



UNIVERSITY OFTM
KWAZULU-NATAL

INYUVESI
YAKWAZULU-NATALI

**DEVELOPMENT OF AN INTEGRATED MODEL AND
SYSTEM TO ENABLE OPTIMAL EFFICIENCY OF
THE HARTRAO LLR SIGNAL PATH**

SPHUMELELE COLIN NDLOVU

2017

Development of an Integrated Model and System to enable Optimal Efficiency of the HartRAO LLR Signal Path

by

Sphumelele Colin Ndlovu

MSc (Physics)

206522092

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy,
in Land Surveying (Geomatics)

School of Engineering,
College of Agriculture, Engineering and Science,
University of KwaZulu-Natal
Durban
South Africa
2017

Under the supervision of,

Supervisor: Prof Ludwig Combrinck

Co-supervisors: Dr Mulemwa Akombelwa and Prof Naven Chetty

Quote

“One never notices what has been done; one can only see what remains to be done.”

- Marie Curie

Preface

The research contained in this thesis was completed by the candidate while based at the Hartebeesthoek Radio Astronomy Observatory under the Space Geodesy Programme and registered under the Discipline of Land Surveying, School of Engineering of the College of Agriculture, Engineering and Science, University of KwaZulu-Natal, Howard College Campus, South Africa. The research was financially supported by the South Africa's National Research Foundation (NRF) and the Department of Science and Technology (DST).

The contents of this work have not been submitted in any form to another university and, except where the work of others is acknowledged in the text, the results reported are due to investigations by the candidate.

Declaration 1: Plagiarism

I,, declare that

1. This research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
 - a. Their words have been re-written but the original information attributed to them has been referenced
 - b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed:

Date:

As the candidate's supervisor I agree/disagree to the submission of this thesis.

Name: Prof Ludwig Combrinck

Signed:

Date:

As the candidate's co-supervisor I agree/disagree to the submission of this thesis.

Name: Dr Mulemwa Akombelwa

Signed:

Date:

As the candidate's co-supervisor I agree/disagree to the submission of this thesis.

Name: Prof Naven Chetty

Signed:

Date:

Declaration 2: Publications

The author's role in each paper and presentation is indicated. The symbol (*) indicates the corresponding author.

1. **Ndlovu S*** 2016. How long until we lose the moon? Science Today, 14 November, 2016, Mail & Guardian, Pretoria, South Africa
2. **Ndlovu SC***, Combrinck L, Chetty N, Botha RC and Akombelwa M 2016. Development of an integrated model and system to enable optimal efficiency of the HartRAO-LLR system. Paper presentation to South African Institute of Physics, 61st Annual Conference, 4th to 8rd July, 2016, Cape Town, South Africa. Paper presented by RC Botha.
3. **Ndlovu SC***, Combrinck L, Akombelwa M and Chetty N 2016. Variable Link Equation Parameters and Expected Photon Returns for the HartRAO Lunar Laser Ranger - Improving our Understanding of the Earth-Moon System. *South African Journal of Geology* 119(1), 77-82.
4. **Ndlovu SC***, Combrinck L, Nkosi NP and Botha RC 2015. An Integrated Software Based Analytical Model for the Signal Path Efficiency of the HartRAO Lunar Laser Ranger Optical System. Paper presentation to South African Institute of Physics, 60th Annual Conference, 29th June to 3rd July, 2015, Port Elizabeth, South Africa. Paper presented by SC Ndlovu. (I was awarded the first prize for the best PhD presentation)
5. Dhansay T*, Serper A, Linol B, **Ndlovu S**, Perumal L and de Wit M 2015. Transdisciplinarity within South Africa's global change research: How (well?) are we doing? *South African Journal of Science* 111, 1-4.
6. **Ndlovu SC***, Combrinck L, Akombelwa M and Chetty N 2014, The importance of the HartRAO Lunar Laser Ranging system for modelling

long-term Global Change in the Southern Hemisphere. Paper presentation to the 2nd National Conference on Global Change, 1st to 5th December, 2014, Port Elizabeth, South Africa. Paper presented by SC Ndlovu.

- 7. Ndlovu SC***, Combrinck L, Exertier P, Akombelwa M and Chetty N 2014. An Estimation of the Number of Expected Returned Photons for the HartRAO Lunar Laser Ranger System. Paper presentation to the 19th International Workshop on Laser Ranging, 27th to 31st October, 2014, Annapolis, MD, USA. Paper presented by SC Ndlovu.

Signed:

Date:

Abstract

The Lunar Laser Ranger (LLR) system under development at the Hartebeesthoek Radio Astronomy Observatory (HartRAO) in South Africa is being built to accurately measure the Earth-Moon distance (at 1 cm level) through the use of short laser pulses, a single photon detection system, an accurate timing system and other sophisticated components. This LLR system is unique in Africa and indeed in the entire Southern Hemisphere. The system utilizes a 1 m diameter optical telescope, which was donated to the project by the Observatoire de la Côte d’Azur of France. In this work, the author discusses the development of an integrated model that will be utilized to obtain optimal efficiency of the HartRAO-LLR system. The model is used to estimate the expected number of returned photons by considering a number of parameters which affects the laser beam pulses as they traverse the atmosphere from the LLR telescope to the Moon and back to the telescope. Factors such as the apparent Earth-Moon range, atmospheric extinction, laser beam characteristics, optical path efficiencies and others, affect the estimated (predicted by software) and actual (measured) number of returned photons for the HartRAO-LLR station. The estimated average signal return rate (which is dependent on a number of factors) of the HartRAO-LLR ranges between 0 to 12 photons per minute, which is in agreement with the available data from five globally distributed LLR stations. It also correlated with the estimated returns that were obtained using least squares parameter estimations. They were in agreement by an average difference of 0.00272. Our estimated signal returns are strongly affected by two-way atmospheric extinction (atmospheric and cirrus cloud transmissions), variations in the laser beam incident angle on the retroreflectors located on the Moon as well as the varying Earth-Moon range. A new parameter, named lunar reflectivity ranging between 0 and 1, was introduced in the link budget equation to consider the effects of Moon Phases on the returned photons.

Modelling the returned number of photons and comparing these to the actual number received leads to an understanding of the effects of numerous variables on the total laser path efficiency. Total system efficiency can be improved as well, as particular atmospheric conditions will not allow LLR to be successful on certain days. For these days, the system can be utilized for other purposes such as maintenance or satellite laser ranging.

Acknowledgements

First and foremost, I want to thank my supervisor Prof Ludwig Combrinck. It has been an honour to be working under his supervision. He has taught me so many things which includes assembling electronic components, developing software for embedded systems, writing for scientific publications and presenting scientific papers at national and international conferences. I appreciate all his contributions of time, ideas, and funding to make my PhD experience productive and stimulating. I am also thankful for the excellent example he has provided as a successful geodesist and professor.

My co-supervisors, Dr Mulemwa Akombelwa (School of Engineering, UKZN) and Prof Naven Chetty (School of Chemistry and Physics, UKZN) are thanked for their contribution in this work.

A special dedication to my beautiful partner, Promise Shabangu, and our two sons, Siyanda and Siyabulela Ndlovu. Their encouragement and understanding provided me with an opportunity to still belong to the family regardless of the number times I could not have breakfasts or suppers with them. I will always be grateful to them for their continuous support. I will also never forget the funny discussions I had with Promise about space. I consider her presence in my life as an achievement on its own.

I thank people from the South Africa's National Research Foundation (NRF) and the Department of Science and Technology (DST) for awarding me with the doctoral Professional Development Programme (PDP). Mrs Gaaratwe Dibetle, former HR officer at the Hartebeesthoek Radio Astronomy Observatory, is also thanked for making sure that I always received my PDP salary on time. Miss Zanele Ntshakala, former SCM officer at HartRAO, is also thanked for her assistance with the procurements of electronic components and both my international and national travels.

Mr André van der Merwe, Mr Attie van Wyk and Mr Richard Moralo, all from the HartRAO’s mechanical workshop, played a role in manufacturing and fine tuning some of my project’s valuable hardware components. I am grateful to the HartRAO’s Space Geodesy group (Nokwazi Nkosi, Cilence Mughemezulu, Paul Barasa, Philemon Tsela and Webby Miyoba). The group discussions and progress meetings helped me to understand more about the HartRAO Lunar Laser Ranger and my role in the project. The Inkaba yeAfrica community, which was also a home to the HartRAO Space Geodesy group, is also thanked for funding some of my conference fees and providing me with an opportunity to present my papers in their workshops. Mrs Elronah Schaap (Smit), Operations Manager for Inkaba yeAfrica (IyA), is thanked for her contributions in making sure that I applied and received my IyA funding. Dr Khadija EL Bouchefry is also thanked for going through the first draft of my thesis. Her comments and suggestions ensured readability of my thesis.

I would also like to thank Mr Keith Jones (Engineering Manager at HartRAO) for allowing me to freely use the electronics workshop. Former HartRAO electronics technicians, Mrs Lerato Nkadimeng and Mrs Charity Moralo, helped me a lot with my electronic components. They taught me many things about electronic components. I also thank Mr Tejas Vedalankar together with the former and current SKA-SA internship students at HartRAO (Donald Mokoena, Gcwelumusa Nduli, Tankiso Moso, Blessing Zondi, Sthembiso Mhlongo, Ntokozo Dlamini and Tshepo Monere) for assisting me with the designs and mounting of some of the electronics components.

My late maternal grandmother, Bafikile maNgubo Khuboni, helped me with the very first university fee of R500 so that I could pay for my university acceptance offer using her pension money. I wouldn’t be where I am if she never used her last

cent to make sure that I enrolled at a university. I will always be grateful to her. May her beautiful soul continue to rest in peace. *Lala uphumule nkosazana yasema Bheleni.*

Last but not least, I thank my very big family, my friends and the community of Emaswazini in Elandskop (*Ngaphezulu*) for their continuous support. Their presence and words of encouragement made, and will always make, me to want more in life.

#EducationMustRise
#UbuntuMustRise
#RacismMustFall
#CorruptionMustFall
#CrimeMustFall

Dedication

I dedicate this work to my family. A special feeling of gratitude to my wonderful mother, Ms Thangithini Khuboni-Ndlovu, for her support and words of encouragement. She taught me and my brother at a very young age that education will be our future. I thank her for being the greatest person in my life. I also thank her for being a wonderful and greatest grandmother to my two sons. My brother, Ntuthuko Ndlovu, believed in me more than I believed in myself. That alone, made me want more in life and keep my head up straight because I knew that someone was learning from me. He never forgot to remind me that I was and still am his role model.

Contents

Preface	ii
Declaration 1: Plagiarism	iii
Declaration 2: Publications	vii
Abstract	ix
Acknowledgments	xi
Dedication	xiv
1 Introduction	1
1.1 Background	1
1.2 Scientific motivation	3
1.3 Aim of the research	4
1.4 Scientific objectives	6
1.5 Laser Ranging	7
1.5.1 Satellite Laser Ranging	11
1.5.2 Lunar Laser Ranging	14
1.5.3 Astronomical seeing	18
1.5.4 Lunar reflector mirrors	24

1.5.5	Effects of Lunar Phases on LLR	27
1.6	Applications of LLR	28
1.7	HartRAO-LLR	30
1.8	Scientific contribution	33
1.9	Thesis overview	35
2	The HartRAO-LLR station	37
2.1	Introduction	37
2.2	HartRAO station	39
2.2.1	Background	39
2.2.2	Location	41
2.2.3	Current role	43
3	HartRAO-LLR system overview and the proposed methods to enable optimal efficiency	45
3.1	Introduction	45
3.2	System overview	46
3.2.1	HartRAO-LLR Laser	49
3.2.2	The timing system	51
3.2.3	The detection system	53
3.2.4	The telescope	54
3.2.5	System monitor and control	55
3.3	Telescope configuration	57
3.3.1	Beam expander method	58
3.3.2	Cassegrain telescope method	64
3.3.3	Coudé telescope method	66
3.3.4	The HartRAO ranging simulator	68

3.4	Measurement challenges	71
4	Simulations and their results	72
4.1	Introduction	72
4.2	System parameters	74
4.3	Least Squares simulations	80
4.4	Parameter estimation	82
4.4.1	System value	83
4.4.2	The reflector optical cross section	84
4.4.3	Atmospheric transmission effects	87
4.4.4	Cirrus cloud effects	89
4.4.5	Lunar reflectivity effects	92
4.5	Expanded radar link equation	95
4.6	Returned photons data	97
5	Discussion	106
5.1	Introduction	106
5.2	Uncertainty sources	107
5.2.1	The HartRAO-LLR laser	112
5.2.2	Detectors	113
5.2.3	Lunar librations	114
5.2.4	Retroreflectors	117
5.2.5	Range	120
5.2.6	Lunar reflectivity	124
5.2.7	Signal-to-Noise ratio	127
5.3	Model implementations	135

6 Conclusion	137
6.1 Summary	137
6.2 Concluding remarks	140
6.3 Future work	141
References	143
Appendix A - Microcontroller source code	157
Appendix B - Graphical User Interface source code	172
Appendix C - Simulator source code	193

List of Figures

1.1	The SLR (MOBLAS6) tracking a satellite from HartRAO. The green laser is visible as the eye (and camera) integrates the short pulses.	12
1.2	The LLR stations that currently carry out laser ranging to retroreflector arrays on the Moon surface. Adapted from: (NASA 2014)	16
1.3	The relationship between turbulence strength and altitude using the general H-V model of changing parameters, three values of moderate to strong wind speed ($w = 10, 21, \text{ and } 30 \text{ m/s}$) and two values of nominal ground turbulence levels $A = 1.7 \times 10^{-14}$ and $A = 1.7 \times 10^{-13} \text{ m}^{-2/3}$ (Andrews and Phillips 2005).	22
1.4	(a) Corner cube prisms or retroreflectors. (b) Apollo 14 retroreflector square array which is similar to that of Apollo 11. (c) Hexagonal Apollo 15 retroreflector array. (d) The Lunokhod that the Soviets landed on the Moon.	24
1.5	The locations of the five retroreflectors on the Moon surface. Source: NASA	25
1.6	The lunar reflectors statistics with the Apollo 15 as the highest reflector. Source: NASA, Accessed on the 18th of June 2013.	27

1.7	The Apache Point Observatory LLR distribution of returns for both day and night observations.	28
1.8	The diagram of the HartRAO-LLR system. Laser pulses for both SLR and LLR propagate through the same coudé path. The returned signal is detected at the back of the telescope. Adapted from (Nkosi 2015)	32
1.9	The ex-OCA 1 m telescope is now installed on its foundations at HartRAO. (Source: Combrinck 2011).	33
2.1	International Laser Ranging Service SLR stations with only two stations in Africa (closed Helwan in Cairo, Egypt and the current active site at Hartebeesthoek in Johannesburg, South Africa). Source: NASA	40
2.2	A Google map view of the Hartebeesthoek Radio Astronomy Observatory (25.8900° S, 27.6853° E). The 26 m radio telescope is clearly visible in near mid-centre, to the west of Moblas6 and the LLR telescope run-off shelter.	41
2.3	The collocated geodetic techniques of HartRAO; GNSS receiver, VLBI telescopes (26-m and 15-m), SLR and LLR. The LLR control room is in front of the SLR and the LLR run-off enclosure is to the right. The anemometer and MET4 meteorological station support the geodetic measurements.	44
3.1	Basic description of the main components and signal path (optical and electrical) of the LLR system.	47

3.2	A Pololu MC3390 dual motor control (in green) before being assembled together with the Arduino Mega2560 (in blue) to control a DC motor. The accurate position of the motor can be obtained via the incremental rotary encoder and positions can be displayed in the 2X16 Liquid Crystal Display or the GUI.	48
3.3	HartRAO's new LLR (100 mJ) and SLR (0.45 mJ) laser sources. The enclosures prevent dust contamination of the laser optics. These laser sources never operate simultaneously as they both use the same signal path to range to targets.	50
3.4	Schematic diagram of the fundamental architecture of hardware components and connections of the HartRAO-LLR.	53
3.5	Schematic diagram of the measured (green asterisk) and proposed (red asterisk) system parameters of the HartRAO Lunar Laser Ranger.	56
3.6	The interior of the HartRAO-LLR control room, includes a server rack, which is not shown in the figure, the telescope's pointing and testing computer software and a large television screen used to monitor the position of the objects. The day and night world map is displayed on the bigger screen.	57
3.7	Schematic diagram of the adjustable and motorised 0.2 m (aperture diameter) beam expander for the HartRAO-LLR system. Source: OCA	59

3.8	The electronic components (mounted on a metal plate) used for test purposes to control the eyepiece lens of the beam expander. The microcontroller, motor and the incremental rotary encoder play a significant role in making sure that any desired eyepiece lens position is achieved. The integrated Pololu motor controller, LCD screen and relay switches are for position monitoring and motor control purposes.	61
3.9	The RS232 GUI interface which sends commands to the code that is uploaded to the MEGA2560.	64
3.10	Schematic diagram of a Cassegrain telescope representing the proposed optics arrangement for the HartRAO-LLR telescope. . . .	65
3.11	Schematic diagram of a transceiver coudé telescope with the secondary and primary mirrors aligned parallel to each other and the flat tertiary mounted in the middle of the telescope.	67
3.12	The Graphical User Interface to estimate the number of returned photons per minute using known and unknown parameters of the HartRAO-LLR system.	70
4.1	Schematic diagram of the Earth-Moon distance determined from the radius of the Earth, slant range and Moon's radius. The green dashed line represents the slant range, R , when measured during LLR, R_E and R_M are in solid green lines representing Earth and Moon radii, respectively.	73
4.2	The relationship between the telescope aperture and the angular resolution.	75
4.3	The effects of the atmospheric turbulence on a propagating laser beam at different r_0	77

4.4	The diagram of the relationship between the transmitted beam diameter and the number of returned photons per minute.	78
4.5	The logarithmic graph of the difference of least squares estimates and radar link budget equation estimates over a range of retroreflector cross section. The average difference between these two types of estimates was found to be 0.00272.	82
4.6	The estimated number of returned photons under different weather conditions using the Apollo arrays data on Mirrors Cross-Section. .	85
4.7	The normalised cross-section as a function of incident angle for fused silica retroreflectors. The 50% and 0% efficiency points for fused silica ($n=1.455$) are 13° and 45° respectively.	86
4.8	The expected number of returned photons per minute as calculated for the HartRAO-LLR system; graph (a) was obtained when the atmospheric transmittance was 0.9 under favourable weather conditions and graph (b) when it was 0.02 for unfavourable weather conditions.	88
4.9	Best fit representation curve for any computed single scattering co-albedo.	90
4.10	The relationship between the lunar integral phase function and lunar phase angle (Cox 2002).	92
4.11	The 2006 – 2015 stacked data for Apache Point Observatory LLR distribution of returns for both day and night observations during a lunar month.	94

4.12	The relationship between the lunar reflectance efficiency and the lunar age (days). The plot was derived from the available distribution of the number of Apache returns during the periods 2006-2015.	95
4.13	A comparison of the number of estimated LLR returns from HartRAO where the blue graph indicates that the lunar reflectivity efficiency was not considered (RLE) and the orange graph indicates the inclusion of the lunar reflectivity parameter (ERLE).	96
4.14	The variations in the estimated number of returned photons for HartRAO-LLR system due to changes in the atmospheric transmittance and lunar reflectivity ($L_r = 0.35, 0.70$ and 1.0). . . .	97
4.15	Statistics of the total detected photon returns from each lunar retroreflector array between 1988 and 2015.	100
4.16	The annual (2006 to 2015) total returned photons, for each lunar retroreflector array, obtained from the Apache Point Observatory Lunar Laser-ranging Operation.	102
4.17	The annual (2009 to 2015) total returned photons, for each lunar retroreflector array, obtained from the Observatoire de la Côte d’Azur.	103
4.18	The annual (2003 to 2015) total returned photons, for each lunar retroreflector array, obtained from the Matera Laser Ranging Observatory. There are no data available for Luna 17 and Luna 21 arrays.	104
4.19	The annual (1988 to 2015) total returned photons, for each lunar retroreflector array, obtained from the McDonald Observatory. . . .	105

5.1	The GUI of the transmitted and additional data in the developed HartRAO mathematical tool that estimates the number of returned photons.	109
5.2	Lunar Laser Ranging measurement setup with basis vectors. Source: Hofmann et al. 2010	110
5.3	Retroreflector orientation difference as it depends on the lunar libration and on the initial orientation of the array as compared to the mean orientation of the Earth centre as seen from the Moon. . .	116
5.4	The schematic diagram of lunar retroreflector mounting.	118
5.5	A comparison of the number of LLR returns when ranging from the OCA station to all five retroreflector arrays from 2009 to 2015. . .	120
5.6	The relationship between the slant range variations and the number of LLR returns for Apache Point Observatory. The data used was for year 2015. The function ($y = -7 \times 10^{-25}x^6 + 2 \times 10^{-18} x^5 - 2 \times 10^{-12}x^4 + 1 \times 10^{-6}x^3 - 0.3021x^2 + 48917x - 3 \times 10^9$, where y = number of returns and x = Earth-Moon distance) was generated to estimate the relationship between the slant range and the number of returns.	121
5.7	The stacked averaged data from 2009 to 2015 for the Observatoire de la Côte d'Azur LLR distribution of returns for both day and night observations during a lunar month.	126
5.8	The comparison between the number of LLR returns and the estimated signal-to-noise ratio for the OCA station observations (2009 to 2015).	128

5.9	The quadratic fit residual for lunar return photons presented as range offset (ns) over time (s). The thin streak is the returned photons embedded in the background of randomly-timed arrivals. Source: PHYSICS-UCSD	129
5.10	A comparison of the annual (2006 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the APOLLO system.	131
5.11	The annual (2009 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the OCA LLR system.	132
5.12	The annual (2003 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the Matera LLR system.	133
5.13	The annual (1988 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the McDonald LLR system.	134

List of Tables

1.1	The specifications of the lasers installed at HartRAO for S/LLR. . .	31
2.1	The parameters of the VLBI / Geodetic telescopes at HartRAO (Nickola 2012).	43
3.1	The specifications of the installed laser system at HartRAO for S/LLR.	49
3.2	The specifications timing systems used in different active LLR stations.	51
3.3	The type and specifications of detectors used at 532 nm wavelength for the five active LLR stations.	55
3.4	The incremental rotary encoder electrical and mechanical specifications.	62
3.5	The calculated output laser beam parameters for far field objects. .	66
4.1	The estimated photon returns from different transmitting methods.	74
4.2	The varying parameters that influences the number of photons reflected by Apollo 11.	78
4.3	The atmospheric effects on laser ranging accuracy at different elevations (in degree).	79

4.4	The total number of estimated returned photons for the HartRAO station when varying the Cirrus cloud parameter and fixing all the other parameters, in the radar link equation, at their maximum except the atmospheric transmission which was at 0.41.	91
4.5	Apache Point Observatory individual ranging statistics for each retroreflector (7 April 2009 and 21 December 2016).	93
4.6	Total of the returned photons over time, from the four active LLR stations, for each retroreflector. The estimations from the HartRAO station are also included.	99
5.1	The maximum values for the HartRAO-LLR station positional displacements (pole tide, Earth tide, ocean loading and atmospheric loading). Source: McCarthy and Petit 2003	107
5.2	The fixed and varying parameter values used in the radar link equation to estimate the HartRAO number of returned photons for lunar laser ranging.	108
5.3	Minimum/median/maximum LLR returned photons per minute for each retroreflector array for period 1969 - 2014. The estimated return rate for the HartRAO-LLR system is also included. Source (http://polac.obspm.fr/llrdatae.html)	123
5.4	Statistics of range error per station per pass for nominal and strong turbulence estimation cases. Source: Dirkx et al. 2015	124

Nomenclature

APD	Avalanche Photodiode Detector
APOLLO	Apache Point Observatory Lunar Laser-ranging Operation
CCR	Corner Cube Retroreflector
CMB	Core/Mantle Boundary
COM	Centre-of-Mass
CSIR	Council for Scientific and Industrial Research
DORIS	Doppler Orbitography and Radiopositioning Integrated by Satellite
DST	Department of Science and Technology
EOP	Earth Orientation Parameters
FRD	Foundation for Research Development
FWHM	Full-Width Half Maximum
G	Gravitational Constant
GNSS	Global Navigation Satellite Systems
GSFC	Goddard Space Flight Center

HartRAO	Hartebeesthoek Radio Astronomy Observatory
IDS	International DORIS Service
IfE	Institut fuer Erdmessung
IGS	International GNSS Service
ILR	Interplanetary Laser Ranging
ILRS	International Laser Ranging Service
ITRF	International Terrestrial Reference Frame
LASER	Light Amplification by Stimulated Emission of Radiation
LLR	Lunar Laser Ranging
LRR	Lunar Retroreflector
LURE	Lunar Ranging Experiment
M	Mass of the Earth
MCP	Micro-Channel Plate
MLRO	Matera Laser Ranging Observatory
MLRS	McDonald Laser Ranging System
MTBF	Mean Time Between Failures
NASA	National Aeronautics and Space Administration
Nd:YAG	Neodymium-Doped Yttrium Aluminum Garnet
NGSLR	Next Generation Satellite Laser Ranging

NP	Normal Point
NRF	National Research Foundation
OCA	Observatoire de la Côte d’Azur
PDF	Probability Distribution Function
S/LLR	Satellite/Lunar Laser Ranger
S/LLR	Satellite/Lunar Laser Ranger
SEP	Strong Equivalence Principle
SRF	Selenocentric Reference Frame
TDC	Time-To-Digital Converter
TIR	Total Internal Reflection
TOF	Time-of-Flight
TRF	Terrestrial Reference Frame
USA	United States of America
VGOS	VLBI Global Observing System
WEP	Weak Equivalence Principle

Chapter 1

Introduction

1.1 Background

A state-of-the-art Lunar Laser Ranging (LLR) system, with Satellite Laser Ranging (SLR) capabilities which are commonly known as Next Generation SLR (NGSLR), is under development at the Hartebeesthoek Radio Astronomy Observatory (HartRAO) in South Africa. It consists of a 1 m Cassegrain telescope and associated peripheral equipment. The telescope was donated to the project by the Observatoire de la Côte d’Azur (OCA) located in France and it is currently the only one of its kind in Africa and indeed the entire Southern Hemisphere. It will measure the distance to the Moon through the use of laser pulses. The measurements that will be obtained from the LLR system will be used for studies of Earth-Moon dynamics and tests of the Special and General Theories of Relativity.

The five active LLR stations in the world are OCA, Apache Point Observatory Lunar Laser-ranging Operation (APOLLO) in the United States of America (USA), Matera Laser Ranging Observatory (MLRO) in Italy, Wettzell

Observatory in Germany and McDonald Observatory in the USA, which are all located in the Northern Hemisphere. The HartRAO-LLR system is expected to improve data coverage to the International Laser Ranging Service (ILRS) and strengthen the geometry of the network. However, there are many factors that can affect the HartRAO-LLR data quantity and quality. These factors can result from deformations of the receiving/transmitting optics, atmospheric perturbations, site selection and quality of the laser pulses (Ndlovu et al. 2016; Nkosi et al. 2016).

In laser ranging, laser pulses propagate through fluctuating atmospheric conditions that can influence laser beam divergence. A transmitted laser pulse with a beam divergence of 3 to 4 arcseconds, as observed from the ground station on Earth, can spread to a diameter of approximately 7 km on the Lunar/Moon surface (Dickey et al. 1994). The optical retroreflector mirrors placed on the lunar surface during the Apollo missions (Apollo arrays 11, 14 and 15) and Lunokhod missions (Lunokhod arrays 17 and 21) reflect transmitted laser pulses back to Earth. The distance, the small size and design of the mirrors contribute to the weak returned signal but still detectable with the use of advanced photon detection systems placed inside the “Earth-fixed” receiving telescope. It is assumed that these mirrors are covered by dust as the returns are lower than expected (Wohl et al. 2011).

The aim of this work, which will be more comprehensively discussed in Chapter 2, was to develop an integrated model and system that will enable optimal efficiency for the HartRAO Lunar Laser Ranging signal path. The latest available engineering capabilities have allowed significant developments such as large telescopes, adaptive optics, physical computing, data analysis software etc. To date, the effects of atmospheric turbulence on propagating laser beams are

still a major problem in the space geodetic community, especially when ranging to distant objects. The mostly considered efficient method of minimising the effects of atmospheric turbulence effects on laser ranging is to select a site with better astronomical seeing conditions, i.e. clear, stable skies (minimal atmospheric turbulence) and favourable weather conditions. The use of the above mentioned method together with quality optics and a “perfectly” collimated laser beam can enable optimal efficiency of a Lunar Laser Ranger signal path.

1.2 Scientific motivation

Accuracy and range precision in Lunar Laser Ranging are the most important factors that yield improvements especially in the determination of fundamental physics parameters such as the Weak Equivalence Principle (WEP), Strong Equivalence Principle (SEP), de Sitter precision and the time variation of Newton’s gravitational constant G (Murphy et al. 2001). Combrinck (2007) initiated a HartRAO-LLR project that will add to the current effort of precisely determining Earth-Moon distances to sub-centimetre accuracy. The HartRAO-LLR project is expected to contribute to improvement in the determination of the aforementioned parameters through improved LLR network geometry and high accuracy ranging data.

Through LLR, it has been determined that the Moon’s orbit is not displaced by a violation of the SEP at approximately 5 mm level of precision, based on 2-3 cm raw range precisions, constraining the SEP violation at the level of about 5 parts in 10^4 (Murphy 2000). Thus an LLR station with high-quality data should allow high level (less than 1 cm) determinations of the SEP-induced polarisation of the lunar orbits, thereby achieving a few parts in 10^5 precision on the potential SEP. Physics theories such as string-motivated theories, quintessence, and other

alternatives to General Relativity, predict that the violation of the SEP results from new particles like dilaton (hypothetical particle with extra dimensions) and moduli (geometric space) that are coupled with gravitational strength (Murphy 2000).

Available engineering capabilities have allowed significant developments such as large telescopes, adaptive optics, physical computing, data analysis software etc. While the effect of atmospheric turbulence on propagating laser beams is still a major problem in the optical community, especially when ranging to distant objects. The other effects are attributed to the target accuracy, distribution of detection times, retroreflector tilts and optical path. These factors can result in a returned signal which is weak but still detectable if the advanced single photon detection systems are placed inside the Earth “fixed” receiving telescope and aligned properly.

1.3 Aim of the research

The aim of this work was to enable optimal signal-path-efficiency for the HartRAO-LLR system by developing an integrated software-based analytical model and optical system. This integrated model and system used the existing link budget equation to estimate the number of returned photons for given conditions and LLR system parameters. The equation is one of the essential mathematical tools that can be considered when developing an integrated system and model for LLR. This mathematical tool can be used to estimate and visualize the relationship between the number of returned photons (observed and computed) and the varying link budget equation parameters.

The existing link equation can be used to predict the laser ranging system

efficiency and is based on an assumed accuracy of all parameters which influence the returned signal, presented as an estimate of expected number of returned photons. However, an introduction of the lunar reflectivity parameter in the equation will help to improve our understanding of the signal-to-noise ratio. Enabling optimal efficiency in the HartRAO-LLR signal path will yield an improvement in the return-energy of the laser so that ranges to the lunar corner cube retroreflectors can be measured accurately.

Since the current LLR development is currently the only of its kind in the Southern Hemisphere and the entire Africa, it must be developed such that it meets the requirements of ILRS, which applies to any LLR developments in the world. The requirements are drafted on the ILRS website (<http://ilrs.gsfc.nasa.gov/network>) as:

- ▷ ILRS stations will range only to satellites for which they have authorization from the ILRS or the satellite owner
 - ILRS authorization is constituted by an approved Mission Support Request Form
- ▷ ILRS stations will adhere to the Restricted Tracking Procedures which are;
 - The corner cubes may not be visible under certain geometric conditions. Tracking outside the effective pass interval does not harm the satellite but no returns are possible.
 - Some satellites are equipped with optical sensors that may be damaged by the SLR laser beam if the station is within the field of view of this optical sensor.
 - The level of power some satellites' detectors can handle is limited. The

station must reduce the power delivered to the satellite detectors to a safe level or forgo ranging the satellite.

- ▷ ILRS stations will keep their site logs (ftp site) and configuration files (ftp site) current.
- ▷ ILRS stations will maintain aircraft avoidance procedures and will document these procedures on the ILRS site logs.
- ▷ ILRS stations will adhere to the ILRS Data Product Delivery requirements.

1.4 Scientific objectives

The effect of thermal and density fluctuations on propagating laser beams was one of the main areas of interest in this study. Therefore, it was necessary to develop scientific objectives which would be relevant to investigating how to minimise these effects and so to enable optimal efficiency of a lunar laser ranger signal path for the HartRAO-LLR. The objectives were to:

1. Develop an advanced mathematical tool, utilising C++ code, which will estimate the number of returned photons that are initially transmitted as laser pulses through an “Earth-fixed” telescope to fall on, and get reflected by, the retroreflector arrays placed on the Moon surface. The primary output will be a scientifically accurate reflection of all parameters which will provide a scientifically justifiable estimate of the expected photon return success rate.
2. Expand the existing link equation to consider the effects, which can be caused by the lunar phases as the photons are returned from the Moon surface through the LLR signal path. This is expected to improve the system signal-to-noise ratio.

3. Develop a model and data analysis software to enable correct beam expander settings for precise beam collimation through the atmosphere.

The combination of these objectives will result in a new, holistic approach towards an integrated S/LLR system that will facilitate highly accurate range determination. This holistic approach will take into account all factors which modify 'fixed' and variable parameters. It will also enable estimation in a least squares sense of certain parameters (e.g. the optical transparency of Cirrus clouds), which affects the accuracy of the link equation.

1.5 Laser Ranging

In the late 1950's, the Light Amplification by Stimulated Emission of Radiation (commonly known as the LASER) was first introduced (Gordon 1959). Lasers play a significant role in weaponry (McAulay 2011), remote sensing, target designation, adaptive optics, medicine and laser ranging, which is the main focus of this work. In geodesy, laser ranging is defined as a technique that measures the distance between Earth and its orbiting artificial satellites (SLR) or the Moon (LLR) using short laser pulses. This technique is considered as the most accurate for obtaining highly-precise distance measurements that can be used in geodetic analysis (Anderson et al. 1996; Combrinck 2010).

During laser ranging, a pulsed laser light is transmitted through a ground telescope (or a beam expander) to fall on retroreflector equipped Earth-orbiting artificial satellites or one of the five retroreflector arrays placed on the lunar surface. These retroreflector arrays, also known as corner cube retroreflector (CCR) arrays, ensure that any transmitted laser signal that falls on its surface can be reflected

back to Earth and directed to a photon detection system, which measures the round trip time-of-flight (TOF). The measured distance, s , between the two objects is approximately calculated from (Bender et al. 1973),

$$s = \frac{1}{2} (c \times t), \quad (1.1)$$

where c is the speed of light in vacuum and t is the round trip TOF, which is about 2.5 seconds for Lunar Laser Ranging.

Degnan (1993) used the radar link budget equation to calculate the expected photons for SLR systems. This equation can also be used to estimate the number of received photons considering various thermal and system parameters. The link equation is the mathematical tool that is used to calculate the mean number of photoelectrons, n_{pe} , received by the photon detector (Degnan 1993), and is written as,

$$n_{pe} = \eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \sigma \left(\frac{1}{4\pi R^2} \right)^2 A_r \eta_r T_a^2 T_c^2, \quad (1.2)$$

where,

- η_q is the detector quantum efficiency,
- E_T is the laser pulse energy in Joules,
- h is the Planck's constant = $6.6260695 \times 10^{-34}$ m² kg/s,
- $\lambda = 532$ nm is the laser wavelength,
- $c = 299792458$ (m/s) is the speed of light in vacuum,
- η_t is the transmit optics efficiency,
- G_t is the unit-less transmitter gain,
- σ (m²) is the satellite optical cross-section,
- R is the slant range to the target in km,

- $A_r = \pi r^2$ is the effective area of the telescope receive aperture,
- η_r is the efficiency of the receive optics,
- T_a is the one-way atmospheric transmission and,
- T_c is the one-way transmissivity of cirrus clouds.

The calculated slant range, R , for Earth-orbiting satellites is calculated from ground to target using (Degnan 1993):

$$R = -H \cos \theta_{zen} + \sqrt{H^2 \cos^2 \theta_{zen} + 2R_E(h_s + h_t) + h_s^2 - h_t^2}, \quad (1.3)$$

where $H = R_E + h_t$ is a combination of the radius of Earth (R_E) and the station height (h_t) above sea level, h_s is the satellite altitude above the sea level, and θ_{zen} is the zenith angle (complement of the elevation angle E) of the satellite as observed from the ground station. The equation for determining the Earth-Moon distance is complicated and requires several iterations. For data analysis purposes an approximate range as described by Dickey et al. (1994) can be expressed as,

$$R = |\mathbf{r}| + \mathbf{R}_M \cdot \hat{\mathbf{r}} - \mathbf{R}_E \cdot \hat{\mathbf{r}}, \quad (1.4)$$

where r is the distance between the surfaces of Earth and Moon. The average radii of the Earth (R_E) and Moon (R_M) are 6371 km and 1700 km, respectively.

The transmitter gain (no obscuring structures present) and the general transmitter gain (obscuring structures are present) depend on the laser divergence and pointing error (which is the measure of accuracy for whether the laser points where it is supposed to) as provided by Degnan (1993) as,

$$G_t(\theta) = \frac{8}{\theta_t^2} \exp \left[-2 \left(\frac{\theta}{\theta_t} \right)^2 \right], \quad (1.5)$$

where θ_t is the far field divergence half-angle between the beam centre and the $1/e^2$ intensity point and θ is the beam pointing error. In this work, Equation (1.5) was used for SLR calculations and the general transmitter gain was used for LLR calculations to accommodate obscuring structures present. The general transmitter gain described by Degnan (1993), is written as,

$$G_t = \frac{4\pi A_t}{\lambda^2} g_t(\alpha_t, \beta = 0, \gamma_t, X = 0), \quad (1.6)$$

where $A_t = \pi a_t^2$ is the area of the transmitting aperture and $g_t(\alpha_t, \beta, \gamma_t, X)$ is a geometric factor which is fully described by Klein and Degnan (1974) as,

$$g_t(\alpha_t, 0, \gamma_t, 0) = \left(\frac{2}{\alpha_t^2} \right) \left(e^{-\alpha_t^2} + e^{-\gamma_t^2 \alpha_t^2} \right), \quad (1.7)$$

where $\alpha_t = \frac{a_t}{w}$ is the radii ratio of the primary transmitting aperture (a_t) to the Gaussian beam waist (w), the point along the propagation direction where the laser beam radius is minimum, at the transmitting primary, $\gamma_t = \frac{b_t}{a_t}$ is the quotient of the radius of an obscuring secondary mirror (b_t), (if any) and the radius of the primary transmitting aperture.

Klein and Degnan (1974) reported that in order to maximise the transmitter gain for an arbitrary obscuration ratio, the choice of the transmit primary radius to beam radius ratio is approximated by

$$\alpha_t \sim 1.12 - 1.30\gamma_t^2 + 2.12\gamma_t^4, \quad (1.8)$$

for $\gamma_t < 0.4$. For the HartRAO-LLR telescope, the estimated value for γ_t is 0.3 since $b_t = 0.3$ m and $a_t = 1$ m. By using Equation (1.6), the transmitter gain decreases with a squared increase in wavelength. This illustrates that collimation

for shorter wavelength beams is a stringent requirement in order to strengthen the beam quality for the success of laser ranging. The effect of atmospheric turbulence on laser beam propagation is another complex factor that needs to be considered. Detailed discussion of this effect is beyond the scope of this research. However, for full details on the effects of atmospheric turbulence on laser beam propagation, the interested readers are referred to (Obukhov 1941; Tatarski 1961; Fried 1965 and 1966; Roddier 1981 and 1989)

1.5.1 Satellite Laser Ranging

The SLR technique originates back to the early 1960's. The very first successful experiments were conducted at NASA-Goddard Space Flight Center (GSFC) in Greenbelt, Maryland. The experiments were led by Dr. Henry H. Plotkin. Plotkin's team, included Dr. John Degnan as a researcher, worked with a ruby laser which was part of the first GSFC laser tracking systems (NASA 2014). The results of these experiments were reported by Smith (1964). The SLR technique measures the distance between the ground station and the artificial Earth-orbiting satellites equipped with special retroreflector mirrors. Within a global network, SLR stations provide instantaneous range measurements at millimetre level precision which can be accumulated to provide accurate orbits (Degnan 2012). The success of SLR has contributed to studies of Earth, atmosphere and ocean systems (Botai et al. 2015).

In the past three decades, the global SLR network has evolved into a powerful source of data for studies of the solid Earth and its ocean and atmospheric systems (McGarry 2012). In addition, SLR provides precise orbit determination for spaceborne radar altimeter missions mapping the ocean surface (which are used to model global ocean circulation), for mapping volumetric changes in

continental ice masses, and for land topography. It provides a means for sub-nanosecond global time transfer, and a basis for special tests of the Theory of General Relativity.

The Mobile Laser Ranging System - 6 (MOBLAS6) SLR located in South Africa (Figure (1.1)) contributes to the ILRS data. It arrived at HartRAO in June 2000 from the USA. Its installation and collocation tests took place between June and August 2000, followed by on-site and off-site (at NASA GSFC) staff training. This is an ongoing operation with collaboration between NASA and HartRAO. The system uses a 0.75 m Cassegrain telescope built on a 12 m mobile trailer. The trailer houses the 100 mJ, 532 nm, 180 ps pulse length laser and peripheral equipment.



Figure 1.1: The SLR (MOBLAS6) tracking a satellite from HartRAO. The green laser is visible as the eye (and camera) integrates the short pulses.

The SLR technique helps to accurately determine geocentric position of Earth

satellites, allowing for the precise calibration of radar altimeters and separation of long-term instrumentation drift from secular changes in ocean topography (Cheng and Tapley 2004). It also facilitates measurement of the temporal variations of the Earth’s gravity field and to monitor the motion of the station network with respect to the geocentre together with the capability to monitor vertical motion in an absolute system. This makes it unique for modelling and evaluating long-term climate change (Müller et al. 2008).

Modern SLR systems play a pivotal role in the international laser network. A typical older (developed during the 1970’s and continuously upgraded) SLR system transmits short laser pulses with energy of between 10 mJ and 100 mJ at a pulse repetition frequency ranging between 5 Hz and 20 Hz, whereas modern SLR systems have lower power levels and higher firing rates of up to 2 kHz (Botai et al. 2015). These systems have a much more improved sub-centimetre single shot precision of between 1 mm and 3 mm for the achieved normal points. A normal point (NP) data is defined as the primary ILRS station data product, which gradually replaced on-site sampled data and later fullrate data. These NP data are sometimes referred to as quick-look data, because they are generated very shortly after a laser ranging session and transmitted rapidly (within a few hours) to the data centres.

In 2002, the ILRS Governing Board adopted a minimum data requirement for SLR normal points. The requirements were as follows: daytime NPs and nighttime NPs with minimum of 6 and 3 data points, respectively (<https://ilrs.cddis.eosdis.nasa.gov>). Fewer data points would be acceptable on lower satellites (5-second normal points) from those ranging systems with lower pulse repetition rates where these minimum requirements are not practical. The timing accuracy of SLR equipment is no longer a limiting factor for the absolute

accuracy of the range measurements (Degnan 1993). One major source of error is propagation delays which results from atmospheric perturbations (Dirkx et al. 2015).

1.5.2 Lunar Laser Ranging

The first successful tests of LLR were reported by Smullin and Fiocco (1962) at the Massachusetts Institute of Technology using a laser with millisecond pulse length. The Crimean Astrophysical Observatory also reported on the success of LLR observations using a Q-switched ruby laser, pulsed output laser beam (Grasyuk et al. 1964; Kokurin et al. 1966). The improvements in data accuracy to approximately 25 cm between 1969 and 1975 was achieved following the installation of retroreflector mirrors on the lunar surface during the manned Apollo missions and the unmanned Soviet Lunokhod missions (Bender et al. 1973; Nordtvedt, 1988). All these missions were deployed between 1969 and 1973 (Bender et al. 1973).

The installation of the retroreflectors on the surface of the Moon helped to achieve high-accuracy range measurements, as perceived in 1964. This included a much improved lunar orbit, determination of the location of the retroreflectors with respect to the lunar centre of mass, study of the lunar librations (angular motions about the centre of mass due to gravitational torques on the Moon), determinations of the locations of ground stations on the Earth from which range observations were made and an accurate check on gravitational theory, through a search for deviations from the calculated range after all known parameters in the problem had been adjusted (Bender et al. 1973).

The Apollo 11 and 14 reflector packages are each equipped with 100 solid fused silica corner reflectors, 3.8 cm in diameter and mounted in a 46 cm square aluminium panel. While the Apollo 15 reflector array has similar specifications, it

is three times larger with 300 corner reflectors mounted in a hexagonal panel. The Lunokhod arrays (Luna 17 and Luna 21), which were a series of Soviet Robotic Lunar Rovers, have 14 triangularly faced corner cubes of 11-cm edge. Both Rovers were equipped with TL laser retroreflectors, AF-3L UV/visible astrophotometers to measure light levels and Rubin 1 photodetectors for laser detection experiments. Each Rover had a total height of about 135 cm and a mass of 840 kg. The Rovers had two speeds, approximately 1 and 2 km/h (NASA 2010).

The ILRS observatories (Figure (1.2)) that currently carry out Lunar Laser Ranging are all in the Northern Hemisphere (APOLLO, McDonald, Matera, Wettzell and Grasse). Those LLR stations highlighted in green circles are operational, whereas the inactive LLR stations are Mount Stromlo Observatory in Australia (highlighted in yellow) and the Lunar Ranging Experiment (LURE) Observatory in Hawaii (highlighted in red). The wildfire in Mount Stromlo destroyed the observatory and damaged equipment which included their Lunar Laser Ranger (www.ga.gov.au). This observatory was rebuilt for geodetic measurements but it is currently running without LLR facilities. The LURE Observatory only operated from 1974 until 1994 (<https://ilrs.cddis.eosdis.nasa.gov/network>). The location of the LLR system under development in South Africa is indicated in white on the map.

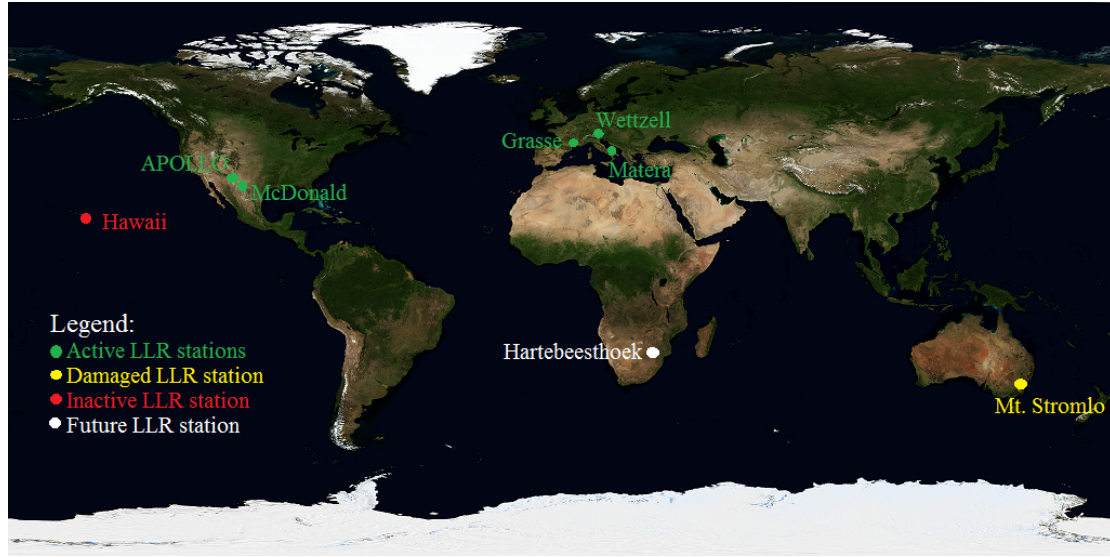


Figure 1.2: The LLR stations that currently carry out laser ranging to retroreflector arrays on the Moon surface. Adapted from: (NASA 2014)

Early range precision of the LLR data was approximately 25 cm from 1970 to 1975, it was followed by a range precision of between 12 cm to 16 cm until 1984. Subsequently, in the mid-1980s the McDonald Laser Ranging System (MLRS) operation was transferred to a dedicated 0.76 m telescope (also used for SLR) with a 200 ps Nd:YAG (neodymium-doped yttrium aluminum garnet) laser operating at 10 Hz and 150 mJ per pulse. This station achieved a range precision of between 2 to 3 cm during the mid-to-late 1980s (Nordtvedt 1988). The MLRS precision has been the precision benchmark for LLR until 1994 and these results were verified by the French LLR system located at OCA in the early 1990s (Dickey et al. 1994). At that time, the Observatoire de la Côte d’Azur used a 1.5 m telescope and a 70 ps Nd:YAG laser firing at 10 Hz and 75 mJ per pulse. This station has been the leading Lunar Laser Ranging station in the world and has contributed about half of the total range measurements to date (Samain et al. 1998). In recent years, these two stations (MLRS and OCA) have been the only contributors to lunar range data with typical return rates and normal points of about 0.002 photons per

pulse and 15 photons for the MLRS and 0.01 photons per pulse for the MLRS and 40 photons for the OCA, respectively (Murphy et al. 2008).

The Wettzell Laser Ranging System in Germany and the Matera Laser Ranging Station in Italy are the two modern stations that have demonstrated lunar capability. Both stations are operational for LLR at present. The Wettzell Laser Ranging System (also used for SLR) uses a 0.75 m telescope on an Alt/Az mount for transmitting and receiving and a 50 ps Nd:YAG laser, with three amplifier stages, firing at 20 Hz and 180 mJ per pulse. The Matera Laser Ranging Observatory (MLRO) subsystems and components were carefully chosen to exploit the best of currently available technology. Their system consists of a 1.50 m Cassegrain telescope and a 50 ps Nd:YAG laser operating at 10 Hz and 100 mJ per pulse. Both stations have contributed to LLR data and also been used for LLR model packages together with the other stations to enable independent high-precision LLR data analysis packages that should be able to fit parameters which are sensitive to LLR measurement, including dynamic parameters of the Earth-Moon system (Tian and Yeh 2016).

The existence of the Apache Point Observatory Lunar Laser-ranging Operation (APOLLO) further improved the measurements to a millimetre precision benchmark consequent of its large 3.5 m telescope aperture size and a multiple photon detection system (Murphy et al. 2008; Battat et al. 2009). The APOLLO system uses a 90 ps full-width at half-maximum (FWHM) Nd:YAG laser operating at 20 Hz and 115 mJ per pulse. It obtains photon return rates of around one photon per pulse, so that the requisite number of photons for 1 mm normal points may be collected on few-minute timescales. To date, the best performance has been approximately 8900 returned photons from the Apollo 15 array in a period of 8 minutes. The average photon return rate for this period is

about 0.9 photons per shot, with peak rates of 2.5 photons per pulse. Approximately 80% of the returned photons arrive in multi-photon bundles with the largest containing 10 photons. Since about 10 years ago, this station brought LLR solidly into the multi-photon regime for the first time (Murphy et al. 2008).

Combrinck (2007, 2011 and 2014) in collaboration with OCA and NASA proposed the development of a new LLR station in the Southern Hemisphere that would also provide Satellite Laser Ranging (SLR). This station will contribute to fundamental physics, the ILRS and to the maintenance of the International Terrestrial Reference Frame (ITRF). After successful development and refurbishment of the equipment, the LLR system will be tested and first operated at HartRAO. Once completed and fully operational, the HartRAO's LLR system will be moved to Matjiesfontein in the Karoo (a semi-desert region in South Africa) where the astronomical seeing conditions are better than at HartRAO, and cloud cover is less, allowing for more efficient ranging.

1.5.3 Astronomical seeing

In this section, a brief discussion on the theory of the effects of atmospheric turbulence on laser ranging is provided. Models of atmospheric turbulence were related to astronomical seeing by Obukhov (1941), Tatarski (1961), Fried (1965 and 1966) and Roddier (1981 and 1989). The literature provides a better understanding of the formation of turbulence in the atmosphere and quantifies its effects on the propagation of light. Atmospheric turbulence quantification data can be used to determine an appropriate site for laser ranging.

The Rytov theory of turbulence states that thermal fluctuations in the atmosphere can cause optical turbulence effects (irradiance scintillation) on a propagating laser beam (Andrews and Phillips 2005; Mayer and Young 2008).

Such effects cause perturbations of the amplitude of the propagating beam along the optical path. However, these perturbations can be solved using the Rytov weak turbulence fluctuation theory (Mayer and Young 2008), which is expressed as,

$$\sigma_I^2 = \frac{\langle I^2 \rangle}{\langle I \rangle^2} - 1, \quad (1.9)$$

where σ_I^2 is the scintillation index (normalised variance of irradiance fluctuations), I is the irradiance of the beam and $\langle \rangle$ denotes an ensemble average. The scintillation index for weak turbulence is less than 1 and it can be identical to the variance of the log intensity, also known as the Rytov variance. These fluctuations are written in (Yuskel 2005) as,

$$\sigma_I^2 = BC_n^2 k^{7/6} L^{11/6}, \quad (1.10)$$

where C_n^2 is the refractive index structure-function commonly known as the strength of turbulence, B is a constant equal to 1.23 for plane waves and 0.5 for spherical waves, k is the wavenumber of the propagating laser beam defined as $2\pi/\lambda$, λ is the wavelength of the laser beam and L is the propagation path length between the transmitter and the target.

The varying atmospheric perturbations on the propagating beam degrades the beam quality and hence, the number of returned photons is reduced along the optical signal path. The optical signal propagation is affected by the Earth's atmosphere. This leads to irradiance fluctuations which can amount to several metres in the measured distance. The normalised variance of irradiance fluctuations depends on three parameters, see Equation (1.10). Assuming that the wavenumber and the optical path length remain constant, the scintillation index becomes proportional to the strength of turbulence. This is due to the

dependence on C_n^2 to the refractive index of air, which is considering the aerosols, thermal and density fluctuations in the atmosphere, is not constant.

Turbulence in the atmosphere is described by three major characteristics; the strength of atmospheric turbulence, C_n^2 , the outer scale of turbulence, L_0 , and the inner scale, l_0 . These characteristics results from thermal and density fluctuations in the atmosphere. A small temperature change associated with the inner and outer scales of turbulence form the lower and upper boundaries of the inertial-convective range (Wang et al. 2011), such that the temperature structure-function, D_T , becomes,

$$D_T(R) = \langle (T_1 - T_2)^2 \rangle = \begin{cases} C_T^2 R^{2/3}, & l_0 \ll R \ll L_0 \\ C_T^2 R^2 l_0^{-4/3}, & R \ll l_0 \end{cases}, \quad (1.11)$$

where D_T is the ensemble average of the squared temperature differences, T_1 and T_2 are the temperatures at two points separated by the distance R and C_T^2 , which is the temperature structure-function.

Temperature-induced fluctuations in the atmosphere have properties of statistical homogeneity and isotropy with the inertial sub-range between L_0 and $l_0 = 5.8(D^3/\epsilon)$, D is the diffusivity of heat in air and ϵ is the average energy dissipation rate. These fluctuations lead to the formation of the refractive index structure-function (Wang et al. 2011), which is written as,

$$D_n(R) = \langle (n(R_1) - n(R_2))^2 \rangle = \begin{cases} C_n^2 R^{2/3}, & l_0 \ll R \ll L_0 \\ C_n^2 R^2 l_0^{-4/3}, & R \ll l_0 \end{cases}, \quad (1.12)$$

where R_1 and R_2 denote the distances at which the refractive index is separated by two points, and the other parameters are as defined in Equation (1.11). In this case, the inner scale of the refractive index is $l_0 = 7.4(\nu^3/\epsilon)$, where ν is the

kinematic viscosity of the fluid. The refractive index structure constant is given by Gladstone's relation (Wang et al. 2011), and is written as,

$$C_n^2 = \left(79 \times 10^{-6} \frac{P}{T^2}\right)^2 \cdot C_T^2, \quad (1.13)$$

where P is the pressure in millibars (mBar) and T is the absolute temperature. The strength of turbulence is classified as weak if $C_n^2 \leq 10^{-17} \text{m}^{-2/3}$, moderate if $10^{-17} < C_n^2 < 10^{-13}$ and strong if $C_n^2 \geq 10^{-13} \text{m}^{-2/3}$ (Mayer and Young 2008).

