared Sky temp ($^\circ$C)')
ax.set_xlabel('CT1 infrared Sky temp ($^\circ$C)')
ax.text(2, 6, r'$\rho$=%f' %(corr13), fontsize=10)
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("flagskyCT1CT3")
fig.show()

# removing nan values from CT1 and CT4 data
CTs14=[(i,j) for i,j in zip(CT1sky,CT4sky) if str(i) != 'nan' and str(j) != 'nan']
# splitting CTs14 to CT1 and CT4 clean list
x14=[i[0] for i in CTs14]
y14=[i[1] for i in CTs14]
# calculating the Pearson Coefficient of CT1 and CT3
corr14, p14 = pearsonr(x14, y14)
print(corr14,p14)

# Plotting CT 1 vs CT 4


fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CT1sky,CT4sky,color='red')
ax.set_title('CT1 vs CT4 after flagging')
```

```
    ax.set_ylabel('CT4 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT1 infrared Sky temp ($^\circ$C)')
    ax.text(2, 6, r'$\rho$=%f' %(corr14), fontsize=10)
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("flagskyCT1CT4")
    fig.show()


     # removing nan values from CT2 and CT4 data
    CTs24=[(i,j) for i,j in zip(CT2sky,CT4sky) if str(i) != 'nan'
   and str(j) != 'nan']
    # splitting CTs24 to CT2 and CT4 clean list
    x24=[i[0] for i in CTs24]
    y24=[i[1] for i in CTs24]
    # calculating the Pearson Coefficient of CT2 and CT4
    corr24, p24 = pearsonr(x24, y24)
    print(corr24,p24)

    # Plotting CT 2 vs CT 4

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CT2sky,CT4sky,color='green')
    ax.set_title('CT2 vs CT4 after flagging')
    ax.set_ylabel('CT4 infrared Sky temp ($^\circ$C)')
    ax.set_xlabel('CT2 infrared Sky temp ($^\circ$C)')
    ax.text(2, 6, r'$\rho$=%f' %(corr24), fontsize=10)
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig("flagskyCT2CT4")
    fig.show()
```

```
# removing nan values from CT2 and CT3 data
CTs23=[(i,j) for i,j in zip(CT2sky,CT3sky) if str(i) != 'nan'
and str(j) != 'nan']
# splitting CTs23 to CT2 and CT3 clean list
x23=[i[0] for i in CTs23]
y23=[i[1] for i in CTs23]
# calculating the Pearson Coefficient of CT2 and CT3
corr23, p23 = pearsonr(x23, y23)
print(corr23,p23)

# Plotting CT2 vs CT3

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CT2sky,CT3sky,color='yellow')
ax.set_title('CT2 vs CT3 after flagging')
ax.set_ylabel('CT3 infrared Sky temp ($^\circ$C)')
ax.set_xlabel('CT2 infrared Sky temp ($^\circ$C)')
ax.text(2, 6, r'$\rho$=%f' %(corr23), fontsize=10)
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("flagskyCT2CT3")
fig.show()

# removing nan values from CT3 and CT4 data
CTs34=[(i,j) for i,j in zip(CT3sky,CT4sky) if str(i) != 'nan'
and str(j) != 'nan']
# splitting CTs34 to CT3 and CT4 clean list
x34=[i[0] for i in CTs34]
y34=[i[1] for i in CTs34]
# calculating the Pearson Coefficient of CT3 and CT4
```

```
corr34, p34 = pearsonr(x34, y34)
print(corr34,p34)

# plottinfg CT3 vs CT4

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CT3sky,CT4sky,color='grey')
ax.set_title('CT3 vs CT4 flagging')
ax.set_ylabel('CT4 infrared Sky temp ($^\circ$C)')
ax.set_xlabel('CT3 infrared Sky temp ($^\circ$C)')
ax.text(2, 6, r'$\rho$=%f' %(corr34), fontsize=10)
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("flagskyCT3CT4")
fig.show()

############################ calibrated plots after flagging of data ##########################

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CTtime,CT1sky,color='blue')
ax.set_title('CT1 infrared Sky temp after calibration')
ax.set_ylabel('infrared Sky temp ($^\circ$C)')
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("calskyCT1")
fig.show()
```

```


fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CTtime,CT2sky,color='red')
ax.set_title('CT2 infrared Sky temp after calibration')
ax.set_ylabel('infrared Sky temp ($^\circ$C)')
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("calskyCT2")
fig.show()

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CTtime,CT3sky,color='green')
ax.set_title('CT3 infrared Sky temp after calibration')
ax.set_ylabel('infrared Sky temp ($^\circ$C)')
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("calskyCT3")
fig.show()


fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.scatter(CTtime,CT4sky,color='purple')
#ax.set_title('CT4 infrared Sky temp after calibration')
ax.set_ylabel('infrared Sky temp ($^\circ$C)')
```

```
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("calskyCT4")
fig.show()

##### finding the Average & standard deviation of all CT 1-4
radiometer data ######

CTs14ave=[]
CTs14std=[]

for i,k,m,n in zip(CT1sky,CT2sky,CT3sky,CT4sky):
    CTs=[i,k,m,n]
    aveCTs=np.mean(CTs)
    sttdCTs=np.std(CTs)
    CTs14ave.append(aveCTs)
    CTs14std.append(sttdCTs)



CTs14ave=zero_to_nan(CTs14ave)
CTs14stde=zero_to_nan(CTs14std)