The turbulence structure-function does not only vary with thermal and density fluctuations but also with height above the ground and time of the day. A commonly used optical model for profiling the strength of the atmospheric turbulence parameter is the Hufnagel-Valley turbulence model (Wang et al. 2011), written as,

$$C_n^2(h) = 0.00594 \left(\frac{w_{rms}}{27}\right)^2 \left(10^{-5}h\right)^{10} e^{-h/1000} + 2.7 \times e^{-h/1500} + A e^{-h/100}, \quad (1.14)$$

where h is the height above the ground, w_{rms} is the root-mean-square wind speed and $A = C_n^2(0)$ is the ground level turbulence strength function. The ground turbulence level has little effect above 1 km and the wind speed governs the profile behaviour primarily in the vicinity of 10 km (Figure (1.3)).

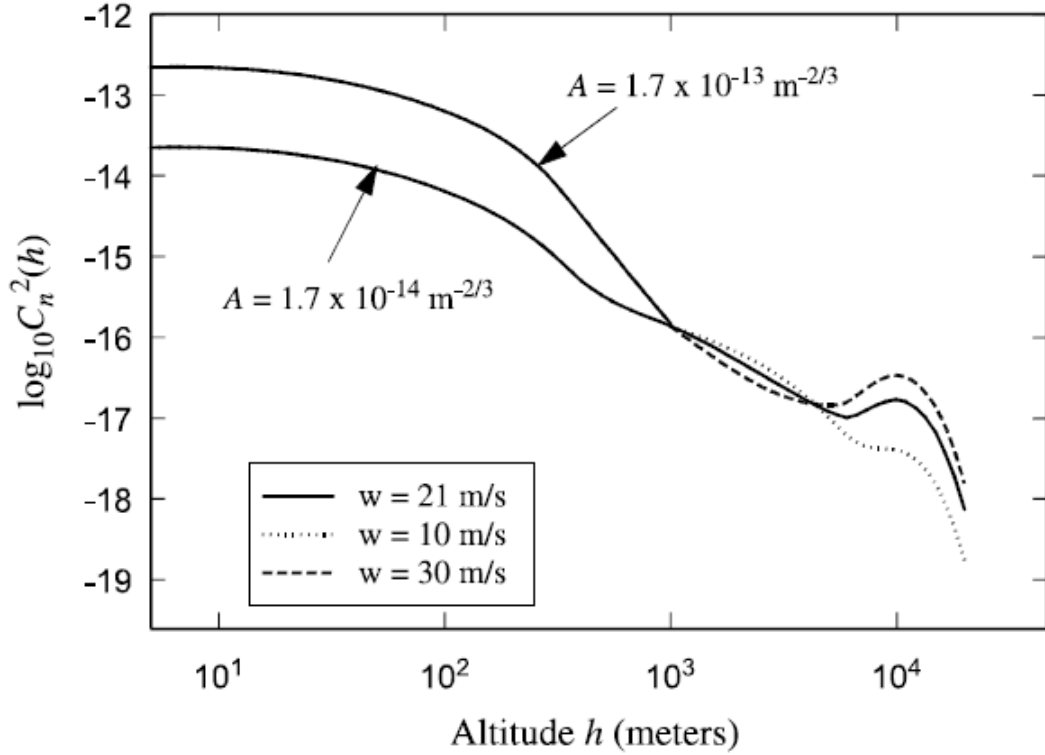


Figure 1.3: The relationship between turbulence strength and altitude using the general H-V model of changing parameters, three values of moderate to strong wind speed ($w = 10, 21$, and 30 m/s) and two values of nominal ground turbulence levels $A = 1.7 \times 10^{-14}$ and $A = 1.7 \times 10^{-13} \text{ m}^{-2/3}$ (Andrews and Phillips 2005).

A recent study on the influence of atmospheric turbulence on planetary transceiver laser ranging used the Hufnagel-Valley turbulence profile model to calculate the effects of turbulence on range accuracy, precision and parameter estimation performance (Dirkx et al. 2015). The study revealed that turbulence induced laser-signal strength variations were coupled with a 3 to 4 mm error between weak and strong turbulence conditions. It also suggested that the strong accuracy variations were due to variations in the ground turbulence strength function, with the mean wind velocity having little to no variations influence. The primary contributor to turbulence-induced accuracy and detected pulse fraction variations were found to be the turbulence-induced pointing error.

The vertical distribution of atmospheric turbulence is given by the refractive index structure parameter C_n^2 which allows for predicting atmospheric optical quality in terms of astronomical seeing. The integrated value of the index of refraction structure parameter for all atmospheric layers gives the total effect of the atmospheric turbulence. It can be obtained by making use of the integral of Equation (1.13) and measuring the atmosphere's temperature profile C_T^2 as well as the meteorological parameters, pressure P [mBar] and absolute temperature, T [K]. The HartRAO index of refraction structure parameter can be used to obtain the structure-function of optical signal fluctuations, D_s . This structure varies with the optical path length and the scales of turbulence (Born et al. 1975), and is written as,

$$D_s = 2.91 C_n^2 r^{5/3} z \left(1 - 0.8 (2\pi r / L_0)^{1/3} \right), \quad (1.15)$$

where r is the separation of l_0 , the inner scale and L_0 , the outer scale and z is the optical path length.

The Fried parameter r_0 is determined from the C_n^2 profile by Roddier (1981) and Vernin and Muñoz-Tuñón (1992) as,

$$r_0(\lambda, \gamma, h) = \left(16.7 \lambda^{-2} \frac{1}{\cos \gamma} \int_0^\infty C_n^2(h) dh \right)^{-3/5}, \quad (1.16)$$

where the turbulence strength C_n^2 varies as a function of height h above the telescope, γ is the angle relative to zenith and λ is the wavelength of light. Usually γ is chosen at zenith with $1/\cos \gamma = 1$. Therefore, the expression for

atmospheric seeing (Vernin and Muñoz-Tuñón, 1992), and is written as ,

$$\varepsilon_{FWHM} = 0.98 \frac{\lambda}{r_0} = 5.25 \lambda^{-1/5} \left(\int_0^\infty C_n^2(h) dh \right)^{-3/5}. \quad (1.17)$$

Knowing the wavelength of observation λ and height above the telescope h , the integrated value of the atmospheric seeing ε_{FWHM} is determined for a site. The atmospheric seeing at a site can be estimated by the inclusion of a turbulence model to obtain $C_n^2(h)$ profiles for a specific site.

1.5.4 Lunar reflector mirrors

On the 20th of July 1969, a team of Apollo astronauts landed on the Moon surface for the very first time in history (Faller et al. 1971; Dickey et al. 1994). They placed the first retroreflector array (commonly known as the Apollo 11) on the Moon surface. Thereafter, two more manned Apollo missions (Apollo 14 and 15) were deployed in February and August 1971, respectively. Each reflector consists of an array of solid, circularly cut fused silica corner cube retroreflectors 3.8 cm in diameter. The properties which includes shape, material and size of the retroreflectors (Figure (1.4)) fulfil the purpose of lunar laser ranging, described in Section 1.1 (Murphy 2013; Murphy et al. 2014).

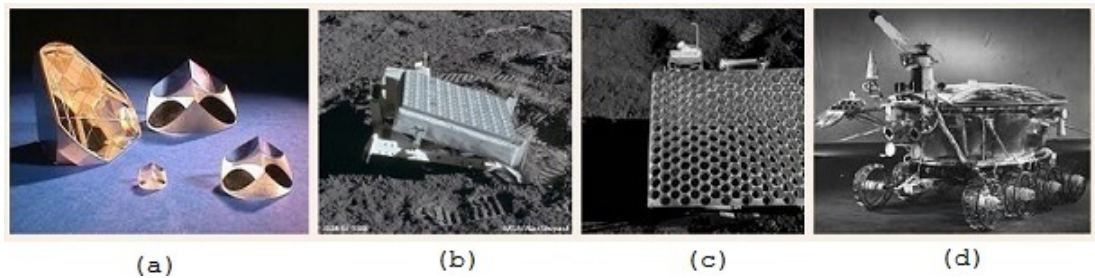


Figure 1.4: (a) Corner cube prisms or retroreflectors. (b) Apollo 14 retroreflector square array which is similar to that of Apollo 11. (c) Hexagonal Apollo 15 retroreflector array. (d) The Lunokhod that the Soviets landed on the Moon.

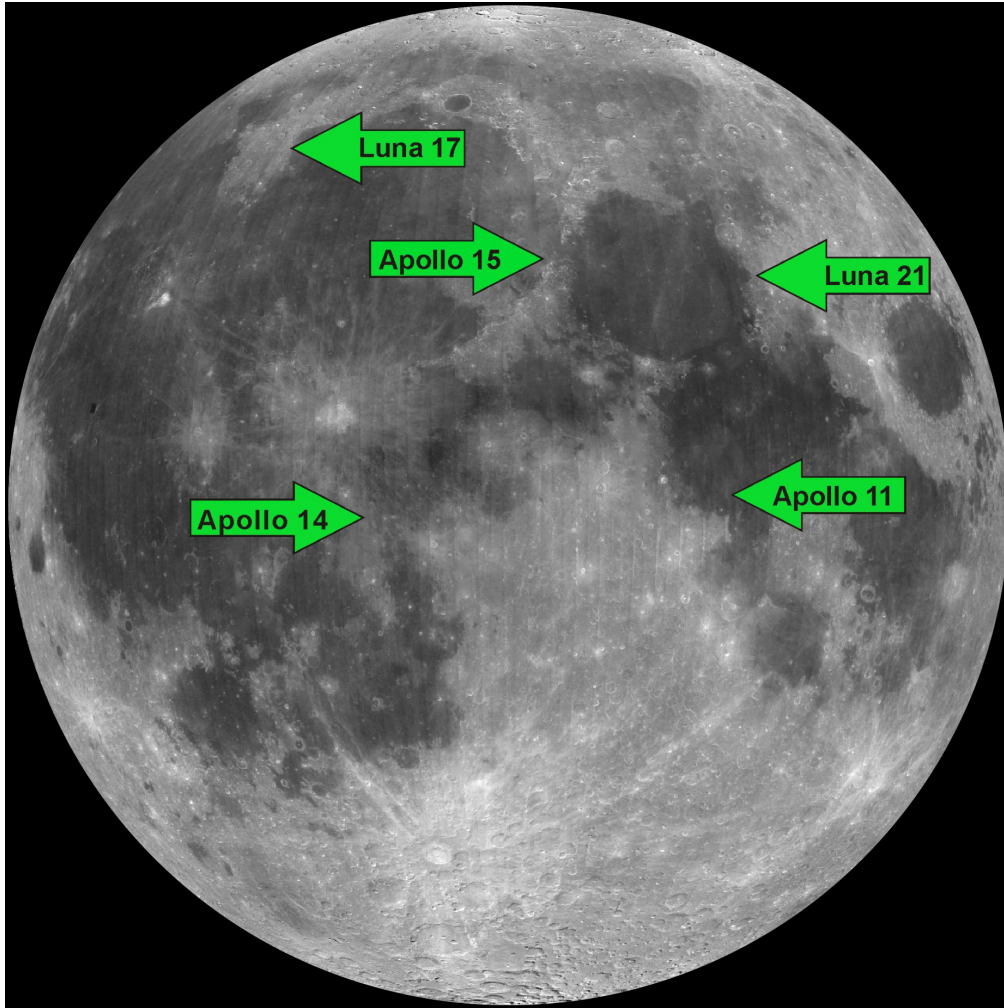


Figure 1.5: The locations of the five retroreflectors on the Moon surface. Source: NASA

The two unmanned Soviet Lunokhod missions (Luna 17 or Lunokhod 1 and Luna 21 or Lunokhod 2) were placed on the Moon in November 1970 and January 1973, respectively (Bender et al. 1973). A total of five retroreflectors are located on the Moon surface (Figure (1.5)). The main function of these five lunar reflectors was (and still is) to support LLR operations for studying Earth-Moon dynamics. Recent studies focusing on these reflectors reveal a long-term degradation on the optical reflectors (Murphy et al. 2010; 2014). Lunar dust accumulation is believed to be responsible for these optical degradations as the retroreflectors' effectiveness

vary with lunar phases (Murphy et al. 2014).

The detrimental effects on the laser signals caused by the incident angle of the laser on retroreflectors from an Earth station (ignoring lunar libration) is estimated by Degnan (2012) as:

$$\theta_{inc} = \alpha \tan \left(\frac{R_E}{R_{EM}} \right), \quad (1.18)$$

where R_E is as defined earlier in Equation (1.3), R_{EM} is the average Earth-Moon distance and α is the magnitude of the angular displacement in the far field divergence pattern. In general, the laser signal is reduced by half or more if the retroreflector cube diameter, D_{cc} , satisfies the inequality discussed by Degnan (2012) as:

$$\alpha = \frac{2\nu}{c} > \theta_{1/2} = \frac{\lambda x_{1/2}}{\pi D_{cc}} \quad \text{or} \quad D_{cc} > D_{1/2} = \frac{1.6\lambda c}{\pi\alpha}.$$

The global ranging statistics (Figure (1.6)) indicate that the Apollo 15 array reflects more signal than the other reflectors. This high number of reflected laser signals is due to the bigger size of the array and its central position on the Moon surface.

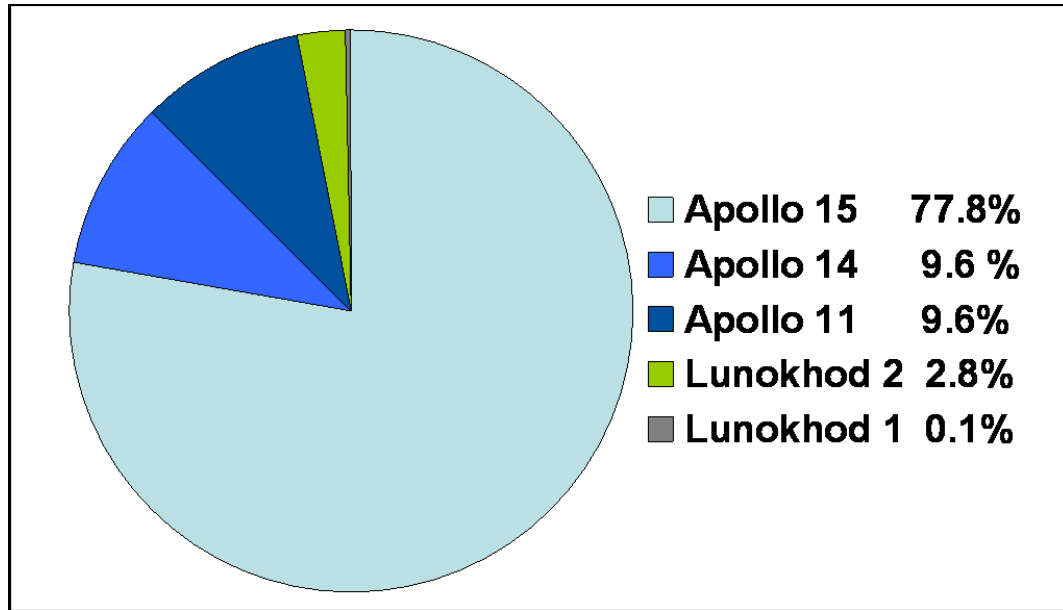


Figure 1.6: The lunar reflectors statistics with the Apollo 15 as the highest reflector. Source: NASA, Accessed on the 18th of June 2013.

1.5.5 Effects of Lunar Phases on LLR

The Moon goes through 8 major phases; New Moon, Waxing Crescent Moon, First Quarter Moon, Waxing Gibbous Moon, Full Moon, Waning Gibbous Moon, Third Quarter Moon and Waning Crescent Moon. These are different ways the Moon looks from Earth over about a month, also known as lunar month. The cause of these Moon phases is owed to the Moon orbits around the Earth and the part of the Moon that is illuminated by the light from the Sun. As the retroreflectors are mounted on the Moon surface, the Moon phases have been found to have some effects on the LLR observations (Chapront and Francou 2003).

The LLR observations are carried out about 25 days within a lunar month with no observations around New Moon (Chapront and Francou 2003). The observations are grouped together around the First Quarter (evening observations) and Third Quarter (morning observations). There are no observations which are carried out at Full Moon because of heavy noise except

the measurements during an eclipse in year 2000 at CERGA (Chapront and Francou 2003). These observations were also observed using the Apache Point Observatory data of January 2010 (Figure (1.7)).

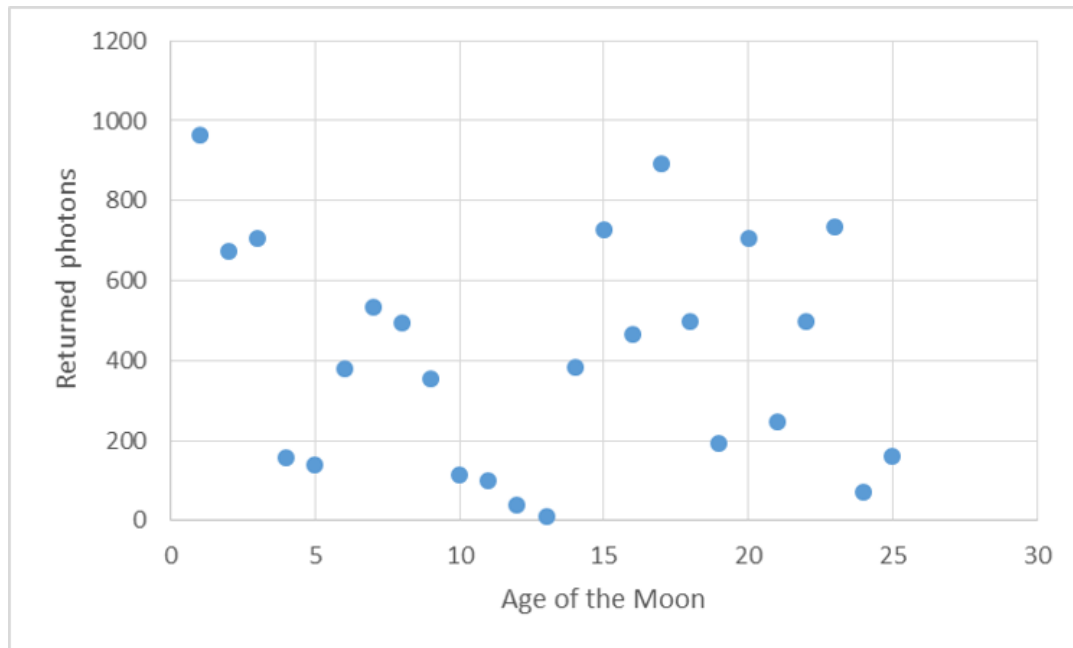


Figure 1.7: The Apache Point Observatory LLR distribution of returns for both day and night observations.

1.6 Applications of LLR

Early and current LLR measurements have contributed immensely to a wide range of scientific fields that include astronomy, lunar science, geodesy, gravity and fundamental physics. The advantage of LLR is that it provides the means for testing Einstein's Strong Equivalence Principle (SEP), which states that the outcome of any local experiment (gravitational or not) in a freely falling laboratory is independent of the velocity of the laboratory and its location in space-time (Nordtvedt 1988 and Hofmann et al. 2013). A violation of the SEP

would change the orbit of the Moon about the Earth-Moon centre of mass to be polarised in the direction of the Sun (Nordtvedt 1988). Thus, for a violation of the SEP, the ratio of the gravitational mass, M_G , to inertial mass, M_I , depends on the self-energy, U_G (Nordtvedt 1988), and this is expressed as,

$$\frac{M_G}{M_I} - 1 = \eta \frac{U_G}{Mc^2}, \quad (1.19)$$

where η is the Nordtvedt coefficient (which measures both the superposition of gravitational effects and how much space curvature is produced by unit rest mass), M is the mass of a body and c is the speed of light in vacuum.

Previous LLR data analysis gave $\left(\frac{M_G}{M_I} - 1\right) = (2 \pm 5) \times 10^{-13} \text{ yr}^{-1}$ to be equivalent to $\eta = -0.0005 \pm 0.0011$ (Dickey et al. 1994). A recent LLR analysis using data between 1970 and 2009 gave an improved $\left(\frac{M_G}{M_I} - 1\right) = (-7 \pm 3.8) \times 10^{-13} \text{ yr}^{-1}$ and $\eta = -0.0006 \pm 0.00052$ (Williams et al. 2009; Hofmann et al. 2010). The improvement can be associated, but not limited to, the improved LLR measurements. The uncertainty in the Nordtvedt coefficient, η , assumes no violation of the Weak Equivalence Principle (Dickey et al. 1994). The aforementioned results can be interpreted as a test for the parameterised post-Newtonian (PPN) parameters β , which is a measure of the superposition of gravitational effects and is commonly determined from the precession of the argument of perigee, and γ , which is a measure of how much space curvature is produced by unit mass. These two PPN parameters equal 1 if the General Theory of Relativity is valid. Both these PPN parameters can be used to determine the Nordtvedt coefficient (Dickey et al. 1994), expressed as:

$$\eta = 4\beta - 3 - \gamma. \quad (1.20)$$

In the field of astronomy, LLR data contributes to the lunar ephemeris (listing

of lunar coordinates with respect to the Earth as a function of time) by providing a determination of radial distance variations with 6 mm accuracy. It has the potential to improve the angular-rate and the mean distance uncertainties (Samain et al. 1998). The contribution of LLR data to planetary ephemerides is essential for the positioning of the ephemeris in the fundamental astronomical reference frame at the milliarcsecond level since accurate ephemerides are crucial for spacecraft navigation, mission planning, asteroid position determination and precise reference frame connection (Samain et al. 1998).

The advantage of LLR in geodynamics is that it provides a faster determination of Universal Time (Earth rotation) when compared to Very Long Baseline Interferometry, commonly known as VLBI (Samain et al. 1998). Very Long Baseline Interferometry is a type of astronomical interferometry used together with multiple radio telescopes on Earth to collect signals from an astronomical radio source such as a quasar. Analysis of LLR data in geodynamics also provides an accurate determination of the LLR station's coordinates and motion (due to tectonic plate motion), the product of the gravitational constant (G), the mass of the Earth (M), and the tidal acceleration of the Moon. In addition, it yields data about the exchange of angular momentum between the solid Earth and the atmosphere.

1.7 HartRAO-LLR

The HartRAO-LLR system consists of two TME_{00} laser sources, one for Lunar Laser Ranging and the other for Satellite Laser Ranging. Both of these laser sources have a wavelength of 532 nm each and produce laser pulses with different repetition rates (Table (1.1)). The standard deviation for the pulse jitter for these lasers is less than 0.1 ns. The SLR laser beam propagates through the optical

path and is transmitted through a mounted 0.2 m in diameter beam expander which is properly aligned parallel with the 1 m receiving telescope to ensure that the transmitted laser signals are received through the 1 m aperture. For LLR, the laser beam is transmitted through the 1 m transceiver optical telescope and directed to the retroreflector mirrors mounted on the Moon surface.

Table 1.1: The specifications of the lasers installed at HartRAO for S/LLR.

Laser specifications	Satellite Laser Ranging	Lunar Laser Ranging
Output energy, mJ	0.45	100
Repetition rate, Hz	1000	20
Initial beam diameter, mm	3	12
Pointing stability, μrad	<30	<50
Output beam diameter, m	0.2	1.0

Each laser pulse propagates through the same coudé path which forms the laser ranging optical train. This optical train starts from the optical table (where the laser sources are placed) and ends at the transmitting telescope. The total optical path length of the coudé path is 12 m. It includes the optical components that ensure that the transmitted laser beam remains collimated and directed to the distant retroreflector mirrors to ensure that laser signals are reflected to the “Earth-fixed” receiving telescope. The total transmission efficiency of the HartRAO telescope was determined to be 90 % (Nkosi et al. 2016). However, this telescope is only capable of transmitting 71% of a 532 nm laser light for a maximum return-rate of 7 photons per minute under favourable weather conditions (Nkosi et al. 2016). The complete optical layout (Figure (1.8)) consists of two laser sources, 0.45 mJ for SLR and 100 mJ for LLR, mounted

inside a box to prevent dust contamination, reduce vibrations and thermal variations.

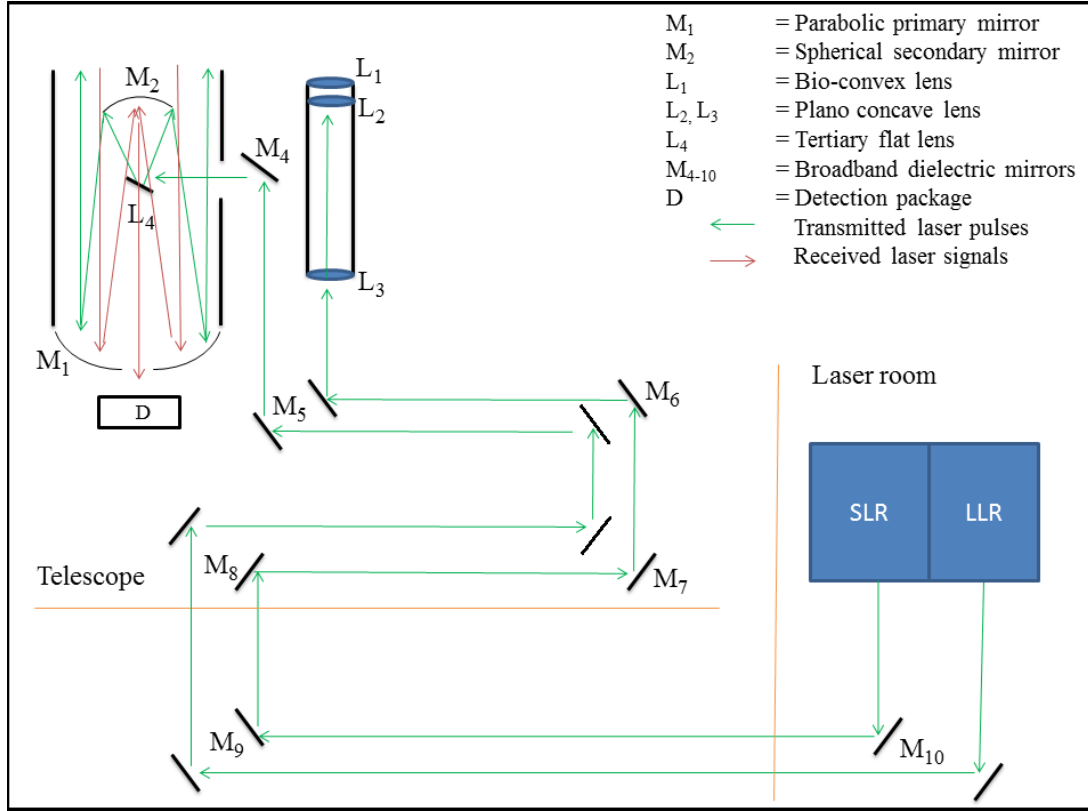


Figure 1.8: The diagram of the HartRAO-LLR system. Laser pulses for both SLR and LLR propagate through the same coude path. The returned signal is detected at the back of the telescope. Adapted from (Nkosi 2015)

The 1 metre Cassegrain optical telescope mount configuration is an Azimuth-Elevation configuration and consists of transceiver (transmitting and receiving) optical mirrors. A hyperbolic 0.3 m secondary mirror mounted on a metal spider structure attached to the front of the telescope tube directs the returned signal to the detector mounted at the back of the telescope. The telescope (Figure (1.9)) is equipped with servo drives and a high accuracy steering and pointing software that maximises high precision tracking capabilities (Combrinck 2014). The entire system will be integrated with a solid state photon detector, event

timer, precipitation and visibility sensors, range gate generator, start diode and programmable pulse repetition frequency (PRF) system.



Figure 1.9: The ex-OCA 1 m telescope is now installed on its foundations at HartRAO. (Source: Combrinck 2011).

1.8 Scientific contribution

An advanced mathematical tool, utilising C++ code, was developed and used to estimate the expected number of returned photons for the HartRAO's LLR system. The tool allowed the user to select a targeted satellite/retroreflector and adjust parameters. This tool will be used to evaluate computed and observed photon return efficiency, using as departure point the existing link budget equation, with the option to add and estimate parameters in the least squares sense. The mathematical tool also included the ranging degrading factors such as the variations of the angle of incidence of the laser on retroreflectors, laser energy variations and efficiency of retroreflectors. The slant range is one of the biggest degrading parameters on the number of received photons due to its 4th order of magnitude. Atmospheric transmission and cirrus cloud transmittance have an

influence on the number of returned photons degradation. Under unfavourable weather conditions, a decline in the number of returned photons is observed.

A new varying parameter was introduced in the radar link budget equation. This parameter addresses the effects of the brightness of the Moon surface as observed from Earth, which is considered as one of the factors that also plays its role in reducing the number of observed LLR returns. The introduction of this parameter in the radar link budget equation helps with a better understanding of the signal-to-noise ratio when estimating the expected number of LLR returns.

Another mathematical tool, utilising C++ code, was successfully developed to enable correct beam expander settings for precise eyepiece lens position (using microprocessors and software programme utilising C++ code). The developed tool controlled hardware and software embedded systems that accurately drives the movement of the beam expander's eyepiece lens.

Uncertainty sources associated with LLR were discussed. It was described that most of these uncertainties originate from the integrated measuring system with optical, mechanical and electronic components. The errors from the aforementioned components, together with the atmospheric fluctuations and operational factors, determine the measurement uncertainty of the LLR. Available literature systematically addressed classification of uncertainty sources and their effects on laser ranging. The LLR link budget parameters (laser pulse energy, detection efficiency, effective cross-section area of retroreflectors, range and environmental transmission) and their associated uncertainties were used to estimate the total number of returned photons.

1.9 Thesis overview

A brief but thorough literature review was presented in this chapter as an introduction. In Chapter 1, a review of the background theory and scientific application of laser ranging is provided. It also includes an overview on Satellite Laser Ranging and the HartRAO-LLR system. The effects of astronomical seeing on laser ranging is also discussed. A literature review on the efficiency of the lunar retroreflectors is presented and the chapter is concluded by a section on the thesis's overview.

The basic structure for the rest of this work is as follows;

Chapter 2: In this chapter, descriptions of the problem statement, aim and scientific objectives of the research project are provided. A section on the HartRAO station is provided to highlight the importance of the station's location, station background information and its current role.

Chapter 3: A full description of the methodology that was followed to generate data, for the development of a model and system that enables optimal efficiency of a lunar laser ranger signal path, is provided in this chapter. HartRAO-LLR subsystems are also fully described.

Chapter 4: Simulations and results of this research work, in the form of tables, pictures and graphs, is presented under this chapter.

Chapter 5: Discussion of results and implementations based on the results of optimal efficiency of a lunar laser ranger signal path is presented under this chapter.

Chapter 6: A summary, concluding remarks and a brief discussion on future work on LLR systems in the Southern Hemisphere is discussed in detail in this

chapter.

Chapter 2

The HartRAO-LLR station

2.1 Introduction

The HartRAO-LLR system is being built to accurately measure the Earth-Moon distance at 1 cm level through the use of short laser pulses and sophisticated components that are each discussed in more detail in Chapter 3. This system requires an optimally efficient signal path in order to ensure that transmitted laser pulses are targeted on the lunar retroreflector mirrors and are subsequently reflected from these mirrors. The development of an integrated model and system will help to enable optimal efficiency for the HartRAO-LLR signal path. As this is a core requirement for any LLR system; without signal path optimal efficiency, the number of returned photons could be zero.

An integrated model was utilized to obtain optimal efficiency (optical and electrical) of the HartRAO-LLR system. The model calculates the expected number of returned photons by considering a number of parameters which affects the laser beam pulses as they traverse the atmosphere from the LLR telescope to the Moon and back again to the telescope. This is achieved by modelling the

effects of thermal and density fluctuations of the atmosphere on the apparent Earth-Moon range, atmospheric extinction, laser beam characteristics, optical path efficiencies and other factors on the number of returned photons. These factors affect the estimated (predicted by software) and actual (measured) number of returned photons for the HartRAO-LLR station. The model also estimates the expected number of photon returns for given observational parameters to improve the LLR's system functionality. An improved system functionality reduces adverse effects (beam divergence) on the transmitted laser beam. Although adverse effects cannot be easily minimised, the software-based analytical model for HartRAO's LLR will enable evaluation of specific portions of the receiving and transmitting paths, as well as certain atmospheric parameters.

In this chapter, the relevance of the HartRAO-LLR and its role in global context are discussed. The scientific motivation for this research, the aim and specific objectives are discussed in the Conclusion. The Matjiesfontein site will be equipped with collocated space geodesy systems, (Global Navigation Satellite Systems (GNSS), New Generation Satellite Laser Ranging (NGSLR), VLBI Global Observing System (VGOS), Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS)) to form a node of the proposed IAG Global Geodetic Observing System (Combrinck 2007). The main aim of this work was to develop an integrated model and system that will enable optimal efficiency of the HartRAO-LLR signal path. The HartRAO-LLR system must be developed such that it meets all the requirements of the ILRS.

2.2 HartRAO station

2.2.1 Background

The Hartebeesthoek Radio Astronomy Observatory was originally built by NASA to track the probes for space beyond Earth orbit explorations. The former name of HartRAO was "Deep Space Instrumentation Facility 51", abbreviated as DSIF51. Construction of this station began in January 1961 and it became operational in July 1961, supporting Ranger 1 pre-launch tests (Tsiao 2007). The 26 m antenna was originally built with an aluminium mesh surface, and operated at a frequency of 960 MHz, i.e. a wavelength of about 30 cm. The lightweight mesh surface was quite transparent optically until the installation of new solid surface panels of the antenna was completed in September 2003. These panels permitted operation at frequencies up to 24 GHz, doubling the usable frequency range of the telescope.

The South African Council for Scientific and Industrial Research (CSIR) operated the station until its shutdown in 1974. Thereafter, it became a radio astronomy observatory, operating firstly under the CSIR, then the Foundation for Research Development (FDR), which is now the National Research Foundation (NRF) since 1999. HartRAO participates and contributes to many global networks. These include the ILRS SLR network (Figure (2.1)), VLBI (radio astronomy and geodetic antenna), International GNSS Service (IGS) and International DORIS Service (IDS), that are simultaneously used to precisely determine ranges for low Earth-orbiting satellites. The HartRAO-LLR system, will now also form part of the ILRS Network.

The HartRAO-LLR system consists of an optical telescope which is equipped with a highly accurate monitoring, steering and pointing software (Combrinck 2011 and 2014; Combrinck and Botha 2013; Ndlovu et al. 2014). The HartRAO-LLR

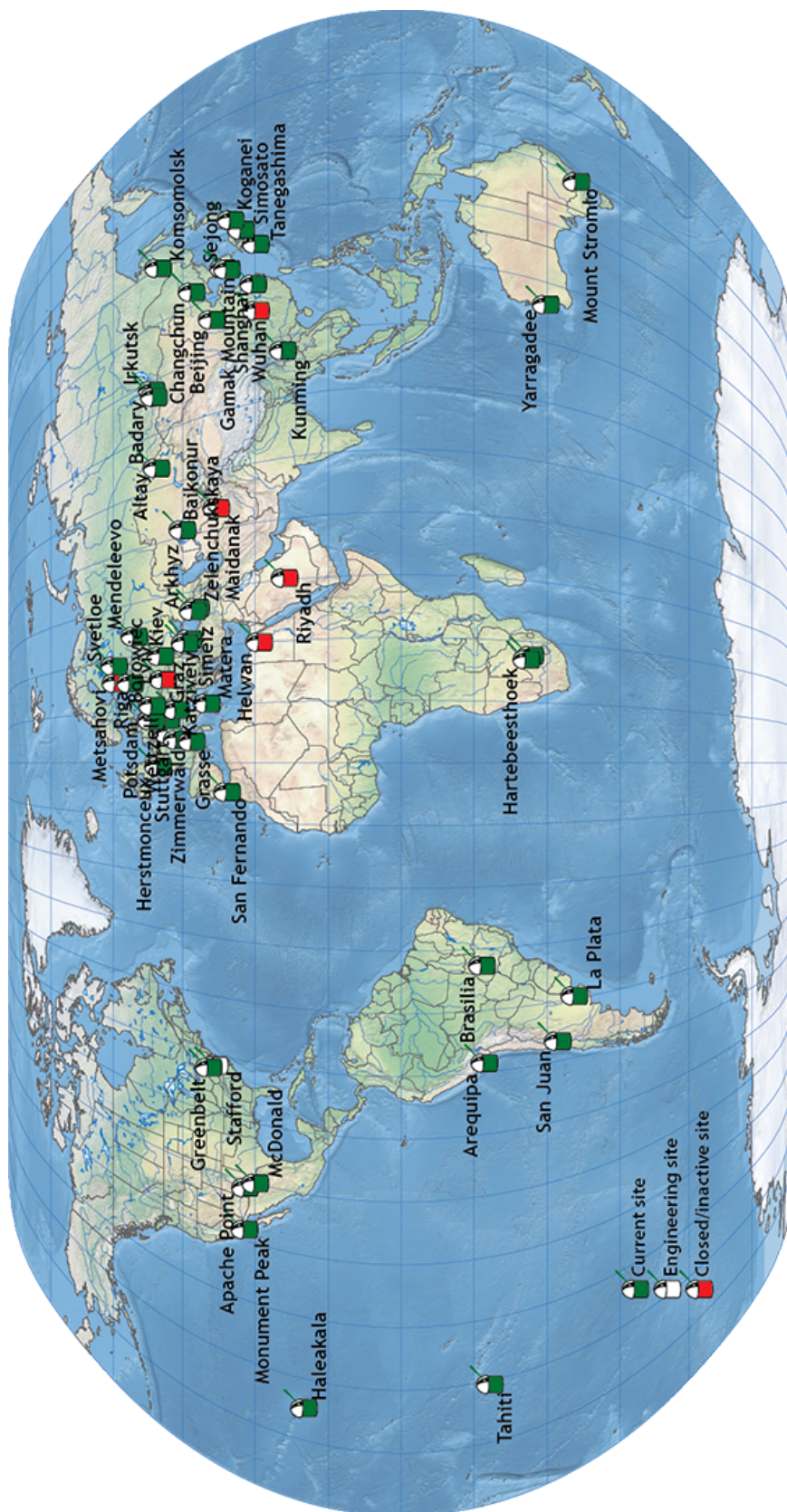




Figure 2.2: A Google map view of the Hartebeesthoek Radio Astronomy Observatory (25.8900° S, 27.6853° E). The 26 m radio telescope is clearly visible in near mid-centre, to the west of Moblas6 and the LLR telescope run-off shelter.

system will now also form part of the International Laser Ranging Service Network. Its role is to improve the data quantity and separate latitude dependencies. This improvement will enhance the current effort to determine highly accurate Earth-Moon distances for the various scientific purposes.

2.2.2 Location

The Hartebeesthoek Radio Astronomy Observatory facility is located in a natural bowl (1400 m above sea level) of hills at Hartebeesthoek (25.8900° S, 27.6853° E) just South of the Magaliesberg mountain range (Figure (2.2)) in the Gauteng Province of South Africa, about 65 km west of Johannesburg. The site is poor for SLR/LLR due to pollution and relatively high cloud cover (compared to Matjiesfontein).

The HartRAO telescopes are generally isolated from terrestrial radio frequency interference as the nearest town, Krugersdorp, is located 32 km away from the Observatory. These telescopes (26 m and 15 m) are both astronomical instruments consisting of radio receivers and antenna systems used to detect radio-frequency radiation emitted by extra-terrestrial sources. They are also used for geodetic VLBI operations, using radio signals from quasars to measure movements of the Earth's crust and the rotation of the Earth. Measurements of the time difference between the arrivals of the radio waves at different antennas on the Earth surface are taken. Both HartRAO telescopes play an important role in the current and future VLBI network since, in particular, the precision of the Earth Orientation Parameters (EOP) are highly dependent on this observatory (Mayer et al. 2014). The 26 m telescope was part of the systems that were built by NASA in 1961 and the 15 m was built as a Square Kilometre Array prototype. Table (2.1) contains a list of some of the HartRAO radio telescope specifications.

Table 2.1: The parameters of the VLBI / Geodetic telescopes at HartRAO (Nickola 2012).

Parameter	26 m Antenna	15 m Antenna
Owner and Operator	HartRAO	HartRAO
Year of construction	1961	2007
Telescope mount	Offset equatorial	Azimuth/Elevation
Receiving feed	Cassegrain	Prime focus
Main reflector diameter, m	25.914	15
Focal length, m	10.886	7.5
Focal ratio	0.42	0.5
Surface error of reflector, mm	0.5	1.6
Short wavelength limit, cm	1.3	2
Pointing resolution, $^{\circ}$	0.001	0.001
Slew rate on each axes, $^{\circ}/s$	HA: 0.5, Dec: 0.5	Az: 2, El: 1

2.2.3 Current role

The existing VLBI antennas are collocated with several GNSS stations, SLR (MOBLAS-6) a French DORIS system, an underground vault containing geophysical instruments and the LLR (under development). These collocated geodetic techniques form a multi-technique fundamental station by different reference points. The measurements of these reference points play a crucial role in the computation and combination of the ITRF. As the distribution of most of the space geodetic techniques is sparse in the Southern Hemisphere, particularly in Africa, the HartRAO collocated site (Figure (2.3)) and the derived local ties therefore play an important role in determining site coordinates and velocities,

earth rotation parameters, atmospheric correction and geocentre motion. These measurements are essential for computing quality and reliable ITRF, gravity field modelling as well as for defining geodetic data for South Africa (Botai et al. 2015).

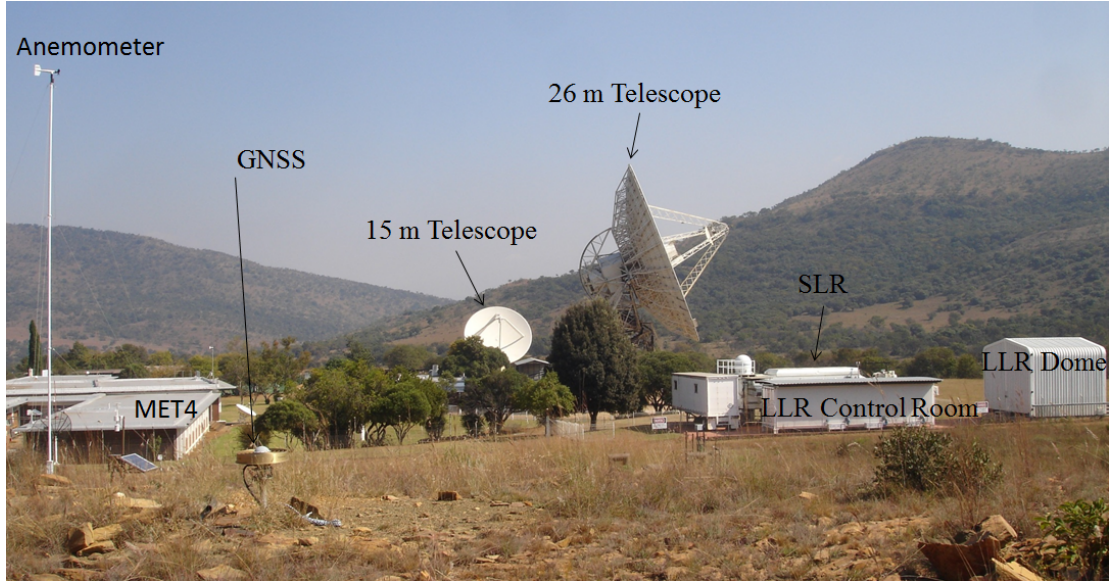


Figure 2.3: The collocated geodetic techniques of HartRAO; GNSS receiver, VLBI telescopes (26-m and 15-m), SLR and LLR. The LLR control room is in front of the SLR and the LLR run-off enclosure is to the right. The anemometer and MET4 meteorological station support the geodetic measurements.

The VGOS antenna at HartRAO, which is currently under development, will be collocated with the existing geodetic equipment. This radio telescope of approximately 13 metres in diameter will be used to help improve the accuracy across global baselines from around 1 cm down to 1 mm (Mey and Combrinck 2016). This fast antenna will provide many more observations to support higher temporal and spatial resolution in estimating the effect of the troposphere at each station. This effect is known, from simulations, to be one of the largest source of noise (Shabala et al. 2015).

Chapter 3

HartRAO-LLR system overview and the proposed methods to enable optimal efficiency

3.1 Introduction

The estimated average signal return rate of the HartRAO-LLR can be strongly affected by two-way atmospheric extinction (atmospheric and cirrus cloud transmissions), variations in the laser beam incident angle on the retroreflectors located on the Moon as well as the varying Earth-Moon range. Modelling the returned number of photons and comparing it to the actual received number of photons will lead to an understanding of the effects of numerous variables on the total laser path efficiency. Hence, the total system efficiency can be improved as well. Particular (adverse) atmospheric conditions will not allow ranging to the Moon to be successful. The estimation software can therefore be used as a prediction tool to schedule LLR sessions; less suitable days can be allocated to

SLR or maintenance. A description of the HartRAO-LLR subsystems is provided as well as a discussion on the different methods and procedures that were considered in order to understand the factors that influence the estimated number of returned photons expected from the HartRAO-LLR system.

3.2 System overview

The HartRAO-LLR system is software centric to allow rapid changes to specific functions, system parameter monitoring, telescope pointing and steering, as well as preliminary data quality checks (Combrinck 2014). The schematic diagram (Figure (3.1)) describes the signal path (optical and electrical) and the main components of a LLR system. The Moon is occupied with a total of 5 retroreflector arrays that can be tracked by an optical telescope. Such a telescope is equipped or integrated with software and hardware components that enable accurate tracking of the Moon position and location of lunar mirrors.

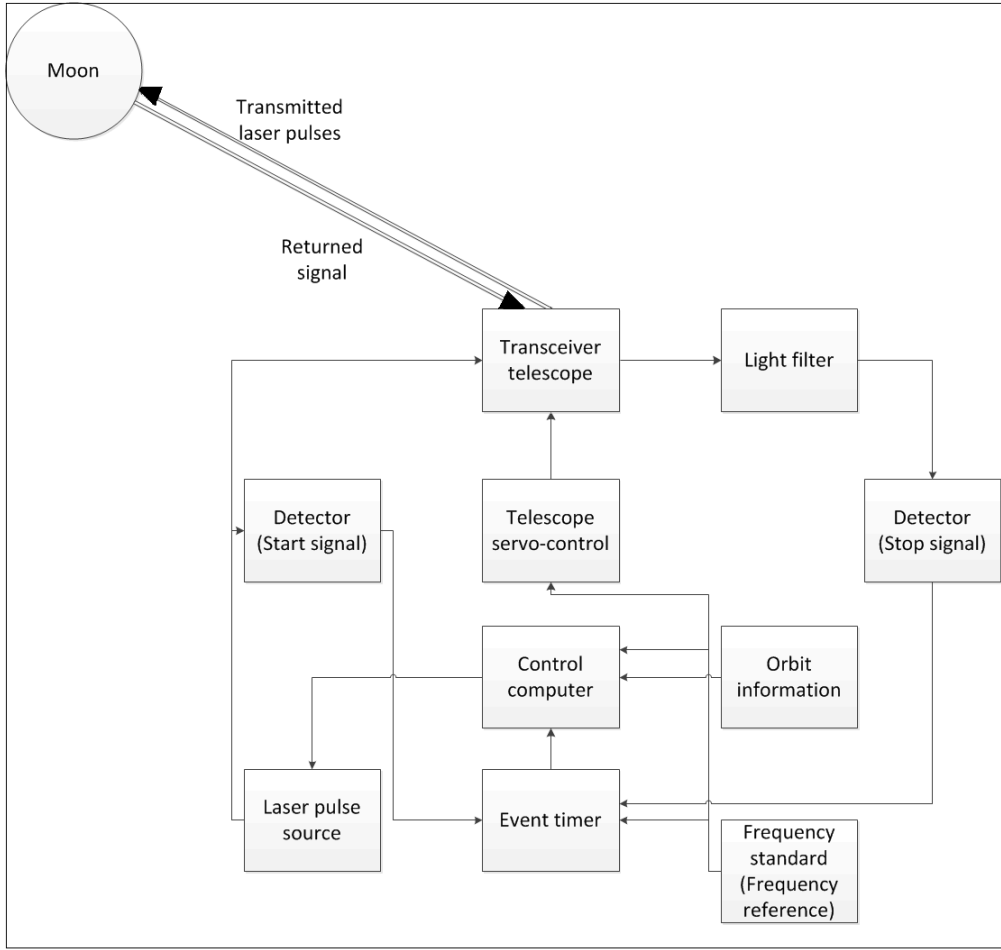


Figure 3.1: Basic description of the main components and signal path (optical and electrical) of the LLR system.

The telescope simultaneously transmits and receives short laser pulses at about 10 to 20 Hz for LLR and about 1 kHz for SLR. As the pulse is emitted from the laser pulse generator, it triggers an event timer at a certain time and that time is recorded. On its return from the corner cube reflector array, it re-enters the telescope and triggers the event timer again to record the time of arrival. This procedure allows the calculation of the distance between Earth and Moon which is half the product of the round TOF and the speed of light in vacuum. Satellites orbiting the Earth are also measured with the same principle of calculating the distance between the two bodies using the TOF and the speed of light in vacuum

(Degnan 1993).

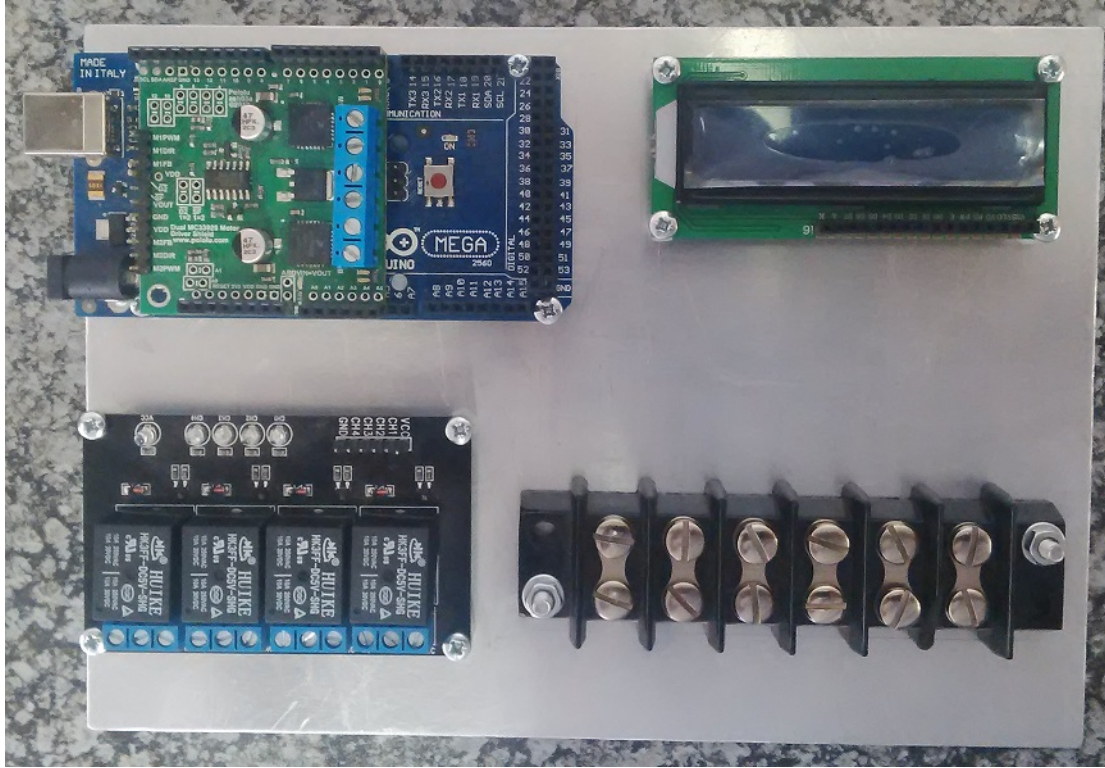


Figure 3.2: A Pololu MC3390 dual motor control (in green) before being assembled together with the Arduino Mega2560 (in blue) to control a DC motor. The accurate position of the motor can be obtained via the incremental rotary encoder and positions can be displayed in the 2X16 Liquid Crystal Display or the GUI.

In this section, a discussion is provided on the development of a technique which used C++ code and computer micro-controllers (Figure (3.2)) to control hardware and software embedded systems that accurately drives the movement of the beam expander's eyepiece lens. The development of an advanced mathematical tool also utilising C++ code to analyse LLR system and atmospheric parameters is also fully described. The primary output of the advanced mathematical tool is a scientifically accurate reflection of all parameters which will influence the LLR return signal, presented as an estimate of expected photon return rate.

3.2.1 HartRAO-LLR Laser

The HartRAO-LLR system consists of two TME₀₀ laser pulse sources, one for Lunar Laser Ranging and the other for Satellite Laser Ranging. Both these laser sources have a wavelength of 532 nm and produce laser pulses of different repetition rates (Table (3.1)). The output beam diameters of the transmitted lasers are; 1 m for LLR and 0.2 m for SLR. For LLR, the laser beam is transmitted through the transceiver 1 m optical telescope and directed to the retroreflector mirrors mounted on the Moon surface. The exit beam refractor for SLR is mounted and properly aligned parallel with the 1 m receiving telescope.

Table 3.1: The specifications of the installed laser system at HartRAO for S/LLR.

Laser Specifications	Satellite Laser Ranger	Lunar Laser Ranger
Maximum output energy, mJ	0.5	100
Pulse repetition rate, Hz	1000	20
Beam diameter, mm	~3	~12
Beam pointing, μ rad	<30	<50
Aperture, m	0.2	1.0
Pulse energy stability, %	~0.8	<1.0
Pre-pulse contrast	>200:1	>200:1
Pulse duration stability, ps	± 1	± 3
Polarization	Linear, >100:1	Linear, vertical >110:1
Laser head (W×L×H), mm	455×1035×242	453×1224×250
Typical warm-up time, min	30	30

Each laser pulse is transmitted via a coudé path which forms part of the optical

train which starts from the optical table and ends at the exit of the transmitting telescope. The optical train includes the optical components that will play a key role in ensuring that the transmitted laser is directed to the distant retroreflector mirrors and laser signals are reflected to the “Earth-fixed” receiving telescope. It consists of an average transmit and receive optics efficiency of about 90 % which has a direct effect on the number of transmitted and returned photons. The laser pulse sources (SLR and LLR) are mounted inside the white boxes that prevent dust contamination, minimise external vibrations and reduce thermal variations (Figure (3.3)).

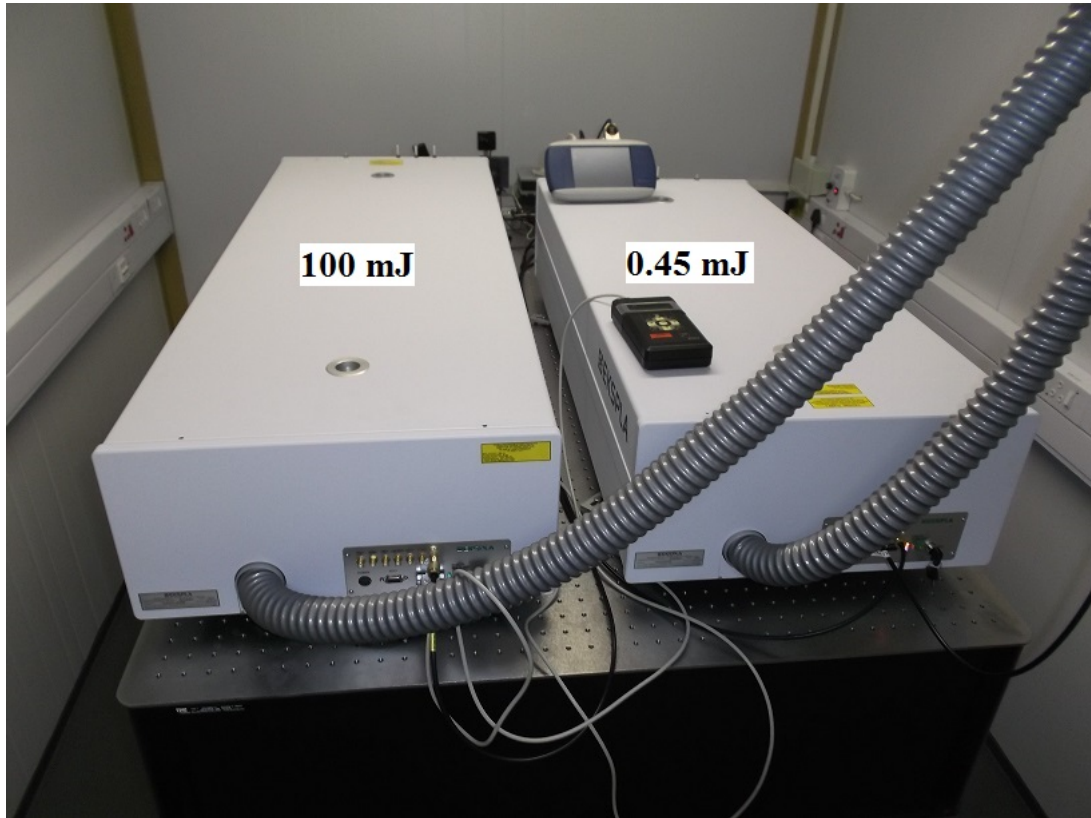


Figure 3.3: HartRAO’s new LLR (100 mJ) and SLR (0.45 mJ) laser sources. The enclosures prevent dust contamination of the laser optics. These laser sources never operate simultaneously as they both use the same signal path to range to targets.