#######====== Atmoscope PWV for CALIBRATION with CT's
=====######

# reading in and processing Atmoscope data

y=6.5*(10**-3)
Atmoscopetime=[]
skytemp_atmoscope=[]

```

```
    for file in sorted(glob.glob('/home/thecuriosvambo/Documents/codes/hesspart/*.dat')):
        with open(file) as f:
            # skipping and passing the Header titles and not reading them
            for line in islice(f, 1, None):
                pass

                # Extracting the temperature and cloud height and converting it into a useful number
                temperature=float(line.split()[2])
                cloud_altitude=float(line.split()[7])

                # skipping and not reading inthe data when the instruments where at the inital point and not recording useful data
                if  temperature <= -50.0:
                    continue

                # extracts the date and time from the file
                Datetime=line.split()[0]
                # extract date and converst it to a float (useful number)
                date=Datetime.split('_')[0]
                year=int(date.split('-')[0])
                month=int(date.split('-')[1])
                day=int(date.split('-')[2])
                # extracts the time and converts it into a useful number
                time=Datetime.split('_')[1]
                hour=int(time.split(':')[0])
                minute=int(time.split(':')[1])
                sec=int(time.split(':')[2])

                # Conversion to julian dates
```

```
            dateHuman = datetime(year, month, day, hour, minute)
            Atmoscopetime.append(dateHuman)


            # Calculating the PWV from the ambient temp,cloud altitude altitude and gradient temperature.
            T0=temperature  # ( degrees Celscius)
            h=cloud_altitude  #(m)

            # MWS 3 & 485-Sensors with Microprocessor without datalogger manual

            # Formula to calculate the cloud temperature, page 9 section 3.2.6 The clouds Sensor WKS 485
            T=T0-h*y    # (K)   T=cloud temperature temperature, T0=ambient temp, y=gradient temperature
            #T= T - 273.15  # (Degrees celcius)

            skytemp_atmoscope.append(T)

f.close()




# ##############################################################################
# ###########################################################################
# ###########################################################################
```

```
# ##########################################################################

################### CT AND ATMOSCOPE DATA CALIBRATION #####################

# ##########################################################################
# ##########################################################################

# CT sky temp data
allCTdata=[i for i in zip(CTUTC,CTs14ave,CTs14std) if i[1] > -90]

CTdata=[(i[0],i[1]) for i in allCTdata]
CTsttd=[(i[0],i[2]) for i in allCTdata]


newCTdata=[i for i in zip(CTtime,CTs14ave,CTs14std) if i[1] > -90]

CTtime=[i[0] for i in newCTdata]
CTs14ave=[i[1] for i in newCTdata]
CTs14std=[i[2] for i in newCTdata]

# Atmoscope data sky temp and calibrated PWV
AtmoscopePWV=[ np.exp(2.08874992)*np.exp(0.07109687*i)  for i in skytemp_atmoscope]  ### As gain from the first relationship, between the atmoscope and Aeronet
Atmoscopedata=[i for i in zip(Atmoscopetime,skytemp_atmoscope)]
AtmoscopePWVdata=[i for i in zip(Atmoscopetime,AtmoscopePWV)]
```

```
# finding data that overlap between Atmoscope and CT

AtmosCT=list(set(i for i,j in AtmoscopePWVdata).intersection(x
for x,y in CTdata))

# Overlapping Atmoscope PWV

Atmoscopemapping = dict((a, b) for a, b in AtmoscopePWVdata)
Atmoscoperesult = [Atmoscopemapping[x] for x in sorted(AtmosCT)]

# Overlap CT sky temp and Standard deviation
CTmapping = dict((a, b) for a, b in CTdata)
CTresult = [CTmapping[i] for i in sorted(AtmosCT)]

CTsttdmapping = dict((a, b) for a, b in CTsttd)
CTsttdresult = [CTsttdmapping[i] for i in sorted(AtmosCT)]

print("The number of points with same time is %i"  %(len(AtmosCT
)))

# removing nan values from overlapping CT,Atmoscope and standard
deviation results

XY=[i for i in zip(CTresult,Atmoscoperesult,CTsttdresult) if str
(i[0]) != 'nan' ]

X=np.array([i[0] for i in XY])
Y=np.array([i[1] for i in XY])
S=np.array([i[2] for i in XY])

print("The number of points without 'nan' with the same time is
%i"  %(len(X)))

#########Ploting averaged sky temp calibrated plot
```

```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.errorbar(CTtime,CTs14ave,yerr=CTs14std,color='red',fmt='o',
alpha=1)
#ax.set_title('CT mean IR sky temperature at H.E.S.S')
ax.set_ylabel('infrared Sky temp ($^\circ$C)')
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("CTrawdata")
fig.show()


### Ploting overlapping results against each other to find
relationship
fig = plt.figure()
ax = fig.add_subplot(111)
ax.errorbar(X,Y,xerr=S,color='red',fmt='o')
ax.set_title('Atmoscope PWV vs CT mean IR sky temp')
ax.set_ylabel('Atmoscope PWV (mm)')
ax.set_xlabel('CT mean infrared Sky temp ($^\circ$C)')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("Atmos-CT")
fig.show()

X = np.array(X, dtype=float)
Y = np.array(Y, dtype=float)
S = np.array(S, dtype=float)

lin=np.polyfit(X, np.log(Y), 1, w=np.sqrt(Y))
```

```
XX = np.linspace(-100, -20, 500)
YY=[np.exp(lin[1])*np.exp(lin[0]*i) for i in XX]


# Finding the relationship between CT sky temp and Atmoscope PWV

#lin=np.polyfit(X,Y,1)
print("A=%f" %(np.exp(lin[1])))
print("B=%f" %(lin[0]))
print("the equation is given by y=Ae^(Bx)")
print("y=%f*e^%fx" %(np.exp(lin[1]),lin[0]))


#Ploting the fit onto the grapgh


fig = plt.figure()
ax = fig.add_subplot(111)
ax.errorbar(X,Y,xerr=S,color='red',fmt='o',alpha=0.3,zorder=1,
label="fitted data")
ax.set_title('Atmoscope PWV vs CT mean IR sky temperaturature')
ax.plot(XX,YY,'blue',label='fitted line y=%f*e^%fx' %(np.exp(lin
[1]),lin[0]))
ax.set_ylabel('Atmoscope PWV (mm)')
ax.set_xlabel('CT mean infrared Sky temp ($^\circ$C)')
ax.grid(True)
ax.legend()
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("Atmos-CTfitt")
fig.show()
```

```
CTPWVdata=open('CTPWV.csv','w')  # Opening new file to write CT PWV results as given by fit

#CTPWV=[ float(lin[0])*i+float(lin[1])  for i in CTs14ave] # PWV as given by fit
CTPWV=[ np.exp(lin[1])*np.exp(lin[0]*i)  for i in CTs14ave]

#Error propagation from Sky temp to PWV
CTPWVstd=[np.sqrt(((lin[1]*lin[0])*(np.exp(lin[0]*i)))*(p**2)) for i,p in zip(CTs14ave,CTs14std)]

for i,j,k in zip(CTtime,CTPWV,CTPWVstd):
    CTPWVdata.write('%s,%5.2f,%5.2f\n' %(i,j,k)) #wirting on to File



# Plotting CT PWV results
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([],[])
ax.errorbar(CTtime,CTPWV,yerr=CTPWVstd,color='blue',fmt='o', alpha=1)
ax.set_title('mean callibrated CT PWV at H.E.S.S')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('time')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig("CTcallibrateddata")
fig.show()

print ("Calibration Completed")
```

Appendix C

Processing script in python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Sep  7 17:49:16 2019

@Author: Frans Lott N
@alias: thecurioswambo
contact: +264 81 3129813
email:franslott8@gmail.com
"""

#importing packages

from datetime import datetime
import numpy as np
import matplotlib.pyplot as plt
import math
import matplotlib.cm as cm
import csv

###############################

###### Plotting option ######