3.2.2 The timing system

Timing systems continuously measure time intervals between transmitted laser pulses and returned laser signals, in the form of detectable photons, in real time. The two types of timing systems generally used for laser ranging are time interval counters and event timers. Time interval counters generally measure the time of flight between the station to the target and back again, whereas the event timer continuously records the epochs (a period of time) of signals received by both the start and stop detectors and places them into a buffer. The obtained epochs are then matched by range gate prediction.

Event timers are widely used in most laser ranging stations to advance the system's functionality (Fan et al. 2006). Event timers allow the increase of firing frequencies so that data quantity is increased. Each laser ranging station has a unique timing system (Table (3.2)) to provide high precision during laser ranging measurements. The Apache Point uses a clock count plus a Time-to-Digital Converter (TDC), which measures the time change values indicated in the table to a resolution of 25 ps.

Table 3.2: The specifications timing systems used in different active LLR stations.

Station	Timer	Model	Resol. [ps]	Prec. [ps]
Apache Point	Clock count + TDC	Custom	25	13
Grasse	Event	Not provided	<1	<2
Matera	Event	TD811	50/channel (4)	~30
McDonald	Event	Not provided	1.2	7
Wettzell	Event	WLRS-ET	1.22	2.0

The four laser ranging stations used event timers. Grasse is the only station with an event timer resolution which is less than 1 ps and a precision that is less than 2 ps. The HartRAO-LLR station will use a newly developed 4393A rubidium timing reference system by Microsemi with an accuracy to sub-picoseconds ($< 10^{-12}$) level (Munghemezulu et al. 2016). This system is expected to improve the measurement of the time-of-flight of the laser pulses and limit instrumentation error dependency. The proposed hardware connection diagram of the HartRAO-LLR in Figure (3.4) indicates that the laser fires while simultaneously providing a high signal-to-noise determination of the laser start time to be referenced at the photon detection time of the few photons returning from the calibration retroreflector. The GPS-slaved timing system also provides a reference to Coordinated Universal Time.

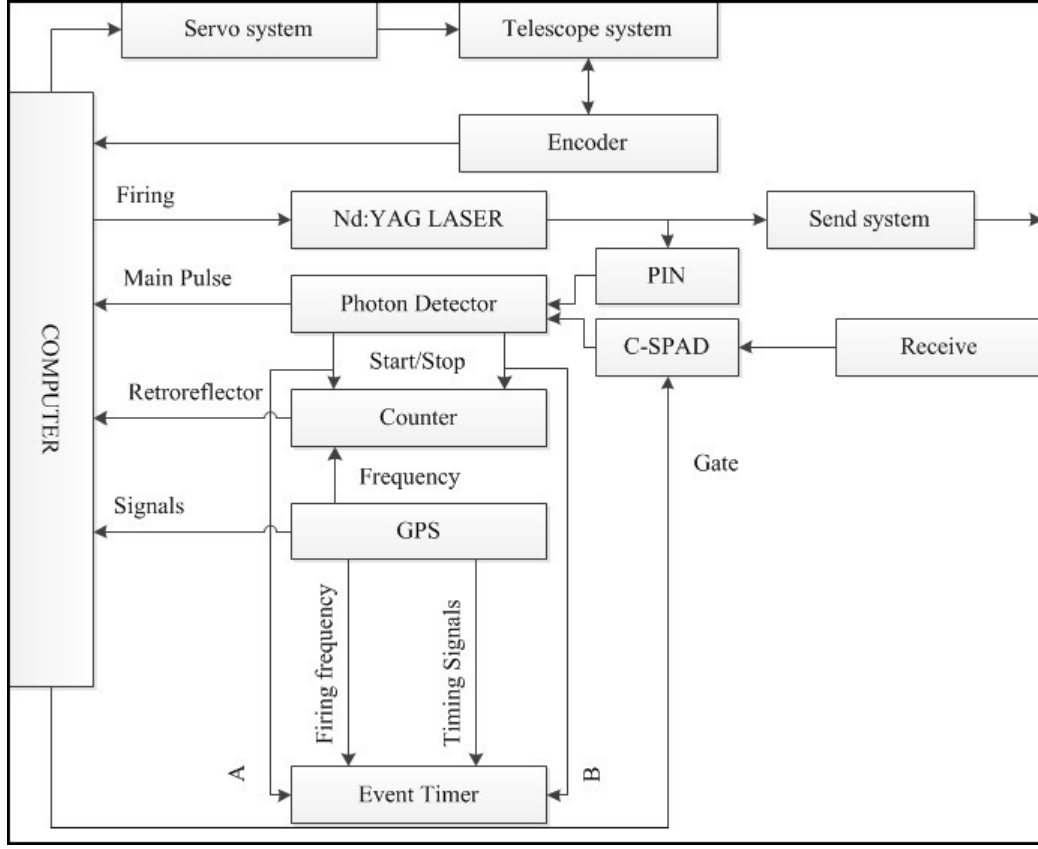


Figure 3.4: Schematic diagram of the fundamental architecture of hardware components and connections of the HartRAO-LLR.

3.2.3 The detection system

The HartRAO-LLR system is based on measuring the TOF of short laser pulses reflected off the lunar retroreflectors. The system is expected to yield centimetre to sub-centimetre precision. Thus, high precision LLR will require high photon return rates, ϵ , (Murphy et al. 2002) that can be expressed as the link efficiency,

$$\epsilon = \eta^2 f \eta_q \left(\frac{nd^2}{R^2 \Phi^2} \right) \left(\frac{D^2}{R^2 \phi^2} \right), \quad (3.1)$$

where η is the two way telescope or atmospheric transmission efficiency, f is the receiver throughput which is dominated by a narrow-band filter, η_q is the

quantum detector efficiency, n is the number of retroreflectors in the array, d is the diameter of the retroreflector, R is the distance to the surface of the Moon, Φ is the atmospheric divergence, ϕ is the retroreflector divergence and D is the diameter of the collecting telescope.

Lunar Laser Ranging photon return rates are generally dominated by signal loss from divergence of the transmitted and returned laser signals. Optimally collimated laser beams can be limited by the influence of the atmosphere to a divergence of between one to a few arcseconds (Murphy et al. 2002). The use of a high-quantum-efficiency single-photon counting system in the HartRAO-LLR will help to detect the weak photon returns. The quantum efficiency, η_q is given by Takeuchi et al. (1999) as,

$$\eta_q = \left(\frac{N_S}{1 - N_S\tau} - \frac{N_B}{1 - N_B\tau} \right) / N_{in}, \quad (3.2)$$

where N_S is the observed count with input signal photons, N_B is the observed dark-count without input photons, τ is the dead time, $N_{in} = \frac{P\lambda}{ch}$ is the number of incident photons and P is the power of light incident on the detector. In Table (3.3), information on the types of detectors used in different active LLR stations is provided. Most of these stations use a Micro-Channel Plate (MCP) for photon counting to have sufficient gain to deliver a useful output pulse for a single detected photon.

3.2.4 The telescope

The HartRAO telescope and its subsystems (Figure (3.5)) is capable of transmitting and receiving laser signals along the same optical path (the coudé

Table 3.3: The type and specifications of detectors used at 532 nm wavelength for the five active LLR stations.

Station	Detector	QE [%]	Gain	Rise [ps]	Jitter [ps]	FOV ["]
Apache Point	SPAD	30	Geiger	500	50	0.35/pixel
Grasse	APD	20	10^9	200	20	7 to 20
Matera	MCP	15	10^5	<120	30	60
McDonald	MCP	11	10^5	350	100	Selectable
Wettzell	MCP	>11	10^5	<95	100 - 130	30

path). This special type of optical configuration is a very complicated matter, as the partially transmitted laser signal may be instantly detected by the receiving photon detection package. For HartRAO's LLR, three methods are proposed and evaluated to achieve the optimal output beam diameter for LLR. One of the methods is to transmit through the beam expander and the other two methods expands the laser so that it fills the 1 m aperture of the HartRAO-LLR telescope.

In general, mirrors used for high power laser applications must withstand large amounts of radiation, some of which will be transformed into mirror heating. After evaluations of the transmission optics of the laser coudé path, Nkosi et al. (2016) found that the HartRAO-LLR telescope is capable of accomplishing its desired mission of ranging to the Moon at high accuracy. It was also found that the telescope mirror structure was efficient enough to guarantee that its surface will not be significantly distorted during laser ranging.

3.2.5 System monitor and control

The HartRAO-LLR telescope is equipped with highly accurate steering and pointing software (Combrinck 2011 and 2014; Combrinck and Botha 2013;

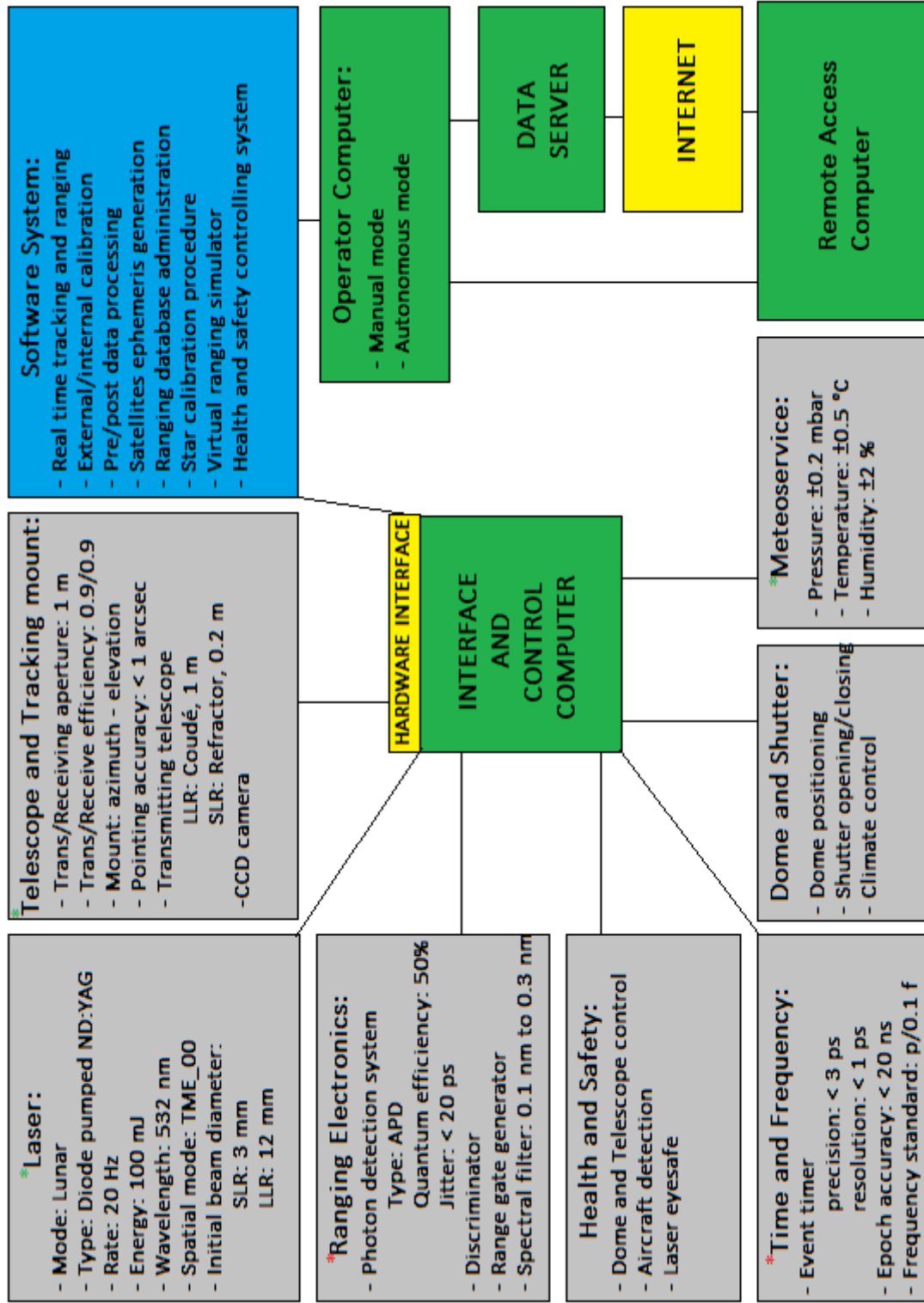


Figure 3.5: Schematic diagram of the measured (green asterisk) and proposed (red asterisk) system parameters of the HartRAO Lunar Laser Ranging.

Ndlovu et al. 2014). This software will improve the ranging accuracy to sub-centimetre level to enhance the current effort to determine highly accurate Earth-Moon distances to improve our understanding of the Earth-Moon system. The whole LLR system is controlled and monitored from the LLR control room (Figure (3.6)) which is located a distance of 12 m from the main telescope.



Figure 3.6: The interior of the HartRAO-LLR control room, includes a server rack, which is not shown in the figure, the telescope's pointing and testing computer software and a large television screen used to monitor the position of the objects. The day and night world map is displayed on the bigger screen.

3.3 Telescope configuration

A transceiver telescope's optical configuration is a very complicated matter, as the telescope's receive and transmit optics are mounted along the same optical path within the telescope system. The shapes, sizes and configurations of optical components (lenses and mirrors) inside a transceiver telescope determine the

output beam diameter. A collimated and “perfectly” shaped laser beam is one of the key features for successful laser ranging to distant retroreflector equipped objects such as the Moon and Earth-orbiting satellites. Some of the transmitted laser signal may need to be filtered to avoid damaging the detection system as it partially detects some of the transmitted laser beam. For HartRAO’s LLR, three methods were proposed to achieve the optimal beam diameter that would yield high photon rate. All these methods were evaluated in terms of optimal beam diameter and final laser energy output. A combination of optimal beam diameter and output energy, transmit and receive optics, as well as detection system efficiency results in maximising photon returns.

In Subsection 3.3.1, in the case of method 1, the proposal is to use the 0.2 m beam expander to transmit laser pulses and use the 1 m telescope to collect the returned signal. The sections that follow focusses on method 2 and method 3, which evaluate the signal path efficiencies of the output beam diameter when using a Cassegrain and coudé configuration, respectively. All the investigated methods are evaluated in terms of the relationship between the transmitted optimal beam diameter and the far field beam divergence that will enable successful photon returns during laser ranging.

3.3.1 Beam expander method

A laser beam expander (Figure (3.7)) is an optical system designed to increase the diameter of a collimated input beam to a larger collimated output beam. Beam expanders are used in various optical applications such as laser scanning, interferometry, remote sensing and laser ranging. Contemporary laser beam expander designs are afocal systems that are developed from well-established optical telescope fundamentals. In such systems, the object rays, located at

infinity, enter and exit parallel to the optical axis of the internal optics and hence the focal length to the entire system is infinite.

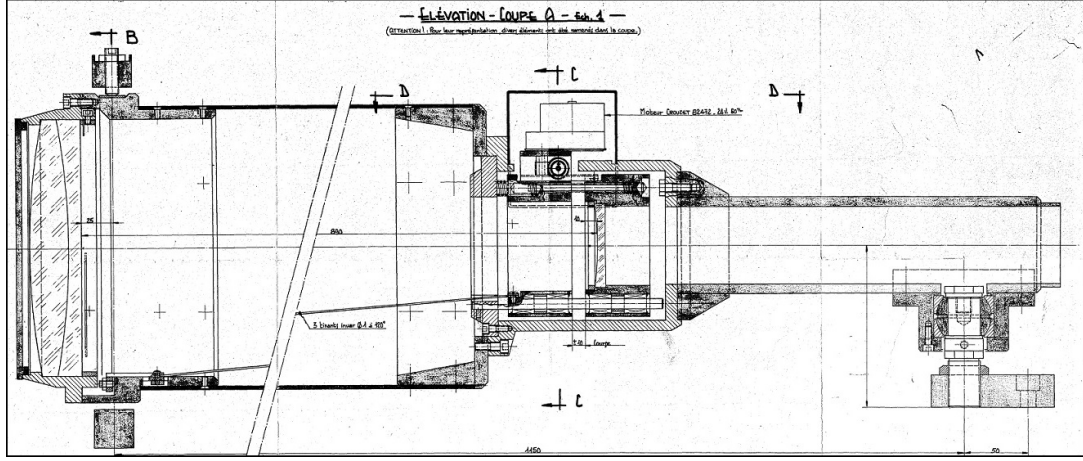


Figure 3.7: Schematic diagram of the adjustable and motorised 0.2 m (aperture diameter) beam expander for the HartRAO-LLR system. Source: OCA

The basic behaviour of a Gaussian beam along an optical path in a vacuum can be determined by the spot size, Rayleigh range, radius of curvature and beam divergence (Alda 2003), and this behaviour can be expressed as,

$$S(x) = S_0 \sqrt{1 + \left(\frac{x}{x_R}\right)^2}, \quad (3.3)$$

$$S(\pm x_R) = \sqrt{2 \times S_0}, \quad (3.4)$$

$$R(x) = x \left[1 + \left(\frac{x_R}{x}\right)^2 \right], \quad (3.5)$$

$$\theta \simeq \frac{\lambda}{\pi S_0}, \quad (3.6)$$

where $S(x)$ is the spot size radius at object distance x , S_0 is initial spot size radius when $x = 0$, x_R is the Rayleigh range (the distance from the beam waist along the

propagation direction where the initial beam radius doubles its cross section), $R(x)$ is the radius of curvature, θ is the divergence of the beam and λ is the wavelength of the laser beam. The corresponding time-average laser intensity (irradiance) distribution is expressed as,

$$I = I_0 \left(\frac{S_0}{S(x)} \right)^2 \exp \left(\frac{-2r^2}{S^2(x)} \right), \quad (3.7)$$

where I_0 is the measured intensity of the beam at its waist and r is the radial distance from the centre axis of the beam.

An adjustable and motorised 0.2 m aperture beam expander was initially planned to be used in a model to transmit laser pulses to the lunar surface. This beam expander has a high optics efficiency of about 90 % that enables a collimated and a “perfectly” shaped laser beam to minimise beam divergence through the atmosphere for accurate and precise targeting of the retro reflectors. For a perfect laser beam the value for $M^2 = 1$, where M square factor $M^2 \geq 1$ describes the deviation of the beam from a perfect Gaussian beam. The photon return rate is very low when transmitting to the lunar mirrors through the 0.2 m beam expander because of the long Earth-Moon distance and a small initial beam diameter. Most of the signal is lost on its way to the Moon surface. The other disadvantage of using a beam expander is that it requires a proper parallel aligned mount with the receiving telescope to ensure that the returned laser signals are detected at the back of the 1 m receiving telescope. A beam expander also requires a model and data analysis software to enable correct beam expander settings (using microcontrollers (Figure (3.8))) for precise beam collimation.

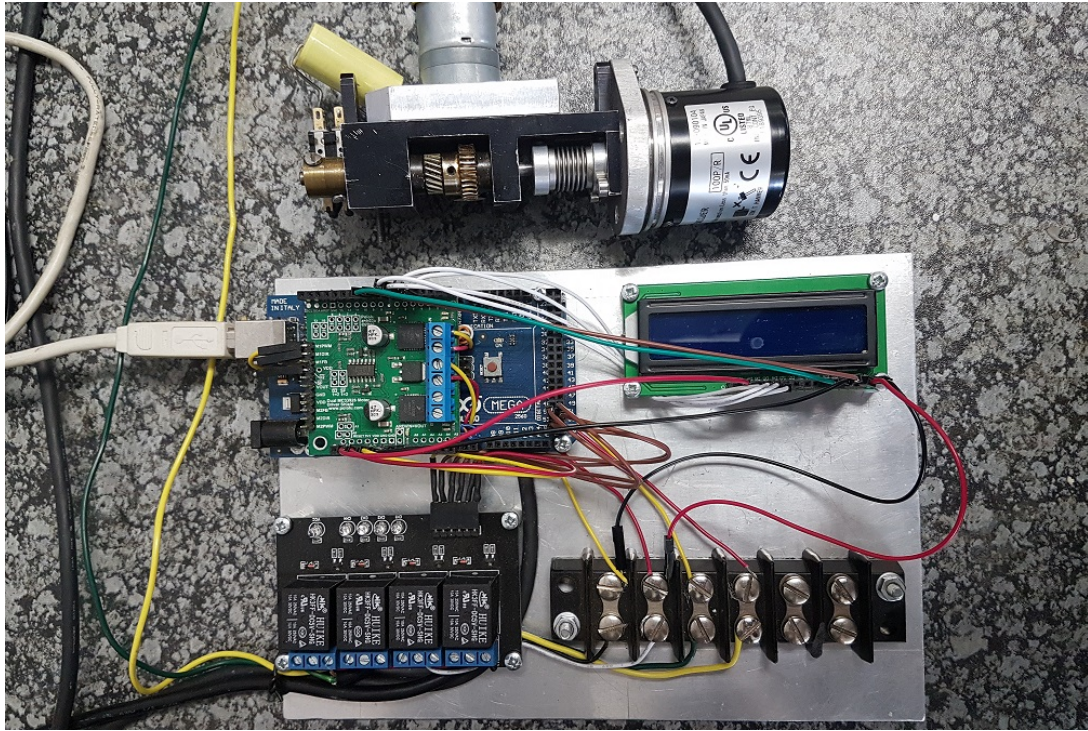


Figure 3.8: The electronic components (mounted on a metal plate) used for test purposes to control the eyepiece lens of the beam expander. The microcontroller, motor and the incremental rotary encoder play a significant role in making sure that any desired eyepiece lens position is achieved. The integrated Pololu motor controller, LCD screen and relay switches are for position monitoring and motor control purposes.

The motor specifications:

- Size 52mm (L) x 25mm (diameter)
- 4 mm-diameter D-shaped shaft
- Key specs at 6 V: 120 RPM and 80 mA free-run and 2.2 A stall
- Optimal Supply Volt 6V (Supply Range 3 – 9 V)

The motor and the incremental rotary encoder (Table (3.4)) play a significant role in making sure that any desired eyepiece lens position is achieved. Both these components were controlled through the microcontroller.

Relay 4 Channel specifications

- Supply Voltage: 5V

Table 3.4: The incremental rotary encoder electrical and mechanical specifications.

Electrical	Mechanical
- Input voltage 5VDC / 8 ÷ 24 VDC	- Shaft diameter ø4 g6 EL 30 E ø6 EL 30 H/I/E
- Input current with no output load Max 100 mA	- Enclosure rating IP54 standard
- Source and sink current 50 mA for channel	- Shaft speed 3000 RPM
- Output types NPN / NPN OPEN COLLECTOR	- Max shaft load 5N (0.5 Kp) axial 5N (0.5 Kp) radial
- Output frequency 100 KHz Max	- Shock 50 G for 11 msec - Vibrations 10G 10 ÷ 2000 Hz
- Frequency response F = RPM x Resolution	- Bearings life 109 revolutions - Bearings n° 2 ball bearings
	- Shaft material Stainless steel AISI303

- Each input opto-isolated with 817C opto-isolator
- LED indication for each Relay's Status
- Can be driven directly with Arduino

These electronic components were used for test purposes to determine the output beam diameter to propagate through the atmosphere. In the “absence” of atmospheric disturbances, the initial transmitted beam diameter determines the output beam divergence as the two are inversely proportional to each other (Equation (3.6)). Estimates from a 0.2 metre laser beam transmitted to the Moon provided a 650 m spot size radius, $1.69 \mu rad$ beam divergence and $0.09I_0$, where I_0 is the initial intensity of the transmitted beam. An accurate and precise estimation of the laser's spot size on the Moon is a highly useful feature that can determine if there would be any laser signal returns. Such returns depend on the effect of atmospheric turbulence fluctuations. A study by Yaoheng and Hesheng (2002) revealed that when the number of returned laser photons is less than 1 for a single transmitted laser pulse, a technical plan can be needed to increase the number of returned photons. That technical plan involved system integration with tilt sensors that allow their Gaussian laser beam to accurately and centrally fall on the lunar mirrors to maximise the number of returned photons that can be detected through the optical telescope.

The RS232 GUI is a Graphical User Interface written in C++, which is used to send commands to the electronic components via the MEGA2560 (Figure (3.9)). A baudrate of 115200 is used and the communications port is selected from the GUI. The code sample for this GUI is provided in Appendix E. This GUI helps to send commands that will accurately position and determine the direction of the eyepiece lens for precise laser collimation.

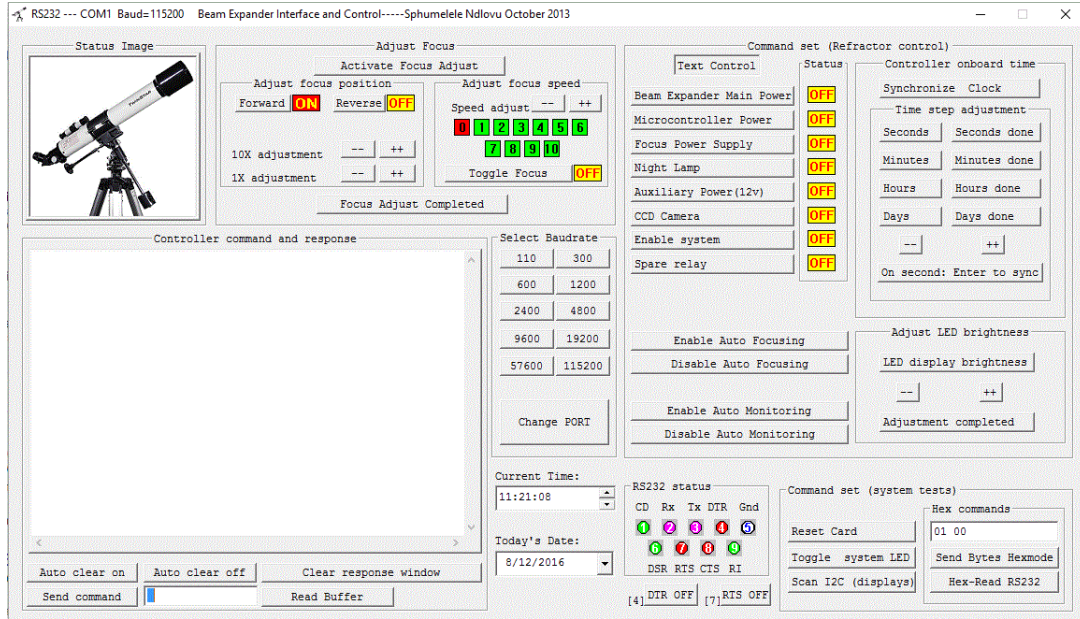


Figure 3.9: The RS232 GUI interface which sends commands to the code that is uploaded to the MEGA2560.

3.3.2 Cassegrain telescope method

Atmospheric turbulence is the most contributing factor to the decreased number of returned photons during laser ranging. Recent studies, on the effects of thermal turbulence on propagating laser beams, by Ndlovu and Chetty (2014; 2015) indicated that a laser beam propagating through air can easily undergo scintillation, beam wander and energy redistribution. Substantial improvements on reducing the internal effects that may be caused by the hardware and software during laser ranging have been achieved from the operating LLR stations since the beginning of LLR. Hence, the proof of this is the improvement in data accuracy that has been achieved since the beginning of laser ranging. However, the effects of atmospheric turbulence on propagating laser beams, with different beam sizes, is still a subject of interest.

The proposed model made use of a 1 m Cassegrain telescope to transmit and

receive laser signals for the HartRAO-LLR. For optimal efficiency, proper arrangement and alignment of the transceiver optics must be achieved so that the correct output laser beam diameter is obtained. A correct laser beam diameter is classified as such if it has the right shape and is properly collimated to enable successful laser ranging. Two ways of achieving correct laser beam settings are discussed in this section. The first method (method 2a) focuses on transmitting the laser pulses by illuminating one side of the transmitting mirror to avoid any structural obscuration. The other method (method 2b) considers transmitting an annular laser pulse, where the whole 1 m mirror is illuminated by the transmitted laser. In this case, 10% of the signal is obscured by the 0.3 m secondary mirror (Figure (3.10)).

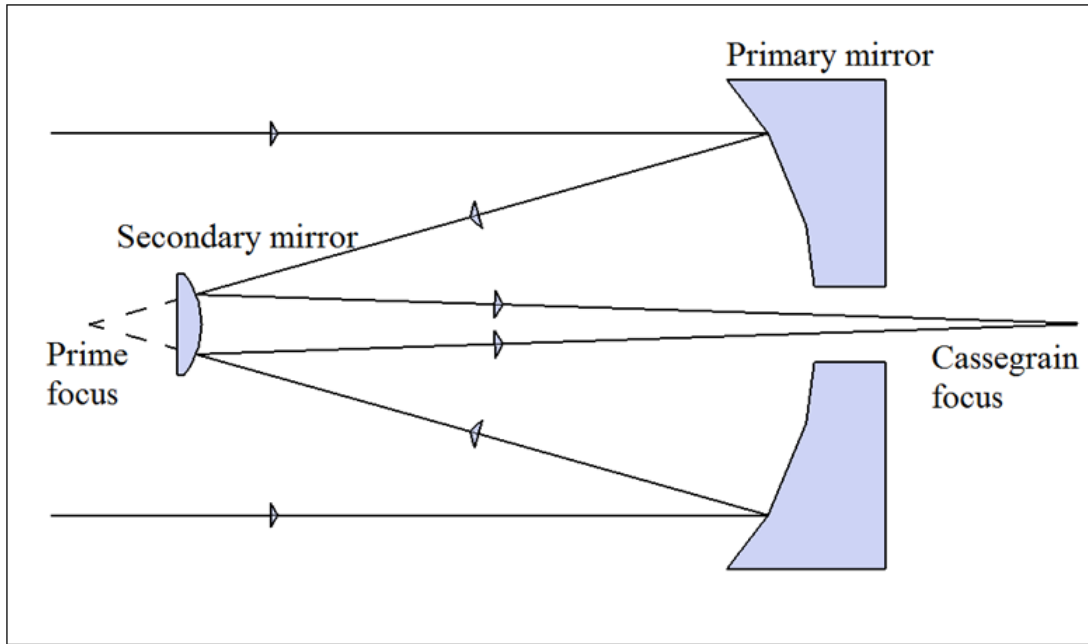


Figure 3.10: Schematic diagram of a Cassegrain telescope representing the proposed optics arrangement for the HartRAO-LLR telescope.

The final output beam diameter depends on the initial transmitted beam waist and the angle of incidence of the initial beam on the surface of the

transmitting mirrors. This dependence helps in determining the actual number of transmitted photons that can be reflected by the mirrors on the Moon and be detected by the ground telescope. It can be achieved by relating the number of photons directed to the reflectors with the laser energy redistribution at the specified distance. A comparison of the two methods obtained using a Cassegrain telescope configuration is provided in Table (3.5).

Table 3.5: The calculated output laser beam parameters for far field objects.

Transmitter	Spot size radius	Beam Divergence (μrad)	Irradiance
Method 2a (0.3 m)	425 m	1.13	$0.14I_o$
Method 2b (1.0 m)	182 m	1.96	$0.45I_o$

Doubling the diameter of the transmitted beam decreases beam divergence by a factor of 10, and thus more photons are reflected by the corner cube reflectors.

3.3.3 Coudé telescope method

A coudé focus gives a narrower field of view and is used together with heavy instruments that normally do not need a wide field of view (Stark et al. 1997). The model used in this section is similar to that used in Section 3.3.1 except that transceiver optics inside the telescope are of coudé configuration (Figure (3.11)).

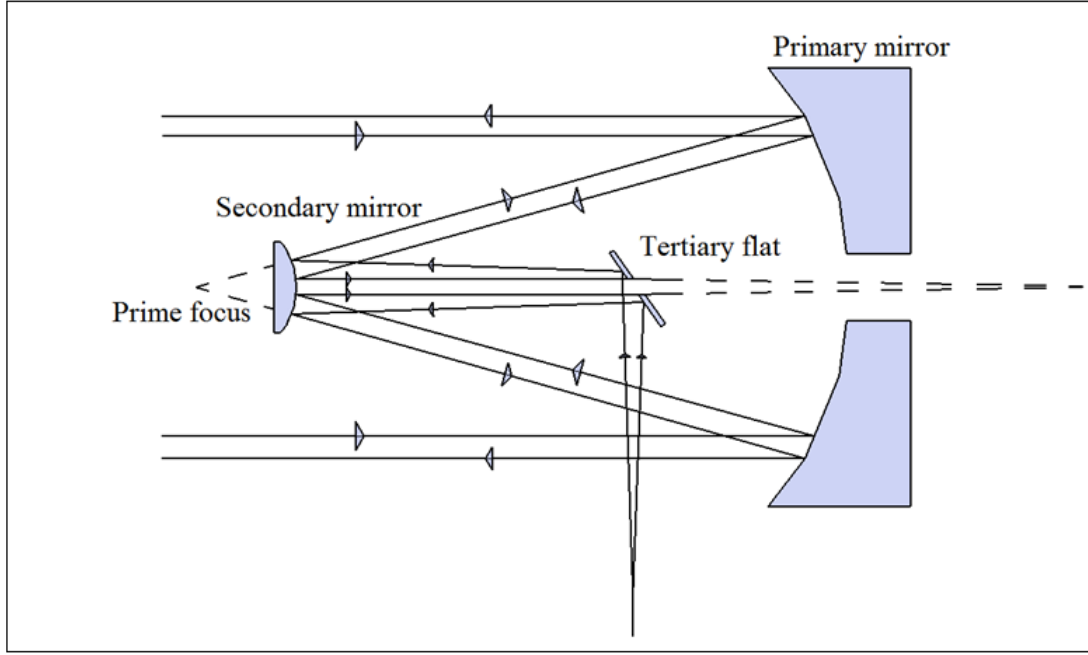


Figure 3.11: Schematic diagram of a transceiver coudé telescope with the secondary and primary mirrors aligned parallel to each other and the flat tertiary mounted in the middle of the telescope.

Instead of transmitting laser pulses from the back of the telescope, a flat tertiary mirror is mounted in-between the primary and secondary mirrors to transmit laser pulses. The Cassegrain focus at the back of the telescope is used to mount the photon detection system. All the partially transmitted laser signals are instantly detected as the start photons. The returned photons are also detected at the same location. The total transmitted laser beam diameter is equal to 1.0 m. Hence, the corresponding beam divergence is $1.96 \mu\text{rad}$ as mentioned earlier in Table (3.5) in Subsection (3.3.2). This method proves to be more favourable compared to Method 2. This is because the HartRAO telescope was initially designed by OCA to receive through a coudé focus. Thus, there is enough “room” for a coudé telescope configuration.

The initial beam diameter from the laser source for the HartRAO-LLR system is 12 mm. This diameter is expanded to 1 m when ranging using the full telescope

aperture. It is achieved by first expanding the beam diameter from 12 mm to 115 mm so that the flat tertiary lens in Figure (3.11) is fully illuminated by the laser pulse. This configuration will help in ensuring that the 0.3 m secondary aperture is fully illuminated. This will also ensure that the parabolic primary aperture is illuminated by the laser and the signal is directed to the Moon reflectors. A larger beam has higher chances of hitting the reflectors since it can be better collimated and therefore does not diverge as much as a smaller beam diameter. The only disadvantage with the discussed configuration is the presence of the obscuring secondary mirror that reduces laser energy by 10 % when transmitting as well as when receiving. A small beam divergence associated with the 1 m beam diameter and the mechanical structure of the telescope allows this method to enable optimal signal efficiency for the HartRAO Lunar Laser Ranging.

3.3.4 The HartRAO ranging simulator

An advanced mathematical tool, utilising C++ code, was developed and used to estimate the expected number of returned photons for the HartRAO's LLR system (Ndlovu et al. 2014; 2015; 2016). The tool allows the user to select a targeted satellite/lunar mirrors and adjust variable parameters. It will be used to evaluate computed and observed photon return efficiencies, using as departure point the existing link budget equation, with the option to add and estimate parameters in the least squares sense (Figure (3.12)). Other parameters such as Temperature, Humidity, Pressure, Moon Position, Moon Direction, Moon's mean anomaly etc. were directly linked to the mathematical tool from an Excel Spreadsheet. The cross section values for the retroreflector equipped satellites and lunar retroreflectors were calculated by Degnan (2012). The atmospheric transmittance and cirrus transmission values were varied between 0.02 to 0.9 and 0.1 to 1, respectively. The

calculated transmitter gain for HartRAO's system is calculated from Equation (1.5) and found to be 7.9×10^{10} .

The attenuating effects on the returned laser signal resulted from thermal and density fluctuations of the atmosphere. The long Earth-Moon distance is also one of the biggest degrading factors on the number of received photons due to the 4th power magnitude of the slant range. The atmospheric transmission and cirrus cloud cover transmittance have a influence on the number of returned photons. Additional research is still necessary to verify the accuracy of the tool. This includes the use of the currently operating LLR stations' system parameters to evaluate computed and observed photon return efficiency. The data can be used to construct Normal Points. A Normal Point is normally constructed from 5 - 100 returned photons, spanning 10 - 45 minutes of observations (Dickey et al. 1994).

A simple step by step guide to use the mathematical tool (the Ranging Simulator);

Step-1 Select the laser ranging Station;

Step-2 Choose the System to be used;

Step-3 Select the targeted Satellite;

Step-4 Press one of the Visualisation buttons to display a graph of your choice;

Step-5 Adjust the Transmitted parameters;

Step-6 Adjust the Additional parameters;

Step-7 Take note of the received data and;

Step-7 Press the Save Current Data button to save the obtained data into a text file. A pop-up window will appear, then choose location for the named text file and select save.

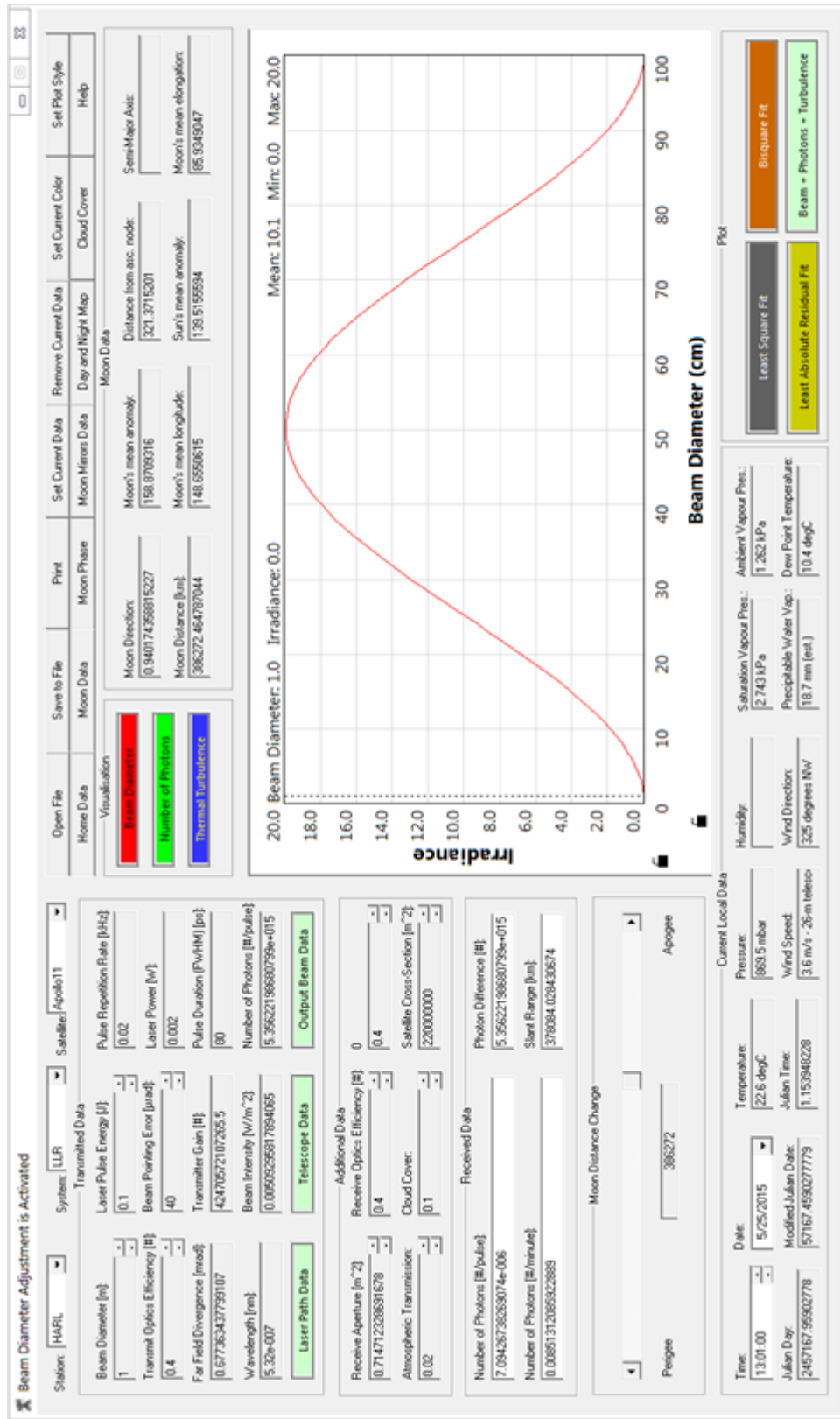


Figure 3.12: The Graphical User Interface to estimate the number of returned photons per minute using known and unknown parameters of the HartRAO-LLR system.

The saved data can be used later on for further scientific investigations.

3.4 Measurement challenges

HartRAO’s significant LLR measurement challenges are to (1) obtain centimetre to sub-centimetre accuracy using a single photon detection package; (2) determine the optical telescope configuration; (3) distribution of start and stop photon detectors; (4) determine accurate laser beam angle of incidence on retroreflectors and (5) maximise the number of returned photons.

The HartRAO photon detection system’s resolution will play a role in realising centimetre to sub-centimetre accuracy. Generally, a 25 ps resolution can be averaged to approximately a 7 ps uncertainty that can yield a targeted accuracy. However, this is only possible if there are some variations present that could spread the measurements over the several or more than 25 ps measurement “bins” (Michelsen 2010). Thus, averaging sufficient measurements can provide a fairly reasonable value. The HartRAO-LLR parameter variations originate from different factors such as effects of thermal fluctuations on the propagating laser, range variations and angle of incidence on retroreflectors.

Chapter 4

Simulations and their results

4.1 Introduction

The behaviour of atmospheric fluctuations, slant range, lunar retroreflector optical cross section and optics efficiencies of the S/LLR system determine the number of photons that can be obtained when using a single photon detector (Dirkx et al. 2015). It is evident from Equation (1.2), described in Chapter 1, that the returned number of photons varies more with changes in the atmospheric transmission and cloud cover than any other parameter. However, the largest number of returned photons is lost due to the fourth power slant range parameter which is inversely proportional to the number of detectable photons.

There are a number of parameters that are taken into consideration when determining the Moon's position in space such as the Moon's altitude, zenith angle and slant range. As the Earth-Moon distance varies with the Moon orbits, the slant range (R) helps to accurately determine the actual Earth-Moon distance which is measured from the centres of the two bodies (Figure (4.1)).

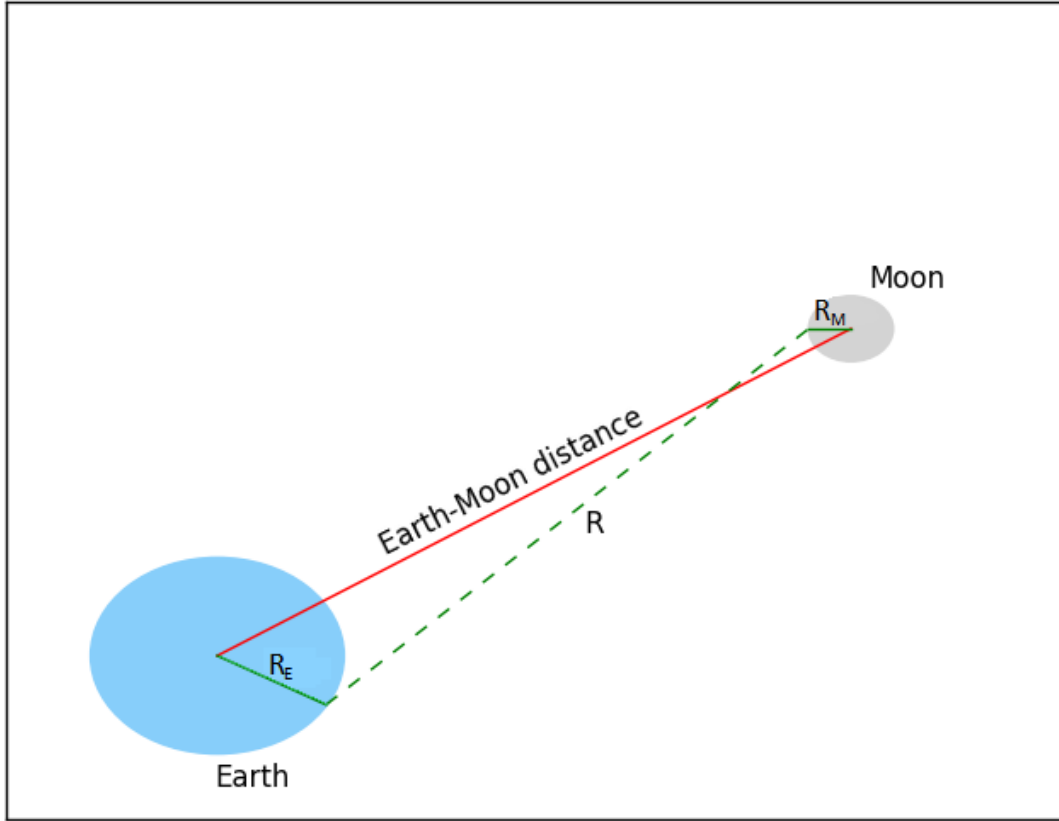


Figure 4.1: Schematic diagram of the Earth-Moon distance determined from the radius of the Earth, slant range and Moon's radius. The green dashed line represents the slant range, R , when measured during LLR, R_E and R_M are in solid green lines representing Earth and Moon radii, respectively.

The coudé telescope configuration type was found to be ideal for the HartRAO-LLR system. The exact size and placement of the flat mirror will have to be determined; it is expected to be placed in line with the elevation axis. The LLR telescope phase centre will then be the intersection of the elevation rotational axis with the rotation axis of the azimuth mount. This phase centre point will thus be centred on the flat mirror. A combination of optimal beam diameter and output energy, transmit and receive optics, as well as detection system efficiency will result in maximising photon returns.

4.2 System parameters

A comparison of photon returns when using three different beam diameter methods revealed that the coudé telescope is the most suitable configuration for the HartRAO-LLR. The preliminary results were obtained from a computer program (Ndlovu et al. 2014; Ndlovu et al. 2015; Ndlovu et al. 2016), utilising C++, which estimated the returned number of photons by using known and estimated parameters and different output beam diameters (Table (4.1)). The wavelength of the laser pulse is the same for all telescope configurations. The laser output energy for lunar laser ranging is 100 mJ. For calculations in Table (4.1), the output laser energy is expressed in percentages to indicate the amount of energy which is capable of leaving the LLR system under different transmitting configurations.

Table 4.1: The estimated photon returns from different transmitting methods.

	Method 1	Method 2a	Method 2b	Method 3
Maximum output beam, m	0.2	0.35	1.0	1.0
Output beam shape	Gaussian	Gaussian	Annular	Annular
Primary aperture, m	0.2	1	1	1
Obscuring mirror, m	0	0	0.3	0.3
Output energy, %	100	100	90	90
No. of photons/minute	0.003	0.005	5.75	5.75

Both method 2b and method 3 result in 5.75 returned photons per minute on average. The number of these reflections translates to high-precision Earth-Moon distance measurements. The angular resolution which is the diameter of the

optical intensity across the point spread function for imaging through the atmosphere (Figure (4.2)) is inversely proportional to the Fried parameter (a measure of the quality of optical transmission through the atmosphere due to fluctuations in its refractive index) expressed as,

$$\theta_A \propto k \frac{\lambda}{r_0}, \quad (4.1)$$

where θ_A is the angular resolution, k is the proportionality constant, λ is the beam wavelength and r_0 is the Fried parameter. If the diameter of the telescope aperture, D , is smaller than the Fried parameter, the seeing is given by the Rayleigh criterion (Argyle 2004), expressed as,

$$\theta_A = 1.22 \times \frac{\lambda}{D}. \quad (4.2)$$

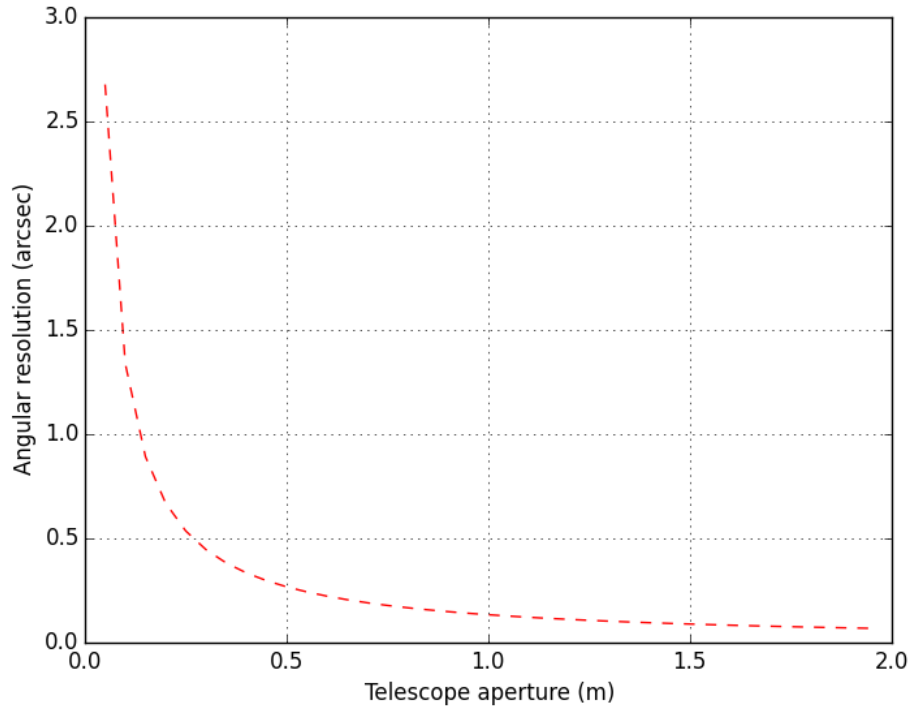


Figure 4.2: The relationship between the telescope aperture and the angular resolution.

Changes in atmospheric turbulence cause fluctuations in the refractive index of the Earth's atmosphere. These fluctuations result in a number of effects on the propagating laser beam which include beam wander, beam spread and scintillation. The variance of intensity fluctuation caused by scintillation is ≤ 0.02 , hence the scintillation has no significant effect on laser ranging (Churnside 1992). For laser ranging, only the beam wander and beam spread are mostly considered because of their effects on changing the angle of arrival of the wavefront from the transmitted beam (Yaoheng and Hesheng 2002).

The statistics for quantifying the atmospheric effects on the propagating laser beam are obtained from, (Yaoheng and Hesheng 2002)

Short-term beam wander:

$$\langle \rho_C^2 \rangle = \frac{10.22z^2}{k^2 r_0^{5/3} D^{1/3}} \quad (4.3)$$

Short-term beam spread:

$$\langle \rho_S^2 \rangle = \frac{4z^2}{k^2 D^2} + \frac{D^2}{4} \left(1 - \frac{z}{R(z)} \right)^2 + \frac{17.6z^2}{k^2 r_0^2} \left[1 - 0.48 \left(\frac{r_0}{D} \right)^{1/3} \right]^{6/5} \quad (4.4)$$

Long-term beam spreading:

$$\langle \rho_L^2 \rangle = \frac{4z^2}{k^2 D^2} + \frac{D^2}{4} \left(1 - \frac{z}{R(z)} \right)^2 + \frac{17.6z^2}{k^2 r_0^2} \quad (4.5)$$

where $k = \frac{2\pi}{\lambda}$ is the wavenumber, D is the laser transmitter diameter, z is the laser propagation axis and coordinate, R is the radius of curvature of the laser beam and r_0 is Fried's coherent length.

The corresponding short-term beam wander (Figure (4.3)), for quantifying the atmospheric turbulence effects on the 1 m HartRAO-LLR laser system, was calculated from Equation (4.3). The effects of atmospheric turbulence degrades

laser ranging precision with several millimetre to centimetre scale (Yaoheng and Hesheng 2002). Propagating laser beams through the atmosphere result in a series of effects because of the fluctuations in the refractive index of the fluctuating Earth’s atmosphere. These fluctuations occur on a time scale of ms and are related to the Fried’s coherence length r_0 .

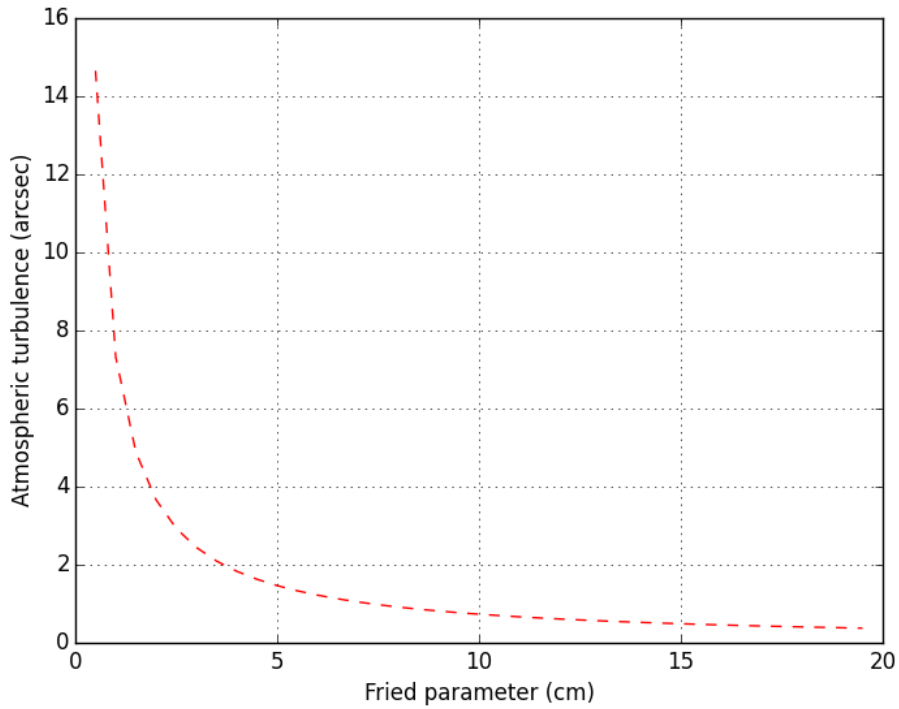


Figure 4.3: The effects of the atmospheric turbulence on a propagating laser beam at different r_0 .

The influence of beam divergence on a transmitted laser beam reduces the number of photons that illuminate the corner cube reflectors and are subsequently reflected back to the “Earth-fixed” receiving telescope. The relationship between beam divergence and the diameter of the transmitted laser beam plays a pivotal role in determining beam spread as it propagates through the atmosphere under various scenarios (Table (4.2) and Figure (4.4)).

Table 4.2: The varying parameters that influences the number of photons reflected by Apollo 11.

Parameter	Poor value	Optimal value
Transmit optics efficiency	0.4	0.9
Slant range (km)	396929	370013
Detector quantum efficiency	0.4	0.5
Receive optics efficiency	0.4	0.9
Atmospheric transmission	0.02	0.9
Cirrus transmission (Cloud cover)	0.1	1

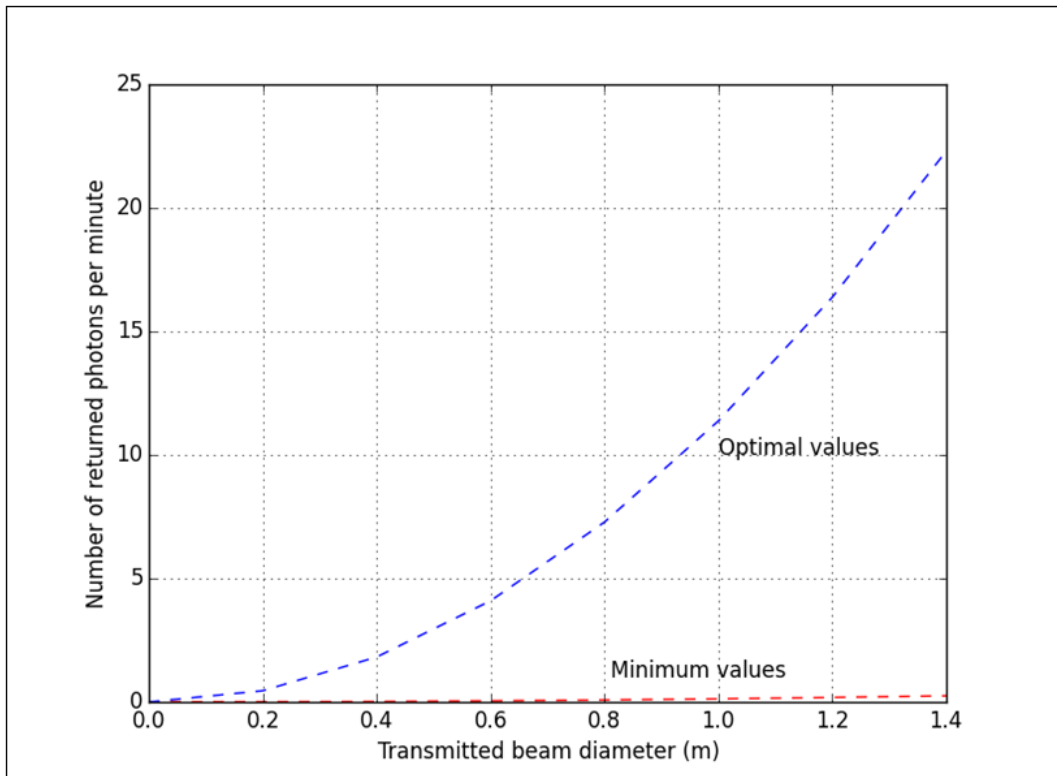


Figure 4.4: The diagram of the relationship between the transmitted beam diameter and the number of returned photons per minute.

The influence of atmospheric turbulence effects on laser ranging at elevation above 10 degrees can be corrected using classical corrective formulas (Marini and Murray 1973). This consideration can be done without considering the turbulence random movement of the refractive index of the atmosphere. A random path deviation that result from the fluctuations of the refractive index for a round trip laser can be used to calculate the covariance of the average path deviation in order to achieve laser ranging accuracy as described by Gardner (1976) as,

$$\Delta L^2 = \frac{3.127 C_n^2(0) L_o^{5/3} h_T}{\sin(E_L)}, \quad (4.6)$$

where $C_n^2(0)$ is the ground turbulence structure parameter, L_o is the turbulence strength outer scale, h_T is the atmospheric scale height and E_L is the elevation angle. For a station with good seeing conditions, the accuracy of the laser ranging affected by the atmospheric turbulence can be negligible (Yaoheng and Hesheng 2002). The effects of atmospheric effects on laser ranging depends on seeing and elevation angle (Table (4.3)). The observing stations with small elevation angle will only experience small differences in turbulence-induced measurement fluctuations (Dirkx et al. 2015).

Table 4.3: The atmospheric effects on laser ranging accuracy at different elevations (in degree).

Atmospheric turbulence strength, C_n^2	$E_L = 10^\circ$	$E_L = 30^\circ$	$E_L = 60^\circ$
$\sim 10^{-13} \text{ m}^{-2/3}$, strong turbulence	10.33	6.09	4.60
$\sim 10^{-15} \text{ m}^{-2/3}$, moderate turbulence	0.83	0.45	0.37
$\sim 10^{-17} \text{ m}^{-2/3}$, weak turbulence	0.17	0.10	0.08

4.3 Least Squares simulations

There are many methods which can be used for parameter estimations. This includes the method of least squares, Kalman filtering, robust regression and bias correction (Zhang 1997). In this work, the method of least squares for parameter estimations was considered. The nature of this work involved multiple parameter estimations with reasonable a-priori start-up values of the parameters in the link budget equation. These unknown parameters are estimated by determining their numerical values that minimise the sum of the observed values and the functional portion of the model (the link budget equation). In general, the criterion for least squares that provide the parameter estimates is:

$$\varepsilon = \sum_{i=1}^n \left[n_i - n_{pe(i)}(R, \sigma, T_a, T_c) \right]^2, \quad (4.7)$$

where ε is the least (sum of) squares, n_i is the observed number of the detected LLR photon returns and $n_{pe(i)}$ is the mathematical model (the link budget equation) that calculates the number of detectable photoelectrons given by Degnan (1993). The least squares minimisation for linear models is normally calculated using calculus but, for nonlinear models, the minimisation is done using iterative numerical algorithms. For the link budget model, the least squares parameter estimates were computed by minimising Equation (4.7) to

$$\varepsilon = \sum_{i=1}^n \left[n_i - \left(\eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \sigma \left(\frac{1}{4\pi R^2} \right)^2 A_r \eta_r T_a^2 T_c^2 \right) \right]^2. \quad (4.8)$$

Taking the partial derivatives of ε with respect to R , σ , T_a and T_c , and setting each partial derivative equal to zero, the following estimators for the parameters

were achieved;

$$R = \left[\left(\eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \sigma \left(\frac{1}{4\pi} \right)^2 A_r \eta_r T_a^2 T_c^2 \right) \div \sum_{i=1}^n n_i \right]^{1/4}, \quad (4.9)$$

$$\sigma = \sum_{i=1}^n n_i \div \left(\eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \left(\frac{1}{4\pi R^2} \right)^2 A_r \eta_r T_a^2 T_c^2 \right), \quad (4.10)$$

$$T_a = \sqrt{\sum_{i=1}^n n_i \div \left(\eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \sigma \left(\frac{1}{4\pi R^2} \right)^2 A_r \eta_r T_c^2 \right)}, \quad (4.11)$$

$$T_c = \sqrt{\sum_{i=1}^n n_i \div \left(\eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \sigma \left(\frac{1}{4\pi R^2} \right)^2 A_r \eta_r T_a^2 \right)}. \quad (4.12)$$

Here;

R is the distance between the surfaces of the Earth and Moon,

σ is the optical cross-section of the retroreflectors,

T_a is the atmospheric transmit efficiency and

T_c is the cloud cover transmittance.

The four unknown parameters, solved from the resulting system of four equations, determine the signal return rate. A high signal return rate can be translated to a better statistical determination of the lunar range. It is evident from these four equations that none of the parameter estimators are independent of each other. The relationship between the estimators can be analogous since the model being dealt with here is very complicated, includes statistically linear and nonlinear models. It was difficult to make a mental estimate of the accuracy of the parameter estimates due to parameter variations which are not always linear. However, comparing these estimates to that obtained from the direct calculations from the link budget equation, it was found that both these results were in agreement by an average difference of 0.00272 (Figure (4.5)).

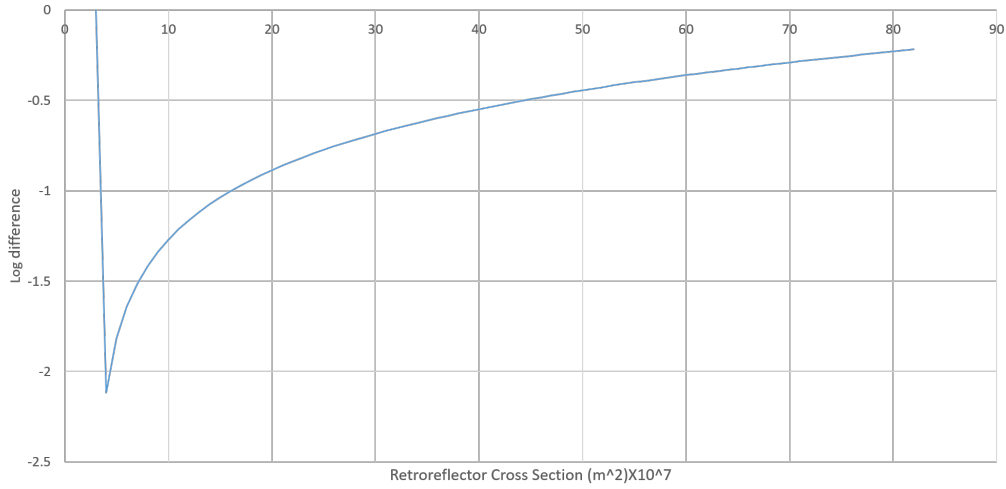


Figure 4.5: The logarithmic graph of the difference of least squares estimates and radar link budget equation estimates over a range of retroreflector cross section. The average difference between these two types of estimates was found to be 0.00272.

4.4 Parameter estimation

In this section, parameter values that were obtained when the link budget equation was used to estimate photons under varying atmospheric conditions are provided. The radar link budget equation consists of three features that ensure the success of laser ranging i.e. the values of the transmitted signal, the effects on the two-way signal and the returned signal. A new parameter was introduced in this work to simplify the number of system parameters that are treated as “fixed” system parameters. The retroreflector values were calculated from a mathematical model which is provided in Subsection 4.5.2. This section is concluded with the estimations of the transmittances of the atmosphere and cirrus clouds.

4.4.1 System value

Every existing LLR system is integrated with the modern engineering equipment to increase chances of receiving laser photons returns (Degnan 1993), expressed as,

$$C_{system} = \left(\frac{1}{4\pi}\right)^2 \eta_q \left(E_T \frac{\lambda}{hc}\right) \eta_t G_t A_r \eta_r, \quad (4.13)$$

where all the parameters are as defined in Equation (1.2) in Section 1.2. The C_{system} value can be used, together with the environmental fluctuations efficiencies, to estimate the number of returned photons that are reflected by retroreflectors. A relationship between the estimated number of returned photons (observed and computed) and the varying link budget parameters provide a better understanding of the behaviour of propagating laser beams for high precision measurements.

All the HartRAO-LLR estimations were done to enable signal path optimal efficiency that will yield an improvement in the return-energy of the laser for accurate ranges to the corner cube retroreflectors. The optimal signal path efficiency determined by the systems' and astronomical seeing parameters describes the relation between laser ranging and returned signal which is detected by the receiving telescope. Such a relation will result in a new, holistic approach towards an integrated LLR system, which determines ranges of very high accuracy. This holistic approach takes into account all factors which modify 'fixed' and variable parameters. It also enables the estimation of certain parameters (e.g. the optical transparency of Cirrus clouds, atmospheric transmission), which affect the accuracy of the link equation.

4.4.2 The reflector optical cross section

A software program, utilising Microsoft Foundation Classes (MFC)/C++, was developed at HartRAO for the LLR's signal path parameter estimation (Ndlovu et al. 2014; 2015; 2016). This was achieved by using the existing link budget equation where system and environmental parameters were varied for more realistic returned photon estimations. The influence of angles of incidence on reflectors varied with the active optical cross section as illustrated by Degnan (2012). In Figure (4.6), the relationship between the lunar mirrors cross section and the estimated HartRAO photons per minute returns under different weather conditions is presented. A larger cross-section reflects more number of returned photons.

The number of returned linearly photons varies with the lunar reflector cross section. A study by Degnan (2012) on the effective area of corner cubes has revealed that, at arbitrary incident angle, the area is reduced by the factor,

$$\eta(\theta_{inc}) = \frac{2}{\pi} \left(\sin^{-1} \mu - \sqrt{2} \tan \theta_{ref} \right) \cos \theta_{inc}, \quad (4.14)$$

where η is the effective area of the retroreflector, the quantity $\mu = \sqrt{1 - \tan^2 \theta_{ref}}$, θ_{ref} is the internal refracted angle as determined by Snell's law, i.e.

$$\theta_{ref} = \sin^{-1} \left(\frac{\sin \theta_{inc}}{m} \right), \quad (4.15)$$

and m is the refractive index of the retroreflectors. In this case, the retroreflector's optical material type is fused silica so $m = 1.45$. Thus, the peak optical cross-section is $5.8 \times 10^6 \text{ m}^2$ for the popular 38 mm diameter retroreflectors as measured in the centre of the reflected lobe which decreases as the incident angle increases

Apollo 11 and Apollo 14	Cross-Section*10 ⁶ m ²	Photons (Bad Weather)	Photons (Good Weather)
	0.0096	4.64656e-007	0.000185862
	0.00734	3.55268e-007	0.000142107
	220	0.0101	4.5285
	332	0.0151	6.8362
	796	0.0385277	15.4111
	956	0.046272	18.5088
Apollo 15	Cross-Section*10 ⁶ m ²	Photons (Bad Weather)	Photons (Good Weather)
	0.0147	7.11504e-007	0.000284602
	0.0246	1.19068e-006	0.000476272
	0.0288	1.39397e-006	0.000557587
	526	0.0271	9.4945
	659	0.0318967	12.7587
	997	0.0482564	19.3026
	2390	0.11568	46.272
	2870	0.138913	55.5651

Figure 4.6: The estimated number of returned photons under different weather conditions using the Apollo arrays data on Mirrors Cross-Section.

(Degnan 2012), i.e.

$$\sigma_{eff}(\theta_{inc}) = \eta^2(\theta_{inc}) \frac{\pi^3 \rho D_r^4}{4\lambda^2}, \quad (4.16)$$

where ρ is the reflectivity of the retroreflector which is typically equal to 0.78 for aluminium-coated back faces and 0.93 for uncoated Total Internal Reflection (TIR) surfaces, D_r is the retroreflector diameter and λ is the wavelength.

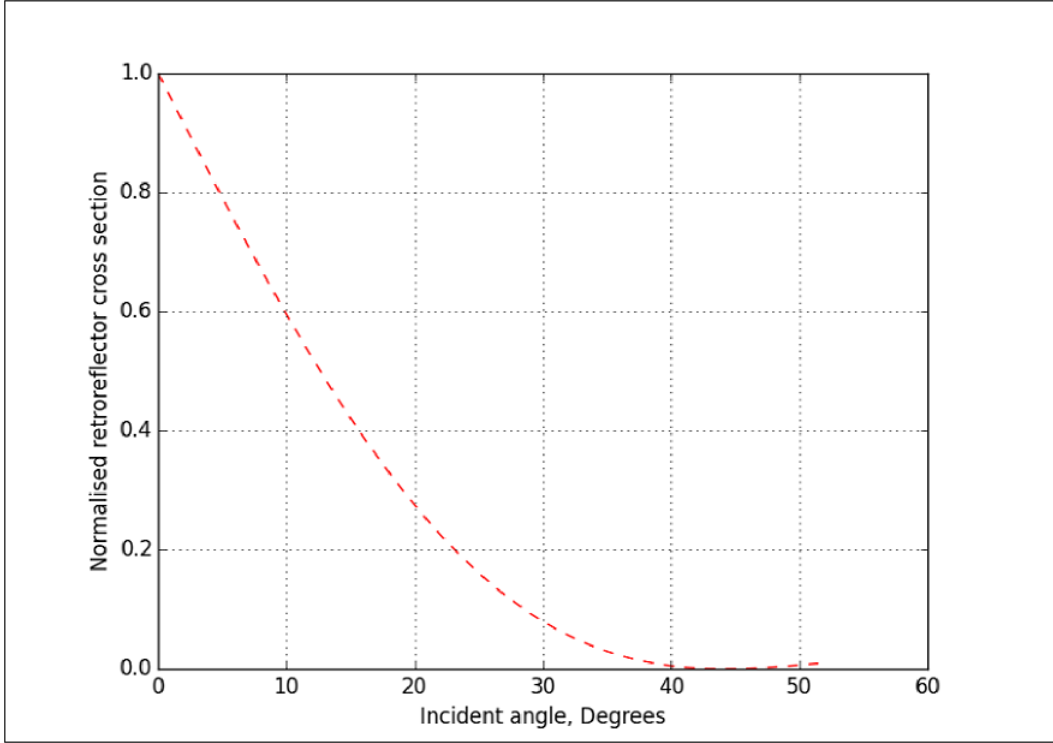


Figure 4.7: The normalised cross-section as a function of incident angle for fused silica retroreflectors. The 50% and 0% efficiency points for fused silica ($n=1.455$) are 13° and 45° respectively.

A normalised cross-section as a function of incident angle for fused silica retroreflectors can be used to illustrate the dependence of the number of returned photons on the retroreflector surface as the incidence angle varies. The angle of incidence of the transmitted laser beam at which this type of retroreflector reflects light is greater than zero (Figure (4.7)). Murphy et al. (2012) emphasised the relationship between photon rate and statistical determination of the

reflector range. It was found that most LLR stations prefer Apollo 15 mirrors as the target. The Apollo 15 array is the most convenient reflector to target due to the fact that it is three times larger than the first two Apollo arrays, it is located near the distinct topography (the Hadley-Apennine landing site) and it is less affected by lunar libration variations because of its location. Even the Lunokhod arrays deliver lower returned photons than the Apollo 15 array. The MLRS, OCA and APOLLO stations target the Apollo 15 array 83%, 80% and 50% of the times, respectively (Murphy et al. 2013).

Dust settling on the lunar reflector mirrors is considered to be one of the factors that reduces lunar reflector efficiency over time by a factor of 10, plus an additional decrease at full Moon (Murphy et al. 2010). The decrease at full Moon is caused by an enhancement of solar energy absorption by the lunar retroreflectors and their housing. During this period, the “Earth-fixed” photon detectors tend to detect too much background noise which becomes difficult to separate from the returned photons. The LLR observations at APOLLO which normally last less than an hour are confined to 75% of the lunar phase distribution, from $D = 45^\circ$ to $D = 315^\circ$, where D is the synodic phase relative to new moon at $D = 0$ (Murphy et al. 2010).

4.4.3 Atmospheric transmission effects

Atmospheric transmittance is characterized either by the transmission coefficient, which is equal to the fraction of radiation that passes through the atmosphere as rays fall vertically, or by the turbidity factor that indicates the extent to which the transmittance of an actual atmosphere under given conditions differs from the transmittance of an ideal atmosphere. Atmospheric transmission varies at different points on the Earth’s surface. Fluctuations in humidity and the dust content in

the air at a given point throughout the year determine the annual atmospheric transmittance at that point.

The minimum atmospheric transmission coefficient, equal to 0.02, is observed under unfavourable weather conditions, and the maximum transmission coefficient, equal to 0.9, is observed under favourable weather conditions as described in Table (4.2). Corresponding graphs (Figure (4.8)) for favourable and unfavourable weather conditions provide the expected number of returned photons per minute as calculated using the parameters of the HartRAO-LLR system. Our favourable or unfavourable weather conditions are determined by the sky visibility which, in this case, is affected by high cloud cover (e.g. Cirrus), air quality and temperature.

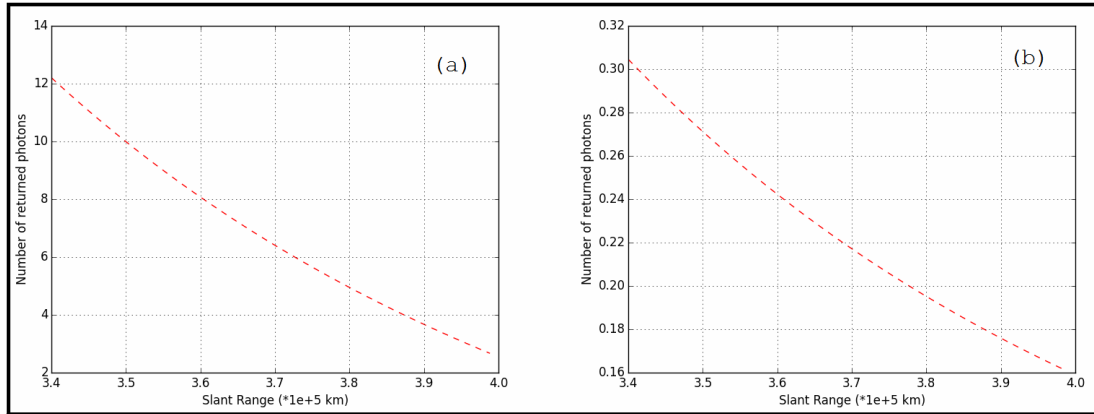


Figure 4.8: The expected number of returned photons per minute as calculated for the HartRAO-LLR system; graph (a) was obtained when the atmospheric transmittance was 0.9 under favourable weather conditions and graph (b) when it was 0.02 for unfavourable weather conditions.

A good site for laser ranging is determined by the magnitude of atmospheric transmittance which depends on the behaviour and quantity of different particles in the atmosphere (e.g. aerosols, dust, water vapour, thermal and density variations and air mass). The fluctuations in the atmosphere cause a propagating laser beam to diverge and adversely affect lunar laser ranging data quantity and

quality. Nickola (2012) investigated these fluctuations through determining the astronomical seeing conditions for LLR site selection and characterisation. Astronomical seeing conditions on-site were determined using a double star separation technique; seeing at Matjiesfontein was approximately 2 arcseconds. Even though site characterisation with respect to seeing requires prolonged periods of time to determine seasonal variations over time, this currently measured value indicates that Matjiesfontein could be a suitable site for laser ranging. At this site, the propagating laser beam through the atmosphere is expected to experience less beam degradation compared to HartRAO which is located in close proximity to Johannesburg industries.

4.4.4 Cirrus cloud effects

Cirrus clouds are thin, wispy white clouds that appear at high altitude (above 6 km) and consist of ice crystals. Such clouds are globally distributed at all latitudes over land or sea at any season of the year and the quantity increases with air traffic (Stordal et al., 2005). If present, laser ranging may be affected as these clouds undergo continuous changes in area coverage, thickness, texture, and position. The magnitude of cirrus cloud transmissivity is equal to 1 if no cirrus cloud is present. If there are any visible cirrus clouds present, which may be produced by weather disturbances in mid-latitudes, the minimum corresponding transmissivity magnitude is 0.1 (Degnan, 1993).

The measurement of the microphysical properties of cirrus clouds associated with dimension of ice crystals and water content in the mid-latitudes was proved to correlate with temperature (Heymsfield 1975 and 1977; Gerber et al. 1998; Larsen et al. 1998; Gallagher et al. 2005; Lawson et al. 2006). The distribution

of ice crystal size in cirrus clouds (Liou et al., 1990), and is represented as,

$$n(D) = \begin{cases} A_1 \times D^{B_1} \times IWC, & D \leq D_0 \\ A_2 \times D^{B_2} \times IWC, & D > D_0 \end{cases}, \quad (4.17)$$

where $n(D)$ is in units of $\text{m}^{-3} \mu\text{m}^{-1}$, D is the crystal maximum dimension in μm , IWC is the ice water content in gm^{-3} , B_1 and B_2 are slopes of curves, A_1 and A_2 are coefficients related to ice crystal concentrations at 100 and 1000 μm , respectively, and the crystal maximum dimension criterion is calculated from,

$$D_0 = \left(\frac{A_2}{A_1} \right)^{1/(B_1-B_2)}. \quad (4.18)$$

The single scattering albedo, the ratio of the scattering cross section to the extinction cross section, was computed from the parameter z (Figure (4.9)).

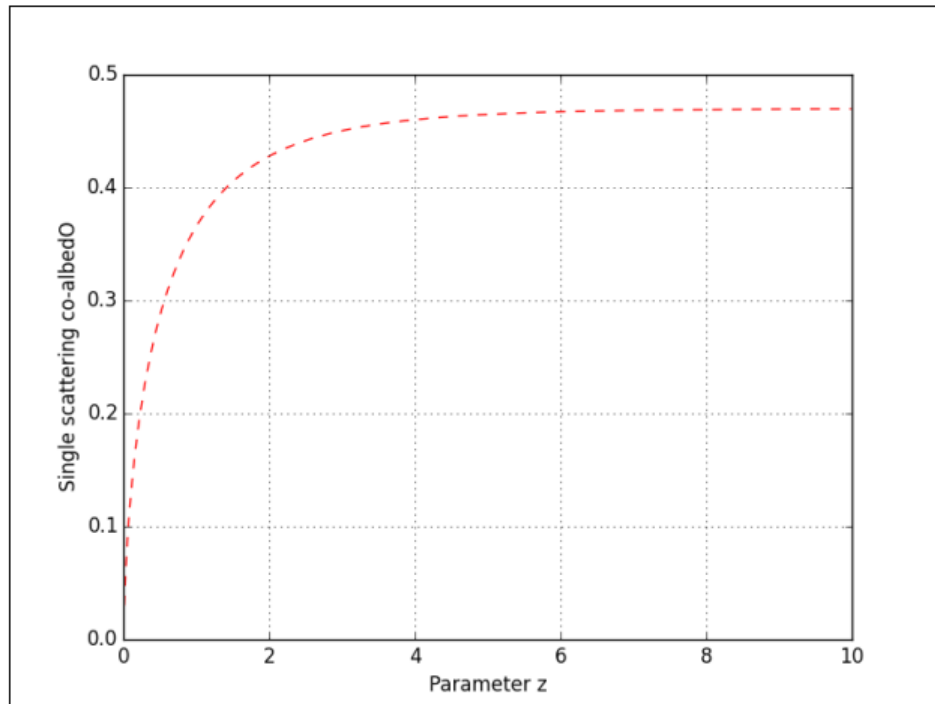


Figure 4.9: Best fit representation curve for any computed single scattering co-albedo.

The expression in Equation (4.19) was used to compute the scattering albedo (Liou et al., 1990):

$$\omega = 1 - 0.47 \left[1 - e^{az^b} \right], \quad (4.19)$$

where ω is the single scattering albedo defined as the fraction of reflected laser signal by cirrus clouds back to Earth, z is the physical parameter of the extinction cross-section which ranges from 0.01 to 10, $a = -1.5051$ and $b = 0.6789$. In the case of strong absorption, the scattering cross section is related to Fraunhofer diffraction and external reflections. The influence of the presence of Cirrus clouds on the laser signal (transmitted and received) is observed from the measurements of the total returned photons (Table (4.4)).

Table 4.4: The total number of estimated returned photons for the HartRAO station when varying the Cirrus cloud parameter and fixing all the other parameters, in the radar link equation, at their maximum except the atmospheric transmission which was at 0.41.

Transmittance, T_c^2	Photon return rate
0	0
0.1	0.67
0.2	1.34
0.3	2.02
0.4	2.69
0.5	3.37
0.6	4.04
0.7	4.72
0.8	5.39
0.9	6.06
1.0	6.70

4.4.5 Lunar reflectivity effects

For over 350 years, it has been known that the Moon is characterized by certain photometric properties, due to the special structure of its surface (Kopal 1961). At full Moon, the distribution of brightness is approximately the same at the centre as it is near the edges. In Figure (4.10), the photometry of the Moon can be used to determine the variation of its integral brightness as a function of the phase angle.

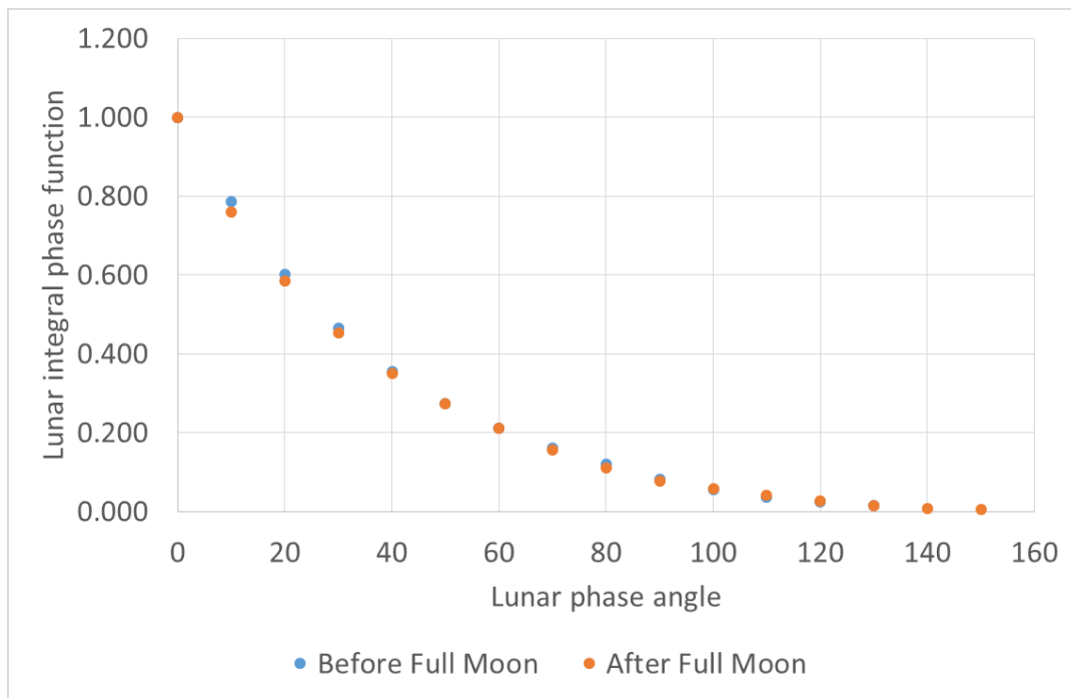


Figure 4.10: The relationship between the lunar integral phase function and lunar phase angle (Cox 2002).

The continuous 7-year period of data between 7 April 2009 and 21 December 2016 has been statistically compiled to illustrate the effects of lunar phases on LLR observations. The experiment was successfully operated on about 2388 ranging runs during that 7-year interval. These 2388 runs were ranged on the five lunar corner reflectors. In Table (4.5), breakdown of the individual statistics for each of the lunar arrays is provided.

Table 4.5: Apache Point Observatory individual ranging statistics for each retroreflector (7 April 2009 and 21 December 2016).

Reflector	Surface area, mm ²	Total measurements	Detected returns
Apollo 11	450 × 450	451	120656
Luna 17	440 × 190	186	36846
Apollo 14	450 × 450	476	122938
Apollo 15	1040 × 630	1125	396180
Luna 21	440 × 190	150	14627

The brightness of the Moon surface as observed from Earth is considered as one of the factors that also plays its role in reducing the number of observed LLR returns (Chapront et al. 2000). Most observations of the returns are observed during the First Quarter Moon and Third Quarter Moon. The observed data (Figure (4.11)) of LLR returns from the existing and active stations proved that the lunar phases, which is translated to lunar reflectivity, affects the observed LLR returns (Chapront and Francou 2003). The effects of lunar phases or reflectivity on laser ranging observations during a lunar month were observed.

A function to be used to account for the effects of lunar reflectivity during laser ranging is a necessity. This function is introduced as an efficiency parameter in the radar link equation to improve estimations of LLR returns. The function was generated from the data points from the Apache Point Observatory due to the number of experiments that have been conducted since the installation of its LLR system. This observatory has been consistent in measuring the Earth- Moon distance since 2006 (Murphy et al. 2008). The stacked data for LLR returns since 2006 to 2015 was used to generate a function that relates the number of LLR returns and Moon age (days). The obtained function can be expressed as,

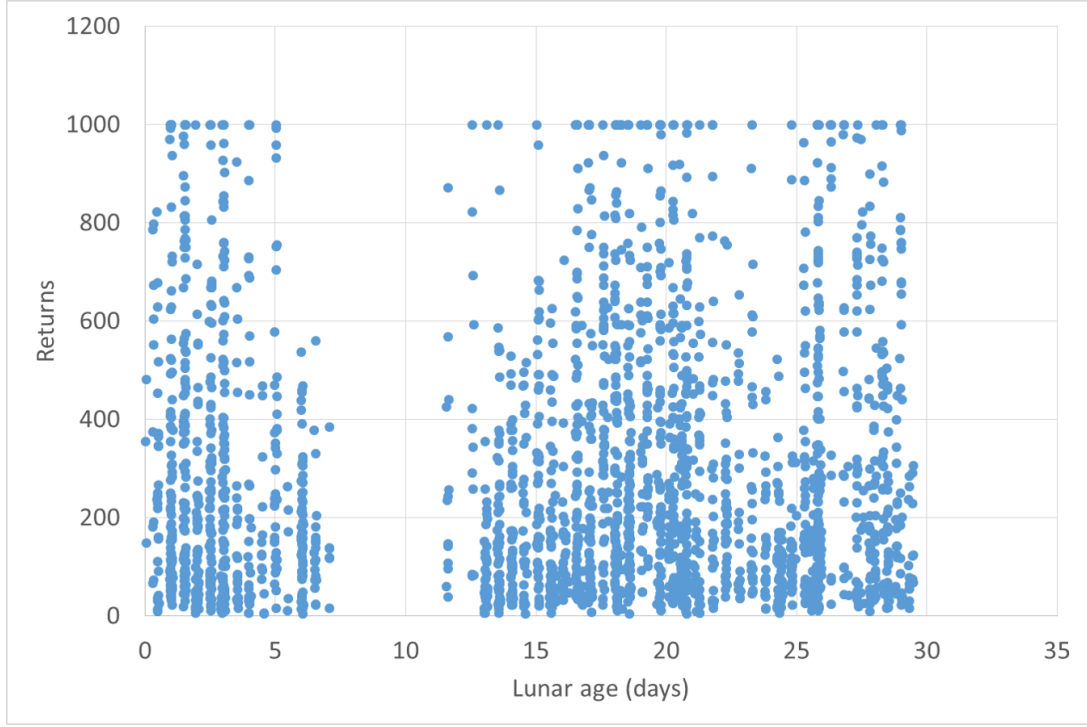


Figure 4.11: The 2006 – 2015 stacked data for Apache Point Observatory LLR distribution of returns for both day and night observations during a lunar month.

$$\begin{aligned}
 n_p = & -0.0001M_{PE}^6 + 0.0101M_{PE}^5 - 0.3849M_{PE}^4 + 6.7718M_{PE}^3 \\
 & -52.714M_{PE}^2 + 131.66M_{PE},
 \end{aligned} \tag{4.20}$$

where M_{PE} is the Moon age.

The function to estimate the effects of the lunar reflectivity on LLR observations was expressed as,

$$\begin{aligned}
 L_r = & 3 \times 10^{-8}M_{PE}^6 - 3 \times 10^{-6}M_{PE}^5 + 6 \times 10^{-5}M_{PE}^4 + 3 \times 10^{-4}M_{PE}^3 \\
 & -1.5 \times 10^{-2}M_{PE}^2 + 5.4 \times 10^{-3}M_{PE} + 0.9995,
 \end{aligned} \tag{4.21}$$

The maximum lunar reflectance efficiency is equal to 1.00 when the lunar surface is completely dark, as observed from Earth (Figure (4.12)). Ideally, this is expected to yield optimal LLR returns. However, the reflectors are in darkness and the observations are normally cancelled because of poor contrast. During this period (New Moon), the LLR system can be used for SLR observations or to prepare the LLR data for transmission, repair equipment and making system improvements.

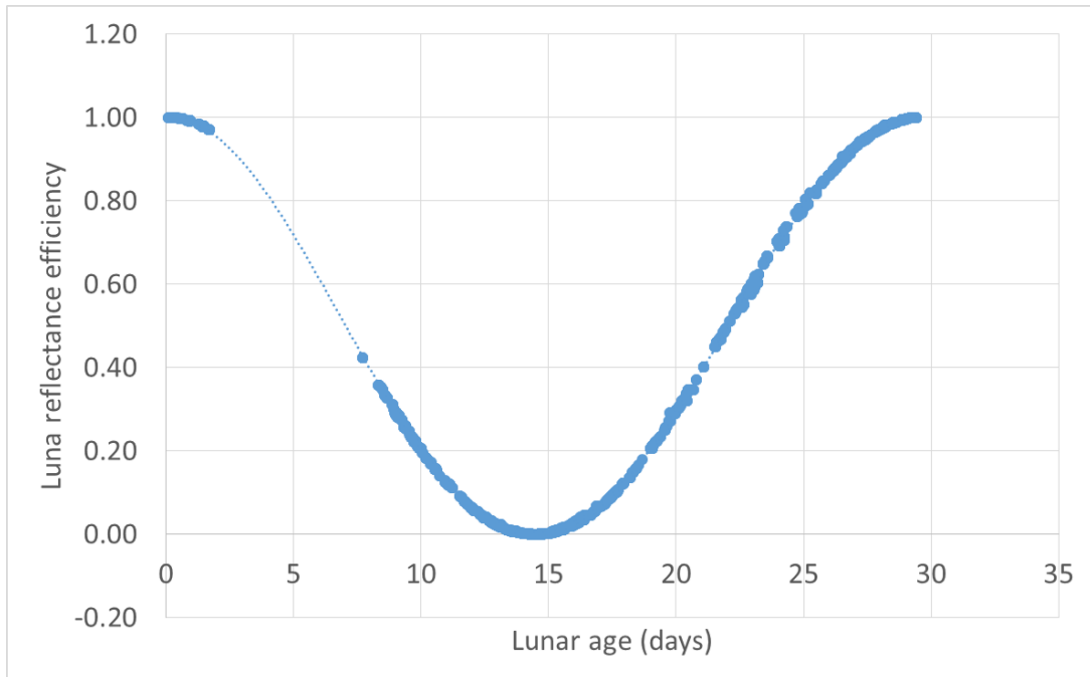


Figure 4.12: The relationship between the lunar reflectance efficiency and the lunar age (days). The plot was derived from the available distribution of the number of Apache returns during the periods 2006-2015.

4.5 Expanded radar link equation

The new parameter was introduced to account for the effects of lunar phases during laser ranging. This parameter expanded the existing radar link equation discussed by (Degnan 1993), now expressed as,

$$n_{pe} = \eta_q \left(E_T \frac{\lambda}{hc} \right) \eta_t G_t \sigma \left(\frac{1}{4\pi R^2} \right)^2 A_r \eta_r T_a^2 T_c^2 L_r. \quad (4.22)$$

The minimum value for $L = 0$ at Full Moon, which is an indication that there will be no observations or returns during that period of that lunar phase. The varying parameters in the radar link budget equation now includes the obtained lunar reflectivity efficiency values. The relationship between the lunar reflectivity and the number of expected returns obtained using the radar link equation (RLE) and the expanded radar link equation (ERLE) are provided in Figure (4.13). In Figure (4.14), the variations in the estimated number of returned photons for HartRAO LLR due to changes in the atmospheric transmittance and lunar reflectivity.

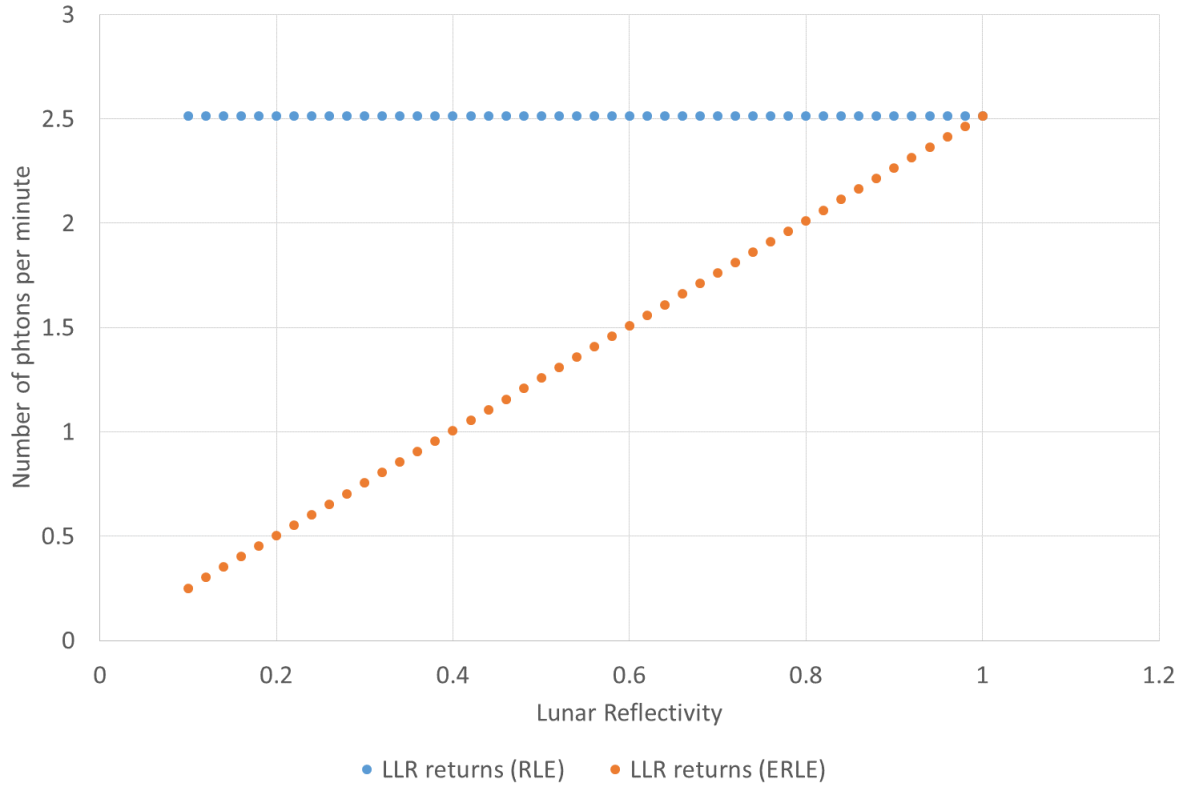


Figure 4.13: A comparison of the number of estimated LLR returns from HartRAO where the blue graph indicates that the lunar reflectivity efficiency was not considered (RLE) and the orange graph indicates the inclusion of the lunar reflectivity parameter (ERLE).

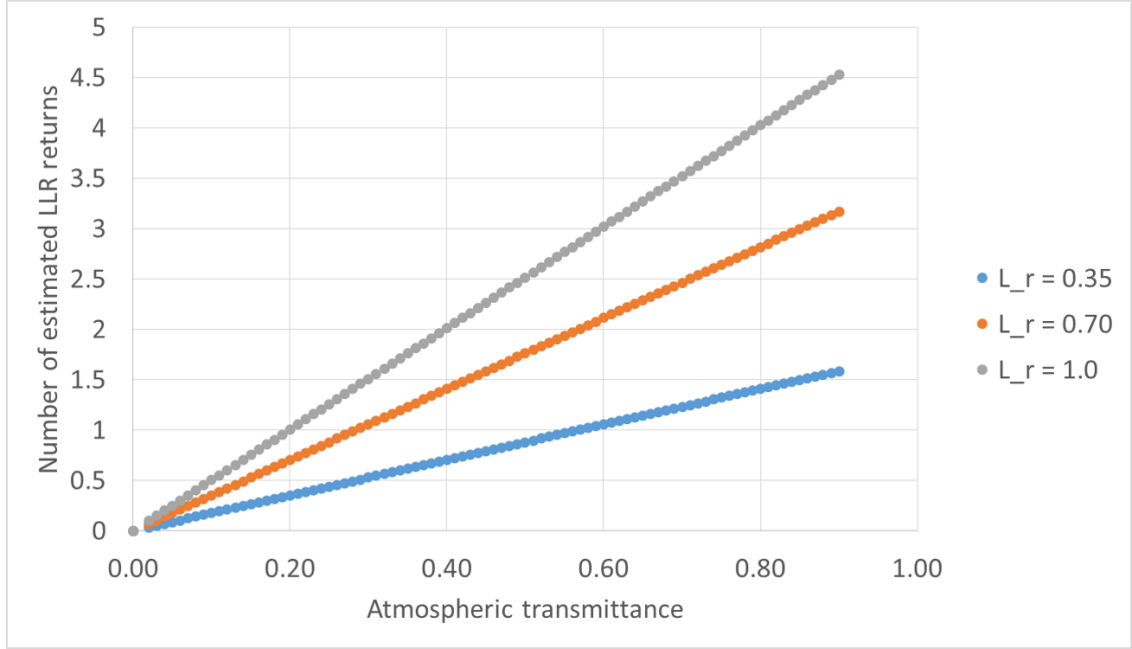


Figure 4.14: The variations in the estimated number of returned photons for HartRAO-LLR system due to changes in the atmospheric transmittance and lunar reflectivity ($L_r = 0.35, 0.70$ and 1.0).

The grey graph ($L_r = 1.0$), in Figure (4.14), assumes maximum reflection of returns and this was assumed in the original radar link budget equation which assumed absence of the effects of lunar reflectivity on the estimated LLR signals. The number of estimated LLR returns, using the expanded radar link equation, are now taking the effects of lunar reflectivity into account. This has a direct relationship with the total LLR system efficiency. It also provides an understanding of the effects of the Moon phases or lunar reflectivity during LLR observations.

4.6 Returned photons data

The measurement principle used in Lunar Laser Ranging is to fire laser pulses towards the targeted retroreflector array on the Moon. The aim is to receive back localised and recognizable laser signals and to measure the duration of the

round-trip travel of the light. Although the concept seems elementary, it is in fact a real technical challenge. At first, the performances of the observations depend on the quality of the time measurements. On average the duration of the round-trip flight is 2.5 seconds varying according to the lunar distance, the mean distance Earth-Moon being 385000 km. Achieving a precision of 1 cm for the separation between transmitter and reflector requires an accuracy of the order of 0.1 nanosecond (10^{-10} second) in the measurement of the round-trip travel duration (Chapront and Francou 2003).

The LLR measurement of time is based on a very stable high frequency signal generated by a station clock whose the frequency accuracy is better than 10^{-12} yielding an uncertainty below 10 picoseconds (10^{-11} second) over the journey of the light (Chapront and Francou 2003). Though, there are several factors that affect the precision of the measurements (i.e. affecting the number of returned photons). This includes the atmosphere which induces a time delay which is difficult to estimate rigorously, probably between 50 and 100 picoseconds; it depends on temperature, pressure and humidity. The libration of the Moon makes an oscillation of the orientation of the array which produces, in worse cases, an unbiased scatter in distance of a few centimetres (about 200 picoseconds in the time of flight). There are other sources of uncertainties such as the photodiodes connected to the timer or the return detector.

The determination of the Earth-Moon distance is not straight forward. As the surface of the Moon reflects light from the Sun, a detection of photons from the Sun or the Earth's atmosphere will always be possible. So, filtering of light is always needed. Spectral filtering only passes those photons to the detector which are very close to the transmitted laser frequency. Spatial filtering only considers the photons that are originating from a small region around the target. This means

keeping the field of view as small as possible. Temporal filtering is used to consider photons, which lie very close to the expected round trip travel time. Mastering the processes of measuring as well as analysing the number of returned photons yields accurate measurements of the total number of returned photons that can be obtained for a station.

The total number of the returned photons over time for each retroreflector array and station site is provided in Table (4.6). The total returned photons for each station is consistent with the sizes and positions of the lunar retroreflector arrays (<http://ilrs.gsfc.nasa.gov/network>). The Apollo 15 array is large in size and it is well positioned, more to the centre of the Moon. Hence, it yields more data for each station site. The Luna 17 array is able to reflect laser photons back to Earth. Most stations started to range to this “lost and found” array in year 2010 but yet its returned photons for APOLLO measurements are consistent with that of Luna 21.

Table 4.6: Total of the returned photons over time, from the four active LLR stations, for each retroreflector. The estimations from the HartRAO station are also included.

Station	Aperture size (m)	Time span (years)	Total returned photons		
			Apollo 11&14	Apollo 15	Luna 17 & 21
APOLLO	3.5	2006-2015	120656 & 122938	396180	36846 & 14627
OCA	1.54	2009-2015	7615 & 6651	50504	1140 & 237
Matera	1.50	2003-2015	114 & 58	2005	no data
McDonald	0.76	1988-2015	2634 & 3378	38627	5 & 114
HartRAO	1.0	2018-2020	~330 & ~325	~1125	~170 & ~150

The estimations for the expected total returned photons for the HartRAO station were computed using the known system parameters and the HartRAO station annual meteorological data. The estimated returned photons for the HartRAO-LLR assumed a 100 % station availability and an average return rate of 5 photons per minute for Apollo 11 and 14 lunar arrays. This average return rate was based on the obtained estimations from the HartRAO-LLR software for returned photons. Based on the total returned photons from the four active LLR stations for each retroreflector (Figure (4.15)), the total number of estimated photons from the HartRAO station were also consistent with the size, location and efficiency of each retroreflector array. The returned photons to be measured from the HartRAO station are expected to have an improvement on ILRS data contribution. This is because the station has a favourable geographical location with an average astronomical seeing of about 2 arcseconds.

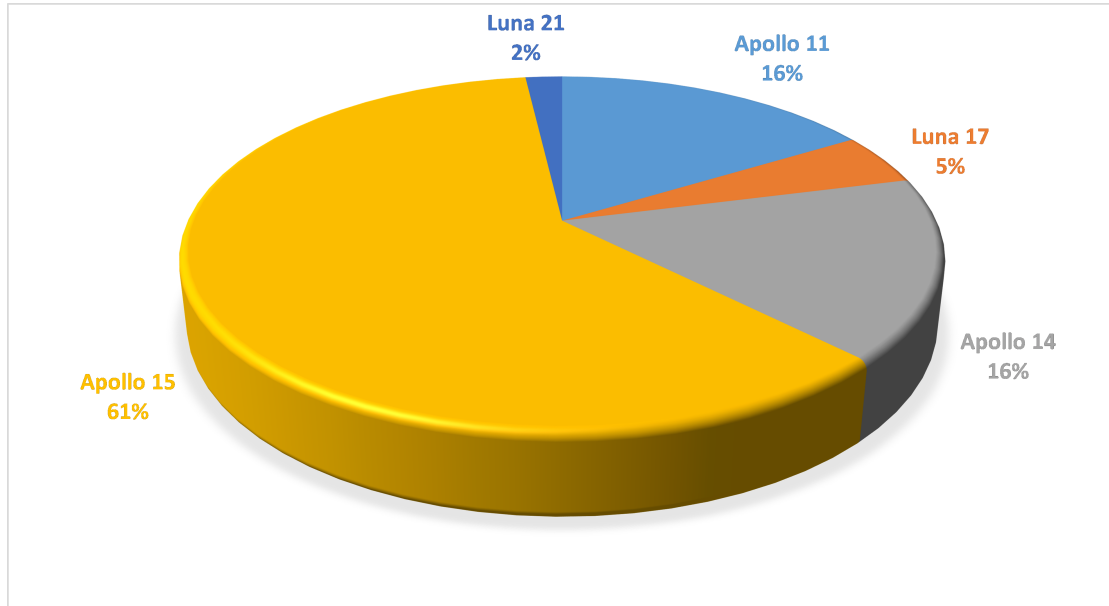


Figure 4.15: Statistics of the total detected photon returns from each lunar retroreflector array between 1988 and 2015.

It is observed from available Apache LLR data

(<http://ilrs.gsfc.nasa.gov/network>), for the total number of returned photons since 2006 to 2015 (Figure (4.16)), that this station is capable of collecting a greater number of returned photons owing to its large receiver telescope and detection system. Considering the age (of more than 45 years) of the lunar retroreflector arrays, all these arrays are still capable to reflect laser signals that are transmitted from the ground stations. The performance upgrades on the existing ground stations have improved the ranging accuracy by more than two orders of magnitude, i.e., a factor of 140 (Currie et al. 2008; Currie et al. 2009).

Most of the detected returned photons are reflected by the Apollo 15 array which is larger in surface area when compared to the other arrays. Its position on the Moon surface also plays a vital role in maximising measurements. To this day, the existing Apollo retroreflector arrays contribute a significant fraction of the limiting errors in the range measurements through lunar librations, described later in Section 5.2.3. The improved recent data (Figure (4.17) to Figure (4.19)) was obtained from the improved performances of the ground stations. The Apollo 15 is the most “ranged to” array and Luna 21 array is the least considered array for almost all the ground LLR stations.

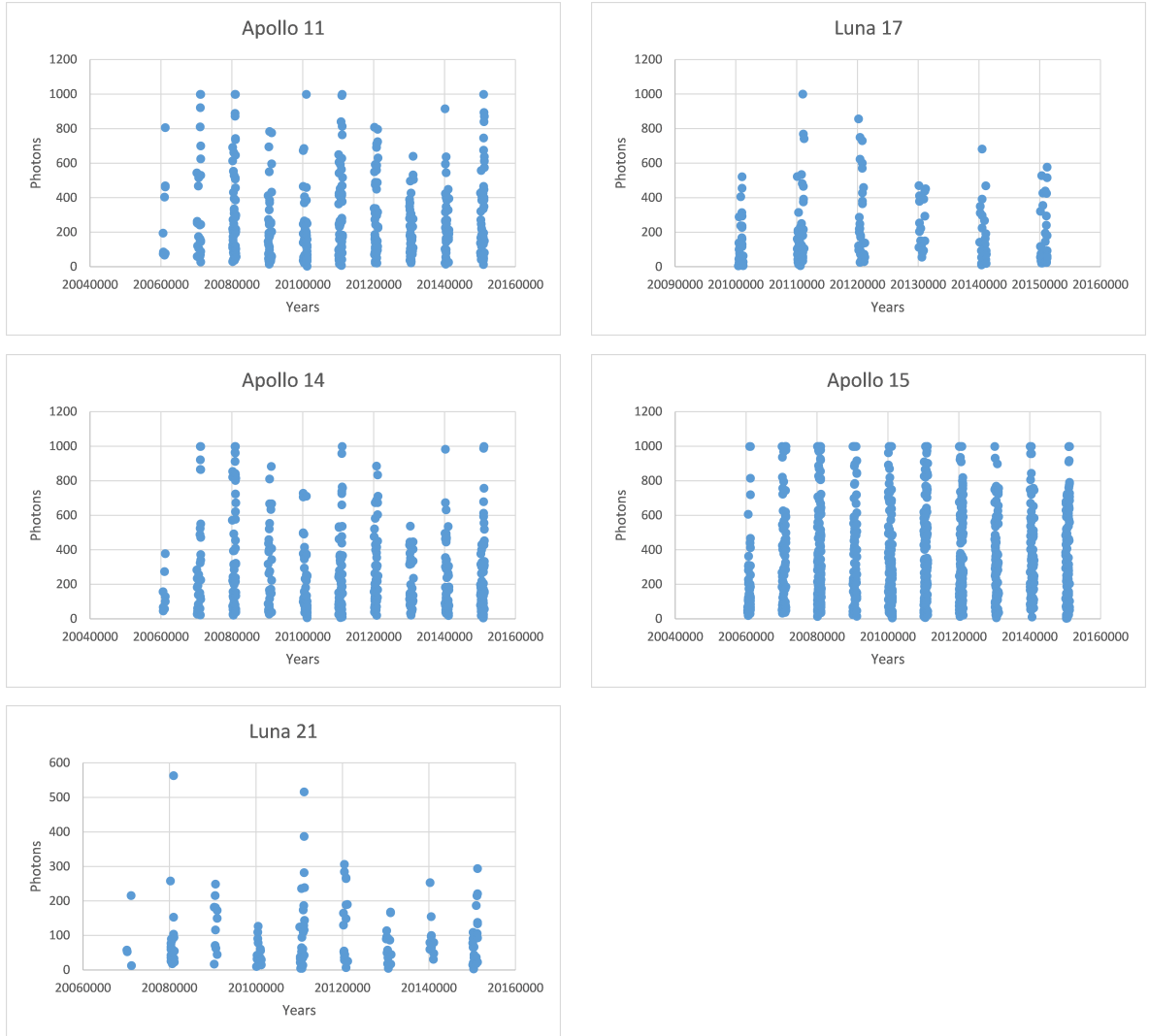


Figure 4.16: The annual (2006 to 2015) total returned photons, for each lunar retroreflector array, obtained from the Apache Point Observatory Lunar Laser-ranging Operation.