```

```
print("enter 1: for CT (night) results")
print("enter 2: for AERONET (day) results")
##############################

select=int(input('select:')) ########### recieve plotting input
bin_edges=[2.5,3,3.5,4,4.5,5]  ####list of bin adges
bin_edges1=[2.5,3,3.5,4,4.5,5]  ####list of bin adges
bin_edges3=[.5,1,1.5,2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8]
bin_edges2=[.5,1,1.5,2.5,3,3.5,4,4.5,5,5.5,6]
#### plotting CT results #######

if select == 1:
    fin=open('CTPWV.csv','r')      # Opening CT PWV file


    CTPWV=[]                # empty list for storing CT PWV data
    CTPWVstd=[]             # empty list storing CT PWV standard deviation data
    CTtime=[]               # empty list for storing correspinding CT data time


    #proccessing data from file (loading data)
    for line in fin:

        # extracts date and time strings, and converting it into a number
        Datetime=line.split(',')[0]
        date=Datetime.split()[0]
        year=int(date.split('-')[0])
        month=int(date.split('-')[1])
        day=int(date.split('-')[2])
        time=Datetime.split()[1]
        hour=int(time.split(':')[0])
        minute=int(time.split(':')[1])
```

```
     sec=int(time.split(':')[2])

     # creating a time from extracted data
     dateHuman = datetime(year, month, day, hour, minute)

     # extracting PWV and Standard deviation and converting it into float

     PWV=float(line.split(',')[1])
     std=float(line.split(',')[2])

     # storing time,PWV and standard deviation data to empty list
     CTPWV.append(PWV)
     CTPWVstd.append(std)
     CTtime.append(dateHuman)

 # plotting data to visulize how the PWV looks like

 fig = plt.figure()
 ax = fig.add_subplot(111)
 ax.plot([],[])
 ax.errorbar(CTtime,CTPWV,yerr=CTPWVstd,color='blue',fmt='o',alpha=1)
 ax.set_title('mean callibrated CT PWV at H.E.S.S')
 ax.set_ylabel('PWV (mm)')
 ax.set_xlabel('period')
 ax.grid(True)
 fig.autofmt_xdate(rotation=45)
 fig.tight_layout()
 fig.savefig('CTplots/CTdata')
 fig.show()

 # a function to convert all zeros to nan values incase of plotting
 def zero_to_nan(values):
```

```
        return [float('nan') if x==0 else x for x in values]

# creating a list equivalent to number of CT years data called YCT
YCT=list(range(1,17))

# Creating a list equivalent to number of months in a year
M=list(range(1,13))
# Gamsberg Measured PWV in 1994 and 1995
Gamsberg_94 =[0,0,0,0,0,0,3.20799,2.56114,5.46083,6.21921,7.51292,6.70993]
Gamsberg_95 =[5.90694,0,6.33074,6.46457,6.55379,4.34557,3.52027,0,0,0,0,0]

# a list containing months
months=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

# analysing the PWV data,and standard deviation by year
for j in YCT:
    CTyear=[i for i in zip(CTtime,CTPWV,CTPWVstd) if i[0].year ==2003+j ]
    CTdateyear=[i[0] for i in CTyear]
    CTaveyear=[i[1] for i in CTyear]
    CTstdyear=[i[2] for i in CTyear]

    # plotting by yearly PWV data

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.errorbar(CTdateyear,CTaveyear,CTstdyear,color='yellow', fmt='o')
    ax.set_title('CT mean at H.E.S.S. %i' %(2003+j))
    ax.set_ylabel('Percipitable Water Vapour (mm)')
    ax.set_xlabel('time')
```

```
        ax.grid(True)
        fig.autofmt_xdate(rotation=45)
        fig.tight_layout()
        fig.savefig('CTplots/Plots/Yearly/CT1-4meanatH.E.S.S.%i.jpg' %(2003+j))
        fig.show()

        # analysing data by month of year in YCT
        for m,t in zip(M,months):

            CTmonth=[i for i in CTyear if i[0].month== m ]
            CTdate=[i[0] for i in CTmonth]
            CTave=[i[1] for i in CTmonth]
            CTstd=[i[2] for i in CTmonth]

            # Plotting individual month data

            fig = plt.figure()
            ax = fig.add_subplot(111)
            ax.errorbar(CTdate,CTave,CTstd,color='blue',fmt='o')
            ax.set_title('CT mean PWV at H.E.S.S. in %s %i' %(t,2003+j))
            ax.set_ylabel('Percipitable Water Vapour (mm)')
            ax.set_xlabel('time')
            ax.grid(True)
            fig.autofmt_xdate(rotation=45)
            fig.tight_layout()
            fig.savefig('CTplots/Plots/Monthly/CT1-4meanatH.E.S.S.%s%i.jpg' %(t,2003+j))
            fig.show()

    # scaling function for Mt Gamsberg
    h1=1800                  #  elevation of H.E.S.S. site in meters
    h2=2347                  #  elevation of Mt Gamsberg in meters
    H=2000.                  # Water vapour scale height
```

```
alpha=np.e**-((h2-h1)/H)    #  scaling function

Gamsberg_94=zero_to_nan(Gamsberg_94)   # removing zero values for plotting
Gamsberg_95=zero_to_nan(Gamsberg_95)   # removing zero values for plotting

PWVzero = [0 if math.isnan(x) else x for x in CTPWV] # adding zero if "nan"
stdzero=[0 if math.isnan(x) else x for x in CTPWVstd] # adding zero if "nan"

scaPWV = [ alpha*x for x in CTPWV]
# removing zeros from list to have clean PWV and standard deviation
PWVclean = [i for i in zip(CTtime,PWVzero,stdzero) if i[1] != 0 ]

# creating empty list for storing weighted mean PWV
PWVweightedave=[]
# creating empty list for storing corresponding standard deviation
yerr=[]
# creating empty list for storing corresponding time
month=[]

# analysing data by year
for j in YCT:
    CTyear=[i for i in PWVclean if i[0].year ==2003+j ]
    CTdateyear=[i[0] for i in CTyear]
    CTaveyear=[i[1] for i in CTyear]
    CTstdyear=[i[2] for i in CTyear]


    monthlist=[]
```

```
    allmonthdata=[]

    maxcount=[]
    maxtime=[]

    mincount=[]
    mintime=[]

    # Analysing data by month
    for m,t in zip(M,months):
        CTmonth=[i for i in CTyear if i[0].month== m ]
        CTdate=[i[0] for i in CTmonth]
        CTave=[i[1] for i in CTmonth]
        CTstd=[i[2] for i in CTmonth]
        CTstd=[1 if np.mean(CTstd)==0 else i for i in CTstd]
        allmonthdata.append(len(CTmonth))
        monthlist.append(t)

        # proccesing data as 0 if number of data points is less than 50

        if len(CTmonth) < 50.0 :
            month.append(t)
            yerr.append(0.0)
            PWVweightedave.append(0.0)
            mincount.append(len(CTmonth))
            maxcount.append(0.0)
            maxtime.append(t)
            mintime.append(t)

        # processing data if number of data points is 50 or more
        if len(CTmonth) >= 50.0 :
            weightedave=np.average(CTave,weights=CTstd)
            PWVweightedave.append(weightedave)
            CTweightedstd=np.std(CTave)
```

```
            yerr.append(CTweightedstd)
            month.append(t)
            mincount.append(0.0)
            maxcount.append(len(CTmonth))
            maxtime.append(t)
            mintime.append(t)

            # Plotting monthly histogram

            plt.figure("Monthly Histogram of PWV")
            plt.hist(CTave, color = 'purple', edgecolor = 'black',bins = bin_edges,alpha=1,cumulative=1,normed=True)
            plt.title("PWV Histogram for %s in %i at H.E.S.S." %(t,2003+j))
            plt.xlabel(" Percipitable Water Vapour (mm)")
            plt.ylabel("fraction [%]")
            plt.savefig('CTplots/Hist/Monthly/PWV histograms of month %s year %i.jpeg' %(t,2003+j))
            #plt.show()


    # getting number of data taken for each month with records and writting them to a file
    alldatafile=[i for i in zip(monthlist,allmonthdata)]
    with open('CTplots/datacount/csvfile/alldatafile_{0}.csv'.format(j+2003),'w',newline='') as f:
        w = csv.writer(f)
        w.writerow(['month of year','data points taken'])
        w.writerows(alldatafile)

    # plotting number of monthly data points
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(monthlist,allmonthdata,color='purple')
```

```
        #ax.set_title('number of monthly recordings at H.E.S.S. in %i' %(2003+j))
        ax.set_ylabel('number of recordings')
        ax.set_xlabel('Month')
        ax.grid(True)
        fig.autofmt_xdate(rotation=45)
        fig.tight_layout()
        fig.savefig('CTplots/datacount/plots/recordcount%i.jpg' %(2003+j))
        fig.show()


        # convering all zero data points to nan for plotting
        maxcount=zero_to_nan(maxcount)
        mincount=zero_to_nan(mincount)

        # plotting considered 50 or more points for visualization
        fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.scatter(maxtime,maxcount,color='purple',label='considered value')
        ax.scatter(mintime,mincount,color='grey',label='Unconsidered value')
        ax.set_title('number of considered monthly recordings at H.E.S.S. in %i' %(2003+j))
        ax.set_ylabel('number of recordings')
        ax.set_xlabel('Month')
        ax.grid(True)
        ax.legend()
        fig.autofmt_xdate(rotation=45)
        fig.tight_layout()
        fig.savefig('CTplots/datacount/plots/greyrecordcount%i.jpg' %(2003+j))
        fig.show()