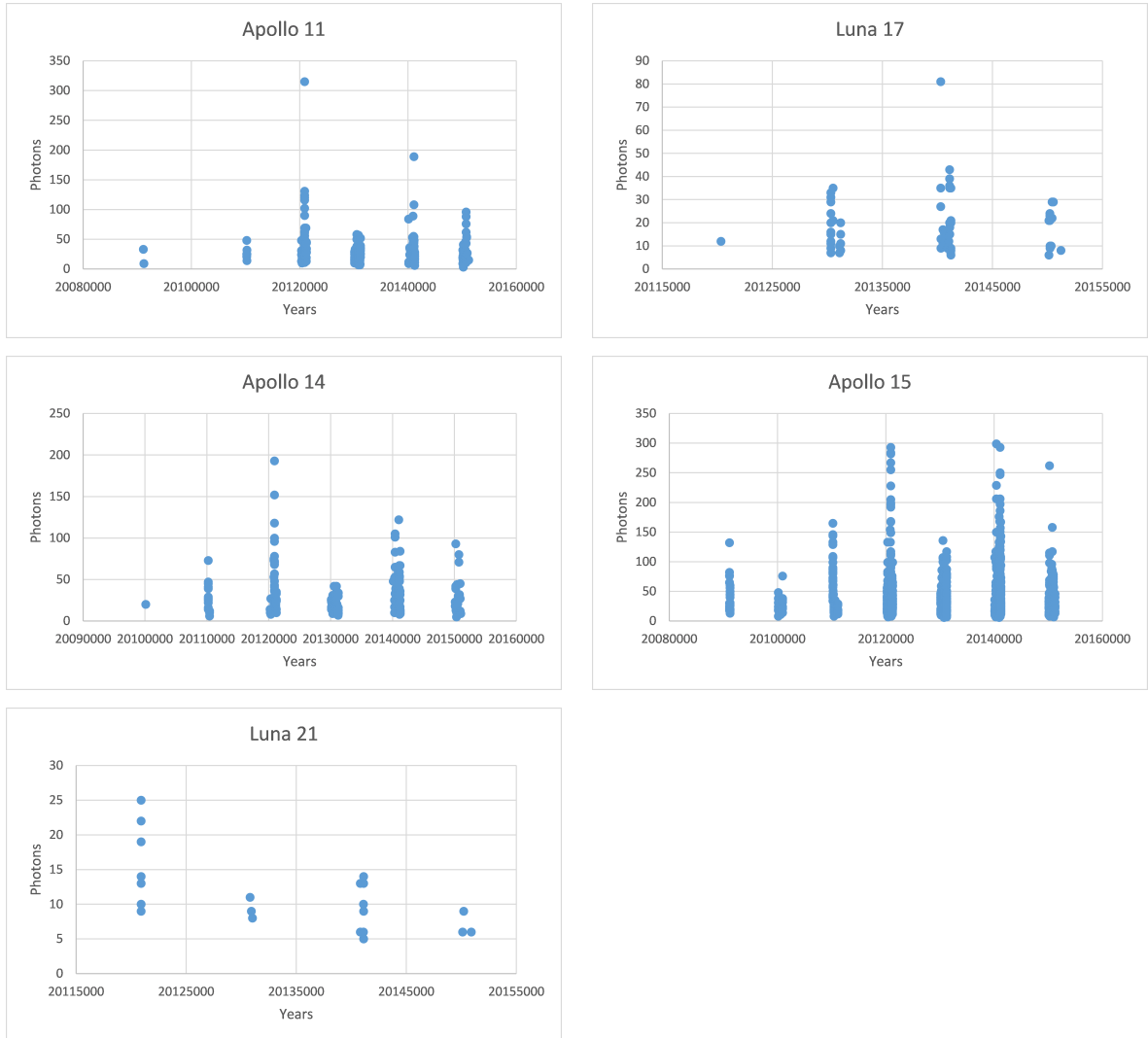


Figure 4.17: The annual (2009 to 2015) total returned photons, for each lunar retroreflector array, obtained from the Observatoire de la Côte d’Azur.

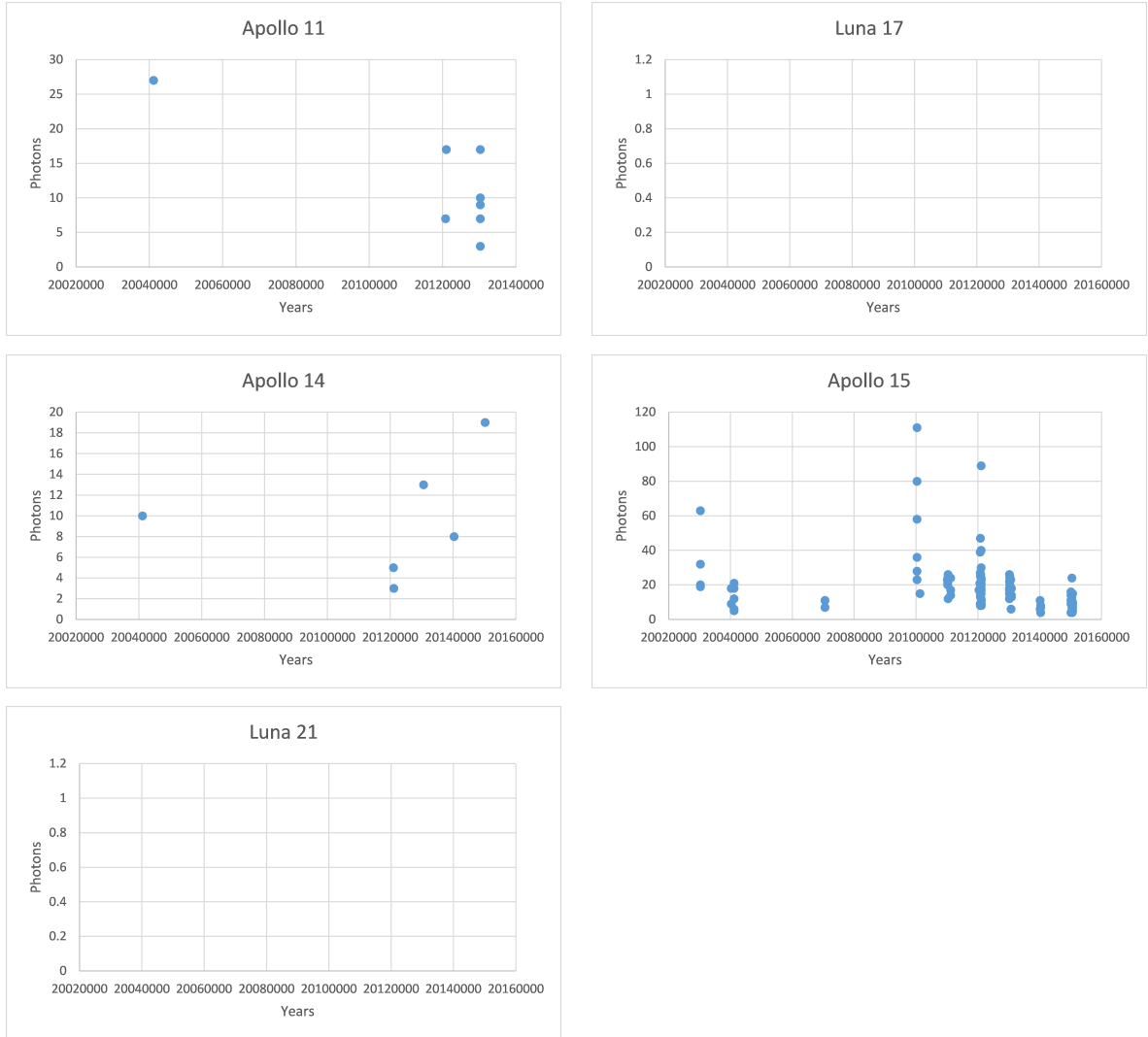


Figure 4.18: The annual (2003 to 2015) total returned photons, for each lunar retroreflector array, obtained from the Matera Laser Ranging Observatory. There are no data available for Luna 17 and Luna 21 arrays.

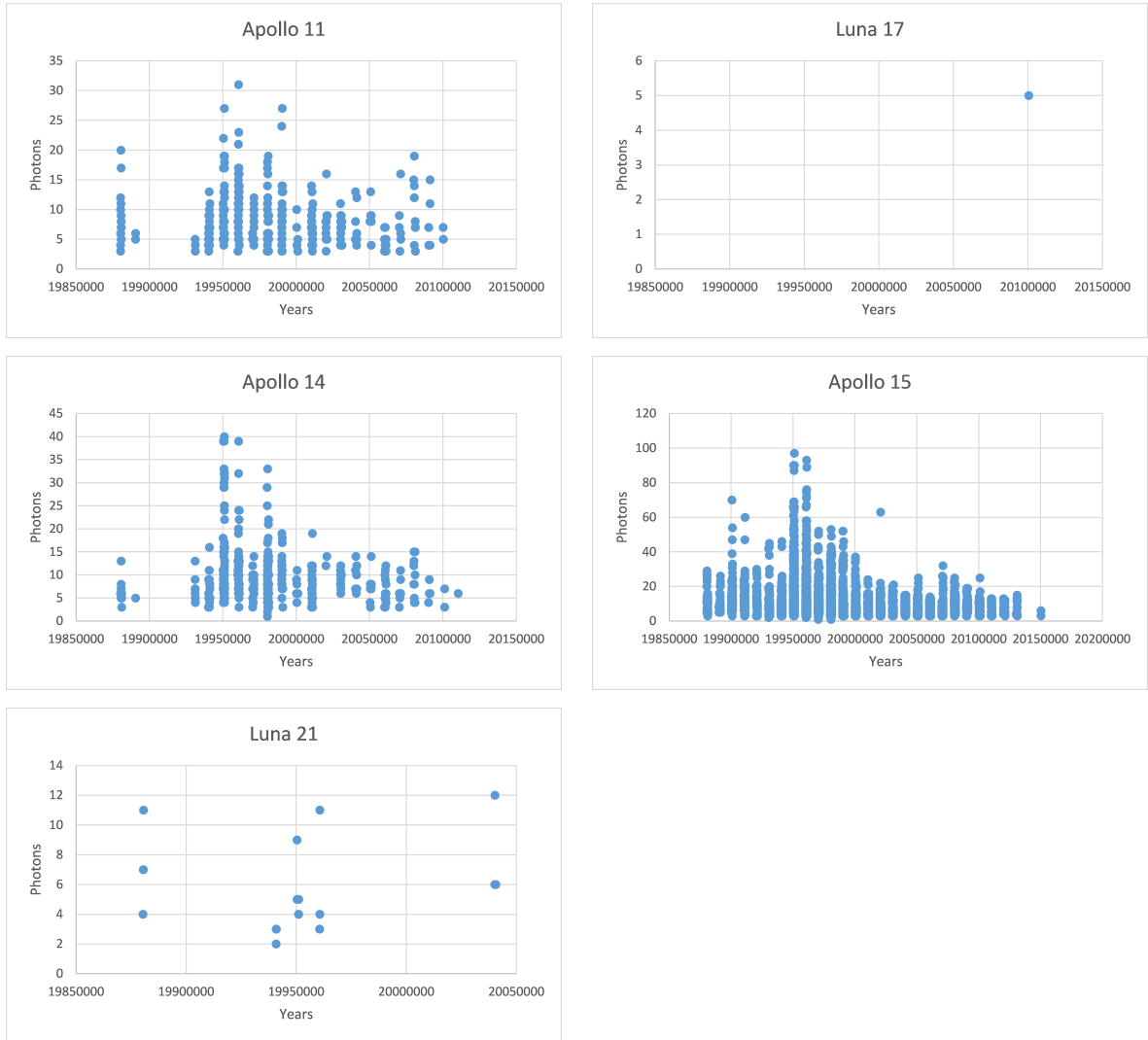


Figure 4.19: The annual (1988 to 2015) total returned photons, for each lunar retroreflector array, obtained from the McDonald Observatory.

Chapter 5

Discussion

5.1 Introduction

An integrated model and system to enable optimal signal path efficiency is a requirement for any LLR system to estimate the number of returned photons. An optimally efficient LLR signal path yields an improvement in the returned photons so that ranges to the lunar retroreflectors are measured precisely and accurately. Schneider et al. (1996) emphasised that the high-precision measurement of the Earth-Moon distance is highly in demand since determination of the exact Earth-Moon distance is a complex undertaking.

The complexity and difficulties of the Earth-Moon distance measurements can originate from the laser path as it is exposed to atmospheric turbulence fluctuations that degrade the beam quality, influence beam wander and thus reduce the number of returned photons, hence the accuracy of the ephemeris is compromised. The obtained results prove that the complexity of the measurements arise from the system uncertainties, parameter estimations and model implementations as discussed in the following Section (5.2, 5.3 and 5.4).

5.2 Uncertainty sources

Measurements from an LLR system are converted to a distance from the Centre-of-Mass (COM) of the Earth to that of the Moon. The understanding of the body rotations and the positional displacements (pole tide, Earth tide, ocean loading and atmospheric loading, Table (5.1)) of the surface sites with respect to the COMs of the respective bodies is vital and can lead to sub-centimetre distance measurements (Murphy et al. 2001).

Table 5.1: The maximum values for the HartRAO-LLR station positional displacements (pole tide, Earth tide, ocean loading and atmospheric loading). Source: McCarthy and Petit 2003

Parameter	Maximum values
Pole tide	several centimetres
Earth tide	tens of centimetres
Ocean loading	few centimetre
Atmospheric loading	0.5 to 3 mm


Pole tide entails a varying elastic response of the Earth’s crust and the Earth tide is the displacement of the solid Earth’s surface caused by the gravity of the Moon and Sun. Ocean loading is a secondary tidal effect, due to the elastic response of the Earth’s crust to ocean tides, producing deformation of the sea floor and a surface displacement of adjacent land. Atmospheric loading is fairly well known as temporal variations in the geographic distribution of atmospheric mass loading of the Earth and deform its surface (McCarthy and Petit 2003). The implementation of these adjustments, of the displacements to the LLR station position, help to precisely determine the range (Combrinck 2010). This is found through iterative solutions of two light-time equations for the uplink and

Table 5.2: The fixed and varying parameter values used in the radar link equation to estimate the HartRAO number of returned photons for lunar laser ranging.

Fixed parameters	Value	Varying parameters
Quantum efficiency	0.5	Cross-section, m ²
Laser energy per pulse, mJ	100	Atmospheric transmission
Wavelength, nm	532	Cirrus cloud cover
Planck's constant	6.626×10^{-34}	Slant range, Km
Speed of light in vacuum	3.00×10^8	Lunar reflectivity
Transmit optics efficiency	0.7	
Receive optics efficiency	0.7	
Primary mirror area, m ²	0.71471	
Transmitter Gain	7.87×10^{10}	

downlink paths.

The transmitted and additional laser signal data (Table (5.2) and Figure (5.1)) during laser ranging experience a delay in the lower part of the atmosphere. This makes measurement of these parameters along the total path difficult and therefore atmospheric models are used to incorporate variables such as SLR/LLR site pressure and temperature and are supported by measured data at the laser site (Botai et al. 2015). Since both Earth and Moon are not perfectly rigid bodies, this can result in a problem with orientation and hence constant offsets as these bodies have fluid interiors which allow flexure in the presence of tidal forces (Murphy et al. 2001).

 Atmospheric Transmission Adjustment is Activated

System: Satellite:

Transmitted Data

Beam Diameter [m]: <input type="text" value="1"/>	Laser Pulse Energy [J]: <input type="text" value="0.1"/>	Pulse Repetition Rate [Hz]: <input type="text" value="20"/>
Transmit Optics Efficiency [#]: <input type="text" value="0.4"/>	Beam Divergence [rad]: <input type="text" value="9.76e-006"/>	Laser Power [W]: <input type="text" value="2"/>
Far Field Divergence [rad]: <input type="text" value="1.76114493827768e-006"/>	Transmitter Gain [#]: <input type="text" value="78688033771.1267"/>	Pulse Duration (FWHM) [ps]: <input type="text" value="80"/>
Wavelength [m]: <input type="text" value="5.32e-007"/>	Beam Intensity [W/m^2]: <input type="text" value="5.09295817894065"/>	Number of Photons [# /pulse]: <input type="text" value="2.67811099340399e+017"/>
<input type="button" value="Laser Path"/>	<input type="button" value="Telescope"/>	<input type="button" value="Output Beam"/>

Additional Data

Receive Aperture [m^2]: <input type="text" value="0.714712328691678"/>	Receive Optics Efficiency [#]: <input type="text" value="0.9"/>	Detector Quantum Efficiency: <input type="text" value="0.5"/>
Atmospheric Transmission: <input type="text" value="0.41"/>	Cloud Cover: <input type="text" value="1"/>	Incident Angle [Deg]: <input type="text" value="0"/>

Figure 5.1: The GUI of the transmitted and additional data in the developed HartRAO mathematical tool that estimates the number of returned photons.

A study, by Dirkx et al. (2015), on the influence of atmospheric turbulence on planetary transceiver laser ranging revealed that the turbulence-induced range error is a significant contributor to the total estimation error budget from a certain turbulence variation amplitude onward. Turbulence-induced errors will, for $C_{n(0)}^2 \leq 10^{-13} \text{ m}^{-2/3}$ and $u < 25 \text{ m/s}$, be of minor concern when compared to inherent sources. However, an increase in turbulence-induced errors is observed with an increase in the ground-level turbulence to $10^{-12} \text{ m}^{-2/3}$. These ranges cannot hold for all system designs and missions. It indicates the existence of turbulence strength values below which the estimation procedure is not

significantly influenced by turbulence.

The LLR analysis model used at Institut fuer Erdmessung (IfE), which is based on Einstein’s theory of gravity, reduced the uncertainty in the variation in G by a factor of 2 when compared to their previous results (Müller et al. 2008). This was mainly due to their use of longer data series and also to a lesser extent, a result of refined modelling. The influence of the mirror cross-sections and slant range on the returned number of photons, described in, depends on the position vector of both the Earth and the Moon and the fact that both bodies are not “fixed”. The basic observation equation for the station–reflector distance d (Figure (5.2)) is described in (Hofmann et al. 2010) as,

$$d = c \frac{\tau}{2} = |\mathbf{r}_{EM} - \mathbf{r}_{obs} + \mathbf{r}_{ref}| + c\Delta\tau, \quad (5.1)$$

where c is the speed of light in vacuum, τ is the pulse travel time, \mathbf{r}_{EM} is the vector between geocentre and selenocentre, \mathbf{r}_{obs} is the position vector of the observatories, \mathbf{r}_{ef} is the position vector of the reflectors and $\Delta\tau$ is a delay due to atmospheric and relativistic effects.

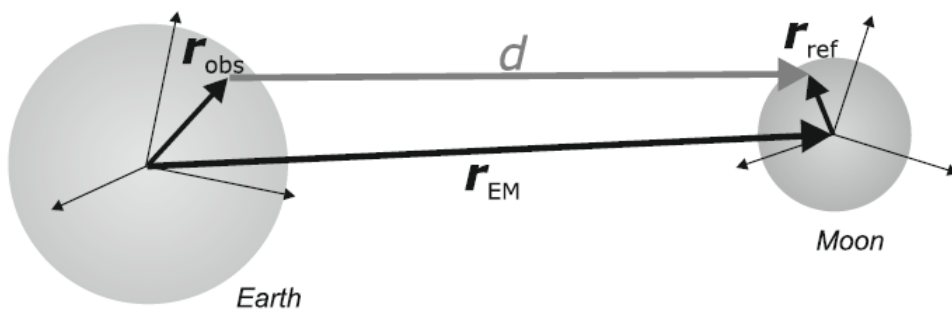


Figure 5.2: Lunar Laser Ranging measurement setup with basis vectors. Source: Hofmann et al. 2010

The vectors in Equation (5.1) can be transformed from their original terrestrial

reference frame (TRF) or selenocentric reference frame (SRF) by applying the expressions described by Hofmann et al. (2010) as,

$$\mathbf{r}_{obs} = \mathbf{R}^e(t)\mathbf{r}_{obs}^{TRF}, \quad (5.2)$$

$$\mathbf{r}_{ref} = \mathbf{R}^m(t)\mathbf{r}_{ref}^{SRF}, \quad (5.3)$$

where $\mathbf{R}^e(t)$ is the rotation matrix between TRF and inertial frame which is calculated from the Earth orientation parameters described by McCarthy and Petit (2004), and \mathbf{R}^m is the rotation between SRF and inertial frame. Hofmann et al. (2010) found that the inclusion of the lunar core refines the modelling of the lunar rotation and leads to an improved LLR data analysis model. A reduction of 4 mm or 1.1 % is achieved in the root mean square of the post-fit residuals of the whole data series.

The other sources of uncertainty in LLR measurements originate from the integrated measuring system with optical, mechanical and electronic components. The errors from the aforementioned components, together with the environmental and operational factors, determine the measurement uncertainty of the LLR. Available literature systematically addresses classification of uncertainty sources and their effects on laser ranging (Battat et al. 2009). The LLR link budget parameters (laser pulse energy, detection efficiency, effective cross-section area of retroreflectors, range and environmental transmission) and their associated uncertainties were used to estimate the total number of returned photons.

5.2.1 The HartRAO-LLR laser

Degnan (1993) discussed the temporal profile for SLR which can be approximated by a Gaussian shape that cannot be influenced significantly by atmospheric turbulence except for pulse lengths on the order of picoseconds or shorter. Anything below these pulse lengths can be affected by atmospheric dispersion, due to refraction, which causes the laser pulse to spread along its propagation path (Dirkx et al. 2015). The Gaussian temporal shape is assumed to remain fixed throughout the pulse path in Interplanetary Laser Ranging (ILR), as opposed to SLR/LLR where it is perturbed by retroreflector reflections (Otsubo and Appleby 2003). The pulse transmission time (i.e. pulse centre at $z = 0$) at $t = 0$, a time τ is introduced by defining (Dirkx et al. 2015):

$$\tau = t - \frac{w}{c_{eff}}, \quad (5.4)$$

where w is the radius of the beam waist, t is the time, c_{eff} is the mean propagation speed of the laser pulse over its trajectory. The τ and z parameters are both zero at the pulse centre.

A discussion on the characteristics of the HartRAO laser in Chapter 3 were described as,

- Type: Nd:YAG/YVO
- Pulse duration: ~ 80 ps
- Wavelength: 1064 nm, 532 nm
- Repetition rate: 20 Hz
- Pulse energy: 100 mJ at 532; 200 mJ at 1064

On laser start-up, the laser is allowed to warm-up for approximately 30 min before the laser is ready for use. This practice is done to avoid the pulse width

experiencing some variations that are caused by temperature inhomogeneity in the laser cavity. The experienced variations, δt , of the laser width can introduce a shift equal to $\delta t/2$ between the pulse centre and the photo-detector electrical response as described by Samain et al. (1998).

5.2.2 Detectors

It is assumed that the HartRAO-LLR single photon detector (not yet implemented) will have a high quantum efficiency of 50 %. It will consists of a very low jitter of $1\sigma < 20$ ps avalanche photodiode detector (APD) and a very high mean time between failures (MTBF) of more than 10 years. The probability of detecting signals (photons) follows Poisson statistics given by Degnan (1993),

$$P(m, n) = 1 - e^{-n} \sum_{m=0}^{m=n_t-1} \frac{n^m}{m!}, \quad (5.5)$$

where $n = n_s + n_n$ is the sum of the mean number of signal and noise of photons within the detector impulse response and n_t is the number of photon detection threshold value.

The detection time of the photons vary with the mean signal photons level for a certain photon detection threshold. A Probability Distribution Function (PDF) for the detection time of the first photon in a temporally Gaussian was derived by Murphy (2001) as,

$$p_{first}^{(1)}(\tau) = \frac{N}{\sqrt{2\pi}\sigma_t} e^{-\frac{\tau^2}{2\sigma_t^2}} \left(\frac{1}{2} - \frac{1}{2} \operatorname{erf} \left(\frac{\tau}{\sqrt{2}\sigma} \right) \right)^{N-1}, \quad (5.6)$$

where $\operatorname{erf}(x)$ is the standard error function (Abramowitz and Stegun 1964; Dirkx et al. 2015), N is the number of detectable photons focused to the detector and may trigger the system, σ_t is the pulse-length standard deviation related to the

pulse full-width half maximum (FWHM) and τ is as defined in Equation (5.4).

5.2.3 Lunar librations

It was discussed by Williams and Boggs (2009) that many of the lunar geophysical parameters affect the 3-axis lunar rotation and orientation, the physical librations, and give LLR sensitivity to those effects. This sensitivity can be illustrated by the equations of motion for the vector rotation of the lunar mantle and fluid core. These three-dimensional rotations are coupled by two interactions at the Core/Mantle Boundary (CMB). The corresponding differential equations are written as,

$$\frac{dI_m\omega_m}{dt} + \omega_m \times I_m\omega_m = T_g + T_{cmb}, \quad (5.7)$$

$$\frac{dI_f\omega_f}{dt} + \omega_m \times I_f\omega_f = -T_{cmb}, \quad (5.8)$$

$$T_{cmb} = K_v(\omega_f - \omega_m) + (C_f - A_f)(\hat{z}g\omega_f) \quad (5.9)$$

where;

- I_m is the mantle moment of inertia matrix including tidal deformation. The mean moment matrix is diagonal with principal moments A_m , B_m and C_m , but the tidal variation matrix is 3×3 .
- ω_m is the spin rate vector for the mantle. The spin rate components are functions of the Euler angels and their rates.
- T_g is the gravitational torque vector from the lunar gravity field, degree 2-4, interacting with the Earth, Sun, Venus, Mars, and Jupiter. The degree-2

field is subject to variations due to tides and spin. Also included is the figure-figure torque from degree-2 Earth \times degree-2 Moon.

- T_{cmb} is the torque vector from two interactions at the fluid-core/solid-mantle boundary. Since the core and mantle have different spin rate vectors, the fluid moves with respect to the solid mantle and forces arise. The CMB dissipation and oblateness forces are the two interactions. Integrating the local torques over the CMB surface gives the total torque. The unit z vector is the mantle principal axis corresponding to moment C_m .

Equation (5.8) is written for a uniform fluid core, assumed to be rotating like a solid, but using the frame of the mantle. The mantle frame is used since it controls the CMB shape and the non-spherical moments of inertia of the fluid.

Several studies have proved that lunar librations influence the retroreflectors on the Moon surface not to, in general, face directly at the Earth (Samain et al. 1998; Scheffer 2005; Courde et al. 2017). For return photon detections which operate in a single photon detection mode, an orientation difference between the normal axis of a retroreflector array and the axis defined by the direction (retroreflector array, telescope) will introduce a dispersion in the measurements (Courde et al. 2017). This orientation difference depends on the lunar libration, and the initial orientation of the array when compared to the mean orientation of the Earth centre as seen from the Moon (Figure (5.3)). Hence, the precision of the returned number of photons depends strongly on the libration.

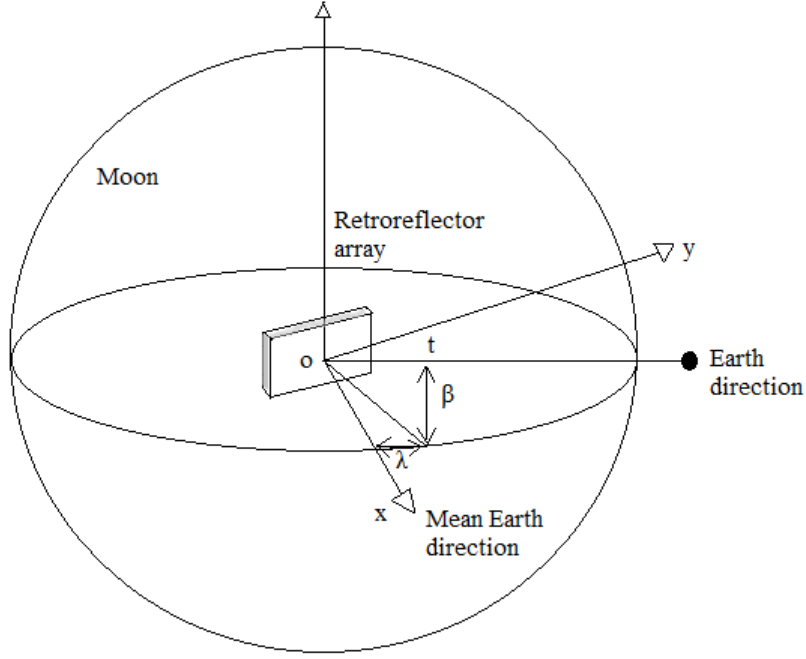


Figure 5.3: Retroreflector orientation difference as it depends on the lunar libration and on the initial orientation of the array as compared to the mean orientation of the Earth centre as seen from the Moon.

The orientation difference is estimated at $\pm 1^\circ$ for the Apollo retroreflector arrays and at $\pm 5^\circ$ for the Lunokhod arrays (Samain et al. 1998), where the reflection point will be known with a specific precision which is expressed in time as $\sigma_{Retroreflectors}$. For simplicity, it is considered that the retroreflector array is located at the centre of the Moon. The vector \mathbf{Ot} defines the instantaneous direction of the Earth telescope as seen from the Moon's centre. At a given date, the orientation \mathbf{Ot} , as compared to the mean direction \mathbf{Ox} , is given by the libration longitude λ and the libration latitude β . The dispersion is then considered proportional to the angle $(\mathbf{Otd}, \mathbf{Ox})$ only if the panels were symmetrical. This can be applied to the Apollo 11 and 14 retroreflector arrays, which are square. Then, (Samain et al. 1998),

$$\sigma_{Retroreflector} = \frac{2D \times \sin(a \times \cos(\cos\beta.\cos\lambda))}{c} \times \frac{1}{2\sqrt{3}}, \quad (5.10)$$

where the term $\frac{1}{2\sqrt{3}}$ is introduced to obtain rms precision and D is the equivalent size of the array approximated as,

$$D \approx \sqrt{L^2 \cos^2 \theta + l^2 \sin^2 \theta}, \quad (5.11)$$

where L is the largest size of the array (in the direction of p), l is the smallest size, $\theta = \frac{\sin\beta}{\cos\beta\sin\lambda} + \alpha$, is the sum of the angles between the projection of t on the planes (\mathbf{yOz}) and (Oy) , and the angle a between p and the plane (\mathbf{xOy}) . The array sizes are defined in (Samain et al. 1998) as, Apollo 15 array = 1040×630 mm², Apollo 11 and 14 arrays = 450×450 mm² and Lunokhod 1 and 2 arrays = 440×190 mm².

The correlation between the theoretical and the observed precisions on Apollo 15 array versus the lunar libration is fully discussed by Samain et al. (1998). It was found that the average precision is proportional to the retroreflector array size, improvement of the global precision will be obtained by increasing the number of echoes on the small arrays. The libration effect on the Apollo 15 orientation lies in an interval from 0 to 350 ps, which is 0 to 150 ps for other retroreflectors. During an LLR observation the retroreflector orientation is considered as fixed. The single-shot precision of the station obtained by calibration ranging gives the combined precision of all the instrumental error sources.

5.2.4 Retroreflectors

Available literature has proved that the dominant source of random uncertainty in modern laser ranging has little to do with the system components, but rather

comes from the varying orientation of the lunar retroreflector arrays (Murphy et al. 2012). Variations in the lunar orientations cause misalignment between the arrays and the mean Earth position. The varying lunar librations causes tilts of the retroreflector arrays. This phenomena contributes the largest uncertainty to any individual photon, typically between $\sigma \sim 15 - 50$ mm (Murphy et al. 2012). Hence, hundreds and thousands of photons are required to reduce the centroid uncertainty to about 1 mm.

The structure of the mechanical mounting of the lunar retroreflectors (LRR) consists of the upper and lower rings (Figure (5.4)), called Teflon rings, to afford all possible thermal insulations. The enclosure (RoE) fit over tabs to enclose the lunar retroreflector. The structure also serves to provide passive thermal control by means of surface properties. Since the reflecting faces of the structure lie on the bottom of the upper mounting then the retroreflector can be treated as structural recession (ZhouS and Li 2011). The structural recession parameters include the length d_2 from the top of the tab to the front face of LRR, and the length d_1 from the front face of the LRR to the RoE (ZhouS and Li 2011).

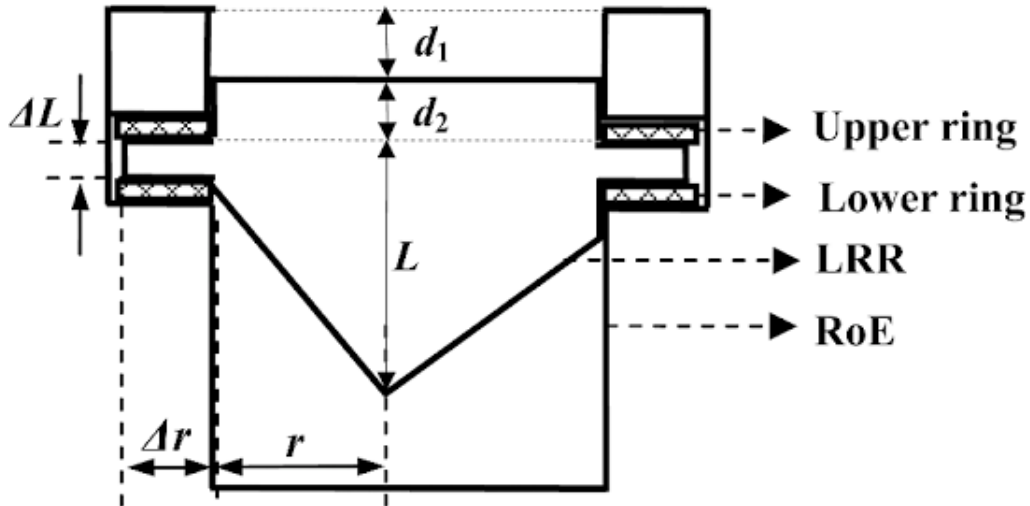


Figure 5.4: The schematic diagram of lunar retroreflector mounting.

The misalignments between the arrays and the mean Earth position are typically around 7° but can be as large as 10° (Murphy et al. 2008). This results in a RMS spread of 15 - 36 mm, corresponding to about 100 - 240 ps of round trip TOF, for the ranges between Earth and the individual array elements (Murphy et al. 2008). This RMS spread contributes to and dominates over the uncertainties associated with the laser pulse width, detector and timing systems. Having a LLR normal point with 16 photons means a limitation of between 4 to 9 mm range precision by the array orientation alone.

The initial return signal strengths for the three Apollo retroreflector arrays came from the number of corner cubes normalized to the 100 corner cubes of 3.8 cm diameter in the Apollo 11 and 14 arrays (Turyshev et al. 2012). McDonald observatory estimated the initial signal strength of the Lunokhod array, when not exposed to sunlight, as comparable to the two small Apollo arrays. Murphy et al. (2008 and 2010) estimated the current signal strength as 1/3 of the Apollo 11 and 14 strengths. The Lunokhod arrays are regarded as a difficult target that can be ranged to only by large telescopes on nights with good quality seeing; its fading appears to be real.

The degradation of the retroreflector's optical performance over time is also considered as a threat to the range precision. It has been suspected that dust contamination on the retroreflectors is the most likely candidate for this degradation (Stubbs et al., 2006). Dust settling on the lunar retroreflector mirrors is considered to be one of the detrimental factors which reduce retroreflectors efficiency over time by a factor of 10, plus an additional decrease at full Moon (Murphy et al. 2010). In Figure (5.5), a comparison of the performance of the retroreflector arrays revealed that all the arrays are still operating normally. The Apollo 15 array contributes to the major parts of the

total observations compared with the other reflectors (Apollo 11, Apollo 14, Lunokhod 17 and Lunokhod 21).

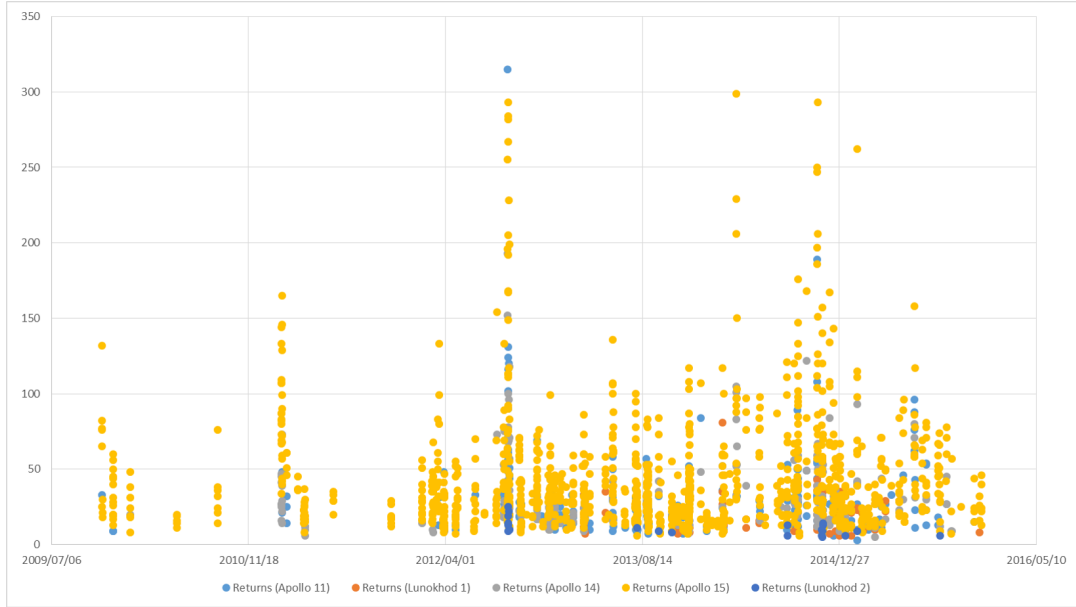


Figure 5.5: A comparison of the number of LLR returns when ranging from the OCA station to all five retroreflector arrays from 2009 to 2015.

5.2.5 Range

The HartRAO-LLR station has the potential to provide sub-cm precision data and due to improved LLR network geometry, will reduce current latitude dependent biases. The combination of a 1 m aperture, 90 % throughput and clear skies will translate to a high photon return rate. However, the very long average Earth-Moon distance reduces the number of returned photons (Figure (5.6)) quadratically as evidenced by Equation (1.2). The measured distance also depends on gravitational physics which is sensitive to the COM separations of the Earth and Moon.

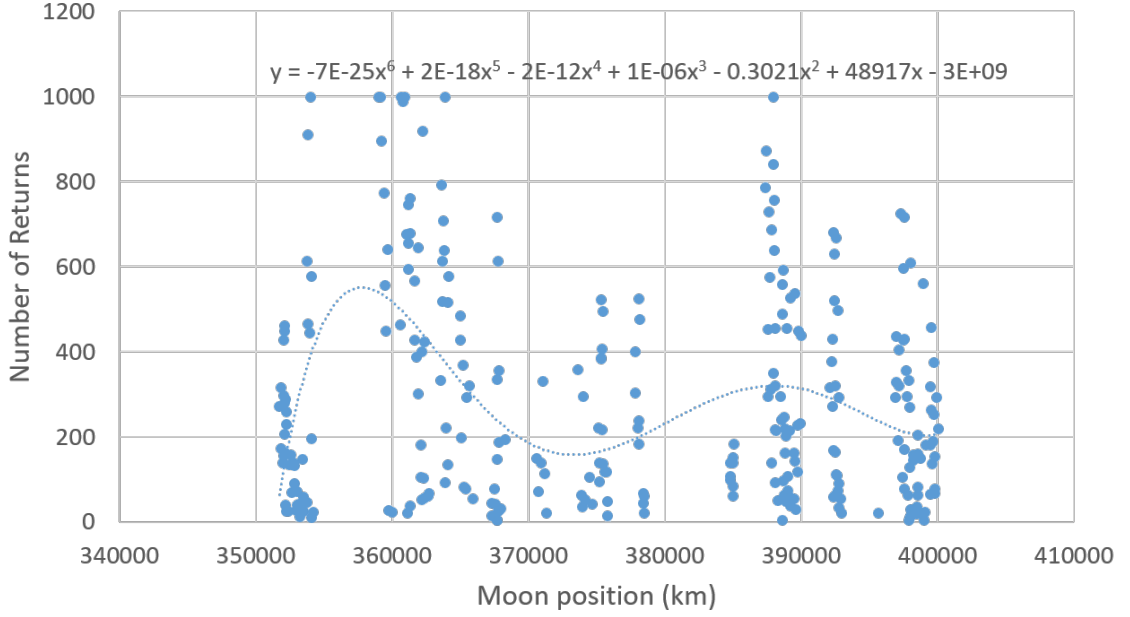


Figure 5.6: The relationship between the slant range variations and the number of LLR returns for Apache Point Observatory. The data used was for year 2015. The function ($y = -7 \times 10^{-25}x^6 + 2 \times 10^{-18}x^5 - 2 \times 10^{-12}x^4 + 1 \times 10^{-6}x^3 - 0.3021x^2 + 48917x - 3 \times 10^9$, where y = number of returns and x = Earth-Moon distance) was generated to estimate the relationship between the slant range and the number of returns.

The mean Earth-Moon distance is 384 000 km, but this distance depends on variations owing to the Moon's orbit which can be strongly distorted by variations in the Moon's centre-to-centre distance from Earth (which varies with mean values of 363396 km to 405504 km) and perturbations due to solar attraction. Dickey et al. (1994), found that the instantaneous eccentricity (deviation of orbit from circularity) varies by a factor of 2 (0.03 to 0.07). These perturbations contribute immensely to the range signatures that give sensitivity to a wide variety of parameters such as the laser beam incident angle on retroreflectors. The strong influence on the lunar orbit perturbations are due to the Sun. This allows the observer to determine the mass ratio of the sun to that of the Earth and Moon from the range data. The relative accuracy in the mass

ratio is found to be,

$$Mass_{sun}/Mass_{(Earth+Moon)} = 329006.311, \quad (5.12)$$

which is in agreement with the mass ratio of 328900.560 that was determined by Dickey et al. (1994).

The minimum, median and maximum LLR returned photons per minute for each retroreflector array and station site is given in Table (5.3). The minimum signal return rate for each station is fairly consistent for different arrays indicating a threshold for finding the signal. The maximum rate is a better indicator of the strength of the return from each retroreflector array. The MLRS2, APOLLO and Grasse (MoE) are fairly recent stations that were able to successfully range to the Lunokhod 1 (Luna 17). The Apache Point Observatory (APOLLO) maximum values are approximately the same for all the retroreflector arrays. There is currently no data from Matera station for Luna 17 and Luna 21 arrays.

Ground station geometry has an influence on the range error. Measurement statistics per station vary in range error between the pass with the largest and that with the smallest mean range error, $\bar{\Delta}s_{pass}$ of a single station (Dirkx et al. 2015). The differences between the passes indicate that the pass-to-pass range error is fluctuating. The turbulence fluctuations cause variations in the range measurement error that cannot be fully absorbed by the bias estimation. Each station has its own specific range of zenith angles at which observations are made. It can be seen from Table (5.4) that the zenith angle range can be very limited for some stations e.g. Station 1 and 4. Therefore, the small ranges of zenith angles that can be obtained for a number of stations are rather common for laser tracking.

Table 5.3: Minimum/median/maximum LLR returned photons per minute for each retroreflector array for period 1969 - 2014. The estimated return rate for the HartRAO-LLR system is also included. Source (<http://polac.obspm.fr/llrdatae.html>)

Station	Receiver Diameter, m	Time Span, years	Apollo 11 (min/med/max)	Apollo 14 (min/med/max)	Apollo 15 (min/med/max)	Luna 17 (min/med/max)	Luna 21 (min/med/max)
#CCRs \times size			100×3.8 cm	100×3.8 cm	300×3.8 cm	14×6.5 cm	14×6.5 cm
Signal strength			$\sim 0.1/1$	$\sim 0.1/1$	$\sim 0.3/3$	$\sim 0.03/\sim 1$	$\sim 0.03/\sim 1$
McDonald 2.7	2.7	1969 - 1985	1/8/27	2/8/38	2/10/121	no data	2/7.5/19
MLRS1	0.76	1983 - 1988	7/14/34	2/14/26	1/5/57	no data	9/9/9
Grasse (Rubis)	1.54	1984 - 1986	2/6/15	2/6/12	2/7/41	no data	2/7/22
Haleakala	1.8	1984 - 1990	2/9/75	5/9.5/30	2/19/249	no data	2/7/27
Grasse (YAG)	1.54	1987 - 2005	5/24/99	3/22/102	3/29/422	no data	3/16/70
MLRS2	0.76	1988 - 2013	3/7/31	1/8/40	1/10/105	5/5/5	2/5/12
Matera	1.50	2003 - 2014	3/10/27	3/9/19	4/16/111	no data	no data
APOLLO	3.5	2006 - 2012	3/179/999	6/163.5/999	5/130/999	5/130/999	5/60/563
Grasse (MeO)	1.54	2009 - 2014	3/25/315	5/25/193	6/32/299	6/15.5/81	5/10/25
HartRAO	1.0	Estimates	5/23/300	4/22/300	15/78/1000	1/12/100	1/10/80

Table 5.4: Statistics of range error per station per pass for nominal and strong turbulence estimation cases. Source: Dirkx et al. 2015

Station	ζ_{\min} [°]	ζ_{\max} [°]	Nominal ($\Delta \bar{s}_{pass}$ [mm])	Strong ($\Delta \bar{s}_{pass}$ [mm])
1	8.3	70.3	4.0	5.2
2	71.3	74.7	1.6	2.1
3	63.4	74.3	2.5	2.4
4	0.2	62.4	4.6	6.1
5	60.0	74.0	1.2	0.9
6	65.2	72.5	1.8	2.2
7	55.0	74.6	2.1	1.8
8	4.3	74.4	2.8	3.5

The determination of the relative geocentric positions of the Moon and Sun is consequential from solar perturbations. The geocentric position of the Moon and the heliocentric positions of the planets can be made internally consistent in their relative orientation (Williams and Standish 1989; Dickey et al. 1994). The LLR experiment is sensitive to the mutual orientation of the Earth’s equatorial planes, the lunar orbit and its ecliptic trajectory. It locates and determines the relationship of the angle between the ecliptic and equator planes for accurate measurements.

5.2.6 Lunar reflectivity

It was discussed by Kopal (1961) that the amount of lunar reflectivity from a given element of the lunar surface depends on its albedo as well as on the angles of incidence (i), reflection (ε) and phase (α). This was evidenced by the data

discussed earlier in Figure (4.10) to Figure (4.12). As measurements by Kopal (1961) were made on the lunar disk at some definite phase-angle, it was reasonable to suppose that the law of reflection be represented by the product of two different functions, expressed as,

$$\xi = \Gamma f(i, \varepsilon, \alpha) \Phi(\alpha).$$

Kopal (1961) also discussed one of two laws of diffuse reflection, which is that of Lambert. This is gathered simply by considering the emission of a solid incandescent body, appearing equally bright in all directions. Hence, the quantity of emitted light varies as the apparent size of emitting area: namely, as $\cos(\varepsilon)$. According to this law, the quantity of light reflected at the angle ε from the area ds illuminated at the incidence angle i is

$$\xi ds = \Gamma \cos(i) \cos(\varepsilon) ds. \quad (5.13)$$

Therefore, the apparent brightness of a given surface obeying this law is

$$j \propto \cos(i). \quad (5.14)$$

In addition, a sphere illuminated by a distant source of light, when viewed at a certain small phase-angle, will appear bright at its centre, but completely dark at the edges. A similar observation can be observed on the planet Mars, the surface of which is really covered by dust; but on the Moon the conditions are apparently quite different (Kopal 1961). Lambert's law is completely devoid of any theoretical foundation, but satisfies fairly well the behaviour of smooth diffusing surfaces. Kopal (1961) found that the reflection from a barium screen with a flat surface compares with Lambert's law at the same angles of incidence

and reflection.

The observations presented in Figure (4.10), which were observed using the Apache Point Observatory data from 2006 to 2015, were grouped together around the First Quarter (evening observations) and Third Quarter (morning observations). This was found to be in agreement with the observations by Chapront and Francou (2003), where the distribution of observations within a lunar month were carried out for about 25 days per lunation with no observation around New Moon. At Full Moon, conducting the experiment is harder due to the lack of contrast as discussed by Silverberg (1974). The obtained data, from the Observatoire de la Côte d’Azur, displayed that there were no observations which were carried out at Full Moon because of heavy noise because of the reflected Sunlight from the surface of the Moon (Figure (5.7)). This heavy noise is usually not observed during Moon eclipses as it was observed in 2000 at CERGA station (Chapront and Francou 2003).

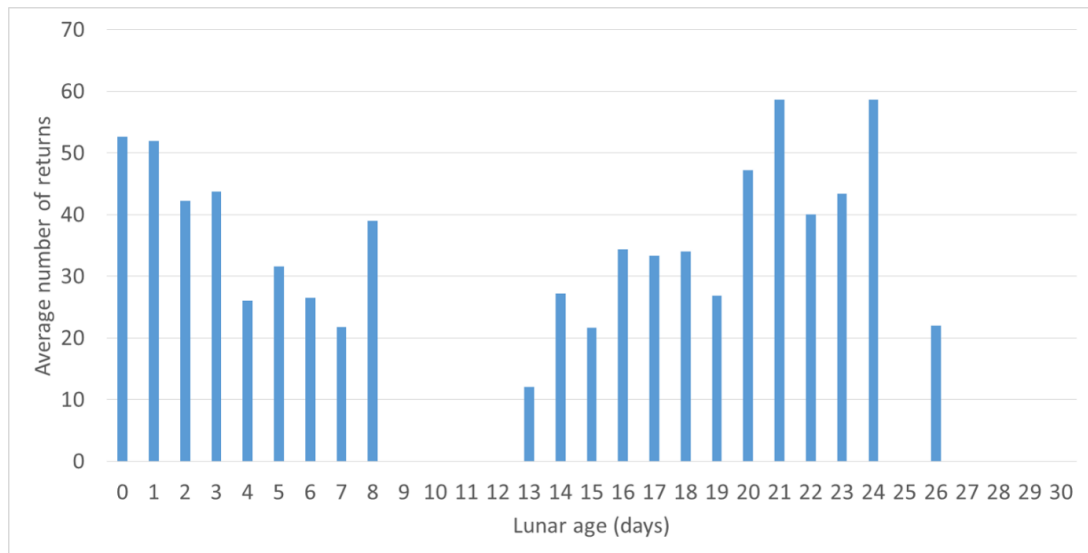


Figure 5.7: The stacked averaged data from 2009 to 2015 for the Observatoire de la Côte d’Azur LLR distribution of returns for both day and night observations during a lunar month.

The inclusion of this new parameter, lunar reflectivity, improved the predicted return values (Figure (4.14)). For instance, in Figure (4.14), the value of a single returned photon corresponded with three different atmospheric transmittance values for each lunar reflectance ($T_a^2 = 0.6$ at $L_r = 0.35$, $T_a^2 = 0.3$ at $L_r = 0.7$ and $T_a^2 = 0.2$ at $L_r = 1.0$). This is an indication of a direct relationship between the brightness of the Moon, contrast and LLR returns. For every conducted LLR experiment, there is a corresponding lunar reflectivity value and other varying parameters (Table (5.2)), which affects laser ranging to result in fewer LLR returns.

5.2.7 Signal-to-Noise ratio

There are many sources of noise that affect laser ranging starting from the optical path to the atmosphere. The existence of the large number of noise sources makes it impossible to identify a returning photon from the Moon without filtering, both physically and mathematically. One of the filtering techniques that provides temporal noise filter is range gating, whereas a pin-hole aperture provides a spatial filter. The latter allows only the proper colour photons to pass through to the receive system. It must be as narrow as possible in wavelength to eliminate as many noise photons as possible, transmitting as many proper photons as possible. These requirements change with lunar phase and sky conditions. Improved filtering result in a significant increase in the amount of LLR data observed.

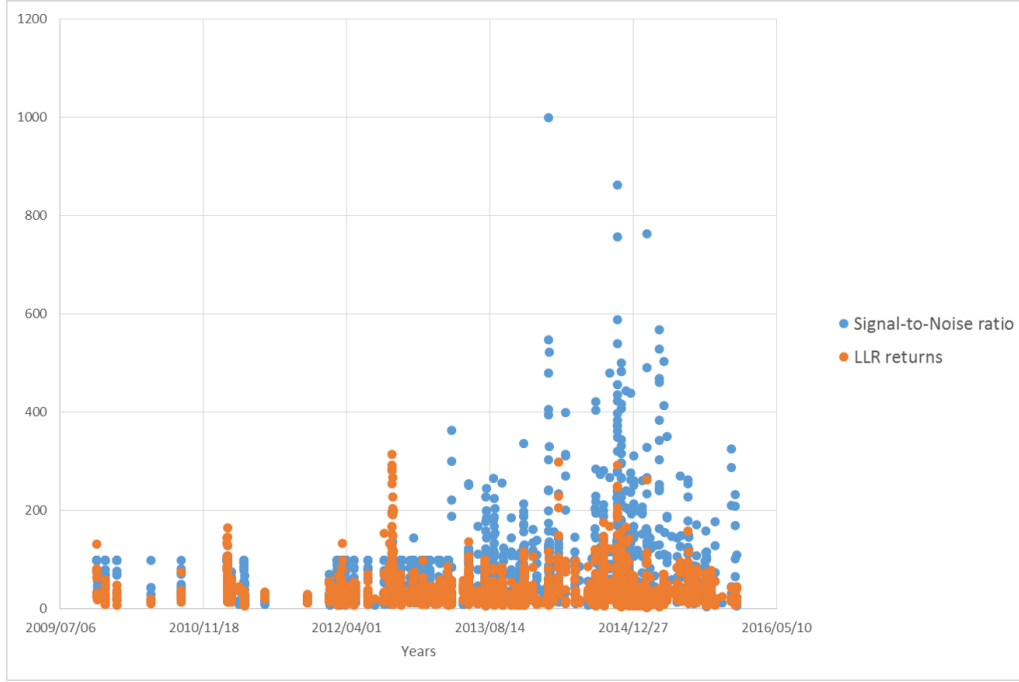


Figure 5.8: The comparison between the number of LLR returns and the estimated signal-to-noise ratio for the OCA station observations (2009 to 2015).

The reflected Sun light from the lunar surface is also a challenge when measuring the Earth-Moon distance using laser pulses. There is a generated plot comparing the estimated number of LLR returns and the estimated signal-to-noise ratio for the OCA station (Figure (5.8)). The OCA LLR system consists of sensitive light filtering to reduce background noise. In turn, the number of reflected laser signals from the lunar retroreflectors is compromised due to the introduction of filters that reduces the weak returned laser signals. A clear representation of the number of lunar return photons embedded in the background of randomly-timed arrivals is a quadratic fit residual graph (Figure (5.9)).

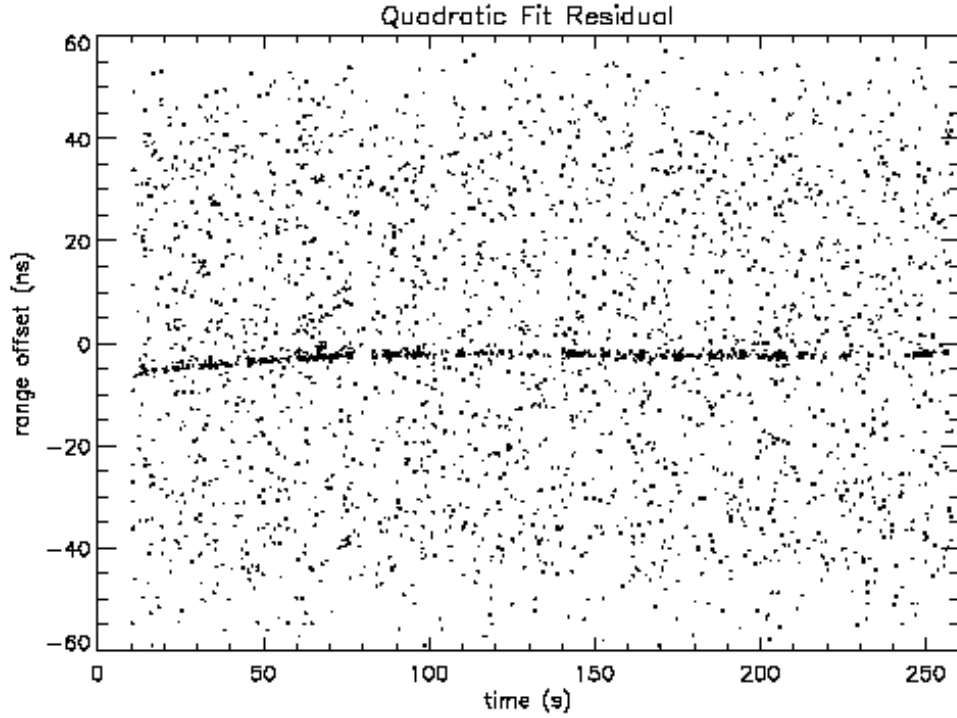


Figure 5.9: The quadratic fit residual for lunar return photons presented as range offset (ns) over time (s). The thin streak is the returned photons embedded in the background of randomly-timed arrivals. Source: PHYSICS-UCSD

The APOLLO LLR system was designed and developed to benefit from a low background photon rate. This was owing largely to the very small accepted field of view of 1.2 arcsec. Because of this, APOLLO was designed to operate in full moon or daylight conditions without significant noise contamination. Though the returned photons are significantly low (about 0.2 photons per pulse) from the full Moon per 100 ns (Murphy et al. 2002). The ability to measure the lunar distance throughout the cycle enables APOLLO to have a very positive impact on LLR analysis. It is evident from the comparison of their signal-to-noise ratio and the returned lunar photons (Figure (5.10)), that the APOLLO system is capable of detecting even a single photon per pulse.

A recent study on lunar eclipse revealed anomalous thermal performance of the Apollo reflectors (Murphy et al. 2014). It was evidenced that dust accumulation

could be responsible for noise effects: reduce signal at all phases; the full-Moon signal deficit; and the double-peaked signal during eclipse. This was concluded from the fact that during illumination, solar thermal absorption by ~50 % dust-covered lunar retroreflectors front surface may provide enough heat to support a several-degree thermal gradient with the retroreflector and this alone is sufficient enough to degrade the central irradiance of the diffraction pattern reflected back to Earth.

The comparison of the signal-to-noise ratios and the returned lunar photons for the other stations (Figure (5.11) to Figure (5.13)) were also consistent with each other. Though the data points for the signal-to-noise ratio are slightly higher than the returned lunar photons. It is evident from these graphs that in order to achieve a reasonable signal-to-noise ratio within an individual Normal Point, there must be average photon return rates of ≥ 0.2 photons per transmitted laser pulse (Murphy et al. 2002).

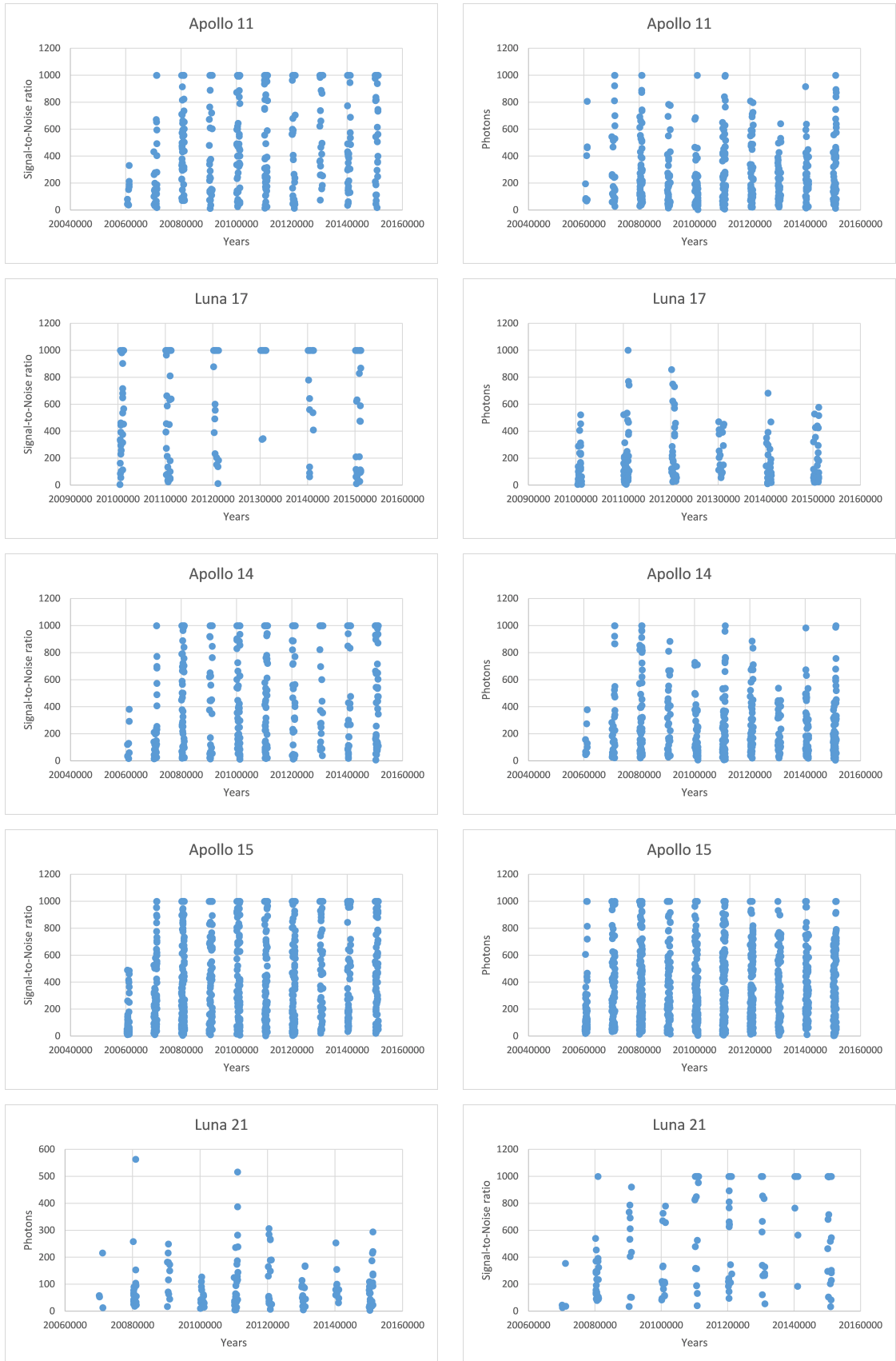


Figure 5.10: A comparison of the annual (2006 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the APOLLO system.

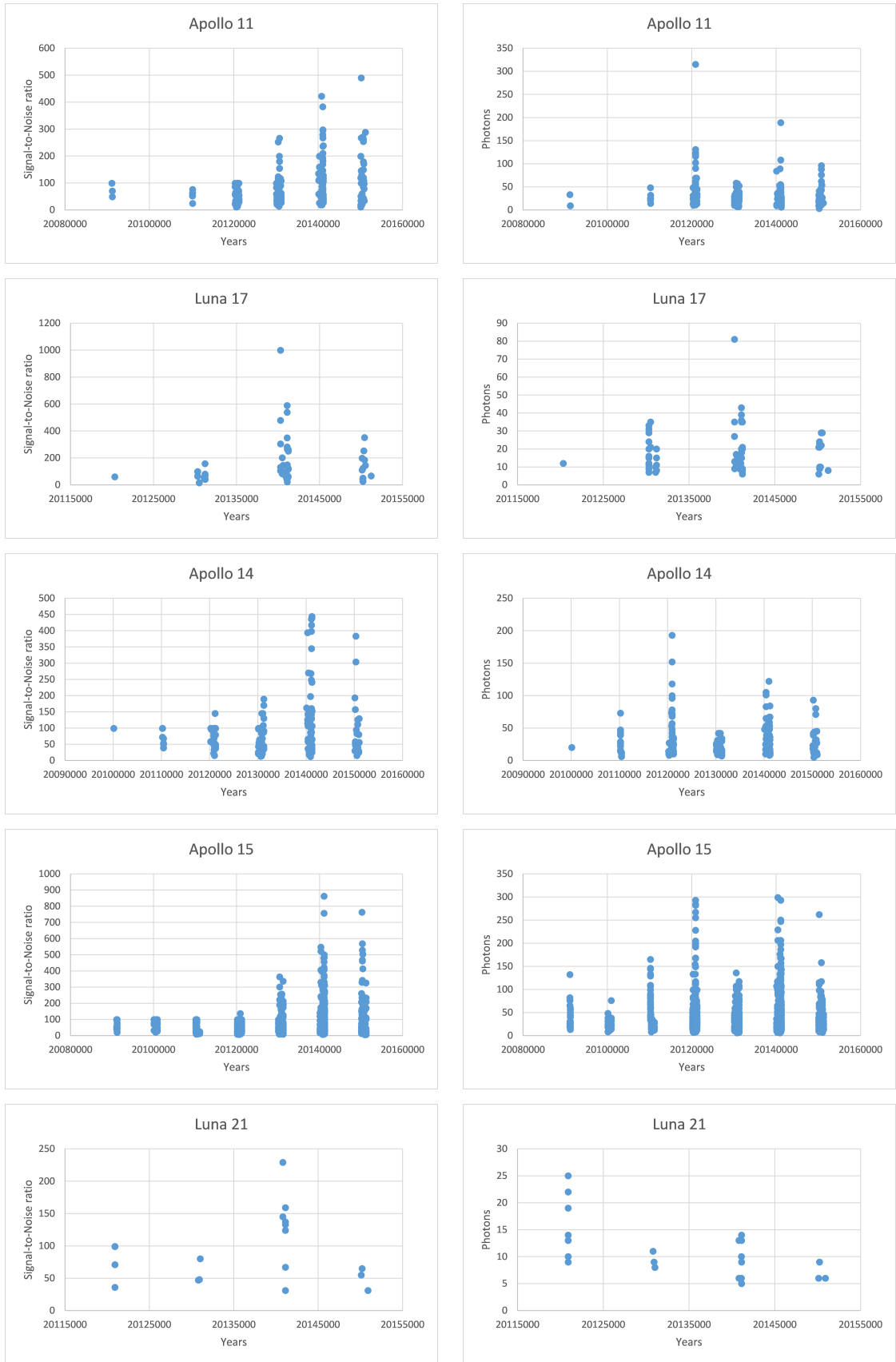


Figure 5.11: The annual (2009 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the OCA LLR system.

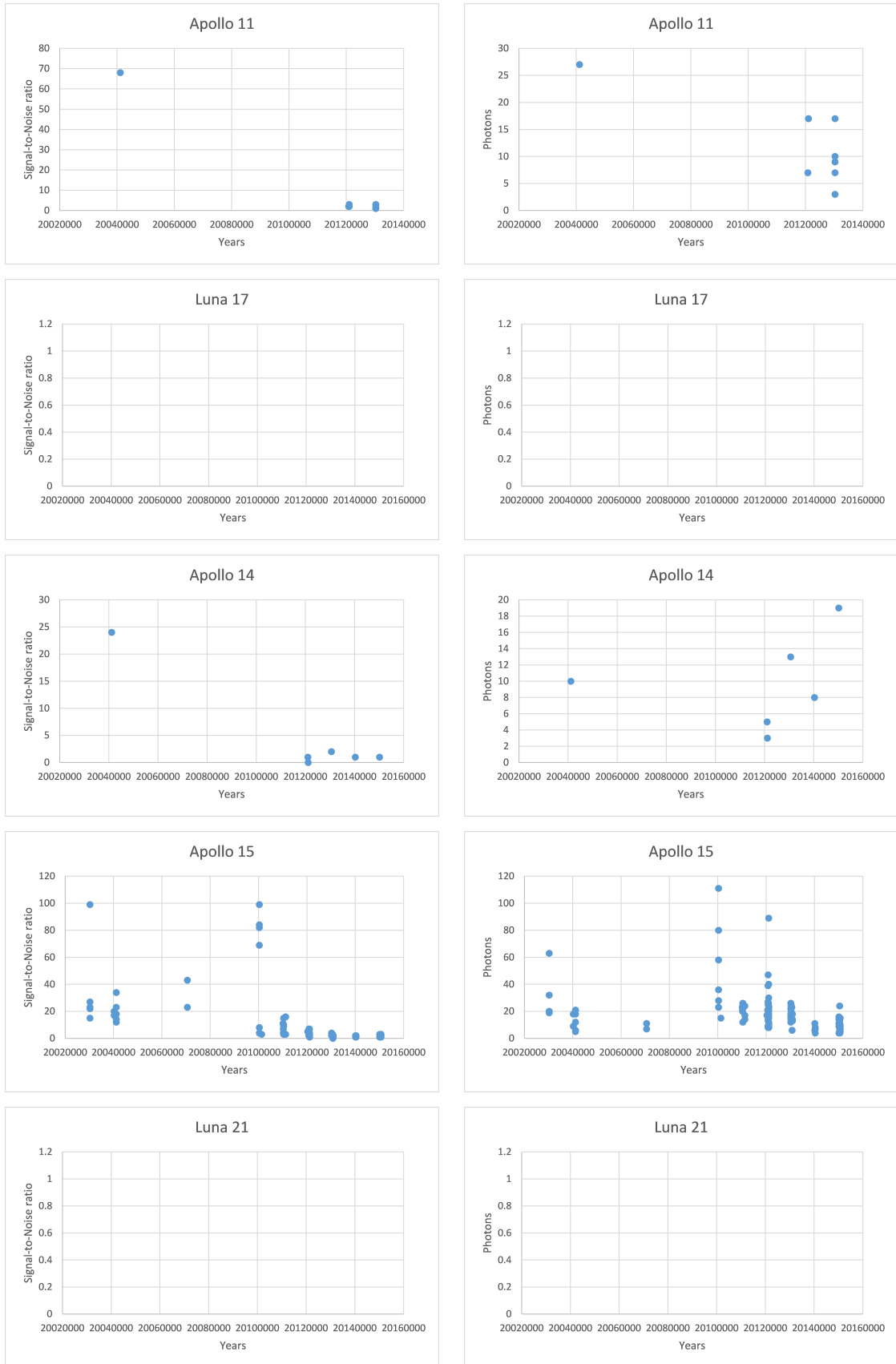


Figure 5.12: The annual (2003 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the Matera LLR system.



Figure 5.13: The annual (1988 to 2015) total signal-to-noise ratio for the returned photons, from each lunar retroreflector array, obtained with the McDonald LLR system.

The HartRAO-LLR system will be able to detect photons using its 1 m aperture and a single photon detection system. The astronomical seeing of less than or equal to 2 arcseconds will also play a key role for large number of lunar returned photons detections. An average return rate of 5 photons per minute is expected from the Apollo 11 and 14 arrays. A much improved photon return rate of about 12 photons per minute is obtained (as expected) when ranging to the Apollo 15. The HartRAO system will also be able to target the laser pulses to the Lunokhod arrays with the Luna 17 yielding more returns as compared to the Luna 21.

5.3 Model implementations

Different mathematical methods were used to estimate parameters to enable optimal signal path efficiency of the HartRAO SLR/LLR station. The use of the radar link budget equation helps to determine system and atmospheric parameters for successful laser ranging. The success of laser ranging is determined by the number of reflected laser signals from the retroreflector corner cube mirrors. The HartRAO-LLR system is still under development so it is difficult to verify the simulated results since observed data are not currently available. Instead, data from other LLR stations was used to verify the estimated parameter values.

The HartRAO-LLR's developed mathematical tool, for returned photons estimations, consists of many fixed and varying parameters. It was observed from the ranging simulator that changing the "varying" parameters can be difficult as there exists other parameters which are affected by this change. At the same time, the noise that accompanies the changes in parameters is sometimes

excluded in the simulated results. This normally reduces data by a factor of about 10 (Michelsen 2010). The known “fixed” system parameters also adds or removes some effects on the simulated photon returns because of the rounding-off errors associated with the computed parameter values.

The developed mathematical tool considered a number of varying, fixed parameters and an additional Lunar Reflectivity parameter. The Lunar Reflectivity parameter is not considered in the existing stations as they already have enough data which provides necessary information for conducting the LLR experiment. The introduction of this new parameter for HartRAO-LLR estimations together with the known and estimated parameters achieved scientifically valid results as compared with data from existing LLR stations over the world.

Chapter 6

Conclusion

6.1 Summary

In this work, a description was provided on the importance of the HartRAO-LLR station. The station is currently the only one in the Southern Hemisphere. It is expected to improve the data quality and quantity contribution to the International Laser Ranging Service (ILRS) and strengthen the geometry of the network. The aim of the research was to develop an integrated model and system that will enable optimal efficiency for the HartRAO-LLR signal path. This was done so that measurements, for accurately determining the Earth-Moon distance, are more precise and can yield improvements in the determination of fundamental physics parameters such as the Weak Equivalence Principle, Strong Equivalence Principle, des Sitter precision and time variations of Newton's gravitational constant (Williams et al. 2009).

The effect of detrimental factors, such as Earth-Moon distance, atmospheric transmission, cirrus cloud cover, transmitter gain, retroreflector optical cross-section and the “fixed” system parameters led to the development of

scientific objectives that enable optimal signal path efficiency of the HartRAO Lunar Laser Ranger. These objectives involved the development of an advanced mathematical tool to analyse the LLR system and atmospheric parameters, expansion of the existing link equation to consider the effects of lunar phases on the returned LLR observations, development of a computer-controlled monitored signal path, development of a computer model and system to enable correct beam expander settings for precise eyepiece lens position (using microprocessors and software programme utilising C++ code).

An overview of the HartRAO-LLR system was provided. The system is software centric to allow rapid changes to specific functions, system parameter monitoring, telescope pointing and steering, as well as preliminary data quality checks. A description of the two TME₀₀ laser pulse sources was provided. One laser is used for Lunar Laser Ranging and the other for Satellite Laser Ranging with 100 mJ/pulse and 0.45 mJ/pulse, respectively. The wavelength for both lasers is 532 nm and the repetition rate of 1 kHz for LLR and 20 Hz for SLR. Both laser pulses can be transmitted, not at the same time, through the same coudé. For LLR, the output beam that leaves the telescope is about 1 m and for SLR, a 200 mm diameter beam expander is used to transmit laser pulses. The returned signals are collected by the 1 m telescope and focused to the detection system for accurate Earth-Moon distance measurements.

An advanced mathematical tool, utilising C++ code, was developed and used to estimate the expected number of returned photons for the HartRAO's LLR system. The tool allowed the user to select a targeted satellite/retroreflector and adjust parameters. This tool will be used to evaluate computed and observed photon return efficiency, using as departure point the existing link budget equation, with the option to add and estimate parameters in the least squares

sense. The mathematical tool also included the ranging degrading factors such as the variations of the angle of incidence of the laser on retroreflectors, laser energy variations and efficiency of retroreflectors. The slant range is one of the biggest degrading parameters on the number of received photons due to its 4th order of magnitude. Atmospheric transmission and cirrus cloud transmittance have an influence on the number of returned photons degradation. Under unfavourable weather conditions, a decline in the number of returned photons is observed.

A new varying parameter was introduced in the radar link budget equation. This parameter addresses the effects of the brightness of the Moon surface as observed from Earth, which is considered as one of the factors that also plays its role in reducing the number of observed LLR returns. The introduction of this parameter in the radar link budget equation helps with a better understanding of the signal-to-noise ratio when estimating the expected number of LLR returns.

Another mathematical tool, utilising C++ code, was successfully developed to enable correct beam expander settings for precise eyepiece lens position (using microprocessors and software programme utilising C++ code). The developed tool controlled hardware and software embedded systems that accurately drives the movement of the beam expander's eyepiece lens.

Uncertainty sources associated with LLR were discussed. It was described that most of these uncertainties originate from the integrated measuring system with optical, mechanical and electronic components. The errors from the aforementioned components, together with the atmospheric fluctuations and operational factors, determine the measurement uncertainty of the LLR. Available literature systematically addressed classification of uncertainty sources and their effects on laser ranging. The LLR link budget parameters (laser pulse energy, detection efficiency, effective cross-section area of retroreflectors, range

and environmental transmission) and their associated uncertainties were used to estimate the total number of returned photons.

6.2 Concluding remarks

Based on the simulations on the state-of-the-art HartRAO-LLR and uncertainty sources, an integrated model to enable optimal signal path efficiency was formulated (Ndlovu et al. 2014; 2015; 2016). The integrated model serves the core part for the HartRAO-LLR system. It will allow the system to improve both the stability and the accuracy of the measurements. The results thereof will then be used to improve our understanding of the Earth-Moon system. Our obtained results take most of the uncertainty sources affecting laser ranging into account and thus will enable the establishment of an uncertainty model which behaves in an identical way as the real system.

In the existing radar link budget equation there was room for expansion as the effects of lunar phases were not integrated in the existing equation. There were two computer based software programs utilising C++ that were successfully developed. One software precisely controlled and accurately adjusted the eyepiece lens of the beam expander for correct beam collimation. The second software used the link budget equation for parameter estimation to study the effects of thermal fluctuations, retroreflector orientation and range variations on returned laser signals (Ndlovu et al. 2014; 2015; 2016). The estimated parameters i.e. received photons helped to determine the total system efficiency.

The effects of the variable link budget parameters and the expected photon returns for the HartRAO Lunar Laser Ranger were investigated. The estimated signal return rate ranged between 0 and 12 photons per minute, which is in

agreement with the data from the already operating LLR stations (Samain et al., 1998; Murphy 2013). The HartRAO’s expected signal return rate is strongly affected by the two-way atmospheric extinctions (atmospheric and cirrus cloud transmissions), variations in the laser incident angle on the retroreflectors and the fourth order of magnitude of the slant range.

The geographic position of the HartRAO station, HartRAO’s new state-of-the-art LLR system (still under development) and the expected number of returned photons allow the HartRAO-LLR station to play a pivotal role in improving the ranging accuracy to a sub-centimetre level. It was found that the most optimal scenario for the HartRAO-LLR system will lead to a photon average return rate of 5 photons per minute. The results from the HartRAO-LLR will enhance the current effort to determine high accurate Earth-Moon distances for various scientific purposes such as lunar ephemeris, determination of Universal Time and rate of change of Newton’s gravitational constant, G .

6.3 Future work

The HartRAO laser ranger is still in its development stage. Thus, there are future efforts that must be directed towards the following aspects:

- More accurate mathematical models for laser ranging are needed for successfully ranging to the artificial satellites and Moon surface with the HartRAO-LLR in order to improve the accuracy to a millimetre level.
- The effective operation of LLR depends on the sensitivity and accurate mount of the detection system. Thus, experimental work on mounting the detection system in the location that will yield optimal results is necessary.
- The development of the analysis software for the HartRAO-LLR data is

required so that scientific parameters such as lunar ephemeris, Universal Time and rate of change of Newton’s gravitational constant are determined.

- The current HartRAO-LLR Nd:YAG laser has 100 mJ output pulse energy at 532 nm. This laser energy and type can be upgraded to improve the current estimated number of returned photons for the station.

The Observatoire de la Côte d’Azur successfully improved their green Lunar Laser Ranging instrumental sensitivity (Martinot-Lagarde et al. 2016). This success was achieved through doubling the pulse energy of their Nd: YAG laser to 200 mJ at 10Hz. The current pulse energy of the HartRAO-LLR can also be doubled in the near future for data quantity improvement. One of the considerations that were also considered by Martinot-Lagarde et al. (2016), would be to replace the dye cell with CR^{4+} :YAG crystal saturable absorber and spatial beam profile improvements. The improvements in the spatial beam profile depends on polarisation, flashlamp configuration and the lens positioning. The combination of these improvements play a pivotal role to future science breakthrough which is linked to quasi-millimetric determination of the Earth-Moon dynamics (Murphy 2013; Martinot-Lagarde et al. 2016).

The OCA station also succeeded with the implementation of IR detection for LLR, which provides new opportunities for the improvement of scientific products. As expected by Courde et al. (2017), LLR in infrared increases the station efficiency by a factor of eight during new and full Moon periods and improves the temporal homogeneity of LLR observations over a synodic month. The best link budget at this wavelength results in a significant increase of the NPs over all the reflectors on the Moon. The success of this technique at OCA as their observations are statistically more homogeneous over all the targets can be experimented, in future, at HartRAO.

References

- Abramowitz** M. and **Stegun** I. A., (1964). Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards, Applied Mathematics Series 55
- ABC** News Online, (2003). Mt Stromlo observatory severely damaged in fires, Retrieved 18 June 2013
- Alda** J., (2003). Laser and Gaussian Beam Propagation and Transformation, Encyclopaedia of Optical Engineering, Marcel Dekker, New York
- Anderson** J. D., Gross M., Nordtvedt K. L. and Turyshev S. G., (1996). The solar test of the Equivalence Principle, *Astrophys. J.* Vol 459, p. 365-370
- Andrews** L. C., and Phillips R. L., (2005). Laser beam propagation through random media. 2nd edition, SPIE 9-14
- Argyle** B., (2004). Observing and Measuring Visual Double Stars, Patrick Moore's Practical Astronomy Series: Springer-Verlag, London Limited, 19 & 86
- Battat** J. B. R., Murphy T. W., Adelberger E. G., Gillespie B., Hoyle C. D., McMillan R. J., Michelsen E. L., Nordtvedt K., Orin A. E., Stubbs C. W., & Swanson H. E., (2009). The Apache Point Observatory Lunar Laser-ranging

Operation (APOLLO): Two Years of Millimeter-precision Measurements of the Earth-Moon Range. Astronomical Society of the Pacific, Vol. 121, p. 29-40

Bender P. L., Currie D. G., Poultney S. K., Alley C. O., Dicke R. H., Wilkinson D. T., Eckhardt D. H., Faller J. E., W. M. Kaula, Mulholland J. D., Plotkin H. H., Silverberg E. C. and Williams J. G., (1973). The lunar laser ranging experiment Science Vol. 182 no. 4109 p. 229-238

Born G. K., Bogenberger R., Erben K. D., Frank F., Mohr F. and Sepp G., (1975). Phase-front distortion of laser radiation in a turbulent atmosphere. Appl. Opt. Vol 14 no. 12 p. 2857-2863

Botai C. M., Combrinck L. and Botai J. O., (2015). Satellite laser ranging measurements in South Africa: Contributions to earth system sciences. South African J. of Sci. Vol. 111(3/4), p. 1-9

Chapront J., Chapront-Touzé M., Francou G., Bidard P., Mangin J., Feraudy D., Furia M., Journet A., Torre J. and Vigouroux G., (2000). New improvements in the reduction of LLR observations. News of the LLR at Grasse with one particular night of Full Moon, In Proceedings of the 12th International Workshop Laser Ranging

Chapront J. and Francou G., (2003). Lunar laser ranging: Measurements, analysis, and contribution to the reference systems. IERS Technical Note, Vol 34 p. 97-116

Cheng M. and Tapley B. D., (2004). Variations in the Earth's oblateness during the past 28 years. J. Geophys. Res. Vol 110 no. B3, doi: 10.1029/2004JB003028

- Churnside** J. H., (1992). Turbulence Effects on the Geodynamic Laser Ranging System, In proceedings of “8th International Workshop on Laser Ranging”
Adapted from: ILRS
- Combrinck** L., (2014). Development of a high accuracy, user-friendly Lunar Laser Ranging telescope steering and pointing software package at HartRAO, In proceedings of “19th International Workshop on Laser Ranging” Adapted from: ILRS
- Combrinck** L. and Botha R., (2013). Challenges and progress with the development of a Lunar Laser Ranging for South Africa, In proceedings of “18th International Workshop on Laser Ranging” Adapted from: ILRS
- Combrinck** L., (2011). Development of a satellite and lunar laser ranging and its future applications 62nd International Astronautical Congress, IAC, Cape Town
- Combrinck** L., (2010). Sciences of Geodesy-I: Satellite Laser Ranging. Springer-Verlag
- Combrinck** L., (2007). White paper towards the establishment of the International Institute for Space Geodesy and Earth Observation (IISGEO), HartRAO
- Courde** C., Torre J. M., Samain E., Martinot-Lagarde G., Aimar M., Albanese D., Exertier P., Fienga A., Mariey H., Metris G., Viot H. and Viswanathan V., (2017). Lunar laser ranging in infrared at the Grasse laser station, A&A Vol 602, A90
- Cox** A. N., (2002). Allen’s Astrophysical Quantities 4th Edition, Springer

- Currie** D. G., Dell’Agnello S., Delle Monache G., Murphy T., Vittori R., Carrier D., Cantone C., Garattini M., Boni A., Martini M., Lops C., Intaglietta N., Bellettini G., Tauraso R., March R., Bianco G. and Rubincam D., (2008). A Lunar Laser Ranging Retro-Reflector Array for NASA’s Manned Landings, the International Lunar Network and the Proposed ASI Lunar Mission MAGIA, Sixteenth International Workshop on Laser Ranging Instrumentation, Poznan Poland, October 13-17,
- Currie** D., Dell’Agnello S. and Delle Monache G., (2009). A Lunar Laser Ranging Retroreflector Array for the 21st Century
- Degnan** J. J., (2012). “A tutorial on retroreflectors and arrays for SLR”, In proceedings of “17th International Workshop on Laser Ranging” Adapted from: ILRS
- Degnan** J. J., (1993). Contributions of space geodesy to geodynamics: Millimeter accuracy satellite laser ranging: A Review, American Geophysical Union
- Dickey** J. O., Bender P. L., Faller J. E., Newhall X. X., Ricklefs R. L., Ries J. G., Shelus P. J., Veillet C., Whipple A. L., Wiant J. R., Williams J. G. and Yoder C. F., (1994). Lunar laser ranging: A continuing legacy of the Apollo program Science Vol 265 no. 5171 p. 482-490
- Dirkx** D., Noomen R., Prochazka I., Bauer S. and Vermeersen L. L. A., (2015). Influence of atmospheric turbulence on planetary transceiver laser ranging, Advances in Space Research Vol. 54 no. 11 p. 2349-2370
- Faller** J. E., Alley C. O., Bender P. L., Currie D. G., Dicke R. H., Kaula W. M., MacDonald G. J. F., Mulholland J. D., Plotkin H. H., Silverberg E. C.

- and Wilkinson D. T., (1971). Laser Ranging Retroreflector, APOLLO 14, Preliminary Science Report, NASA SP-272
- Fan** C., Dong X., Zhao Y. and Han X., (2006). A032-ET Experimental Test on Changchun SLR, In proceedings of “15th International Workshop on Laser Ranging” Adapted from: ILRS
- Fried** D. L., (1966). Optical resolution through a randomly inhomogeneous medium for very long and very short exposures, J. Opt. Soc. Am. Vol 56 p 1372-1379
- Fried** D. L., (1965). Statistics of a Geometric Representation of Wavefront Distortion, JOSA Vol 55 no. 11 p 1427-1431
- Gallagher** M. W., Connolly P. J., Whiteway J., Figueras-Nieto D., Flynn M., Choularton T. W., Bower K. N., Cook C., Busen R. and HÄcker J., (2005). An overview of the microphysical structure of cirrus clouds observed during EMERALD-1, Q. J. R. Meteorol. Soc. Vol 131, p 1143-1169
- Gardner** C. S., (1976). Effects of random path fluctuations on the accuracy of laser ranging system, Appl. Opt. Vol. 15 p. 2539-2545
- Gerber** H., Twohy C. H., Gandrud B., Heymsfield A. J., McFarquhar G. M., DeMott P. J. and Rogers D. C., (1998). Measurements of wave-cloud microphysical properties with two new aircraft probes, Geophys. Res. Lett. Vol 25 no. 8, p 1117-1120
- Gordon** G. R., (1959). The LASER, Light Amplification by Stimulated Emission of Radiation, In Franken, P.A.; Sands R.H. (Eds.). The Ann Arbor Conference on Optical Pumping, the University of Michigan, 15 June through 18 June 1959. p. 128

- Grasyuk** A.Z. et al., (1964). Soviet Phys. Dokl., 9, 162
- Heymsfield** A., (1977). Precipitation Development in Stratiform Ice Clouds: A Microphysical and Dynamical Study, J. Atm. Sci. Vol 34 p 367-381
- Heymsfield** A., (1975). Cirrus Uncinus generating cells and the Evolution of Cirriform clouds. Part II: The structure and circulation of the cirrus Uncinusgenerating head, J. Atm. Vol 32 p 809-819
- Hofmann** F., Müller J. and Biskupek L., (2010). Lunar laser ranging test of the Nordtvedt parameter and a possible variation in the gravitational constant, A&A Vol 522 p 1-3
- Hofmann** F., Müller J., Biskupek L., Mai E. and Torre J. M., (2013). Lunar Laser Ranging - What is it Good for? In proceedings of “18th International Workshop on Laser Ranging” Adapted from: <http://ilrs.gsfc.nasa.gov/docs/2013>
- Klein** B. J. and Degnan J. J., (1974). Optical Antenna Gain. 1: Transmitting Antennas. Appl. Opt. Vol. 13, p. 2134-2140
- Kokurin** Yu. L., Kurbasov V. V., Lobanov V. F., Mozhzherin V. M., Sukhanovski A. N. and Chernykh N. S., (1966). Zhur. Exper. Theor. Fiziki (Pisma) 3, 219
- Kopal** Z., (1961). PHYSICS and ASTRONOMY of the MOON, ACADEMIC PRESS, New York and London
- Larsen** H., Gayet J. -F., Febvre G., Chepfer H. and Brogniez G., (1998). Measurements errors in cirrus cloud microphysical properties, EGS - Springer-Verlang, Ann. Geophysicae Vol 16, p 266-276

- Lawson** R. P., Baker B., Pilson B. and Mo Q., (2006). In Situ Observations of the Microphysical Properties of Wave, Cirrus, and Anvil Clouds. Part II: Cirrus Clouds, J. Atmos. Sc. Vol 63, p 3186-3203
- Liou** K. N., Ou S. C. and Koenig G., (1990). An investigation on the climatic effect of contrail cirrus. Air Traffic and the Environment Background, Tendencies and Potential Global Atmospheric Effects, U. Schumann, Ed., Springer-Verlag, p 154–169
- Marini** J. W. and Murray C. W., (1973). Correction of Laser Range Tracking Data for Atmospheric Refraction at Elevations above 10 Degrees, NASA Technical Report X591-73-351
- Martinot-Lagarde** G., Aimar M., Albanèse D., Courde C., Exertier P., Fienga A., Mariey H., Métrisb G., Rigard-Cerison R., Samain E., Torre J.-M. and Viot H., (2016). Laser enhancements for Lunar Laser Ranging at 532 nm, Results in Physics Vol 6 p. 329-336
- Mayer** D., Böhm J., Combrinck L., Botai J. and Böhm S., (2014). Importance of the Hartebeesthoek Radio Astronomy Observatory for the VLBI network, Acta Geodaetica et Geophysica, Vol 49 no. 3, p 313-325
- Mayer** K. J. and Young C. Y., (2008). Effect of atmospheric spectrum models on scintillation in moderate turbulence, J. Mod. Opt. Vol. 55 no. 7 p 1101-1117
- McAulay** A. D., (2011). Laser to Address Threat of New Nuclear Weapons, Military Laser Technology for Defense: Technology for Revolutionizing 21st Century Warfare

- McCarthy** D. D. and Petit G., (2003). IERS Technical Note No. 32.
<https://www.iers.org/SharedDocs/Publikationen/EN/IERS/Publications>
 Retrieved 18 June 2013
- McGarry** J., (2012). An Overview of Satellite Laser Ranging (SLR), Available
 from: <http://space-geodesy.nasa.gov>
- Mey** P. and Combrinck L., (2016). Preparations for a new VGOS radio telescope
 at HartRAO, 9th IVS General Meeting, Johannesburg, South Africa
- Michelsen** E. L., (2010). Normal Point Generation and First Photon Bias
 Correction in APOLLO Lunar Laser Ranging, University of California,
 PhD thesis
- Munghemezulu** C., Combrinck L. and Botai J. O., (2016). A review of the
 lunar laser ranging technique and contribution of timing systems. S Afr
 J Sci. 112(3/4), Art. #2015-0400, 9 pages. <http://dx.doi.org/10.17159/sajs.2016/20150400>
- Murphy** T. W., McMillan R. J. and Goodrow S. D., (2014). Lunar eclipse
 observations reveal anomalous thermal performance of Apollo reflectors,
 Icarus Vol 231, p 183-192
- Murphy** T. W., (2013). Lunar laser ranging: the millimeter challenge, Rep.
 Prog. Phys. Vol 76 076901 (21pp)
- Murphy** T. W., Adelberg E. G., Battat J. B. R., Hoyle C. D., McMillan R. J.,
 Michelsen E. L., Samad R. L., Stubbs C. W., and Swanson H. E., (2010).
 Long-term degradation of optical devices on the Moon, Icarus Vol. 208 p
 31-35

- Murphy** T. W., Adelberger E. G., Battat J. B. R., Carey L. N., Hoyle C. D., LeBlanc P., Michelsen E. L., Nordtvedt K., Orin A. E., Strasburg J. D., Stubbs C. W., Swanson H. E. and Williams E., (2008). The Apache Point Observatory Lunar Laser-ranging Operation: Instrument description and first detections. Publ. Astron. Soc. Pacific Vol 120, p 20–37
- Murphy** T.W., Adelberger E. G., Strasburg J. D. and Stubbs C. W., (2002). APOLLO: Multiplexed Lunar Laser Ranging,' In: 13th International Workshop on Laser Ranging. (Washington, D. C., USA, 2002), Source: <http://physics.ucsd.edu/~tmurphy/>
- Murphy** T. W., Strasburg J. D., Stubbs C. W., Adelberger E. G., Angle J., Nordtvedt K., Williams J. G., Dickey J. O. and Gillespie B., (2000). The Apache Point Observatory Lunar Laser-Ranging Operation (APOLLO), In Proceedings of the 12th International Workshop on Laser Ranging
- Müller** J., Williams J. G. and Turyshev S. G., (2008). Lunar laser ranging contributions to relativity and geodesy. In: Dittus H, Lammerzähl C, Turyshev SG, editors. Laser, clocks and drag-free control: Exploration of relativistic gravity in space. Berlin: Springer p 457–472
- NASA** Lunar Reconnaissance Orbiter (2010). Source:https://www.nasa.gov/mission_pages/LRO/multimedia/lroimages/ Accessed March 2014
- NASA**, (2014). How Satellite Laser Ranging Got its Start 50 Years Ago, Source:<https://www.nasa.gov/content/goddard> Accessed January 2015
- Ndlovu** S. C. and Chetty N., (2015). Experimental determination of thermal

turbulence effects on a propagating laser beam, CEJP Vol 13 no. 1, p 266-231

Ndlovu S. C. and Chetty N., (2014). Analysis of the fluctuations of a laser beam due to thermal turbulence, CEJP Vol. 12 no. 7 p. 466-472

Ndlovu S. C., Combrinck L., Akombelwa M. and Chetty N., (2016). Variable Link Equation Parameters and Expected Photon Returns for the HartRAO Lunar Laser Ranger - Improving our Understanding of the Earth-Moon System, South African J. of Geo. Vol 119 no.1, p 77-82

Ndlovu S. C., Combrinck L., Nkosi N. P. and Botha R. C., (2015). An Integrated Software Based Analytical Model for the Signal Path Efficiency of the HartRAO Lunar Laser Ranger Optical System. Proceedings of the South African Institute of Physics, 60th Annual Conference, 29th June to 3rd July, 2015, Port Elizabeth, South Africa

Ndlovu S. C., Combrinck L., Exertier P., Akombelwa M. and Chetty N., (2014). An Estimation of the Number of Expected Returned Photons for the HartRAO Lunar Laser Ranger System. Proceedings of the 19th International Workshop on Laser Ranging, 27th to 31st October, 2014, Annapolis, MD, USA.

Nickola M., (2012). Astronomical seeing conditions as determined by turbulence modelling and optical measurement, University of Pretoria, Master of Science Dissertation

Nkosi N., Combrinck W. L. and Akombelwa M., (2016). Optical configuration and optical tests of the HartRAO Lunar Laser Ranger, South African J. of Geo. Vol 119 no.1, 99-108

- Nkosi N.**, Characterization and development of optical components for the Cassegrain telescope and laser beam coudé path of the Lunar Laser Ranger of HartRAO, University of KwaZulu-Natal, Master of Science Dissertation
- Nordtvedt K.**, (1988). Lunar laser ranging and laboratory Eotvos-type experiments, *Physical Review D: Particles and Fields* Vol 37 no. 4, p 1070
- Obukhov A. M.**, (1941). On the distribution of energy in the spectrum of turbulent flow, *Dokl. Akad. Nauk SSSR* Vol 32, p 22-24
- Otsubo T.** and **Appleby G. M.**, (2003). System-dependent center-of-mass correction for spherical geodetic satellites, *J. Geophys. Res. (Solid Earth)* 108, 2201
- Roddier F.**, (1981). The effects of atmospheric turbulence in optical astronomy, *Prog. Optics* Vol 19, p 281-376
- Roddier F.**, (1989). Optical propagation and image formation through the turbulent atmosphere. In *Diffraction-limited imaging with very large telescopes*, ed. DM Alloin, JM Mariotti, Vol 274, p 33-52
- Samain E.**, Mangin, J. F., Veillet, C., Torre, J. M., Fridelance, P., Chabaudie, J. E., Feraudy, D., Glentzlin, M., Pham van, J., Furia, M., Journet, A. and Vigouroux, G., (1998). Millimetric Lunar Laser Ranging at OCA (Observatoire de la Cote d’Azur) *Astronomy and Astrophysics Supplement*, Vol 130, p 235-244
- Scheffer L. K.**, (2005). Better Lunar Ranges with Fewer Photons - Resolving the Lunar Retro-reflectors. Source:<https://arxiv.org/pdf/gr-qc>

- Schneider** M., Müller J., Schreiber U. and Egger D., (1996). High Precision Measurement of the Earth-Moon distance.
Source:<http://www.fesg.bv.tum.de/>
- Shabala** S. S., McCallum J. N., Plank L. and Böhm J., (2015). Simulating the effects of quasar structure on parameters from geodetic VLBI, J. of Geodesy, Vol 89 no. 9, p-873-886
- Silverberg** E. C., (1974). Operation and Performance of a Lunar Laser Ranging Station, Appl. Opt. Vol. 13, no. 3 p. 565-574
- Smith** M. A., (1964). Laser Tracking Success. Published in Flight International, Chief editor M.A. Smith, Number 2908, 86.
- Smullin**, L. D. and Fiocco C., (1962), Optical echoes from the Moon: Nature Vol 194 no. 4835, p 1267
- Stark** A. A., Camberlin R. A., Ingalls J. G., Cheng J. and Wright G., (1997). Optical and mechanical design of the Antarctic Submillimeter Telescope and Remote Observatory, Rev. Sci. Instrum., Vol 68 no. 5, p 2200-2213
- Stordal** F., Myhre G., Stordal E. J. G., Rossow W. B., Lee D. S., Arlander D. W., and Svendby T., (2005). Is there a trend in cirrus cloud cover due to aircraft traffic? Atmos. Chem. Phys., Vol 5, p 2155-2162
- Stubbs** T.J., Vondrak R.R. and Farrell W.M., (2006). A dynamic fountain model for lunar dust. Adv. Space Res. Vol 37, p 59-66
- Takeuchi** S., Kim J., Yamamoto Y. and Hogue H. H., (1999). Development of a high-quantum-efficiency single-photon counting system, Appl. Phys. Lett. Vol 74 no. 8, p 1063-1065

- Tatarski** V. I., (1961). Wave propagation in a turbulent medium, Dover
- Tian** W. and Yeh H.-C., (2016). Development of software for LLR data analysis at TianQin Research Center. In Proceedings of the 20th International Workshop on Laser Ranging
- Tsiao** S., (2007). “Read you loud and clear!” The story of NASA’s spaceflight tracking and data network, NASA SP-4233
- Turyshev** S. G., Williams J. G., Folkner W. M., Gutt G. M., Baran R. T., Hein R. C., Somawardhana R. P., Lipa J. A. and Wang S., (2012). Corner-cube retro-reflector instrument for advanced lunar laser ranging, Source: <https://arxiv.org/abs/1210.7857>
- Vasilyev**, M.V. and Yagudina E. I., (2014), Russian lunar ephemeris EPM-ERA 2012, Sol Syst Res Vol 48 no. 2, p 158 - 165
- Vernin** J. and Muñoz-Tuñón C., (1992). Optical seeing at La Palma Observatory, I. General guidelines and preliminary results at the Nordic Optical Telescope. Astronomy & Astrophysics Vol 257, p 811-816
- Wang** D., Yang Y., Chen C., and Zhuo Y., (2011). Point diffraction interferometer with adjustable fringe contrast for testing spherical surfaces, App. Opt. 50(16), 2342-2348, (2011).
- Williams** J. G. and Standish E. M., (1989). Dynamical Reference Frames in the Planetary and Earth-Moon Systems, Astrophys. & Space Sci. Lib., Vol 154, p 67-90
- Williams** J. G. and Boggs D. H., (2009). Lunar Core and Mantle. What Does LLR See? In Proceedings of the 16th International Workshop on Laser Ranging

- Williams** J. G., Boggs D. H. and Ratcliff J. T., (2009). A Larger Lunar Core?
40th Lunar and Planetary Science Conference, (Lunar and Planetary Science
XL), held March 23-27, in The Woodlands, Texas, id.1452
- Williams** J. G., Turyshev S. G. and Boggs Dale H., (2009) Lunar Laser Ranging
Tests of the Equivalence Principle with the Earth and Moon, Inter. J. of
Mod. Phys. D, Vol 18, p 1129-1175
- Wohl** C. J., Connell J. W., Lin Y., Belcher M. A., Palmieri F. L. and Atkins B.
M., (2011). Generation and Evaluation of Lunar Dust Adhesion Mitigating
Materials, American Institute of Aeronautics and Astronautics, source:
<https://ntrs.nasa.gov/>
- Yaoheng** X. and Hesheng F., (2002). Modification of laser ranging equation
Available from: <http://cddis.nasa.gov/lw13/>
- Yuskel** H., (2005). Studies of the effects of atmospheric turbulence on free space
optical communications, Ph.D. Thesis, Univ. of Maryland
- Zhang** Z., (1997). Parameter Estimation Techniques: A Tutorial with
Application to Conic Fitting. Image and Vision Computing, Vol 15 no.1, p
59-76
- ZhouS** H. and Zheng L., (2011). The phase mode of reflected beam from laser
retroreflector with dihedral angle and flatness error, Opt. Rev. Vol 18 no.
1, p 1-6

Appendix A

The source code of the developed C++ program. This program is for controlling our 0.2 m beam expander's eyepiece lens. The code is a modification of a code which was initially written by Ludwig Combrinck at the Space Geodesy Programme at the Hartebeesthoek Radio Astronomy Observatory. It runs a MEGA2560 micro-controller as part of a Beam Expander control monitoring system. There are two software parts to this system;

- 1) the code that is uploaded to the MEGA2560
- 2) an RS232 GUI interface which sends commands to the code installed in the MEGA2560 (APPENDIX E).

```
// Start date of coding was 25 October 2013 by Sphumelele Ndlovu (PhD candidate) at Hartebeesthoek Radio
  Astronomy Observatory (HartRAO).
// This code is for controlling the beam expander that is used for Lunar Laser Ranger system at HartRAO.
// It is a modification of the code which was initially written by Ludwig Combrinck, Space geodesy
  programme leader, HartRAO.
// I would like to thank him for allowing me to use and modify his software.
```

```
/******
```

Start date of coding (after some preliminary tests) 11 September 2012, Ludwig Combrinck onboard
Agulhas II

Gough Island takeover voyage 2012.

This is code to run a MEGA2560 micro-controller as part of a Beam expander control monitoring system
written to be test implemented on a German built 5" Beam expander (Bernd Spaeker) equipped
with a handmade achromatic lens by Dieter Lichtenknecker. It is hoped that the code can
be used as is for implementation on the 1 m Cassegrain being developed for Lunar Laser Ranging.

There are two software parts to this monitoring system;

- 1) an RS232 GUI interface which sends commands to
- 2) the code that is uploaded to the MEGA2560.

Designed specifically to work with the Adafruit LED 7-Segment backpacks.