```

```
# converting zero to nan
yerr=zero_to_nan(yerr)
PWVweightedave=zero_to_nan(PWVweightedave)
# removing nana values
PWVstdtime=[i for i in zip(month,PWVweightedave,yerr) if str(i[1]) != 'nan']


# scaling H.E.S.S. site PWV for Mt Gamsberg
scPWVweighted=[i*alpha for i in PWVweightedave]
# removing nan values
scPWVstdtime=[i for i in zip(month,scPWVweighted,yerr) if str(i[1]) != 'nan']


avePWV=[]   # empty list to store mean PWV of H.E.S.S.
aveSTD=[]   #  empty lsit for storing corresponding Standard deviation

scavePWV=[]  # empty list for storing scaled mean PWV for Mt Gamsberg
scaveSTD=[]  # empty list storing corresponding standard deviation for scaled vales


# processing non scaled PWV of H.E.S.S. and scaled PWV for Mt Gamsberg
for i in months:
    avestd=[(k[0],k[1],k[2]) for k in PWVstdtime if k[0]==i]
    allPWV=[i[1] for i in avestd ]
    allstd=[i[2] for i in avestd ]
    avePWVave=np.average(allPWV,weights=allstd)
    stdallstd=np.std(allPWV)
    avePWV.append(avePWVave)
```

```
    aveSTD.append(stdallstd)

    scavestd=[(k[0],k[1],k[2]) for k in scPWVstdtime if k[0]==i]
    scallPWV=[i[1] for i in scavestd ]
    scallstd=[i[2] for i in scavestd ]
    scavePWVave=np.average(scallPWV,weights=scallstd)
    scstdallstd=np.std(scallPWV)
    scavePWV.append(scavePWVave)
    scaveSTD.append(scstdallstd)


# Plotting H.E.S.S. site single monthly  PWV values
fig = plt.figure()
ax = fig.add_subplot(111)
ax.scatter(month,PWVweightedave,color='grey',label='monthly weighted value')
ax.errorbar(months,avePWV,yerr=aveSTD,color='blue',fmt='o',label='specific month weighted averages')
ax.set_title('H.E.S.S Weighted monthly averages')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('Month')
ax.grid(True)
ax.legend()
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig('CTplots/AveragedpointHESS.jpg')
fig.show()

# Plotting Mt Gasmberg site single monthly weighted mean PWV values
fig = plt.figure()
ax = fig.add_subplot(111)
ax.scatter(month,scPWVweighted,color='grey',label='monthly weighted value')
```

```
ax.errorbar(months,scavePWV,yerr=aveSTD,color='green',fmt='o',label='specific month weighted averages')
ax.set_title('Mt Gamsberg Weighted monthly averages')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('Month')
ax.grid(True)
ax.legend()
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig('CTplots/AveragedpointGams.jpg')
fig.show()



Gamsberg_94=zero_to_nan(Gamsberg_94)
Gamsberg_95=zero_to_nan(Gamsberg_95)
yerr=zero_to_nan(yerr)
PWVweighted=zero_to_nan(PWVweightedave)


# Plotting H.E.S.S. weighted mean PWV (seasonal variations)
fig = plt.figure()
ax = fig.add_subplot(111)
n=12
q=[PWVweighted[i:i+n] for i in range(0, len(PWVweighted), n)]
x = np.arange(1,13)
markers=['x','*','o','v','^','<','>','1','D','H','s','8','_','|','+','.']
colors = cm.rainbow(np.linspace(0, 1, len(q)))

for j,y, c,m in zip(YCT,q, colors,markers):
    ax.plot(months, y, color=c,linestyle='-',marker=m,label="%i"%(2003+j))
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.5, box.height])
```

```
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
    ax.set_title('H.E.S.S. weighted Means')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('Month of the year')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('CTplots/H.E.S.S.monthlyave.png')
    fig.show()

    # plotting weighted mean scaled values for Mt Gamsberg (Seasonal variations)
    fig = plt.figure()
    ax = fig.add_subplot(111)
    n=12
    q=[scPWVweighted[i:i+n] for i in range(0, len(scPWVweighted), n)]
    x = np.arange(1,13)
    markers=['x','*','o','v','^','<','>','1','D','H','s','8','_','|','+','.']
    colors = cm.rainbow(np.linspace(0, 1, len(q)))

    for j,y, c,m in zip(YCT,q, colors,markers):
        ax.plot(months, y, color=c,linestyle='-',marker=m,label="%i"%(2003+j))
    box = ax.get_position()
    ax.set_position([box.x0, box.y0, box.width * 0.5, box.height])
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
    ax.set_title('Mt Gamsberg weighted Means')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('Month of the year')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('CTplots/MountGamsaveCT.png')
```

```
fig.show()


# plotting weighted means of Mt Gamsberg along with 1994-1995 PWV values
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(months,Gamsberg_94,color='grey',linestyle='-',marker='+',label="1994" )
ax.plot(months,Gamsberg_95,color='purple',linestyle='-',marker='s',label="1995" )
n=12
q=[scPWVweighted[i:i+n] for i in range(0, len(scPWVweighted), n)]
x = np.arange(1,13)
markers=['x','*','o','v','^','<','>','1','D','H','s','8','_','|','+','.']
colors = cm.rainbow(np.linspace(0, 1, len(q)))

for j,y, c,m in zip(YCT,q, colors,markers):
    ax.plot(months, y, color=c,linestyle='-',marker=m,label="%i"%(2003+j))
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.5, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
ax.set_title('Mt Gamsberg weighted Means')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('Month of the year')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig('CTplots/MountGamsave.png')
fig.show()
```

```
heights, bins = np.histogram(CTPWV, bins = bin_edges3)
percent = [i/sum(heights)*100 for i in heights]


plt.figure()
plt.bar(bins[:-1], percent, width = 0.5, linewidth=1,align="edge
", edgecolor='black',color='blue',alpha=1)
plt.xlim(min(bins), max(bins))
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Percipitable Water Vapour (mm)')
plt.ylabel('Fraction  [%]')
plt.xticks()
plt.yticks()
plt.savefig('CTplots/Hist/Monthly/allHESSPWV.jpeg')
plt.title('relative frequencyDistribution Histogram H.E.S.S.',
fontsize=15)
plt.show()


heights1, bins2 = np.histogram(scaPWV, bins = bin_edges2)
percent1 = [i/sum(heights1)*100 for i in heights1]

plt.figure()
plt.bar(bins2[:-1], percent1, width = 0.5, linewidth=1,align="
edge", edgecolor='black',color='green',alpha=1)
plt.xlim(min(bins2), max(bins2))
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Percipitable Water Vapour (mm)')
plt.ylabel('Fraction  [%]')
plt.xticks()
plt.yticks()
plt.savefig('CTplots/Hist/Monthly/allgamsPWV.jpeg')
```

```
    plt.title('Relative frequency Distribution Histogram for Gamsberg',fontsize=15)
    plt.show


    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.errorbar(CTtime,scaPWV,yerr=CTPWVstd,color='green',fmt='o',alpha=1)
    #ax.set_title('mean callibrated CT PWV at H.E.S.S')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('period')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('CTplots/gamsCTdata.jpeg')
    fig.show()


# proccesing areonet values (daily values)

if select == 2:

    fin=open('AeronetPWV.csv','r') # opening Aeronet data file


    CTPWV=[]   # empty list for storing PWV
    CTtime=[]  # empty list for storing corresponding time

    # reading and loading data from file
    for line in fin:
```

```

        # reading in date and time, converting it to a number
        Datetime=line.split(',')[0]
        date=Datetime.split()[0]
        year=int(date.split('-')[0])
        month=int(date.split('-')[1])
        day=int(date.split('-')[2])
        time=Datetime.split()[1]
        hour=int(time.split(':')[0])
        minute=int(time.split(':')[1])
        sec=int(time.split(':')[2])

        # converting it to useful date
        dateHuman = datetime(year, month, day, hour, minute,sec)
        PWV=float(line.split(',')[1])

        # storing PWV data and time to empty list
        CTPWV.append(PWV)
        CTtime.append(dateHuman)

    #plotting data to visualize
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot([],[])
    ax.scatter(CTtime,CTPWV,color='blue',alpha=1)
    ax.set_title('mean callibrated CT PWV at H.E.S.S')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('time')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('Aeronetplots/Aeronetcallibrateddata')
    fig.show()