These displays use I2C to communicate, 2 pins are required to interface. There are 8 multiple
selectable I2C

addresses, so 8 four character displays can be driven from 1 card.

```
--- A2  A1  A0
0x70 : 0   0   0
0x71 : 0   0   1
0x72 : 0   1   0
0x73 : 0   1   1
0x74 : 1   0   0
0x75 : 1   0   1
0x76 : 1   1   0
0x77 : 1   1   1
```

with 3 Address Select pins: 0x70 thru 0x77

```
*/
```

```
#include <Wire.h>
#include "Adafruit_LEDBackpack.h"
#include "Adafruit_GFX.h"
#include "MsTimer2.h"          // MsTimer2 library from the contributed libraries at arduino.cc
#include <avr/boot.h>
#include <LiquidCrystal.h>
```

```
int Card_Input_Pin = A0;      // select the input pin for card voltage
int Position_Input_Pin = 8;
int Card_V_status = 53;      // pwm output for warning LED if card voltage is out of range
int Card_V = 0;              // variable to store the card Vin
int Pulses = 0;
float Tel_LHA;                // Beam expander local hour angle
float Tel_DEC;                // and declination
float Card_Voltage = 0;
float Position = 0;
int test = 5;
float scale_conversion = 5.0/1024.0;
float scale_conversions = 100;
const int FocusPin = 12;      // pin that the focus drive is attached to (to vary between 0-5
  volt) [to be selected]
const int Focus_direction_Pin = 11; //pin which enables focus motor direction changes
boolean drawdot = true;
boolean notdrawdot = false;
boolean auto_on = false;
boolean check_command = false;
boolean sync_clock = false;
```

```

boolean sync_seconds = false;
boolean sync_minutes = false;
boolean sync_hours = false;
boolean sync_days = false;
boolean clock_step = false;
boolean setbrightness = false; //LED display brightness
boolean setfocus = false; // adjust focus
boolean focusON = false; // focus drive enable/disable
boolean setposition = false; // adjust focus
boolean focus_increase = false;
boolean focus_decrease = false;

int tick,i,n;
char seconds = 0;           // Initial values for m,s,h,d matrix display
char minutes = 0;
char hours = 0;
char days = 0;
int int_seconds = 0;        // Initial integer values for m,s,h,d
int int_minutes = 0;
int int_hours = 0;
int int_days = 0;
char adjust_seconds = 0;
char LED_adjust = 0;
char focus_adjust = 5;
int pulses, A_SIG=0, B_SIG=1;

unsigned char RxBuffer[1000];
char m_strEditRead;

int brt = 5; // default 1-16 for brightness level (adjustable from GUI)
int focus = 0; // focus speed default
int position, positiona;
int temp_focus = 0; // temp holding for ramp
int temp_position;

//int pulses, A_SIG=0, B_SIG=1;

Adafruit_7segment matrix1 = Adafruit_7segment();
Adafruit_7segment matrix2 = Adafruit_7segment();

// initialize the library with the numbers of the interface pins
//LiquidCrystal lcd(9, 8, 7, 6, 5, 4);

//creating and writing to a file

void setup()
{
    // set up the LCD's number of columns and rows:
    // lcd.begin(16, 2);
    // Print a message to the LCD.
    // lcd.print("hello, world!");

    tick=0;
    pinMode (FocusPin, OUTPUT);    // pin 12 which sets 0-5 volt level for focus drive speed
    // Arduino sets these to inputs by default (if LED is driven MUST have 220 ohm resistor in serie)
    // set pins to drive 8 channel relay card
    for (int pinindex = 10; pinindex < 53; pinindex++) { // Focus motor enable 4, Focus in/out pin 11
        RR1: 45, 46, 47, 48, 49, 50, 51, 52
        pinMode(pinindex, OUTPUT); // set pins as outputs
    }

    attachInterrupt(0, A_RISE, RISING);
    attachInterrupt(1, B_RISE, RISING);

    Serial.begin(115200);
    Serial.println("Beam expander controlling system V1.25_10_2013 Sphumelele Ndlovu");
}

```

```

//clock
matrix1.begin(0x70); // Start up the LCD matrixes
matrix2.begin(0x71);

matrix1.setBrightness(brt); // Soft lighting at 5V
matrix2.setBrightness(brt);

matrix1.print(0xABCD, HEX); // display some chars in HEX as start-up indicator
matrix2.print(0xDEFF, HEX); //

matrix1.writeDisplay(); // Write matrices to the displays
matrix2.writeDisplay();
delay(2500);

matrix1.clear(); // Clear the matrices
matrix2.clear();

display_time(); // Show initial time

MsTimer2::set(1000, increase_time); // Every second increase_time() is called
MsTimer2::start();

// end of clock
}

void loop() {
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
//lcd.setCursor(0, 1);
// print the number of seconds since reset:
//lcd.print(pulses);

int incomingByte, incomingByte2, incomingByte3, incomingByte4, incomingByte5, i, n;

if (Serial.available() > 0)
{
    incomingByte = Serial.read();

    switch (incomingByte)
    {
        case '0':
            Serial.println(" System LED off");
            digitalWrite(13,LOW);
            // Serial.println("-----");
            break;
        case '1':
            Serial.println(" System LED on");
            digitalWrite(13,HIGH);
            break;
        case 'i':
            i2c_scanner(); // check if i2c devices respond to address calls
            break;
        case 'v': // read onboard voltage
            read_onboard_voltage();
            break;
        case '2': // switch on mains (220 V) to Beam expander system (r) to switch off
            digitalWrite(52,HIGH); //
            Serial.println(" Beam expander mains (220 V) switched ON");
            Serial.println(" 5 Volt; 12 Volt variable, 12 Volt 3 A switched ON");
            break;
        case 'r': // switch off mains (220 V) to Beam expander system (2) to switch on
            digitalWrite(52,LOW); //
            Serial.println(" Beam expander mains (220 V) switched OFF");
            break;
        case '3': // switch on Microcontroller supplies (24 Volt)
            digitalWrite(51, HIGH); //
            Serial.println(" Microcontroller supplies switched ON");
            break;
        case 's': // switch off Microcontroller supplies (24 Volt)
            digitalWrite(51,LOW); //
    }
}

```

```

    Serial.println(" Microcontroller supplies switched OFF");
    break;
case '4': // switch on focus power supply (1 x 12 Volt)
    digitalWrite(50, HIGH); //
    Serial.println(" Focus power supply switched ON");
    break;
case 't': // switch off focus power supply (1 x 12 Volt)
    digitalWrite(50, LOW); //
    Serial.println(" Focus power supply switched OFF");
    break;
case '5': // switch on Night Lamp
    digitalWrite(49, HIGH); //
    Serial.println(" Night Lamp switched ON");
    break;
case 'u': // switch off Night Lamp
    digitalWrite(49, LOW); //
    Serial.println(" Night Lamp switched OFF");
    break;
case '6': // switch on auxiliary outlet (12 Volt)
    digitalWrite(48, HIGH); //
    Serial.println(" Auxuliary output (12 V DC) switched ON");
    break;
case 'w': // switch off auxiliary outlet (12 Volt)
    digitalWrite(48, LOW); //
    Serial.println(" Auxiliary output (12 V DC) switched OFF");
    break;
case '7': // switch on CCD camera power supply
    digitalWrite(47, HIGH); //
    Serial.println(" CCD camera switched ON");
    break;
case 'x': // switch off CCD camera power supply
    digitalWrite(47, LOW); //
    Serial.println(" CCD camera switched OFF");
    break;
case '8': // switch on relay rack 2, bank of 5, connects dc motors to servos etc.
digitalWrite(46, HIGH); // pin to be assigned
    digitalWrite(46, HIGH);
    Serial.println(" System enable ON");
    break;
case 'y': // disable, switch relay rack 2
    digitalWrite(46, LOW); //
    Serial.println(" System enable OFF");
    break;
case '9': // switch on spare relay
    digitalWrite(45, HIGH); //
    Serial.println(" Spare relay ON");
    break;
case 'z': // switch off spare relay
    digitalWrite(45, LOW); //
    Serial.println(" Spare relay OFF");
    break;

case 'Z': // synchronize card clock with pc clock
    sync_clock = true;
    check_command=false;
    Serial.println(" Synchronise clock selected");
    digitalWrite(52, HIGH); // set LED on to display sync clock is active
    while(sync_clock && !check_command)
    {
        if (Serial.available() > 0)
        { incomingByte3 = Serial.read(); switch (incomingByte3)
        {
            case 's': Serial.println(" Seconds (xx) ");
                sync_seconds=true;
                pulse_LED(52); // and leave it on
                while(sync_seconds)
                {
                    if (Serial.available() > 0)
                    { incomingByte4 = Serial.read(); switch (incomingByte4)
                    {

```

```

        case 'u': int_seconds++;
        Serial.println("seconds ++ ");
        seconds=(char)int_seconds; Serial.println(
(int)seconds);
        break;
        case 'd': int_seconds--;
        Serial.println("seconds -- ");
        seconds=(char)int_seconds; Serial.println(
(int)seconds);
        break;
        case 'f': sync_seconds = false;
        Serial.println(" Seconds completed ");
        break;
    }
}
sync_seconds=false;
break;

case 'm': Serial.println(" Minutes (xx) ");
sync_minutes=true;
pulse_LED(52); // and leave it on
while(sync_minutes)
{
    if (Serial.available() > 0)
    { incomingByte4 = Serial.read(); switch (incomingByte4)
    {
        case 'u': int_minutes++;
        Serial.println("minutes ++ ");
        minutes=(char)int_minutes; Serial.println(
(int)minutes);
        break;
        case 'd': int_minutes--;
        Serial.println("minutes -- ");
        minutes=(char)int_minutes; Serial.println(
(int)minutes);
        break;
        case 'f': sync_minutes = false;
        Serial.println(" Minutes completed ");
        break;
    }
}
sync_minutes=false;
break;

case 'h': Serial.println(" Hours (xx) ");
sync_hours=true;
pulse_LED(52); // and leave it on
while(sync_hours)
{
    if (Serial.available() > 0)
    { incomingByte4 = Serial.read(); switch (incomingByte4)
    {
        case 'u': int_hours++;
        Serial.println("hours ++ ");
        hours=(char)int_hours; Serial.println((int)
hours);
        break;
        case 'd': int_hours--;
        Serial.println("hours -- ");
        hours=(char)int_hours; Serial.println((int)
hours);
        break;
        case 'f': sync_hours = false;
        Serial.println(" Hours completed ");
        break;
    }
}
sync_hours=false;

```

```

        break;

        case 'D': Serial.println(" Days (xx) ");
        sync_days=true;
        pulse_LED(52); // and leave it on
        while(sync_days)
        {
            if (Serial.available() > 0)
            { incomingByte4 = Serial.read(); switch (incomingByte4)
              {
                  case 'u': int_days++;
                            Serial.println("days ++ ");
                            days=(char)int_days; Serial.println((int)
days);
                  break;
                  case 'd': int_days--;
                            Serial.println("days -- ");
                            days=(char)int_days; Serial.println((int)
days);
                  break;
                  case 'f': sync_days = false;
                            Serial.println(" Days completed ");
                  break;
              }
            }
        }
        sync_days=false;
        break;

        case 'F': check_command = true;
        sync_clock = false;
        Serial.println(" Synchronising clock completed ");
        int_seconds=0;int_minutes=0;int_hours=0;int_days=0;
        digitalWrite(52,LOW);

        break;
    }
}
break;
case 'A': // set auto monitoring mode
    auto_on=true;
    check_command=false;
    Serial.println(" Auto monitoring has been enabled");
    digitalWrite(53,HIGH); // set LED on to display automode is active
    while(auto_on && !check_command)
    {
        read_onboard_voltage();
        // voltages();
        if (Card_Voltage < 4.0) { // needs to be calibrated
            pinMode(13, OUTPUT);

            Serial.println(" Card voltage below specification: ");
            Serial.println(Card_Voltage);
            check_command = true;
            disable_drive; // switch of relays to power supplies
            (" Disabled drives! ");
            MsTimer2::set(500, flash); // 500ms period
            MsTimer2::start();
        }
        display_time();
        if (Serial.available() > 0)
        { incomingByte2 = Serial.read(); switch (incomingByte2)
          {
              case 'Q': check_command = true;
                        Serial.println(" Auto monitoring has quit Q");
                        digitalWrite(53,LOW); // set LED off to display automode is inactive
                        break;
              case 'p': check_command = true;
                        Serial.println(" Auto monitoring has quit p");
                        digitalWrite(53,LOW); // set LED off to display automode is inactive

```



```

        break;
    }
}

case 'R': // set auto monitoring mode
auto_on=true;
check_command=false;
Serial.println(" Auto focusing has been enabled");
digitalWrite(53,HIGH); // set LED on to display automode is active
while(auto_on && !check_command)
{
    read_position();
    // voltages();
    if (Position < 0) { // needs to be calibrated
        pinMode(13, OUTPUT);

        Serial.println(" Current focus position is unknown: ");
        Serial.println(Position);
        check_command = true;
        disable_drive; // switch of relays to power supplies
        (" Disabled drives! ");
        MsTimer2::set(500, flash); // 500ms period
        MsTimer2::start();
    }
    display_time();
    if (Serial.available() > 0)
    { incomingByte2 = Serial.read(); switch (incomingByte2)
    {
        case 'S': check_command = true;
        Serial.println(" Auto focusing has quit S");
        digitalWrite(53,LOW); // set LED off to display automode is inactive
        break;
        case 'p': check_command = true;
        Serial.println(" Auto focusing has quit p");
        digitalWrite(53,LOW); // set LED off to display automode is inactive
        break;
    }
    }
}
break;

case 'B': // set brightness levels of LED displays
setbrightness=true;
check_command=false;
Serial.println(" Adjusting LED brightness levels has been enabled");
digitalWrite(53,HIGH); // set LED on to display automode is active
while(setbrightness && !check_command)
{
    display_time();
    if (Serial.available() > 0)
    { incomingByte2 = Serial.read(); switch (incomingByte2)
    {
        case 'u': brt++;
        if(brt > 16) brt = 0;
        Serial.println("LED brightness ++ ");
        LED_adjust=(char)brt;
        Serial.println((int)LED_adjust);
        Adjust_LED(brt);

        break;
        case 'd': brt--;
        if(brt < 0) brt = 16;
        Serial.println("LED brightness -- ");
        LED_adjust=(char)brt;
        Serial.println((int)LED_adjust);
        Adjust_LED(brt);

        break;
    }
    }
}

```



```

        digitalWrite(7,HIGH); // set (to be selected)
    } else Serial.println("Adjusting of focus has not been enabled!");
break;

//set up speed

case 'u': if(setfocus){
    focus++;
    if(focus > 10) focus = 10;
    Serial.println("Focus speed ");
    focus_adjust=(char)focus;
    Serial.println((int)focus_adjust);
    Adjust_Focus(focus);
} else Serial.println(" Adjusting of focus has not been enabled!");

break;
case 'd': if(setfocus){
    focus--;
    if(focus < 0) focus = 0;
    Serial.println("Focus speed ");
    focus_adjust=(char)focus;
    Serial.println((int)focus_adjust);
    Adjust_Focus(focus);
} else Serial.println("Adjusting of focus has not been enabled!");

break;

case 's': if(setfocus) {
    focusON = !focusON;
    if (!focusON)
        {digitalWrite(10,LOW); // set off (to be selected) pin low
        //Freewheel: (TB1, Pin 3) Logic "1" (open) - Motor is Enabled Logic "0"
- Motor is de-energized and will coast
        Serial.println("Focus OFF ");
        } else
        { digitalWrite(10,HIGH); // set (to be selected) pin high
        Serial.println("Focus ON ");
        while (focusON = focusON) {
            A_RISE();
            B_RISE();
            A_FALL();
            B_FALL();
            delay(100/focus);
            if(position==pulses || pulses==position){
                focusON = false;
                digitalWrite(10,LOW); // switch off motor driver
                if (pulses--) Serial.println(pulses);
                else Serial.println(pulses);
                Serial.println("Focus position ");
            }
        }
        break;
    }
} else Serial.println("Adjusting of focus has not been enabled!");

break;

case 'f': if(setfocus) {
    setfocus = false;
    focusON = false;
    check_command = true;
    focus = 0;
    Serial.println("Focus adjustment completed at ");
    // Serial.println("Focus speed reset to 0 ");
    focus_adjust=(char)focus;
    if (pulses>0){
        Serial.println(pulses*100/1400 );
    }
    else Serial.println(pulses*-100/1400 );
}

```

```

        //Serial.println(pulses*100/1400);
        Serial.println("%");
        Adjust_Focus(focus);
        digitalWrite(10,LOW); // set off (to be selected) pin low
        //Freewheel: (TB1, Pin 3) Logic "1" (open) - Motor is Enabled Logic "0" ✓
- Motor is de-energized and will coast
        //Serial.println("Focus OFF ");

        digitalWrite(53,LOW); // set (to be selected) LED off to show focus mode ✓
is disabled
    } else Serial.println("Adjusting of focus has not been enabled!");
        break;
    }
}
break;

// empty the buffer of incoming data
while (Serial.available() > 0)
    Serial.read();
Serial.println("-----");
break;

// Ignore carriage return and line feed.
case '\r':
case '\n': Serial.print("Return"); break;
case '\n': Serial.print("Linefeed"); break;

default:
    Serial.print("Unknown command: 0x");
    Serial.println(incomingByte,HEX);
    // Something could be wrong, delete incoming data in buffer
    while (Serial.available() > 0)
        Serial.read();
    Serial.println("-----");
    break;
}
}

if (tick) {
    // If a tick has occurred
    tick = 0; // reset indicator that we have a new time
    display_time(); // and show it off
    write_LHA();
}
}

void i2c_scanner(void)
{
    byte error, address;
    int nDevices;

    Wire.begin();

    nDevices = 0;
    for(address = 0; address <= 127; address++ )
    {
        // The i2c_scanner uses the return value of
        // the Wire.endTransmission to see if
        // a device did acknowledge to the address.
        Wire.beginTransmission(address);
        error = Wire.endTransmission();

        if (error == 0)
        {
            Serial.print("I2C device found at address 0x");
            if (address<16)

```

```

        Serial.print("0");
        Serial.print(address,HEX);
        Serial.print(" ");

        nDevices++;
    }
    else if (error==4)
    {
        Serial.print("Unknow error at address 0x ");
        if (address<16)
            Serial.print("0");
        Serial.println(address,HEX);
    }
}
if (nDevices == 0)
    Serial.println("No I2C devices found. ");
else
    Serial.println("Completed I2C check. ");
}

void read_onboard_voltage()
{
    // read card voltage at Vin
    {
        Card_V = analogRead(Card_Input_Pin);
        matrix1.setBrightness(3);
        Card_Voltage = scale_conversion*Card_V;
        matrix1.print(Card_Voltage,3);
        matrix1.writeDisplay();
    }
}

void read_position()
{
    // read focus position
    {
        Pulses = digitalRead(Position_Input_Pin);
        matrix1.setBrightness(3);
        Position = scale_conversions*Pulses;
        matrix1.print(Position,3);
        matrix1.writeDisplay();
    }
}

void write_LHA()    // Beam expander local hour angle (to be read via DDE from spreadsheet)
{
    Tel_LHA = random(2)+4.23;
    {
        matrix1.setBrightness(3);
        matrix1.print(Tel_LHA,2);
        matrix1.writeDisplay();
    }
}

void display_time () {
    matrix2.writeDigitNum(4,seconds % 10);    // Function to display the time
    matrix2.writeDigitNum(3,seconds / 10);    // Light up 1's seconds
    matrix2.writeDigitNum(1,minutes % 10,true);    // Light up 10's seconds
    matrix2.writeDigitNum(0,minutes / 10);    // Light up 1's minutes (with divisor period)
    // Light up 10's minutes

    if ((hours > 0 && days == 0) || (days > 0)) {
        matrix1.writeDigitNum(4,hours % 10,true);    // Light up 10's minutes
        matrix1.writeDigitNum(3,hours / 10);    // Light up 1's hours (with divisor period)
    }
    // Light up 10's hours

    if (days > 0) {
        matrix1.writeDigitNum(1,days % 10,true);    // Light up 1's days (with divisor period)
        matrix1.writeDigitNum(0,days / 10);    // Light up 10's days
    }
    // Light up 10's days

    matrix1.writeDisplay();    // Take the values written to the matrixes and
    matrix2.writeDisplay();    // write them to the 7-segment displays
}

```

```

}

void increase_time() {
  if(!sync_clock) seconds++; // We're counting
  if (seconds > 59) { // If seconds have rolled over
    seconds = 0; // Reset seconds
    minutes++; // and increase the minutes
    if (minutes > 59){ // If minutes have rolled over
      clock_step !=clock_step; if(clock_step) {adjust_seconds++;
      seconds += 1; Serial.println("Adding 1 sec step
      correction -- ");
      Serial.println((int)adjust_seconds);
      }
      minutes = 0; // reset minutes
      hours++; // increase hours
      if (hours > 23){ // If hours have rolled over
        clock_step !=clock_step; if(clock_step) {adjust_seconds++;
        seconds += 8; Serial.println("Adding 1 sec step
        correction -- ");
        Serial.println((int)adjust_seconds);
        }
        hours = 0; // reset hours
        days++; // increase days
        if (days > 99){ // If days have rolled over
          days = 0; // reset time to starting value
        }
      }
    }
  }
  tick++; // Time has been updated
}

void flash() {
  static boolean output = HIGH;
  digitalWrite(13, output);
  output = !output;
}

void disable_drive() {
  // disable relays to drive power supplies
  // pins to be chosen
  // show 2 RED LEDs that drives are connected or disconnected, otherwise GREEN LED
  // show 1 LED that focus drive has been disconnected (otherwise on GREEN LED)
}

void pulse_LED(int pin)
{
  int count=10;
  while (count--){
    digitalWrite(pin, HIGH); delay(200); digitalWrite(pin, LOW);delay(200);}
    digitalWrite(pin, HIGH);
  }

void Adjust_LED(int brt)
{
  matrix1.setBrightness(brt); // add other matrices here
  matrix2.setBrightness(brt);
}

void Adjust_Focus(int focus)
{
  analogWrite(FocusPin, 15 + (focus*15)); // as max is 255; 16 levels x 15 = 240, i.e.240/255*5 volt =
  4.7 volt, to drive input of Anaheim MBDC050-024031 (5 volt max)
}

void Ramp_Down_Focus(int focus)
{temp_focus = focus;
  while (focus > 0)
    {

```

```

        delay(100);
        focus--;
        // Serial.println("Ramp focus speed down ");
        // focus_adjust = (char)focus; //for debug
        // Serial.println((int)focus_adjust);
        Adjust_Focus(focus);
    }
    // Serial.println("Ramp focus speed down ");
}

void Ramp_Up_Focus(int focus)
{
    focus = 0;
    while (temp_focus--)
    {
        delay(100);
        focus++;
        // Serial.println("Ramp focus speed up ");
        // focus_adjust = (char)focus; // for debug
        // Serial.println((int)focus_adjust);
        Adjust_Focus(focus);
    }
    temp_focus = 0;
    // Serial.println("Ramp focus speed up ");
}

// Interrupt on ramp up position changing state
void A_RISE(){
    detachInterrupt(0);
    A_SIG=1;

    if(B_SIG==0)
        pulses++; //moving forward
    if(B_SIG==1)
        pulses--; //moving reverse
    Serial.println(pulses);
    attachInterrupt(0, A_FALL, FALLING);
}

void B_RISE(){
    detachInterrupt(1);
    B_SIG=1;

    if(A_SIG==1)
        pulses++; //moving forward
    if(A_SIG==0)
        pulses--; //moving reverse
    Serial.println(pulses);
    attachInterrupt(1, B_FALL, FALLING);
}

void A_FALL(){
    detachInterrupt(0);
    A_SIG=0;

    if(B_SIG==1)
        pulses++; //moving forward
    if(B_SIG==0)
        pulses--; //moving reverse
    Serial.println(pulses);
    attachInterrupt(0, A_RISE, RISING);
}

void B_FALL(){
    detachInterrupt(1);
    B_SIG=0;

    if(A_SIG==0)
        pulses++; //moving forward
    if(A_SIG==1)
        pulses--; //moving reverse
    Serial.println(pulses);
}

```

```
    attachInterrupt(1, B_RISE, RISING);
}
void voltages()    // add other monitored voltages here
{
  Serial.println(Card_Voltage);
}
```


Appendix B

The source code of the RS232 GUI interface which sends commands to the code installed in the MEGA2560 is provided in this Appendix. The start date of coding was 25 October 2013 by Sphumelele Ndlovu (PhD candidate) at Hartebeesthoek Radio Astronomy Observatory (HartRAO). This code is for controlling the beam expander that is used for Lunar Laser Ranger system at HartRAO. It is a modification of the code which was initially written by Ludwig Combrinck, Space geodesy programme leader, HartRAO. I would like to thank him for allowing me to use and modify his software.

```
// Start date of coding was 25 October 2013 by Sphumelele Ndlovu (PhD candidate) at Hartebeesthoek Radio Astronomy Observatory (HartRAO). ✓
// This code is the GUI for controlling the beam expander that is used for Lunar Laser Ranger system at HartRAO. ✓
// It is a modification of the code which was initially written by Ludwig Combrinck, Space geodesy programme leader, HartRAO. ✓
// I would like to thank him for allowing me to use and modify his software.
```

```
/******
```

```
Start date of coding (after some preliminary tests) 11 September 2012, Ludwig Combrinck onboard Agulhas II Gough Island takeover voyage 2012.
```

```
This is code to run a MEGA2560 micro-controller as part of a Beam expander control monitoring system written to be test implemented on a German built 5" Beam expander (Bernd Spaeker) equipped with a handmade achromatic lens by Dieter Lichtenknecker. It is hoped that the code can be used as is for implementation on the 1 m Cassegrain being developed for Lunar Laser Ranging.*/
```

```
#include "stdafx.h"
#include "rs232a.h"
#include "rs232aDlg.h"
#include <process.h>
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
#define CR 0x0D
#define LF 0x0A
unsigned char RxBuffer[10000];
boolean auto_clear = false;
boolean Mains_ON = false;
boolean MicrocontrollerPowerSupplies_ON = false;
boolean FocusPowerSupply_ON = false;
boolean NightLamp_ON = false;
boolean AuxiliaryPower_ON = false;
boolean CCD_ON = false;
boolean Enable_ON = false; // 12 volt relay rack on
boolean Spare_ON = false;
boolean InLoop = false;
boolean SystemLED = false;
boolean focusloop = false;
boolean LEDLoop = false;
boolean clockloop = false;
boolean focus_toggle = false; // MUST match compatriot variable in 2560 code
boolean forward = true;
boolean reverse = false;
boolean auto_loop = false;
int position = 0;
int pulse;
int focus = 0; //focus counter, MUST match compatriot variable in 2560 code
int select_task = 0;
```

```
CStdioFile f;
```

```
DWORD dBAUD = 115200;
```

```
////////////////////////////////////
```

```
CRs232aDlg::CRs232aDlg(CWnd* pParent /*=NULL*/)
: CDialog(CRs232aDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CRs232aDlg)

    m_strTx = _T(" ");
    m_strHex = _T("01 00");
    m_strEditRead = _T(" ");

    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);

    }
```

```
void CRs232aDlg::DoDataExchange(CDataExchange* pDX)
```

```
{
    CDialog::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_TX_EDIT, m_strTx);
    DDX_Text(pDX, IDC_HEX_EDIT, m_strHex);
    DDX_Text(pDX, IDC_EDIT_READ, m_strEditRead);
    DDX_Control(pDX, IDC_DATETIMEPICKER1, m_MonthCal1);
    DDX_Control(pDX, IDC_DATETIMEPICKER2, m_MonthCal2);
    //}}AFX_DATA_MAP
}
```

```
BEGIN_MESSAGE_MAP(CRs232aDlg, CDialog)
```

```
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_TX, OnTx)
    ON_BN_CLICKED(IDC_RX, OnRx)
    ON_WM_CHAR()
    ON_BN_CLICKED(IDC_BAUD110, OnBaud110)
    ON_BN_CLICKED(IDC_BAUD300, OnBaud300)
    ON_BN_CLICKED(IDC_BAUD600, OnBaud600)
    ON_BN_CLICKED(IDC_BAUD1200, OnBaud1200)
    ON_BN_CLICKED(IDC_BAUD19200, OnBaud19200)
    ON_BN_CLICKED(IDC_BAUD2400, OnBaud2400)
    ON_BN_CLICKED(IDC_BAUD4800, OnBaud4800)
    ON_BN_CLICKED(IDC_BAUD57600, OnBaud57600)
    ON_BN_CLICKED(IDC_BAUD115200, OnBaud115200)
    ON_BN_CLICKED(IDC_BAUD9600, OnBaud9600)
    ON_BN_CLICKED(IDC_Set_System_LED_Off, OnCommand)
    ON_BN_CLICKED(IDC_I2C_Scanner, OnCommand2)
    ON_BN_CLICKED(IDC_Reset_Card, OnCommand3) // software reset for MEGA2560
    ON_BN_CLICKED(IDC_Enable_Auto, OnCommand4) // enable auto monitoring
    ON_BN_CLICKED(IDC_Disable_Auto, OnCommand5) // disable auto monitoring
    ON_BN_CLICKED(IDC_Sync_Clock, OnCommand6) // synchronize clock
    ON_BN_CLICKED(IDC_Sync_Clock_Seconds, OnCommand7) // seconds
    ON_BN_CLICKED(IDC_Sync_Clock_Minutes, OnCommand8) // minutes
    ON_BN_CLICKED(IDC_Sync_Clock_Hours, OnCommand9) // hours
    ON_BN_CLICKED(IDC_Sync_Clock_Days, OnCommand10) // days
    ON_BN_CLICKED(IDC_Sync_Clock_Seconds_Done, OnCommand11) // escape from ss setting
    ON_BN_CLICKED(IDC_Sync_Clock_Minutes_Done, OnCommand12) // escape from mm setting
    ON_BN_CLICKED(IDC_Sync_Clock_Hours_Done, OnCommand13) // escape from hh setting
    ON_BN_CLICKED(IDC_Sync_Clock_Days_Done, OnCommand14) // escape from days setting
    ON_BN_CLICKED(IDC_Sync_Clock_Stop, OnCommand15) // sync clock on second
    ON_BN_CLICKED(IDC_Sync_Clock_Up, OnCommand16) // units ++
    ON_BN_CLICKED(IDC_Sync_Clock_Down, OnCommand17) // units --
    ON_BN_CLICKED(IDC_Adjust_LED_Brightness, OnCommand18) // adjust LED display brightness
    ON_BN_CLICKED(IDC_LED_brightness_Up, OnCommand19) // units ++
    ON_BN_CLICKED(IDC_LED_brightness_Down, OnCommand20) // units --
    ON_BN_CLICKED(IDC_Adjust_LED_done, OnCommand21) // completed
    ON_BN_CLICKED(IDC_Adjust_Focus, OnCommand22) // adjust LED display brightness
    ON_BN_CLICKED(IDC_Focus_Speed_Up, OnCommand23) // units ++
    ON_BN_CLICKED(IDC_Focus_Speed_Down, OnCommand24) // units --
    ON_BN_CLICKED(IDC_Adjust_Focus_done, OnCommand25) // focus completed
    //ON_BN_CLICKED(IDC_Focus_In, OnCommand26) // focus moves in
    ON_BN_CLICKED(IDC_Focus_Stop, OnCommand27) // focus stop/start
    //ON_BN_CLICKED(IDC_Focus_Out, OnCommand28) // focus moves out
    ON_BN_CLICKED(IDC_Focus_Speed_Up2, OnCommand30) // position units ++
    ON_BN_CLICKED(IDC_Focus_Speed_Down2, OnCommand31) // position units --
    ON_BN_CLICKED(IDC_Forward, OnCommand32) // focus moves forward
    //ON_BN_CLICKED(IDC_Focus_Stop2, OnCommand33) // focus stop/start
    ON_BN_CLICKED(IDC_Reverse, OnCommand34) // focus moves reverse
    ON_BN_CLICKED(IDC_Enable_Auto2, OnCommand35) // enable auto monitoring
    ON_BN_CLICKED(IDC_Disable_Auto2, OnCommand36) // disable auto monitoring

    ON_BN_CLICKED(IDC_CLEAR_LIST, OnClearList)
    //ON_BN_CLICKED(IDC_TX_CR_LF, OnTxCrLf)
    ON_BN_CLICKED(IDC_TX_HEX, OnTxHex)
    ON_BN_CLICKED(IDC_DTR, OnDtr)
    ON_BN_CLICKED(IDC_RTS, OnRts)
    ON_BN_CLICKED(IDC_RX_HEX, OnRxHex)
    ON_BN_CLICKED(IDC_PORT, OnPort)
```

```

//}}AFX_MSG_MAP

ON_BN_CLICKED(IDC_Enable_Auto_Clear, &CRs232aDlg::OnBnClickedEnable_Auto_Clear)
ON_BN_CLICKED(IDC_Enable_Auto_Clear_Off, &CRs232aDlg::OnBnClickedEnable_Auto_Clear_Off)
ON_BN_CLICKED(IDC_Refractor_power_mains, &CRs232aDlg::OnBnClickedRefractorpowermains) // relay for 220v mains ✓
ON_BN_CLICKED(IDC_Microcontroller_power_supplies, &CRs232aDlg::OnBnClickedMicrocontrollerpowersupplies) ✓
// relay for control Microcontroller power supplies (2x24 Volt DC regulated)
ON_BN_CLICKED(IDC_Focus_power_supply, &CRs232aDlg::OnBnClickedFocuspowersupply)// relay for focus power ✓
supply (12 volt DC regulated)
ON_BN_CLICKED(IDC_Night_Lamp, &CRs232aDlg::OnBnClickedNightLamp) // relay for night lamp (red)
ON_BN_CLICKED(IDC_Auxiliary_Power, &CRs232aDlg::OnBnClickedAuxiliaryPower) // power to control panel 12 Volt socket ✓
ON_BN_CLICKED(IDC_CCD_Power, &CRs232aDlg::OnBnClickedCCDPower) // power to CCD
ON_BN_CLICKED(IDC_Enable, &CRs232aDlg::OnBnClickedEnable) // 12 volt relay rack
ON_BN_CLICKED(IDC_Spare_ON, &CRs232aDlg::OnBnClickedSpareON) //spare at this time
//ON_NOTIFY(NM_CUSTOMDRAW, IDC_PROGRESS1, &CRs232aDlg::Positionstatus)
ON_BN_CLICKED(IDC_Focus_Position_Up2, &CRs232aDlg::OnBnClickedFocusPositionUp2)
ON_BN_CLICKED(IDC_Focus_Position_Down2, &CRs232aDlg::OnBnClickedFocusPositionDown2)
ON_NOTIFY(DTN_DATETIMECHANGE, IDC_DATETIMEPICKER1, &CRs232aDlg::OnDtnDatetimechangeDatetimerpicker1)
ON_NOTIFY(DTN_DATETIMECHANGE, IDC_DATETIMEPICKER2, &CRs232aDlg::OnDtnDatetimechangeDatetimerpicker2)
END_MESSAGE_MAP()

```

```

BOOL CRs232aDlg::OnInitDialog()

```

```

{
    CDialog::OnInitDialog();
    SetIcon(m_hIcon, TRUE);
    SetIcon(m_hIcon, FALSE);

    VERIFY(mPin1.SubclassDlgItem(IDC_PIN1,this));
    VERIFY(mPin2.SubclassDlgItem(IDC_PIN2,this));
    VERIFY(mPin3.SubclassDlgItem(IDC_PIN3,this));
    VERIFY(mPin4.SubclassDlgItem(IDC_PIN4,this));
    VERIFY(mPin5.SubclassDlgItem(IDC_PIN5,this));
    VERIFY(mPin6.SubclassDlgItem(IDC_PIN6,this));
    VERIFY(mPin7.SubclassDlgItem(IDC_PIN7,this));
    VERIFY(mPin8.SubclassDlgItem(IDC_PIN8,this));
    VERIFY(mPin9.SubclassDlgItem(IDC_PIN9,this));
    VERIFY(mDTR.SubclassDlgItem(IDC_DTR,this));
    VERIFY(mRTS.SubclassDlgItem(IDC_RTS,this));

    VERIFY(mP_ON_1.SubclassDlgItem(IDC_P_ON_1,this));
    VERIFY(mP_ON_2.SubclassDlgItem(IDC_P_ON_2,this));
    VERIFY(mP_ON_3.SubclassDlgItem(IDC_P_ON_3,this));
    VERIFY(mP_ON_4.SubclassDlgItem(IDC_P_ON_4,this));
    VERIFY(mP_ON_5.SubclassDlgItem(IDC_P_ON_5,this));
    VERIFY(mP_ON_6.SubclassDlgItem(IDC_P_ON_6,this));
    VERIFY(mP_ON_7.SubclassDlgItem(IDC_P_ON_7,this));
    VERIFY(mP_ON_8.SubclassDlgItem(IDC_P_ON_8,this));

    VERIFY(mP_F_1.SubclassDlgItem(IDC_F_ON_1,this));
    VERIFY(mP_F_2.SubclassDlgItem(IDC_F_ON_2,this));
    VERIFY(mP_F_3.SubclassDlgItem(IDC_F_ON_3,this));
    VERIFY(mP_F_4.SubclassDlgItem(IDC_F_ON_4,this));
    VERIFY(mP_F_5.SubclassDlgItem(IDC_F_ON_5,this));
    VERIFY(mP_F_6.SubclassDlgItem(IDC_F_ON_6,this));
    VERIFY(mP_F_7.SubclassDlgItem(IDC_F_ON_7,this));
    VERIFY(mP_F_8.SubclassDlgItem(IDC_F_ON_8,this));
    VERIFY(mP_F_9.SubclassDlgItem(IDC_F_ON_9,this));
    VERIFY(mP_F_10.SubclassDlgItem(IDC_F_ON_10,this));
    VERIFY(mP_F_17.SubclassDlgItem(IDC_F_ON_17,this)); // focus 0 dc
    VERIFY(mP_F_18.SubclassDlgItem(IDC_F_ON_18,this)); // focus toggle on/off
    VERIFY(mP_F_22.SubclassDlgItem(IDC_F_ON_22,this)); // in
    VERIFY(mP_F_23.SubclassDlgItem(IDC_F_ON_23,this)); // out
    VERIFY(mRefractor.SubclassDlgItem(IDC_Refractor,this)); // out

```

```

mx_comport=0; // COM1 per default
ComStatusOld = 0xFF;

```

```

    bool bComRetValue;
    bComRetValue = m_commctrl.SetPortOpen(mx_comport);

    if(bComRetValue == false) SetWindowText("Failed to open RS232 port!");
    else TextNew();

    LoadComBitmaps();
    LoadRelayBitmaps();
    LoadFocusBitmaps();
    UpdateData(FALSE);
    return true;
}

void CRs232aDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this);

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        dc.DrawIcon(x, y, m_hIcon);
        //GetCommCtrl()->ClosePort(mx_comport);

    }
    else
    {
        CDialog::OnPaint();
    }
}

HCURSOR CRs232aDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

BOOL CRs232aDlg::PreTranslateMessage(MSG* pMsg)
{
    if(pMsg->message == WM_KEYDOWN)
    {
        if(pMsg->wParam==VK_ESCAPE) return TRUE;
        if(pMsg->wParam==VK_RETURN) return TRUE;
    }
    if( IsDialogMessage( pMsg ) )
        return TRUE;
    else
        return CDialog::PreTranslateMessage(pMsg);
}

void CRs232aDlg::OnBaud110() { SetBaud(110); }
void CRs232aDlg::OnBaud300() { SetBaud(300); }
void CRs232aDlg::OnBaud600() { SetBaud(600); }
void CRs232aDlg::OnBaud1200() { SetBaud(1200); }
void CRs232aDlg::OnBaud2400() { SetBaud(2400); }
void CRs232aDlg::OnBaud4800() { SetBaud(4800); }
void CRs232aDlg::OnBaud9600() { SetBaud(9600); }
void CRs232aDlg::OnBaud19200() { SetBaud(19200); }
void CRs232aDlg::OnBaud57600() { SetBaud(57600); }
void CRs232aDlg::OnBaud115200() { SetBaud(115200); }
void CRs232aDlg::OnCommand() { SystemLED = !SystemLED;
    if (!SystemLED) SetCommand("0");
    else SetCommand("1");
}

```

```

    } // toggle system LED
void CRs232aDlg::OnCommand2() { SetCommand("i"); } // scan I2C
void CRs232aDlg::OnCommand3() { ResetCard(0); } // software reset of MEGA2560 card, to do power down of
    system first

// enable Auto monitoring of Refractor
void CRs232aDlg::OnCommand4()
{
    if (focusloop) close_focus();
    if (LEDLoop) close_LED();
    if (clockloop) close_clock();
    SetCommand("A");
    InLoop = true;
    auto_loop = true;
    deactivate_clock_sync(); // not allowed while busy
    deactivate_LED_set();
    deactivate_focus();
    SetWindowText("Beam expander auto monitoring is ENABLED");
}

void CRs232aDlg::OnCommand5()
{
    if (auto_loop)
    {
        SetCommand("Q");
        InLoop = false;
        auto_loop = false;
        activate_clock_sync();
        activate_LED_set();
        activate_focus();

        // disable Auto monitoring of Refractor
        SetWindowText("Beam expander auto monitoring is DISABLED");
    }
    else {SetWindowText("Beam expander auto monitoring is not ENABLED");}
}

void CRs232aDlg::OnCommand6()
{
    SetCommand("Z");
    SetWindowText("Synchronizing card clock with PC clock");
    // activate_clock_sync();
    deactivate_LED_set();
    deactivate_focus();
    InLoop = true;
    clockloop = true;
    SetWindowText("Clock synchronisation ENABLED");
} // disable Auto monitoring of Refractor

void CRs232aDlg::OnCommand7() { if(clockloop){
    SetCommand("s");
    SetWindowText("Set seconds");
} else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand8() { if(clockloop){
    SetCommand("m");
    SetWindowText("Set minutes");
} else {SetWindowText("Clock synchronisation NOT ENABLED");}
} // minutes

void CRs232aDlg::OnCommand9() { if(clockloop){
    SetCommand("h");
    SetWindowText("Set hours");
} else {SetWindowText("Clock synchronisation NOT ENABLED");}
} // hours

void CRs232aDlg::OnCommand10() {if(clockloop){
    SetCommand("D");
    SetWindowText("Set days");} // days
else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand11() {if(clockloop)
{ SetCommand("f");
    SetWindowText("Set seconds done");} // ss done
}

```

```

        else {SetWindowText("Clock synchronisation NOT ENABLED");}
    }

void CRs232aDlg::OnCommand12() { if(clockloop)
    {SetCommand("f");
    SetWindowText("Set minutes done");} // mm done
    else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand13() { if(clockloop)
    {SetCommand("f");
    SetWindowText("Set hours done");} // hh done
    else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand14() { if(clockloop)
    {SetCommand("f");
    SetWindowText("Set days done");} // days done
    else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand15()
    { if(clockloop)
        { SetCommand("F");
        SetWindowText("Click to sync on second");
        InLoop = false;
        clockloop = false;
        activate_LED_set();
        activate_focus();
        SetWindowText("Clock synchronisation DISABLED");
        } // sync and done
    else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand16() { if(clockloop)
    {SetCommand("u");
    SetWindowText("Add units");} // ++
    else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand17() { if(clockloop)
    {SetCommand("d");
    SetWindowText("Subtract units");} // --
    else {SetWindowText("Clock synchronisation NOT ENABLED");}
}

void CRs232aDlg::OnCommand18() {SetCommand("B");
    SetWindowText("Adjusting LED readout brightness");
    deactivate_clock_sync();
    deactivate_focus();
    InLoop = true;
    LEDLoop = true;
    SetWindowText("Clock synchronisation DISABLED");
} // sync and done

void CRs232aDlg::OnCommand19() { if(LEDDLoop)
    {SetCommand("u"); SetWindowText("Add units");} // ++
    else {SetWindowText("LED adjustment NOT ENABLED");}
}

void CRs232aDlg::OnCommand20() { if(LEDDLoop)
    {SetCommand("d");
    SetWindowText("Subtract units");} // --
    else {SetWindowText("LED adjustment NOT ENABLED");}
}

void CRs232aDlg::OnCommand21()
    {if(LEDDLoop)
        {SetCommand("f");
        SetWindowText("LED brightness adjustment done");
        activate_clock_sync();
        activate_focus();
        InLoop = false;
        LEDLoop = false;
    }
}

```

```

        SetWindowText("Clock synchronisation ENABLED");
    }    // LED adjustment done
    else {SetWindowText("LED adjustment NOT ENABLED");}
}

void CRs232aDlg::OnCommand22()    { if(FocusPowerSupply_ON)
    {SetCommand("C");
    deactivate_clock_sync();
    deactivate_LED_set();
    deactivate_position_down();
    focusloop = true;
    focus = 0;
    SetWindowText("Focus Adjust ENABLED");
    InLoop = true;
    LoadFocusBitmaps();
    Invalidate();
    UpdateData(true);
    } else{
        SetWindowText("First enable focus power supply!");}
    }    // done

void CRs232aDlg::OnCommand23() { if (focusloop)
    {SetCommand("u");
    focus++;
    if(focus > 10) focus = 10;
    LoadFocusBitmaps();
    Invalidate();
    SetWindowText("Add focus speed units");}
    else{
        SetWindowText("First adjust focus position!"); }
    }

void CRs232aDlg::OnCommand24() { if (focusloop)
    {SetCommand("d");
    focus--;
    if(focus < 0) focus = 0;
    LoadFocusBitmaps();
    Invalidate();
    SetWindowText("Subtract focus speed units");
    }    // --
    else{
        SetWindowText("First adjust focus position!"); }
    }

void CRs232aDlg::OnCommand25() { if(focusloop)
    { if(FocusPowerSupply_ON)
        {SetCommand("f");
        activate_position_down();
        activate_position_up();
        activate_LED_set();
        activate_clock_sync();
        InLoop = false;
        focusloop = false;
        focus = 0;
        //position = 0;
        focus_toggle = false;
        reverse = false;
        forward = true;
        LoadFocusBitmaps();
        Invalidate();
        SetWindowText("Clock synchronisation ENABLED");
        SetWindowText("LED brightness set ENABLED");
        SetWindowText("Focus adjustment done");
        f.Close();
        } else{
            SetWindowText("First enable focus power supply!");}
        }    // focus adjustment done
    else{
        SetWindowText("First enable focus!");}
    }

void CRs232aDlg::OnCommand27() { if (focusloop)
    {
        SetCommand("s");
    }
}

```



```

        SetWindowText("Focus toggled");
        focus_toggle = !focus_toggle;
        LoadFocusBitmaps();
        f.WriteString(m_strEditRead);
        if (focus_toggle==true) {
            delay(1000*focus);
            OnRx();}
        else {delay(100); OnRx();}
        if (position==pulse){ focus_toggle==false; LoadFocusBitmaps();
SetWindowText("Focus position has been reached, select new position");}

        UpdateData(TRUE);
        Invalidate();

    }
    else{
        SetWindowText("First enable focus!");}

}

void CRs232aDlg::OnCommand30() { if (focusloop)
    {SetCommand("a");
    position++;
    if(position > 1400) position = 1400;
    Invalidate();
    SetWindowText("Add focus position units");
    }
    else{
        SetWindowText("First enable focus!"); }
}

void CRs232aDlg::OnCommand31() { if (focusloop)
    {SetCommand("b");
    position--;
    if (position < 0) position = 0;
    Invalidate();
    SetWindowText("Subtract focus position units");
    }
    else{
        SetWindowText("First enable focus!"); }
}

void CRs232aDlg::OnCommand32() { if (focusloop)
    {SetCommand("c");
    forward = true;
    focus_toggle = false;
    reverse = false;
    LoadFocusBitmaps();
    Invalidate();
    SetWindowText("Forward");
    deactivate_position_down();
    activate_position_up();
    }
    else
    {SetWindowText("First enable focus!");}
}

void CRs232aDlg::OnCommand34() { if (focusloop)
    {SetCommand("e");
    if (reverse = true);
    focus_toggle = false;
    forward = false;
    LoadFocusBitmaps();
    Invalidate();
    SetWindowText("Reverse");
    deactivate_position_up();
    activate_position_down();
    }
    else{
        SetWindowText("First enable focus!");}
}

void CRs232aDlg::OnCommand35() { if (focusloop) close_focus();
    if (LEDLoop) close_LED();
    if (clockloop) close_clock();
}

```

```

        SetCommand("R");
        InLoop = true;
        auto_loop = true;
        deactivate_clock_sync(); // not allowed while busy
        deactivate_LED_set();
        deactivate_focus();
        SetWindowText("Beam expander auto focusing is ENABLED");
    }

void CRs232aDlg::OnCommand36()
{
    if (auto_loop)
    {
        SetCommand("S");
        InLoop = false;
        auto_loop = false;
        activate_clock_sync();
        activate_LED_set();
        activate_focus();

        // disable Auto monitoring of Refractor
        SetWindowText("Beam expander auto focusing is DISABLED");
    }
    else {SetWindowText("Beam expander auto focusing is not ENABLED");}
}

void CRs232aDlg::OnBnClickedFocusPositionUp2()
{
    if (focusloop)
    {
        {SetCommand("o");
        10*position++;
        if(position > 1400) position = 1400;
        Invalidate();
        SetWindowText("Add focus position units");
        }
    }
    else{
        SetWindowText("First enable focus!"); }
}

void CRs232aDlg::OnBnClickedFocusPositionDown2()
{
    if (focusloop)
    {
        {SetCommand("G");
        10*position--;
        if (position < 0) position = 0;
        Invalidate();
        SetWindowText("Subtract focus position units");
        }
    }
    else{
        SetWindowText("First enable focus!"); }
}

void CRs232aDlg::SetBaud(DWORD dnewBAUD)
{
    dBAUD = dnewBAUD;
    ComStatus = GetCommCtrl()->SetupBaudrate(dBAUD,mx_comport); // set port
    LoadComBitmaps();
    TextNew();
    Invalidate();
}

//*****

void CRs232aDlg::OnDtr()
{
    int xx=1;
    if( ComStatus & BIT_DTR) xx = 0;
    ComStatus = GetCommCtrl() -> SetComProperties( xx, mx_comport);
    LoadComBitmaps();
    Invalidate();
}

void CRs232aDlg::OnRts()

```

```
{
int xx=11;
if( ComStatus & BIT_RTS) xx = 10;
ComStatus = GetCommCtrl() -> SetComProperties( xx, mx_comport);
LoadComBitmaps();
Invalidate();
}

void CRs232aDlg::OnTx()
{
UpdateData(TRUE);
GetCommCtrl()->SendCString(m_strTx,mx_comport); // set port
}

void CRs232aDlg::SetCommand(CString Command)
{
UpdateData(TRUE);
GetCommCtrl()->SendCString(Command,mx_comport); // set port
OnRx();
flush();
}

void CRs232aDlg::ResetCard(int task)
{int run;
SetWindowText("Powering down.....wait"); delay(800);
run=3;
SPARE_RELAY(); // controlled shutdown of systems
ENABLE();
CCD_CAMERA();
AUX_POWER();
NIGHT_LAMP();
FOCUS_POWER();
Microcontroller_POWER();
MAINS_POWER();
close_focus();
close_LED();
close_clock();
if (select_task==0)
{while (run-->0) OnDtr();
SetWindowText("Control card reset");
} else
{SetWindowText("Powered down...card not reset");
delay(800);
}
}

void CRs232aDlg::SPARE_RELAY()
{
SetWindowText("Spare relay OFF"); SetCommand("z"); delay(500);
Spare_ON = false;
}

void CRs232aDlg::ENABLE()
{
SetWindowText("Enable system OFF"); SetCommand("y"); delay(500);
Enable_ON = false;
}

void CRs232aDlg::CCD_CAMERA()
{
SetWindowText("CCD camera OFF"); SetCommand("x"); delay(500);
CCD_ON = false;
}

void CRs232aDlg::AUX_POWER()
{
SetWindowText("AuxiliaryPower OFF"); SetCommand("w"); delay(500);
AuxiliaryPower_ON = false;
}

void CRs232aDlg::NIGHT_LAMP()
{
SetWindowText("Night Lamp OFF");SetCommand("u"); delay(500);
NightLamp_ON = false;
}
```

```

}

void CRs232aDlg::FOCUS_POWER()
{
SetWindowText("Focus power supply OFF"); SetCommand("t"); delay(500);
FocusPowerSupply_ON = false;
}

void CRs232aDlg::Microcontroller_POWER()
{
SetWindowText("Microcontroller power supplies OFF"); SetCommand("s"); delay(500);
MicrocontrollerPowerSupplies_ON = false;
}

void CRs232aDlg::MAINS_POWER()
{
SetWindowText("Beam expander mains power OFF (220 V)"); SetCommand("r"); delay(500);
Mains_ON = false; LoadRelayBitmaps(); Invalidate();
}

/*
void CRs232aDlg::OnTxCrLf() // send command with carriage return and line feed (not used now)
{
CString str;
UpdateData(TRUE);
str = m_strTx;
str += (char) CR; str += (char) LF;
GetCommCtrl()->SendCString(str,mx_comport); // set port
}*/

void CRs232aDlg::OnTxHex()
{
unsigned char chx;
unsigned char chx1,chx2;
unsigned char txbuf[1000];
int iTxPosition;

UpdateData(TRUE); // from window -> variable
iTxPosition = 0;
int laenge = focus*m_strHex.GetLength();
int x1=0;
while(x1 < 1000)
{
chx = m_strHex.GetAt(x1++);
if (x1 >= laenge);
if(chx == ' ') chx = m_strHex.GetAt(x1++); // if SPACE
if (x1 >= laenge);

chx1=0;
if(chx >= '0' && chx <= '9') chx1 = chx - '0';
if(chx >= 'A' && chx <= 'F') chx1 = chx - 'A' + 0x0A;
if(chx >= 'a' && chx <= 'f') chx1 = chx - 'a';

chx = m_strHex.GetAt(x1++);
chx2 = 0xFF;
if(chx >= '0' && chx <= '9') chx2 = chx - '0';
if(chx >= 'A' && chx <= 'F') chx2 = chx - 'A' + 0x0A;
if(chx >= 'a' && chx <= 'f') chx2 = chx - 'a';

if(chx2 = 0xFF)txbuf[iTxPosition++] = chx1 * 16 + chx2;
}
GetCommCtrl()->SendXBytes(txbuf,iTxPosition,mx_comport); // set port
}

void CRs232aDlg::OnRx()
{
UpdateData(TRUE);

CString str, str1;
unsigned char *ptr1;
int anz; //= m_strEditRead.GetLength();

```

```

if(auto_clear) {OnClearList();}
else auto_clear = false;

ptr1 = RxBuffer;
str.Empty();
anz = GetCommCtrl()->ReadRs232Input(&ptr1,mx_comport);    // Reads a block from the COM port and pops it into
the provided buffer
str += str1;

while(anz--)
{
    if(isprint(*ptr1)) { str += *ptr1; }
    else { str1.Format("\r\n%s",str += *ptr1); //
    }
    ptr1++;
}

//{m_strEditRead += (char) CR;    m_strEditRead += (char) LF;
m_strEditRead += str;
UpdateData(FALSE);

}

void CRs232aDlg::OnRxHex()
{
    CString str,str1,strList;
    unsigned char *ptr1;
    //ptr1 = RxBuffer;

    int xx,x1,xbytes;
    str.Empty();
    xx = GetCommCtrl()->ReadRs232Input(&ptr1,mx_comport);

    if(!xx) return;

    strList.Empty();
    x1=0;
    xbytes = 0;
    while(xx--)
    {
        str1.Format(_T("%02x"),str);
        strList += str1;
        xbytes++;
        if(xbytes == 16)
        {
            m_strEditRead += strList;
            m_strEditRead += (char) CR;
            m_strEditRead += (char) LF;
            xbytes=0; strList.Empty();
        }
    }

    m_strEditRead += strList;
    m_strEditRead += (char) CR;
    m_strEditRead += (char) LF;

    UpdateData(FALSE);
}

void CRs232aDlg::LoadComBitmaps()
{
    if(ComStatusOld == ComStatus) return;
    ComStatusOld = ComStatus;

    if(ComStatus & BIT_DCD) bmPin1.LoadBitmaps(IDB_PIN1_ON);
    else bmPin1.LoadBitmaps(IDB_PIN1_OFF);

    bmPin2.LoadBitmaps(IDB_PIN2);
    bmPin3.LoadBitmaps(IDB_PIN3);
}

```

```

    if(ComStatus & BIT_DTR)
    { bmPin4.LoadBitmaps(IDB_PIN4_ON);
      cbDTR.SetWindowText("DTR OFF");
    }
    else
    {
      cbDTR.SetWindowText("DTR ON");
      bmPin4.LoadBitmaps(IDB_PIN4_OFF);
    }

    bmPin5.LoadBitmaps(IDB_PIN5);    // GND

    if(ComStatus & BIT_DSR) bmPin6.LoadBitmaps(IDB_PIN6_ON);
    else bmPin6.LoadBitmaps(IDB_PIN6_OFF);

    if(ComStatus & BIT_RTS)
    { bmPin7.LoadBitmaps(IDB_PIN7_ON);
      cbRTS.SetWindowText("RTS OFF");
    }
    else
    {
      cbRTS.SetWindowText("RTS ON");
      bmPin7.LoadBitmaps(IDB_PIN7_OFF);
    }

#define BITx_CTS 0x10

    if(ComStatus & BITx_CTS) bmPin8.LoadBitmaps(IDB_PIN8_ON);
    else bmPin8.LoadBitmaps(IDB_PIN8_OFF);

    if(ComStatus & BIT_RI ) bmPin9.LoadBitmaps(IDB_PIN9_ON);
    else bmPin9.LoadBitmaps(IDB_PIN9_OFF);

    bmPin9.LoadBitmaps(IDB_PIN9);
}

void CRs232aDlg::LoadRelayBitmaps()
{
    if(Mains_ON) bmP_ON_1.LoadBitmaps(IDB_P_ON_1);
    else bmP_ON_1.LoadBitmaps(IDB_P_OFF_1);

    if(MicrocontrollerPowerSupplies_ON) bmP_ON_2.LoadBitmaps(IDB_P_ON_2);
    else bmP_ON_2.LoadBitmaps(IDB_P_OFF_2);

    if(FocusPowerSupply_ON) bmP_ON_3.LoadBitmaps(IDB_P_ON_3);
    else bmP_ON_3.LoadBitmaps(IDB_P_OFF_3);

    if(NightLamp_ON) bmP_ON_4.LoadBitmaps(IDB_P_ON_4);
    else bmP_ON_4.LoadBitmaps(IDB_P_OFF_4);

    if(AuxiliaryPower_ON) bmP_ON_5.LoadBitmaps(IDB_P_ON_5);
    else bmP_ON_5.LoadBitmaps(IDB_P_OFF_5);

    if(CCD_ON) bmP_ON_6.LoadBitmaps(IDB_P_ON_6);
    else bmP_ON_6.LoadBitmaps(IDB_P_OFF_6);

    if(Enable_ON) bmP_ON_7.LoadBitmaps(IDB_P_ON_7);
    else bmP_ON_7.LoadBitmaps(IDB_P_OFF_7);

    if(Spare_ON) bmP_ON_8.LoadBitmaps(IDB_P_ON_8);
    else bmP_ON_8.LoadBitmaps(IDB_P_OFF_8);

    /*
    if(ComStatus & BIT_DTR)
    { bmPin4.LoadBitmaps(IDB_PIN4_ON);
      cbDTR.SetWindowText("DTR OFF");
    }
    else
    {
      cbDTR.SetWindowText("DTR ON");
      bmPin4.LoadBitmaps(IDB_PIN4_OFF);
    }
    */
}

```

```

}

void CRs232aDlg::LoadFocusBitmaps()
{
    if(focus == 1) bmp_F_1.LoadBitmaps(IDB_F_ON_1);
    else bmp_F_1.LoadBitmaps(IDB_F_OFF_1);

    if(focus == 2) bmp_F_2.LoadBitmaps(IDB_F_ON_2);
    else bmp_F_2.LoadBitmaps(IDB_F_OFF_2);

    if(focus == 3) bmp_F_3.LoadBitmaps(IDB_F_ON_3);
    else bmp_F_3.LoadBitmaps(IDB_F_OFF_3);

    if(focus == 4) bmp_F_4.LoadBitmaps(IDB_F_ON_4);
    else bmp_F_4.LoadBitmaps(IDB_F_OFF_4);

    if(focus == 5) bmp_F_5.LoadBitmaps(IDB_F_ON_5);
    else bmp_F_5.LoadBitmaps(IDB_F_OFF_5);

    if(focus == 6) bmp_F_6.LoadBitmaps(IDB_F_ON_6);
    else bmp_F_6.LoadBitmaps(IDB_F_OFF_6);

    if(focus == 7) bmp_F_7.LoadBitmaps(IDB_F_ON_7);
    else bmp_F_7.LoadBitmaps(IDB_F_OFF_7);

    if(focus == 8) bmp_F_8.LoadBitmaps(IDB_F_ON_8);
    else bmp_F_8.LoadBitmaps(IDB_F_OFF_8);

    if(focus == 9) bmp_F_9.LoadBitmaps(IDB_F_ON_9);
    else bmp_F_9.LoadBitmaps(IDB_F_OFF_9);

    if(focus == 10) bmp_F_10.LoadBitmaps(IDB_F_ON_10);
    else bmp_F_10.LoadBitmaps(IDB_F_OFF_10);

    // if(focus == 11) bmp_F_11.LoadBitmaps(IDB_F_ON_11);
    // else bmp_F_11.LoadBitmaps(IDB_F_OFF_11);

    // if(focus == 12) bmp_F_12.LoadBitmaps(IDB_F_ON_12);
    // else bmp_F_12.LoadBitmaps(IDB_F_OFF_12);

    //if(focus == 13) bmp_F_13.LoadBitmaps(IDB_F_ON_13);
    //else bmp_F_13.LoadBitmaps(IDB_F_OFF_13);

    //if(focus == 14) bmp_F_14.LoadBitmaps(IDB_F_ON_14);
    //else bmp_F_14.LoadBitmaps(IDB_F_OFF_14);

    //if(focus == 15) bmp_F_15.LoadBitmaps(IDB_F_ON_15);
    //else bmp_F_15.LoadBitmaps(IDB_F_OFF_15);

    //if(focus == 16) bmp_F_16.LoadBitmaps(IDB_F_ON_16);
    //else bmp_F_16.LoadBitmaps(IDB_F_OFF_16);

    if(focus == 0) bmp_F_17.LoadBitmaps(IDB_F_ON_17);
    else bmp_F_17.LoadBitmaps(IDB_F_OFF_17);

    if(focus_toggle == TRUE) bmp_F_18.LoadBitmaps(IDB_F_ON_18);
    else bmp_F_18.LoadBitmaps(IDB_F_OFF_18);

    if(forward == TRUE) bmp_F_22.LoadBitmaps(IDB_F_ON_22);
    else {bmp_F_22.LoadBitmaps(IDB_F_OFF_22);
        bmp_F_23.LoadBitmaps(IDB_F_ON_23);
    }

    if(reverse == TRUE) bmp_F_23.LoadBitmaps(IDB_F_ON_20);
    else {bmp_F_23.LoadBitmaps(IDB_F_OFF_23);
        bmp_F_22.LoadBitmaps(IDB_F_ON_22);
    }
    bmp_Refractor.LoadBitmaps(IDC_Refractor);
}

void CRs232aDlg::OnClearList()
{
    m_strEditRead.Empty();
}