```

```
# a function to convert all zeros to nan values incase of plotting
def zero_to_nan(values):
    return [float('nan') if x==0 else x for x in values]

# a list containing number of years data taken (3)
YCT=list(range(1,4))
# list containing number of months in year (12)
M=list(range(1,13))
# list contaning listof month
months=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

# processing data by yeear
for j in YCT:
    CTyear=[i for i in zip(CTtime,CTPWV) if i[0].year ==2015+j ]
    CTdateyear=[i[0] for i in CTyear]
    CTaveyear=[i[1] for i in CTyear]

    # plotting year data
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(CTdateyear,CTaveyear,color='yellow')
    ax.set_title('Aeronet PWV at H.E.S.S. %i' %(2015+j))
    ax.set_ylabel('Percipitable Water Vapour (mm)')
    ax.set_xlabel('time')
    ax.grid(True)
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('Aeronetplots/Plots/yearly/meanatH.E.S.S.%i.jpg' %(2015+j))
    fig.show()

    # processing data by month
    for m in M:
```

```
        CTmonth=[i for i in CTyear if i[0].month== m ]
        CTdate=[i[0] for i in CTmonth]
        CTave=[i[1] for i in CTmonth]

        # plotting monthly plots
        fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.scatter(CTdate,CTave,color='blue')
        ax.set_title('Aeronet PWV at H.E.S.S. %i %i' %(m,2015+j))
        ax.set_ylabel('Percipitable Water Vapour (mm)')
        ax.set_xlabel('time')
        ax.grid(True)
        fig.autofmt_xdate(rotation=45)
        fig.tight_layout()
        fig.savefig('Aeronetplots/Plots/monthly/H.E.S.S.%i%i.jpg' %(m,2015+j))
        fig.show()



h1=1800  # H.E.S.S elevation in meters
h2=2347  # Mt Gamsberg elevation in meters
H=2000.  # water vapour scale heiht
alpha=np.e**-((h2-h1)/H)

# adding zero if vale is equall to nan
PWVzero = [0 if math.isnan(x) else x for x in CTPWV]
# removing zero PWV with its time
PWVclean = [i for i in zip(CTtime,PWVzero) if i[1] != 0 ]


# empty list for weighted PWV and erro,and time
PWVweightedave=[]
```

```
    yerr=[]
    month=[]

    # processing data by year
    for j in YCT:
        CTyear=[i for i in PWVclean if i[0].year ==2015+j ]
        CTdateyear=[i[0] for i in CTyear]
        CTaveyear=[i[1] for i in CTyear]


        monthlist=[]
        allmonthdata=[]

        maxcount=[]
        maxtime=[]

        mincount=[]
        mintime=[]

        # processing data by month
        for m,t in zip(M,months):
            CTmonth=[i for i in CTyear if i[0].month== m ]
            CTdate=[i[0] for i in CTmonth]
            CTave=[i[1] for i in CTmonth]

            allmonthdata.append(len(CTmonth))
            monthlist.append(t)

            # discarding data of month if its less then 1500
            if len(CTmonth) < 1500.0 :
                month.append(t)
                yerr.append(0.0)
                PWVweightedave.append(0.0)
                mincount.append(len(CTmonth))
                maxcount.append(0.0)
```

```
            maxtime.append(t)
            mintime.append(t)

        # proccesing data of month if its equall to 1500 or more
        if len(CTmonth) >= 1500.0 :
            weightedave=np.mean(CTave)
            PWVweightedave.append(weightedave)
            CTweightedstd=np.std(CTave)
            yerr.append(CTweightedstd)
            month.append(t)
            mincount.append(0.0)
            maxcount.append(len(CTmonth))
            maxtime.append(t)
            mintime.append(t)

            # plotting histo plot of data
            plt.figure("Monthly Histogram of PWV")
            plt.hist(CTave, color = 'purple', edgecolor = 'black',bins = bin_edges,alpha=1,cumulative=1,normed=True)
            plt.title("PWV Histogram for %s in %i at H.E.S.S." %(t,2015+j))
            plt.xlabel(" Percipitable Water Vapour (mm)")
            plt.ylabel("fraction [%]")
            plt.savefig('CTplots/Hist/Monthly/PWV histograms of month %s year %i.jpeg' %(t,2003+j))
            plt.show()


    # getting number of data taken for each month with records and writting them to a file
    alldatafile=[i for i in zip(monthlist,allmonthdata)]
    with open('Aeronetplots/datacount/csvfile/alldatafile_{0}.csv'.format(j+2015),'w',newline='') as f:
        w = csv.writer(f)
        w.writerow(['month of year','data points taken'])
```

```
            w.writerows(alldatafile)

        # plotting number of data recorded per month of year
        fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.scatter(monthlist,allmonthdata,color='purple')
        ax.set_title('number of monthly recordings at H.E.S.S. in %i' %(2015+j))
        ax.set_ylabel('number of recordings')
        ax.set_xlabel('Month')
        ax.grid(True)
        fig.autofmt_xdate(rotation=45)
        fig.tight_layout()
        fig.savefig('Aeronetplots/datacount/plots/recordcount%i.jpg' %(2015+j))
        fig.show()

        maxcount=zero_to_nan(maxcount)
        mincount=zero_to_nan(mincount)
        # plotting considered number of points
        fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.scatter(maxtime,maxcount,color='purple',label='considered value')
        ax.scatter(mintime,mincount,color='grey',label='Unconsidered value')
        ax.set_title('number of considered monthly recordings at H.E.S.S. in %i' %(2015+j))
        ax.set_ylabel('number of recordings')
        ax.set_xlabel('Month')
        ax.grid(True)
        ax.legend()
        fig.autofmt_xdate(rotation=45)
        fig.tight_layout()
```

```
        fig.savefig('Aeronetplots/datacount/plots/greyrecordcount%i.
jpg' %(2015+j))
        fig.show()

    yerr=zero_to_nan(yerr)
    PWVweightedave=zero_to_nan(PWVweightedave)
    PWVstdtime=[i for i in zip(month,PWVweightedave,yerr) if str(i
[1]) != 'nan']

    scPWVweighted=[i*alpha for i in PWVweightedave]
    scPWVstdtime=[i for i in zip(month,scPWVweighted,yerr) if str(i
[1]) != 'nan']


    avePWV=[]
    aveSTD=[]

    scavePWV=[]
    scaveSTD=[]

    for i in months:
        avestd=[(k[0],k[1],k[2]) for k in PWVstdtime if k[0]==i]
        allPWV=[i[1] for i in avestd ]
        allstd=[i[2] for i in avestd ]
        avePWVave=np.average(allPWV,weights=allstd)
        stdallstd=np.std(allPWV)
        avePWV.append(avePWVave)
        aveSTD.append(stdallstd)

        scavestd=[(k[0],k[1],k[2]) for k in scPWVstdtime if k[0]==i]
        scallPWV=[i[1] for i in scavestd ]
        scallstd=[i[2] for i in scavestd ]
        scavePWVave=np.average(scallPWV,weights=scallstd)
        scstdallstd=np.std(scallPWV)
        scavePWV.append(scavePWVave)
```

```
        scaveSTD.append(scstdallstd)


    # plotting monthly mean PWV at h.e.s.s. site
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(month,PWVweightedave,color='grey',label='monthly weighted value')
    ax.errorbar(months,avePWV,yerr=aveSTD,color='blue',fmt='o',label='specific month weighted averages')
    ax.set_title('H.E.S.S Weighted monthly averages')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('Month')
    ax.grid(True)
    ax.legend()
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('Aeronetplots/AveragedpointHESS.jpg')
    fig.show()

    # plotting weighted mean PWV scaled for Mt Gamsberg
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(month,scPWVweighted,color='grey',label='monthly weighted value')
    ax.errorbar(months,scavePWV,yerr=aveSTD,color='red',fmt='o',label='specific month weighted averages')
    ax.set_title('Mt Gamsberg Weighted monthly averages')
    ax.set_ylabel('PWV (mm)')
    ax.set_xlabel('Month')
    ax.grid(True)
    ax.legend()
    fig.autofmt_xdate(rotation=45)
    fig.tight_layout()
    fig.savefig('Aeronetplots/AveragedpointGams.jpg')
```

```python
fig.show()


yerr=zero_to_nan(yerr)
PWVweighted=zero_to_nan(PWVweightedave)



# plotting seasonal variations plot at H.E.S.S. site
fig = plt.figure()
ax = fig.add_subplot(111)
n=12
q=[PWVweighted[i:i+n] for i in range(0, len(PWVweighted), n)]
x = np.arange(1,13)
markers=['x','*','o']
colors = cm.rainbow(np.linspace(0, 1, len(q)))

for j,y, c,m in zip(YCT,q, colors,markers):
    ax.plot(months, y, color=c,linestyle='-',marker=m,label="%i"%(2015+j))
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.5, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
ax.set_title('H.E.S.S. weighted Means')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('Month of the year')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig('Aeronetplots/H.E.S.S.monthlyave.png')
fig.show()

# plotting scaled values for Mt Gamsberg (seasonal variations)
fig = plt.figure()
ax = fig.add_subplot(111)
```

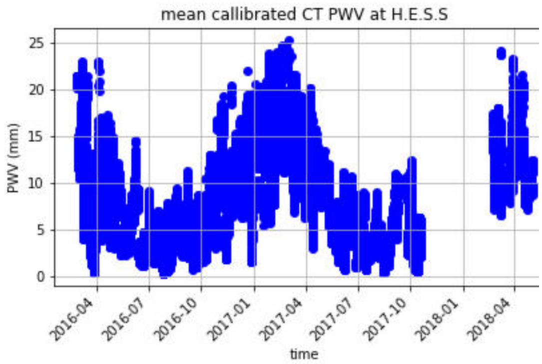
```
n=12
q=[scPWVweighted[i:i+n] for i in range(0, len(scPWVweighted), n)]
x = np.arange(1,13)
markers=['x','*','o']
colors = cm.rainbow(np.linspace(0, 1, len(q)))

for j,y, c,m in zip(YCT,q, colors,markers):
    ax.plot(months, y, color=c,linestyle='-',marker=m,label="%i"%(2015+j))
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.5, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
ax.set_title('Mt Gamsberg weighted Means')
ax.set_ylabel('PWV (mm)')
ax.set_xlabel('Month of the year')
ax.grid(True)
fig.autofmt_xdate(rotation=45)
fig.tight_layout()
fig.savefig('Aeronetplots/MountGamsaveCT.png')
fig.show()
```

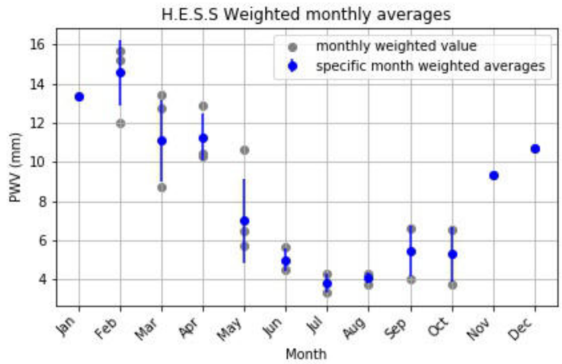
Appendix D

AERONET monthly plots

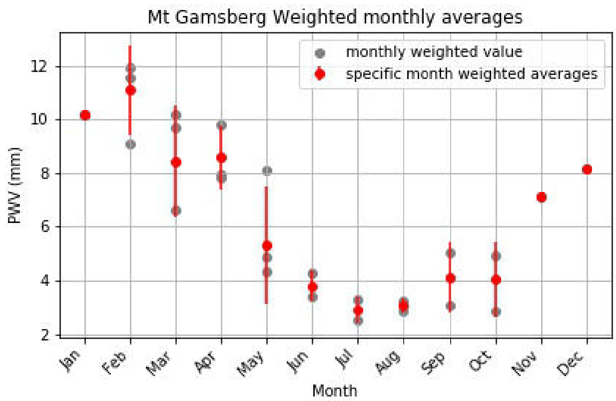
Monthly evaluations for AERONET PWV representing daytime values in contrast to CT values which represent night time values.



(a) PWV recorded at H.E.S.S. site.



(b) Monthly mean PWV at H.E.S.S. site.



(c) Monthly mean PWV scaled for Mt Gamsberg.

Figure D.1: AERONET monthly data analysis for both H.E.S.S. site and scaled values for Mt Gamsberg. AERONET PWV represents daytime values.