```

```
UpdateData(FALSE);
}
```

```
void CRs232aDlg::OnPort()
{
    GetCommCtrl()->ClosePort(mx_comport);
    //if(mx_comport==0) mx_comport++;
    mx_comport++;
    if(mx_comport==10) mx_comport=0; //max 10 for now
    TextNew();
    GetCommCtrl()->SetPortOpen(mx_comport);
}
```

```
void CRs232aDlg::TextNew()
{
    CString str;
    str.Format("RS232 --- COM%d Baud=%d      Beam Expander Interface and Control-----Sphumelele Ndlovu
October 2013",mx_comport+1,dBAUD);
    SetWindowText(str);
    Invalidate();
}
```

```
void CRs232aDlg::OnBnClickedEnable_Auto_Clear()
{
    auto_clear = true;
    SetWindowText("List screen is auto cleared");
    Invalidate();
    flush();
}
```

```
void CRs232aDlg::OnBnClickedEnable_Auto_Clear_Off()
{
    auto_clear=false;
    SetWindowText("List screen reset to manual clear");
    Invalidate();
    flush();
}
```

```
void CRs232aDlg::SetTime()
{
    //boolean set_time = true;
    CTime theTime = CTime::GetCurrentTime();
    CString s = theTime.Format( "%A, %B %d, %Y, %H: %M: %S" );
    // CString s = theTime.Format( "%H %M %S" );
    s.Format(s);
    //while (set_time) SetWindowText(s); } for debugging only
    m_strTx = s;
    //m_strTx = s+"x";
    SetWindowText(m_strTx);
}
```

```
void CRs232aDlg::OnBnClickedRefractorpowermains() // mains (220 V) power
{
    if(!InLoop){
        Mains_ON = !Mains_ON;
        LoadRelayBitmaps();
        Invalidate();
        if (Mains_ON) {SetWindowText("Beam expander mains power ON (220 V)"); SetCommand("2");
        auto_clear = true;}
        else
        {ResetCard(1); //soft off of units
        SetWindowText("Beam expander mains power OFF (220 V)");SetCommand("r");
        MicrocontrollerPowerSupplies_ON = false;
        FocusPowerSupply_ON = false;
        NightLamp_ON = false;
        AuxiliaryPower_ON = false;
        CCD_ON = false;
        Enable_ON = false;
        Spare_ON = false;
        LoadRelayBitmaps();
    }
}
```



```

    //auto_clear = false;
    if(auto_clear) {OnClearList();}
else auto_clear = false;
    }
    CFileDialog *pOpenDlg= new CFileDialog(FALSE, ".txt",
    NULL, OFN_OVERWRITEPROMPT, "Text Files (*.txt)|*.txt|All Files (*.*)|*.*||");
    if(IDOK == pOpenDlg->DoModal())
    {

        CString strFileName = pOpenDlg->GetPathName();
        if(f.Open(strFileName, CFile::modeWrite | CFile::modeCreate))
        {
            //f.WriteString(m_strEditRead + _T("\r\n"));
        }
        //f.Close();
        UpdateData(true);
    }
    delete pOpenDlg;
    //pOpenDlg=NULL;

    } else {SetWindowText("Operation not permitted...complete previous task!");}
}

// Microcontroller power supplies 12Volt and 2* 24 Volt
void CRs232aDlg::OnBnClickedMicrocontrollerpowersupplies()
{
    if(!InLoop){
        if(Mains_ON) {
            MicrocontrollerPowerSupplies_ON = !MicrocontrollerPowerSupplies_ON;
            LoadRelayBitmaps();
            FocusPowerSupply_ON = false;
            SetCommand("t");
            Invalidate();
            if (MicrocontrollerPowerSupplies_ON) {SetWindowText("Microcontroller power supplies ON (2 x 24 Volt and 1 x 12 Volt)");
                SetCommand("3");} else
            {SetWindowText("Microcontroller power supplies OFF"); SetCommand("s"); }
        } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}
    } else {SetWindowText("Operation not permitted...complete previous task!");}
}

// relay for focus power supply (12 volt DC regulated)
void CRs232aDlg::OnBnClickedFocuspowersupply()
{
    if(!InLoop){
        if(Mains_ON) {
            if (MicrocontrollerPowerSupplies_ON){
                FocusPowerSupply_ON = !FocusPowerSupply_ON;
                LoadRelayBitmaps();
                Invalidate();
                if (FocusPowerSupply_ON) {SetWindowText("Focus power supply ON (1 x 12 Volt)");SetCommand("4");
                } else
                {SetWindowText("Focus power supply OFF"); SetCommand("t"); }
            } else {SetWindowText("First switch on microcontroller power");}
        } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}

    } else {SetWindowText("Operation not permitted...complete previous task!");}
}

void CRs232aDlg::OnBnClickedNightLamp() // relay for night lamp (red)
{
    if(!InLoop){
        if(Mains_ON) {
            NightLamp_ON = !NightLamp_ON;
            LoadRelayBitmaps();
            Invalidate();
            if (NightLamp_ON) {SetWindowText("Night Lamp ON");SetCommand("5");} else
            {SetWindowText("Night Lamp OFF"); SetCommand("u"); }
        } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}
    } else {SetWindowText("Operation not permitted...complete previous task!");}
}

void CRs232aDlg::OnBnClickedAuxiliaryPower() // power to control panel 12 Volt socket

```

```

{
    if(!InLoop){
        if(Mains_ON) {
            AuxiliaryPower_ON = !AuxiliaryPower_ON;
            LoadRelayBitmaps();
            Invalidate();
            if (AuxiliaryPower_ON) {SetWindowText("Auxiliary ON"); SetCommand("6"); } else
                {SetWindowText("AuxiliaryPower OFF"); SetCommand("w"); }
            } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}
        } else {SetWindowText("Operation not permitted...complete previous task!");}
    }
}

void CRs232aDlg::OnBnClickedCCDPower() // power to CCD
{
    if(!InLoop){
        if(Mains_ON) {
            CCD_ON = !CCD_ON;
            LoadRelayBitmaps();
            Invalidate();
            if (CCD_ON) {SetWindowText("CCD camera ON");SetCommand("7"); } else
                {SetWindowText("CCD camera OFF"); SetCommand("x"); }
            } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}
        } else {SetWindowText("Operation not permitted...complete previous task!");}
    }
}

void CRs232aDlg::OnBnClickedEnable()
    // enable system (switch on relay rack which connects dc motors to Microcontroller outputs etc.)
{
    if(!InLoop){
        if(Mains_ON) {
            Enable_ON = !Enable_ON;
            LoadRelayBitmaps();
            Invalidate();
            if (Enable_ON) {SetWindowText("Enable system ON");SetCommand("8"); } else
                {SetWindowText("Enable system OFF"); SetCommand("y"); }
            } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}
        } else {SetWindowText("Operation not permitted...complete previous task!");}
    }
}

void CRs232aDlg::OnBnClickedSpareON() //spare at this time
{
    if(!InLoop){
        if(Mains_ON) {
            Spare_ON = !Spare_ON;
            LoadRelayBitmaps();
            Invalidate();
            if (Spare_ON) {SetWindowText("Spare relay ON");SetCommand("9"); } else
                {SetWindowText("Spare relay OFF"); SetCommand("z"); }
            } else {SetWindowText("Main power is OFF! Select ""Beam Expander Main Power"" first. ");}
        } else {SetWindowText("Operation not permitted...complete previous task!");}
    }
}

void CRs232aDlg::delay(unsigned int mseconds)
{
    clock_t goal = mseconds + clock();
    while (goal > clock());
}

void CRs232aDlg::flush()
// pop buffer clean
{
    delay(100);
    OnRx();
}

void CRs232aDlg::activate_clock_sync()
{
    GetDlgItem(IDC_Sync_Clock)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Seconds)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Seconds_Done)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Minutes)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Minutes_Done)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Hours)->EnableWindow(TRUE);
}

```

```

    GetDlgItem(IDC_Sync_Clock_Hours_Done)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Days)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Days_Done)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Down)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Up)->EnableWindow(TRUE);
    GetDlgItem(IDC_Sync_Clock_Stop)->EnableWindow(TRUE);
    SetWindowText("Synchronising of clock enabled");
    delay(1000);
}

void CRs232aDlg::deactivate_clock_sync()
{
    GetDlgItem(IDC_Sync_Clock)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Seconds)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Seconds_Done)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Minutes)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Minutes_Done)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Hours)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Hours_Done)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Days)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Days_Done)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Down)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Up)->EnableWindow(FALSE);
    GetDlgItem(IDC_Sync_Clock_Stop)->EnableWindow(FALSE);
    SetWindowText("Synchronising of clock disabled");
    delay(1000);
}

void CRs232aDlg::activate_LED_set()
{
    GetDlgItem(IDC_Adjust_LED_Brightness)->EnableWindow(TRUE);
    GetDlgItem(IDC_LED_brightness_Down)->EnableWindow(TRUE);
    GetDlgItem(IDC_LED_brightness_Up)->EnableWindow(TRUE);
    GetDlgItem(IDC_Adjust_LED_done)->EnableWindow(TRUE);
    SetWindowText("LED brightness adjustment enabled");
    delay(1000);
}

void CRs232aDlg::deactivate_LED_set()
{
    GetDlgItem(IDC_Adjust_LED_Brightness)->EnableWindow(FALSE);
    GetDlgItem(IDC_LED_brightness_Down)->EnableWindow(FALSE);
    GetDlgItem(IDC_LED_brightness_Up)->EnableWindow(FALSE);
    GetDlgItem(IDC_Adjust_LED_done)->EnableWindow(FALSE);
    SetWindowText("LED brightness adjustment disabled");
    delay(1000);
}

void CRs232aDlg::activate_focus()
{
    GetDlgItem(IDC_Adjust_Focus)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Speed_Down)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Speed_Up)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Speed_Down2)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Speed_Up2)->EnableWindow(TRUE);
    GetDlgItem(IDC_Adjust_Focus_done)->EnableWindow(TRUE);
    //GetDlgItem(IDC_Focus_In)->EnableWindow(TRUE);
    //GetDlgItem(IDC_Focus_Out)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Stop)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_In2)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Out2)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Stop2)->EnableWindow(TRUE);
    SetWindowText("Focus adjustment enabled");
    delay(1000);
}

void CRs232aDlg::deactivate_focus()
{
    GetDlgItem(IDC_Adjust_Focus)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Speed_Down)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Speed_Up)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Speed_Down2)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Speed_Up2)->EnableWindow(FALSE);
    GetDlgItem(IDC_Adjust_Focus_done)->EnableWindow(FALSE);
    //GetDlgItem(IDC_Focus_In)->EnableWindow(FALSE);

```

```

    //GetDlgItem(IDC_Focus_Out)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Stop)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_In2)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Out2)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Stop2)->EnableWindow(FALSE);
    SetWindowText("Focus adjustment disabled");
    delay(1000);
}

void CRs232aDlg::close_focus()
{
    if (focusloop) // get out of focus loop
    {SetCommand("f");
    activate_clock_sync();
    activate_LED_set();
    InLoop = false;
    focusloop = false;
    focus = 0;
    focus_toggle = false;
    LoadFocusBitmaps();
    Invalidate();
    }
}

void CRs232aDlg::deactivate_position_up()
{
    GetDlgItem(IDC_Focus_Speed_Up2)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Position_Up2)->EnableWindow(FALSE);
    SetWindowText("Position up adjustment disabled");
    delay(10);
}

void CRs232aDlg::deactivate_position_down()
{
    GetDlgItem(IDC_Focus_Speed_Down2)->EnableWindow(FALSE);
    GetDlgItem(IDC_Focus_Position_Down2)->EnableWindow(FALSE);
    SetWindowText("Position down adjustment disabled");
    delay(10);
}

void CRs232aDlg::activate_position_up()
{
    GetDlgItem(IDC_Focus_Speed_Up2)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Position_Up2)->EnableWindow(TRUE);
    SetWindowText("Position up adjustment enabled");
    delay(10);
}

void CRs232aDlg::activate_position_down()
{
    GetDlgItem(IDC_Focus_Speed_Down2)->EnableWindow(TRUE);
    GetDlgItem(IDC_Focus_Position_Down2)->EnableWindow(TRUE);
    SetWindowText("Position down adjustment enabled");
    delay(10);
}

void CRs232aDlg::close_LED()
{
    SetCommand("f");
    InLoop = false;
    LEDLoop = false;
}

void CRs232aDlg::close_clock()
{
    SetCommand("F");
    InLoop = false;
    clockloop = false;
}

void CRs232aDlg::OnLbnSelchangeList1()
{
    // TODO: Add your control notification handler code here

```

```
}
```

```
void CRs232aDlg::OnDtnDatetimechangeDatetimestepicker1(NMHDR *pNMHDR, LRESULT *pResult)
{
    CTime ct;
    m_MonthCal1.GetTime(ct);
    m_strDate1.Format(_T("%02d:%02d:%2d"), ct.GetHour(), ct.GetMinute(), ct.GetSecond());
    UpdateData(FALSE);
    *pResult = 0;
}
```

```
void CRs232aDlg::OnDtnDatetimechangeDatetimestepicker2(NMHDR *pNMHDR, LRESULT *pResult)
{
    CTime ct;
    m_MonthCal2.GetTime(ct);
    m_strDate2.Format(_T("%02d/%02d/%2d"), ct.GetMonth(), ct.GetDay(), ct.GetYear());
    UpdateData(FALSE);
    *pResult = 0;
}
```

Appendix C

Here I provide the source code of the integrated software-based analytical model for the signal path efficiency of the HartRAO lunar laser ranger optical system. This is an essential tool that can ensure photon returns when ranging to the Moon. It enhances the current effort to determine highly accurate and precise Earth-Moon distances for various scientific purposes. The model calculates the expected number of returned photons by considering a number of parameters which affects the laser beam pulses as they traverse the atmosphere from the LLR telescope to the Moon and back to the telescope. This is achieved by modelling the effects of thermal and density fluctuations of the atmosphere on the apparent Earth-Moon range, atmospheric extinction, laser beam characteristics, optical path efficiencies and other factors on the number of returned photons. These factors affect the estimated (predicted by software) and actual (measured) number of returned photons for the HartRAO-LLR station.

```

// I Spumelele Ndlovu started working on my program to estimate the number of returned photons using the
// LinePlotTestDlg.cpp file in June 2014.
// The LinePlotTestDlg.cpp file is freely available online:read.pudn.com/downloads10/sourcecode/windows/
// control/39703/LinePlotTestDlg.cpp_.htm

#include <afxdisp.h>
#include "stdafx.h"
#include "LinePlotTest.h"
#include "LinePlotTestDlg.h"
#include "ctime"
#include "LinePlot.h"
#include "homedataDlg.h"
#include "MoonDataDlg.h"
#include "MoonMirrorsDlg.h"
#include "DayNightDlg.h"
#include "MoonPhaseDlg.h"
#include "LaserPathDlg.h"
#include "TelescopeDlg.h"
#include "BeamDiameterDlg.h"
#include "CAplication.h"
#include "CWorkbooks.h"
#include "CWorkbook.h"
#include "CWorksheets.h"
#include "CWorksheet.h"
#include "CRange.h"
#include "CloudCoverDlg.h"
#include "PhotonsData.h"
#include "PredictionsDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

boolean Beam = false;
boolean Turbulence = false;
boolean Photons = false;
boolean BandPandT = false;
boolean Visuals = false;

////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support

//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

////////////////////////////////////
////////////////////////////////////

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT

```

```

}
/////////////////////////////////////////////////////////////////

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{AFX_DATA_MAP(CAboutDlg)
    //}AFX_DATA_MAP
}
/////////////////////////////////////////////////////////////////

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    //}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////
// CLinePlotTestDlg dialog

CLinePlotTestDlg::CLinePlotTestDlg(CWnd* pParent /*=NULL*/)
: CDialog(CLinePlotTestDlg::IDD, pParent)
, m_strTime_Of_Flight(0)
, m_strDiameter(0)
, m_strEnergy(0)
, m_strRepetition(0)
, m_strPower(0)
, m_strWidth(0)
, m_strTxPhotons(0)
, m_Rx_Photon(0)
, m_Difference(0)
, m_SlantRange(0)
, m_wavelength(0)
, m_transmitter_gain(0)
, m_pointing_error(0)
, m_divergence(0)
, m_transmit_optics(0)
, m_Intensity(0)
, m_receive_optics(0)
, m_Quantum_Dect(0)
, m_Atmosphere(0)
, m_Cloud(0)
, m_Aperture(0)
, m_Cross_Section(0)
, m_ScrollBarInfo(_T(""))
, m_QuantumDe(false)
, m_Photon_Second(0)
, m_Local_Time(0)
, m_Clock_Time(0)
, m_direction(0)
, m_currentdate(COLEDateTime::GetCurrentTime())
, m_Julian_Date(0)
, m_Lp(0)
, m_M(0)
, m_Mp(0)
, m_D(0)
, m_F(0)
, m_T(0)
, m_Julian_Day(0)
, m_Local_Temp(_T(""))
, m_Local_Pressure(_T(""))
, m_Local_Humidity(_T(""))
, m_Local_WindS(_T(""))
, m_Local_WindD(_T(""))
, m_Local_SVP(_T(""))
, m_Local_PWV(_T(""))
, m_Local_AVP(_T(""))
, m_Local_DPT(_T(""))
{
    //{AFX_DATA_INIT(CLinePlotTestDlg)
    // NOTE: the ClassWizard will add member initialization here
    //}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32

```



```

    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    m_Min = 0;
    m_Max = 0;
    // m_distance = _T("");
}
////////////////////////////////////

void CLinePlotTestDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CLinePlotTestDlg)
    DDX_Control(pDX, ID_CMB_STYLE, m_cmbStyle);
    DDX_Control(pDX, ID_CMB_STYLE2, m_cmbTarget);
    DDX_Control(pDX, ID_CMB_STYLE3, m_cmbSystem);
    //DDX_Control(pDX, ID_LST_RECEIVED, m_lstReceived);
    //}}AFX_DATA_MAP
    DDX_Text(pDX, IDC_BeamDiameter_Edit, m_strDiameter);
    DDX_Text(pDX, IDC_LaserIntensity_Edit, m_strRepetition);
    DDX_Text(pDX, IDC_LaserPower_Edit, m_strPower);
    DDX_Text(pDX, IDC_PulseWidth_Edit, m_strWidth);
    DDX_Text(pDX, IDC_LaserEnergy_Edit, m_strEnergy);
    DDX_Text(pDX, IDC_RepRate_Edit, m_strTxPhotons);
    DDX_Text(pDX, IDC_Photon_Read, m_Rx_Photon);
    DDX_Text(pDX, IDC_Difference, m_Difference);
    DDX_Text(pDX, IDC_LaserIntensity_Edit3, m_wavelength);
    DDX_Text(pDX, IDC_BeamDiameter_Edit2, m_transmit_optics);
    DDX_Text(pDX, IDC_LaserEnergy_Edit2, m_divergence);
    DDX_Text(pDX, IDC_LaserIntensity_Edit2, m_pointing_error);
    DDX_Text(pDX, IDC_LaserPower_Edit2, m_transmitter_gain);
    DDX_Control(pDX, IDC_Txoptics_eff, m_Txoptics);
    DDX_Control(pDX, IDC_divergence, m_fielddivergence);
    DDX_Control(pDX, IDC_pointing_error, m_pointing);
    DDX_Control(pDX, IDC_diameter, m_output_beam);
    DDX_Control(pDX, IDC_energy, m_total_energy);
    DDX_Control(pDX, IDC_laserpower, m_laserpower);
    DDX_Control(pDX, IDC_reprate, m_width);
    DDX_Text(pDX, IDC_PulseWidth_Edit2, m_Intensity);
    DDX_Control(pDX, IDC_Rxoptics_eff, m_Rx_Optics);
    DDX_Text(pDX, IDC_WindDirection_Read2, m_receive_optics);
    DDX_Control(pDX, IDC_Quantum_Dectetor, m_quantum_detector);
    DDX_Text(pDX, IDC_Quantum_Dect, m_Quantum_Dect);
    DDX_Text(pDX, IDC_Atmospheric_Trans, m_Atmosphere);
    DDX_Control(pDX, IDC_Atm_Trans, m_Atm);
    DDX_Control(pDX, IDC_CloudCover, m_cloud);
    DDX_Text(pDX, IDC_Cloud_Cover, m_Cloud);
    DDX_Text(pDX, IDC_Aperture_Area, m_Aperture);
    DDX_Text(pDX, IDC_WindSpeed_Read2, m_Cross_Section);
    DDX_Text(pDX, IDC_ScrollBar_Info, m_ScrollBarInfo);
    DDX_Control(pDX, IDC_HSlider, m_HSIlnder);
    DDX_Text(pDX, IDC_SlantRange, m_SlantRange);
    DDX_Control(pDX, IDC_Detector, m_Quantum);
    DDX_Text(pDX, IDC_Photons_Minute, m_Photon_Second);
    DDX_DateTimeCtrl(pDX, IDC_Clock, m_Clock_Time);
    DDX_Text(pDX, IDC_Time, m_direction);
    DDX_DateTimeCtrl(pDX, IDC_Time_Local, m_currentdate);
    DDX_Text(pDX, IDC_EDIT18, m_Julian_Date);
    DDX_Text(pDX, IDC_PRGS_HOUR, m_Lp);
    DDX_Text(pDX, IDC_EDIT21, m_M);
    DDX_Text(pDX, IDC_Difference13, m_Mp);
    DDX_Text(pDX, IDC_EDIT22, m_D);
    DDX_Text(pDX, IDC_Difference12, m_F);
    DDX_Text(pDX, IDC_Difference11, m_T);
    DDX_Control(pDX, ID_BTN_LOAD_FROM_FILE2, m_Pop_Home);
    DDX_Text(pDX, IDC_Julian_Day, m_Julian_Day);
    DDX_Text(pDX, IDC_Time3, m_Local_Temp);
    DDX_Text(pDX, IDC_EDIT35, m_Local_Pressure);
    DDX_Text(pDX, IDC_Distance4, m_Local_Humidity);
    DDX_Text(pDX, IDC_Time2, m_Local_WindS);
    DDX_Text(pDX, IDC_EDIT27, m_Local_WindD);
    DDX_Text(pDX, IDC_MDiameter3, m_Local_SVP);
    DDX_Text(pDX, IDC_Distance3, m_Local_PWV);
    DDX_Text(pDX, IDC_EDIT17, m_Local_AVP);
    DDX_Text(pDX, IDC_MTS3, m_Local_DPT);

```

```

    DDX_Control(pDX, ID_BTN_CURR_COLOR2, m_cloudcover);
    DDX_Control(pDX, IDC_Plot_Style, m_Plot_Style);
    DDX_Control(pDX, IDC_BUTTON1, m_Beam);
    DDX_Control(pDX, IDC_BUTTON3, m_Turbulence);
    DDX_Control(pDX, IDC_BUTTON4, m_Photons);
    DDX_Control(pDX, IDC_BandPdT, m_BandPdT);
    DDX_Control(pDX, ID_BTN_SAVE_TO_FILE3, m_Path);
    DDX_Control(pDX, ID_BTN_PRINT3, m_Telescope);
    DDX_Control(pDX, ID_BTN_CURR_UPDATE3, m_Data);
    DDX_Control(pDX, IDC_BandPdT3, m_LeastSquare);
    DDX_Control(pDX, IDC_SCROLLBAR1, m_HScroll);
    DDX_Control(pDX, IDC_Atm_Trans3, m_IncidentAngle);
    DDX_Text(pDX, IDC_Min, m_Min);
    DDX_Text(pDX, IDC_Max, m_Max);
    DDX_Control(pDX, IDC_Reset, m_reset);
    DDX_Control(pDX, IDC_PhotonsData, m_PhotonsData);
    DDX_Control(pDX, IDC_Predictions, m_Predictions);
    DDX_Control(pDX, IDC_HELP, m_Help);
    // DDX_Text(pDX, IDC_Distance, m_distance);
}
////////////////////////////////////

BEGIN_MESSAGE_MAP(CLinePlotTestDlg, CDialog)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_WM_SIZE()
    ON_BN_CLICKED(ID_BTN_CURR_COLOR, OnBtnCurrColor)
    ON_BN_CLICKED(ID_BTN_CURR_UPDATE, OnBtnCurrUpdate)
    ON_BN_CLICKED(ID_BTN_CURR_REMOVE, OnBtnCurrRemove)
    ON_CBN_CLOSEUP(ID_CMB_STYLE, OnCloseupCmbStyle)
    ON_NOTIFY(NM_PLOT_SEL_CHANGE, ID_CTRL_LINE_PLOT, OnPlotSelChange)
    ON_NOTIFY(NM_PLOT_LIMITS_CHANGE, ID_CTRL_LINE_PLOT, OnPlotLimitsChange)
    ON_NOTIFY(NM_PLOT_MOUSE_MOVE, ID_CTRL_LINE_PLOT, OnPlotMouseMove)
    ON_BN_CLICKED(ID_BTN_LOAD_FROM_FILE, OnBtnLoadFromFile)
    ON_BN_CLICKED(ID_BTN_PRINT, OnBtnPrint)
    ON_BN_CLICKED(ID_BTN_SAVE_TO_FILE, OnBtnSaveToFile)
    ON_CBN_SELCHANGE(ID_CMB_STYLE2, &CLinePlotTestDlg::OnCbnSelchangeCmbStyle2)
    ON_CBN_SELCHANGE(ID_CMB_STYLE3, &CLinePlotTestDlg::OnCbnSelchangeCmbStyle3)
    ON_NOTIFY(UDN_DELTAPOS, IDC_Txoptics_eff, &CLinePlotTestDlg::OnDeltaposTxopticseff)
    ON_NOTIFY(UDN_DELTAPOS, IDC_divergence, &CLinePlotTestDlg::OnDeltaposdivergence)
    ON_NOTIFY(UDN_DELTAPOS, IDC_pointing_error, &CLinePlotTestDlg::OnDeltapospointingerror)
    ON_NOTIFY(UDN_DELTAPOS, IDC_diameter, &CLinePlotTestDlg::OnDeltaposdiameter)
    ON_NOTIFY(UDN_DELTAPOS, IDC_energy, &CLinePlotTestDlg::OnDeltaposenergy)
    ON_NOTIFY(UDN_DELTAPOS, IDC_laserpower, &CLinePlotTestDlg::OnDeltaposlaserpower)
    ON_NOTIFY(UDN_DELTAPOS, IDC_reprate, &CLinePlotTestDlg::OnDeltaposreprate)
    ON_EN_CHANGE(IDC_LaserEnergy_Edit2, &CLinePlotTestDlg::OnEnChangeLaserenergyEdit2)
    ON_NOTIFY(UDN_DELTAPOS, IDC_Rxoptics_eff, &CLinePlotTestDlg::OnDeltaposRxopticseff)
    ON_NOTIFY(UDN_DELTAPOS, IDC_Quantum_Dectetor, &CLinePlotTestDlg::OnDeltaposQuantumDectetor)
    ON_NOTIFY(UDN_DELTAPOS, IDC_Atm_Trans, &CLinePlotTestDlg::OnDeltaposAtmTrans)
    ON_NOTIFY(UDN_DELTAPOS, IDC_CloudCover, &CLinePlotTestDlg::OnDeltaposCloudcover)
    ON_WM_HSCROLL()
    ON_NOTIFY(UDN_DELTAPOS, IDC_duration, &CLinePlotTestDlg::OnDeltaposduration)
    ON_NOTIFY(UDN_DELTAPOS, IDC_SPIN14, &CLinePlotTestDlg::OnDeltaposSpin14)
    ON_NOTIFY(NM_CUSTOMDRAW, IDC_HSlider, &CLinePlotTestDlg::OnNMCustomdrawHslider)
    ON_CBN_SELCHANGE(ID_CMB_STATION, &CLinePlotTestDlg::OnCbnSelchangeCmbStation)
    ON_WM_TIMER()
    ON_NOTIFY(DTN_DATETIMECHANGE, IDC_Clock, &CLinePlotTestDlg::OnDtnDatetimechangeClock)
    ON_EN_CHANGE(IDC_Time, &CLinePlotTestDlg::OnEnChangeTime)
    ON_EN_CHANGE(IDC_Distance, &CLinePlotTestDlg::OnEnChangeDistance)
    ON_BN_CLICKED(ID_BTN_LOAD_FROM_FILE2, &CLinePlotTestDlg::OnBnClickedBtnLoadFromFile2)
    ON_BN_CLICKED(ID_BTN_SAVE_TO_FILE2, &CLinePlotTestDlg::OnBnClickedBtnSaveToFile2)
    ON_BN_CLICKED(ID_BTN_CURR_UPDATE2, &CLinePlotTestDlg::OnBnClickedBtnCurrUpdate2)
    ON_BN_CLICKED(ID_BTN_CURR_REMOVE2, &CLinePlotTestDlg::OnBnClickedBtnCurrRemove2)
    ON_BN_CLICKED(ID_BTN_PRINT2, &CLinePlotTestDlg::OnBnClickedBtnPrint2)
    ON_BN_CLICKED(ID_BTN_SAVE_TO_FILE3, &CLinePlotTestDlg::OnBnClickedBtnSaveToFile3)
    ON_BN_CLICKED(ID_BTN_PRINT3, &CLinePlotTestDlg::OnBnClickedBtnPrint3)
    ON_BN_CLICKED(ID_BTN_CURR_UPDATE3, &CLinePlotTestDlg::OnBnClickedBtnCurrUpdate3)
    ON_BN_CLICKED(ID_BTN_CURR_COLOR2, &CLinePlotTestDlg::OnBnClickedBtnCurrColor2)
    ON_BN_CLICKED(IDC_Plot_Style, &CLinePlotTestDlg::OnBnClickedPlotStyle)
    ON_BN_CLICKED(IDC_BUTTON1, &CLinePlotTestDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CLinePlotTestDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON4, &CLinePlotTestDlg::OnBnClickedButton4)
    ON_BN_CLICKED(IDC_BandPdT, &CLinePlotTestDlg::OnBnClickedBandpandT)

```

```

ON_EN_CHANGE(IDC_BeamDiameter_Edit, &CLinePlotTestDlg::OnEnChangeBeamDiameterEdit)
ON_NOTIFY(UDN_DELTAP0S, IDC_Atm_Trans3, &CLinePlotTestDlg::OnDeltaposAtmTrans3)
ON_BN_CLICKED(IDC_Reset, &CLinePlotTestDlg::OnBnClickedReset)
ON_BN_CLICKED(IDC_PhotonsData, &CLinePlotTestDlg::OnBnClickedPhotonsdata)
ON_BN_CLICKED(IDC_Predictions, &CLinePlotTestDlg::OnBnClickedPredictions)
ON_BN_CLICKED(IDC_HELP, &CLinePlotTestDlg::OnBnClickedHelp)
END_MESSAGE_MAP()

////////////////////////////////////
// CLinePlotTestDlg message handlers
////////////////////////////////////

BOOL CLinePlotTestDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    m_HSilder.SetRange(350000, 405000, TRUE);
    m_HScroll.SetScrollRange(350000, 405000, TRUE);
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);        // Set big icon
    SetIcon(m_hIcon, FALSE);       // Set small icon

    // TODO: Add extra initialization here

    m_Txoptics.SetRange(m_transmit_optics, 5);
    m_Txoptics.SetPos(m_transmit_optics);
    if( m_cmbSystem.GetCurSel() == 0 ) {m_output_beam.SetRange(m_strDiameter, 0);}
    else if( m_cmbSystem.GetCurSel() == 1 ) {m_output_beam.SetRange(m_strDiameter, 10);}
    else {m_output_beam.SetRange(m_strDiameter, 10);}
    m_output_beam.SetPos(m_strDiameter);
    m_total_energy.SetRange(m_strEnergy,20);
    m_total_energy.SetPos(m_strEnergy);
    m_laserpower.SetRange(m_strPower,10);
    m_laserpower.SetPos(m_strPower);
    m_Rx_Optics.SetRange(m_receive_optics,5);
    m_Rx_Optics.SetPos(m_receive_optics);
    m_quantum_detector.SetRange(m_Quantum_Dect,20);
    m_quantum_detector.SetPos(m_Quantum_Dect);
    m_Atm.SetRange(0.02,88);
    m_Atm.SetPos(0.02);
    m_cloud.SetRange(0.1,9);
    m_cloud.SetPos(0.1);
    m_IncidentAngle.SetRange(0,2);
    m_IncidentAngle.SetPos(-30);

    CRect rcClient;
    GetWindowRect(&rcClient);
    rcClient.bottom = rcClient.top + 800;
    rcClient.right = rcClient.left + 1350;
    MoveWindow(&rcClient);

    m_Beam.SetFaceColor(RGB(255,0,0), true);
    m_Turbulence.SetFaceColor(RGB(51,51,255), true);
    m_Turbulence.SetTextColor(RGB(255,255,0));
    m_Photons.SetFaceColor(RGB(0,255,0), true);
    m_BandPdT.SetFaceColor(RGB(204,255,204), true);

```

```

m_PhotonsData.SetFaceColor(RGB(204,255,204), true);
m_Predictions.SetFaceColor(RGB(204,255,204), true);
m_Path.SetFaceColor(RGB(204,255,204), true);
m_Telescope.SetFaceColor(RGB(204,255,204), true);
m_Data.SetFaceColor(RGB(204,255,204), true);
m_LeastSquare.SetFaceColor(RGB(96,96,96), true);
m_LeastSquare.SetTextColor(RGB(255,255,255));
// create the line plot control.
m_LinePlot.Create(CRect(0, 0, 1, 1), WS_CHILD | WS_VISIBLE, this, ID_CTRL_LINE_PLOT);

// size all controls correctly;
SizeControls();

UINT ii;

// add a plot to the control.
srand((unsigned)time(NULL));

// add a plot to the control.
FLOATPOINT pData3[100];
for (ii=0; ii<100; ii++)
{
    pData3[ii].x = (float)(ii);
    pData3[ii].y = (float)(-cos((float)(ii/0.7)/16.0f)*10+10);
}
COLORREF crColor3 = RGB (255, 0, 0);
m_LinePlot.Add("Beam", crColor3, CLinePlot::LpLine, pData3, 1);

SetTimer(1, 40, NULL); //Create a running clock

//Read a worksheet

if(!AfxOleInit()) {
    AfxMessageBox("OLE init failed ");
    return true;
}
ColeyVariant
covTrue((short)TRUE),
covFalse((short)FALSE),
covOpt((long)DISP_E_PARAMNOTFOUND, VT_ERROR);

CApplication xlApp;
CWorkbooks xlBooks;
CWorkbook xlBook;
CWorksheets xlSheets;
CWorksheet xlSheet;
CRange xlRange;

VARIANT v = {0}, w = {0}, x = {0}, y = {0}, z = {0}, a = {0}, b = {0}, c = {0}, d = {0};
if(!xlApp.CreateDispatch("Excel.Application")) {
    AfxMessageBox("Failed.");
    return true;
}
UpdateData();
xlBooks = xlApp.get_Workbooks();

xlBook = xlBooks.Open("F:\\2016_PhD\\New motor control\\Lineplot\\geocentric moon.xlsm",
    covOpt , covOpt , covOpt ,
    covOpt , covOpt , covOpt ,
    covOpt , covOpt , covOpt ,
    covOpt , covOpt , covOpt ,
    covOpt , covOpt);

xlApp.put_Visible(false);
xlSheets = xlBook.get_Worksheets();
xlSheet = xlSheets.get_Item(ColeyVariant((short)1));

// Get data from excell cells
xlRange = xlSheet.get_Range( ColeVariant("B21"), ColeVariant("B21"));
v = xlRange.get_Value2();
xlRange = xlSheet.get_Range( ColeVariant("B22"), ColeVariant("B22"));
w = xlRange.get_Value2();

```

```

    xlRange = xlSheet.get_Range( COleVariant(_T("B23")), COleVariant(_T("B23")));
    x = xlRange.get_Value2();
    xlRange = xlSheet.get_Range( COleVariant("B24"), COleVariant("B24"));
    y = xlRange.get_Value2();
    xlRange = xlSheet.get_Range( COleVariant("B25"), COleVariant("B25"));
    z = xlRange.get_Value2();
    xlRange = xlSheet.get_Range( COleVariant("B29"), COleVariant("B29"));
    a = xlRange.get_Value2();
    xlRange = xlSheet.get_Range( COleVariant("B30"), COleVariant("B30"));
    b = xlRange.get_Value2();
    xlRange = xlSheet.get_Range( COleVariant("B30"), COleVariant("B30"));
    c = xlRange.get_Value2();
    xlRange = xlSheet.get_Range( COleVariant("B31"), COleVariant("B31"));
    d = xlRange.get_Value2();

    // Run macros

    // Print data in text boxes
    CString str = v.bstrVal, str1 = w.bstrVal, str2 = x.bstrVal, str3 = y.bstrVal, str4 = z.bstrVal;
    CString str5 = a.bstrVal, str6 = b.bstrVal, str7 = c.bstrVal, str8 = d.bstrVal;

    m_Local_Temp = v.bstrVal;
    m_Local_Pressure = w.bstrVal;
    m_Local_Humidity = x.bstrVal;
    m_Local_WindS = y.bstrVal;
    m_Local_WindD = z.bstrVal;
    m_Local_PWV = a.bstrVal;
    m_Local_SVP = b.bstrVal;
    m_Local_AVP = c.bstrVal;
    m_Local_DPT = d.bstrVal;
    UpdateData(FALSE);

    return true; // return TRUE unless you set the focus to a control
}
////////////////////////////////////

void CLinePlotTestDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}
////////////////////////////////////

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CLinePlotTestDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
}

```

```

    else
    {
        CDialog::OnPaint();
    }
}
////////////////////////////////////

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CLinePlotTestDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}
////////////////////////////////////

CRect CLinePlot::DrawFramework(CDC *pDC, CRect rcBounds, int nFontHeight, bool oIsPrinting)
{
    CPen *ppenOld;
    CFont *pfonOld;
    CFont fonScale;
    CFont fonHorz;
    CFont fonVert;

    // create the fonts to use here.
    LOGFONT lf;
    GetObject(GetStockObject(DEFAULT_GUI_FONT), sizeof(LOGFONT), &lf);
    //lf.lfWeight = FW_NORMAL;
    lf.lfHeight = nFontHeight;
    fonScale.CreateFontIndirect(&lf);
    lf.lfWeight = FW_BOLD;
    lf.lfHeight *= 1.25f;
    fonHorz.CreateFontIndirect(&lf);
    lf.lfEscapement = 900;
    fonVert.CreateFontIndirect(&lf);

    // create a white brush for the background.
    CBrush bkBrush(RGB(255, 255, 255));
    // create a gray pen for the ruler lines.
    CPen penGray(PS_SOLID, 1, RGB(220, 220, 220));
    // create a black brush for the data rectangle.
    CPen penBlack(PS_SOLID, 1, RGB(0, 0, 0));

    // get the height of the caption for the font height.
    int nCaptionHeight = nFontHeight + 4;
    int nSmCaptionHeight = (int)(nCaptionHeight * 0.75f); // GetSystemMetrics(SM_CYSMCAPTION);
    int nSpacer = nCaptionHeight/2;

    // calculate the usable area to draw in.
    CRect rcPlotArea(rcBounds);
    rcPlotArea.DeflateRect(1, 1, 1, 1);

    // calculate the rect for the Y caption.
    CRect rcYCaption(rcPlotArea);
    rcYCaption.right = rcYCaption.left + nCaptionHeight;

    // calculate the rect for the X caption.
    CRect rcXCaption(rcPlotArea);
    rcXCaption.left = rcXCaption.left + nCaptionHeight;
    rcXCaption.top = rcXCaption.bottom - nCaptionHeight;

    // adjust the client area.
    rcPlotArea.left += nCaptionHeight; // + nSpacer;
    rcPlotArea.bottom -= nCaptionHeight + nSpacer;

    // calculate the widest rect for the y scale.
    CString szTestNumMin;
    szTestNumMin.Format("%0.f", m_eYMin);
    int nNumCharsMin = szTestNumMin.GetLength();
    CString szTestNumMax;
    szTestNumMax.Format("%0.f", m_eYMax);
    int nNumCharsMax = szTestNumMax.GetLength();
    int nNumChars = (nNumCharsMin < nNumCharsMax) ? (nNumCharsMax) : (nNumCharsMin);
    CString szScaleTest('9', nNumChars+3);
    // use the font for the scale.

```

```

pfontOld = pDC->SelectObject(&fonScale);
// calculate the nominal height and width of each scale entry.
SIZE siScale = pDC->GetTextExtent(szScaleTest);

// create a starting rect for the scale.
CRect rcStartScale(rcPlotArea);
rcStartScale.right = rcStartScale.left + siScale.cx;
rcStartScale.top = rcStartScale.bottom - siScale.cy;

// shrink the client rect by the height & width of the scale + a little more.
rcPlotArea.left += siScale.cx + nSpacer/2;
rcPlotArea.bottom -= siScale.cy + nSpacer/2;
rcPlotArea.right -= nCaptionHeight;
rcPlotArea.top += nSpacer;

// create a rect to hold the stats at the top.
CRect rcStats(rcPlotArea);
rcStats.bottom = rcStats.top + siScale.cy;

// shrink the client rect away from the stats rect.
rcPlotArea.top += siScale.cy;

// should we calculate a key rect?
if (m_lstPlotData.size() > 1)
{
    // what is the longest key?
    int nMaxKey = 5;
    for (int ii=0; ii<m_lstPlotData.size(); ii++)
    {
        nMaxKey = (m_lstPlotData[ii].m_szName.GetLength() > nMaxKey) ?
            (m_lstPlotData[ii].m_szName.GetLength()) : (nMaxKey);
    }
    int nKeyWidth = (nFontHeight*nMaxKey)/2;
    // create a rect to hold the key
    m_rcKeyArea = rcPlotArea;
    m_rcKeyArea.left = m_rcKeyArea.right-nKeyWidth;
    m_rcKeyArea.right = m_rcKeyArea.right+10;

    // center & shrink the key rect to hold only the number of plots.
    int nYCenter = (m_rcKeyArea.top + m_rcKeyArea.bottom) / 2;
    m_rcKeyArea.top = nYCenter - ((siScale.cy+4) * m_lstPlotData.size()) / 2;
    m_rcKeyArea.bottom = nYCenter + ((siScale.cy+4) * m_lstPlotData.size()) / 2;

    // shrink the client rect away from the key rect.
    rcPlotArea.right -= (nKeyWidth+10);
    // shrink the stats rect away from the key rect.
    rcStats.right -= (nKeyWidth+10);
}

// adjust the space for the x caption so that it is centered.
rcXCaption.left = rcPlotArea.left;
rcXCaption.right = rcPlotArea.right;

// adjust the space for the y caption so that it is centered.
rcYCaption.top = rcPlotArea.top;
rcYCaption.bottom = rcPlotArea.bottom;

// calculate the rect for the min x.
m_rcXMin = rcStartScale;
m_rcXMin.OffsetRect(nSpacer/2, 0);
m_rcXMin.InflateRect(2, 2, 2, 2);

// calculate the rect for the max x.
m_rcXMax = rcStartScale;
m_rcXMax.OffsetRect(rcPlotArea.right - m_rcXMax.right, 0);
m_rcXMax.InflateRect(2, 2, 2, 2);

// calculate the rect for the x-locked icon in the corner.
m_rcXLock.left = rcYCaption.left;
m_rcXLock.top = m_rcXMin.top;
m_rcXLock.right = rcYCaption.right;
m_rcXLock.bottom = m_rcXMin.bottom;

// calculate the rect for the min y.

```

```

m_rcYMin = rcStartScale;
m_rcYMin.OffsetRect(0, rcPlotArea.bottom - m_rcYMin.bottom);
m_rcYMin.InflateRect(2, 2, 2, 2);

// calculate the rect for the max y.
m_rcYMax = rcStartScale;
m_rcYMax.OffsetRect(0, rcPlotArea.top - m_rcYMax.bottom);
m_rcYMax.InflateRect(2, 2, 2, 2);

// calculate the rect for the y-locked icon in the corner.
m_rcYLock.left = m_rcYMin.right - m_rcXLock.Width();
m_rcYLock.top = rcXCaption.top;
m_rcYLock.right = m_rcYMin.right;
m_rcYLock.bottom = rcXCaption.bottom;

// calculate how big each screen interval is.
float eScreenXInterval = (float)(rcPlotArea.Width()) / (float)(m_nXIntervals);
float eScreenYInterval = (float)(rcPlotArea.Height()) / (float)(m_nYIntervals);

// calculate how big each actual interval is.
float eXInterval = (m_eXMax - m_eXMin) / (float)(m_nXIntervals);
float eYInterval = (m_eYMax - m_eYMin) / (float)(m_nYIntervals);

// now we can calculate the ratio of actual data range
// to screen area.
m_eXRatio = (float)(rcPlotArea.Width()) / (m_eXMax - m_eXMin);
m_eYRatio = (float)(rcPlotArea.Height()) / (m_eYMax - m_eYMin);

// calculate the format for the scale.
CString szXFormat;
szXFormat.Format("%%0.%uf", m_uiXPrecision);
CString szYFormat;
szYFormat.Format("%%0.%uf", m_uiYPrecision);

/*
..... now start drawing!
*/

// fill the total background.
pDC->FillSolidRect(&rcBounds, GetSysColor(COLOR_WINDOW));

// draw the sunken edge around the whole control.
if (oIsPrinting==false)
    pDC->DrawEdge(rcBounds, EDGE_SUNKEN, BF_RECT);

// do we have enough room to draw?
if (rcPlotArea.left < rcPlotArea.right)
{
    // draw the y caption
    pfonOld = pDC->SelectObject(&fonVert);
    pDC->SetTextAlign(TA_CENTER | TA_BOTTOM);
    pDC->TextOut(rcYCaption.right + 5, (rcYCaption.top+rcYCaption.bottom)/2, m_szYCaption);

    // draw the x caption
    pfonOld = pDC->SelectObject(&fonHorz);
    pDC->SetTextAlign(TA_CENTER | TA_BOTTOM);
    pDC->TextOut((rcXCaption.left+rcXCaption.right)/2, rcXCaption.bottom -5, m_szXCaption);

    // setup to draw the scale.
    pDC->SetTextAlign(TA_RIGHT | TA_BOTTOM);
    POINT ptXScale;
    POINT ptYScale;
    CString szScale;
    pfonOld = pDC->SelectObject(&fonScale);
    ppenOld = pDC->SelectObject(&penGray);

    // draw each x scale entry & line.
    for (int ii=0; ii<m_nXIntervals+1; ii++)
    {
        szScale.Format(szXFormat, m_eXMin+ii*eXInterval);
        ptXScale.x = (int)(rcStartScale.right+(float)(ii)*eScreenXInterval) + nSpacer/2;
        ptXScale.y = rcStartScale.bottom;
        pDC->TextOut(ptXScale.x, ptXScale.y, szScale);
    }
}

```



```

    pDC->MoveTo(ptXScale.x, rcPlotArea.top);
    pDC->LineTo(ptXScale.x, rcPlotArea.bottom);
}

// is the mouse over the x min?
if (m_uiMouseOver&PLOT_OVER_X_MIN && oIsPrinting==false)
{
    pDC->DrawEdge(m_rcXMin, BDR_RAISEDOUTER, BF_RECT);
}

// is mouse over the x max?
if (m_uiMouseOver&PLOT_OVER_X_MAX && oIsPrinting==false)
{
    pDC->DrawEdge(m_rcXMax, BDR_RAISEDOUTER, BF_RECT);
}

// draw each y scale entry & line.
for (int jj=0; jj<m_nYIntervals+1; jj++)
{
    szScale.Format(szYFormat, m_eYMin+jj*eYInterval);
    ptYScale.x = rcStartScale.right;
    ptYScale.y = (int)(rcStartScale.top-(float)(jj)*eScreenYInterval) - nSpacer/2;
    pDC->TextOut(ptYScale.x, ptYScale.y, szScale);
    pDC->MoveTo(rcPlotArea.left, ptYScale.y);
    pDC->LineTo(rcPlotArea.right, ptYScale.y);
}

// is the mouse over the y min?
if (m_uiMouseOver&PLOT_OVER_Y_MIN && oIsPrinting==false)
{
    pDC->DrawEdge(m_rcYMin, BDR_RAISEDOUTER, BF_RECT);
}

// is the mouse over the y max?
if (m_uiMouseOver&PLOT_OVER_Y_MAX && oIsPrinting==false)
{
    pDC->DrawEdge(m_rcYMax, BDR_RAISEDOUTER, BF_RECT);
}

pDC->SelectObject(&penBlack);

// draw black lines around the data.
pDC->MoveTo(rcPlotArea.left, ptYScale.y);
pDC->LineTo(rcPlotArea.right, ptYScale.y);
pDC->LineTo(rcPlotArea.right, rcPlotArea.bottom);
pDC->LineTo(rcPlotArea.left, rcPlotArea.bottom);
pDC->LineTo(rcPlotArea.left, ptYScale.y);

// is the mouse over the x-lock area?
if (m_uiMouseOver&PLOT_OVER_X_LOCK && oIsPrinting==false)
{
    if (m_oIsMouseDown==true)
        pDC->DrawEdge(m_rcXLock, BDR_SUNKENINNER, BF_RECT);
    else
        pDC->DrawEdge(m_rcXLock, BDR_RAISEDOUTER, BF_RECT);
}

// get the center of the x-lock area.
POINT ptXLock = m_rcXLock.CenterPoint();
// draw the bottom of the lock.
CRect rcXLockBottom;
rcXLockBottom.left = ptXLock.x - 5;
rcXLockBottom.right = ptXLock.x + 6;
rcXLockBottom.top = ptXLock.y - 2;
rcXLockBottom.bottom = ptXLock.y + 7;
pDC->FillSolidRect(rcXLockBottom, RGB(0, 0, 0));
pDC->SetBkColor(GetSysColor(COLOR_WINDOW));
// draw the lock top.
pDC->MoveTo(ptXLock.x - 4, ptXLock.y);
pDC->LineTo(ptXLock.x - 2, ptXLock.y - 7);
pDC->LineTo(ptXLock.x + 2, ptXLock.y - 7);
if (m_oIsXLocked==true)
    pDC->LineTo(ptXLock.x + 4, ptXLock.y);
else

```

```

pDC->LineTo(ptXLock.x + 4, ptXLock.y-6);

// is the mouse over the y-lock area?
if (m_uiMouseOver&PLOT_OVER_Y_LOCK && oIsPrinting==false)
{
    if (m_oIsMouseDown==true)
        pDC->DrawEdge(m_rcYLock, BDR_SUNKENINNER, BF_RECT);
    else
        pDC->DrawEdge(m_rcYLock, BDR_RAISEDOUTER, BF_RECT);
}

// get the center of the y-lock area.
POINT ptYLock = m_rcYLock.CenterPoint();
// draw the bottom of the lock.
CRect rcYLockBottom;
rcYLockBottom.left = ptYLock.x - 5;
rcYLockBottom.right = ptYLock.x + 6;
rcYLockBottom.top = ptYLock.y - 2;
rcYLockBottom.bottom = ptYLock.y + 7;
pDC->FillSolidRect(rcYLockBottom, RGB(0, 0, 0));
pDC->SetBkColor(GetSysColor(COLOR_WINDOW));
// draw the lock top.
pDC->MoveTo(ptYLock.x - 4, ptYLock.y);
pDC->LineTo(ptYLock.x - 2, ptYLock.y - 7);
pDC->LineTo(ptYLock.x + 2, ptYLock.y - 7);
if (m_oIsYLocked==true)
    pDC->LineTo(ptYLock.x + 4, ptYLock.y);
else
    pDC->LineTo(ptYLock.x + 4, ptYLock.y-6);

// should we draw a key?
if (m_lstPlotData.size()>1)
{
    // setup to draw the keys.
    POINT ptLeftBottom;
    CString szName;
    pDC->SetTextAlign(TA_LEFT | TA_BOTTOM);
    CRect rcKey;
    CRect rcColor;
    // draw all the keys
    for (int ii=0; ii<m_lstPlotData.size(); ii++)
    {
        // get the color.
        COLORREF crKey = m_lstPlotData[ii].m_crColor;
        // get the name.
        szName = m_lstPlotData[ii].m_szName;
        // calculate the bottom corner of the text.
        ptLeftBottom.x = m_rcKeyArea.left + siScale.cy + 4;
        ptLeftBottom.y = m_rcKeyArea.top + (ii+1)*(siScale.cy+4) - 2;
        // calculate a rect around the text.
        rcKey.left = m_rcKeyArea.left;
        rcKey.right = m_rcKeyArea.right;
        rcKey.top = m_rcKeyArea.top + (ii)*(siScale.cy+4);
        rcKey.bottom = m_rcKeyArea.top + (ii+1)*(siScale.cy+4) - 1;
        m_lstPlotData[ii].m_rcKeyArea = rcKey;
        // get a rect for the color.
        rcColor = rcKey;
        rcColor.right = rcKey.left + siScale.cy+4;
        rcColor.DeflateRect(4, 4, 4, 4);
        // draw the color;
        pDC->FillSolidRect(&rcColor, crKey);
        pDC->SetBkColor(GetSysColor(COLOR_WINDOW));
        // draw the key text.
        pDC->TextOut(ptLeftBottom.x, ptLeftBottom.y, szName);
        // is this one selected?
        if (ii==m_nSelected && oIsPrinting==false)
        {
            // draw a rect around the key.
            pDC->DrawEdge(rcKey, BDR_SUNKENINNER, BF_RECT);
        }
        // is the mouse over a key?
        else if (ii==m_nMouseOverKey &&
            (m_uiMouseOver&PLOT_OVER_KEY) && oIsPrinting==false)
        {

```

```

        // draw a rect around the key.
        pDC->DrawEdge(rcKey, BDR_RAISEDOUTER, BF_RECT);
    }
}

// draw the position & stats.

CString szPosition;
if (Visuals) {if (Beam){
m_szXCaption = "Beam Diameter (cm/pulse)";
m_szYCaption = "Irradiance";
m_eXMin = 0;
m_eXMax = 100;
m_eYMin = 0;
m_eYMax = 20;}}

else if (Photons){
m_szXCaption = "Atmospheric transmission (%)";
m_szYCaption = "Number of Photons";
m_eXMin = 0;
m_eXMax = 100;
m_eYMin = 0;
m_eYMax = 30;}}

else if (Turbulence){
m_szXCaption = "Turbulence Structure (%)";
m_szYCaption = "Altitude (km)";
m_eXMin = 0;
m_eXMax = 100;
m_eYMin = 0;
m_eYMax = 24;}}

else{}}

if (BandPantT){ if (m_nSelected == 0){
m_szXCaption = "Beam Diameter (m)";
m_szYCaption = "Irradiance";
m_eXMin = 0;
m_eXMax = 100;
m_eYMin = 0;
m_eYMax = 10;}}

else if (m_nSelected == 1){
m_szXCaption = "Atmospheric transmission (%)";
m_szYCaption = "Number of Photons";
m_eXMin = 0;
m_eXMax = 100;
m_eYMin = 0;
m_eYMax = 30;}}

else if (m_nSelected == 2){
m_szXCaption = "Turbulence Structure (%)";
m_szYCaption = "Altitude (km)";
m_eXMin = 0;
m_eXMax = 100;
m_eYMin = 0;
m_eYMax = 24;}}

else{}}

int nCount = m_lstPlotData.size();
if (nCount>0 && m_nSelected>=0 && m_nSelected<nCount)
{
    if (m_nCursor >= 0 && (UINT)(m_nCursor) < m_lstPlotData[m_nSelected].m_uiPointCount)
    {
        FLOATPOINT ptDisplay;
        ptDisplay.x = m_lstPlotData[m_nSelected].m_pptData[m_nCursor].x;
        ptDisplay.y = m_lstPlotData[m_nSelected].m_pptData[m_nCursor].y;

        if (Visuals){if (Photons){ szPosition.Format("Time: %0.1f    Number of photons: %0.1f",
            ptDisplay.x, ptDisplay.y);}}
    }
}

```

```

        else if (Turbulence){ szPosition.Format("Turbulence structure: %0.1f    Altitude: %0.1f",
            ptDisplay.x, ptDisplay.y);}
        else {szPosition.Format("Beam Diameter: %0.1f    Irradiance: %0.1f",
            ptDisplay.x, ptDisplay.y);}}
        if (BandPandT){if (m_nSelected == 1){ szPosition.Format("Time: %0.1f    Number of photons: %0.1f",
            ptDisplay.x, ptDisplay.y);}
        else if (m_nSelected == 2){ szPosition.Format("Turbulence structure: %0.1f    Altitude: %0.1f",
            ptDisplay.x, ptDisplay.y);}
        else {szPosition.Format("Beam Diameter: %0.1f    Irradiance: %0.1f",
            ptDisplay.x, ptDisplay.y);}}

        pDC->SetTextAlign(TA_LEFT | TA_BOTTOM);
        pDC->TextOut(rcStats.left+1, rcStats.bottom-1, szPosition);
    }
    float eYMin = m_lstPlotData[m_nSelected].m_eYDataMin;
    float eYMax = m_lstPlotData[m_nSelected].m_eYDataMax;
    float eYMean = m_lstPlotData[m_nSelected].m_eYDataMean;
    szPosition.Format("Mean: %0.1f    Min: %0.1f    Max: %0.1f",
        eYMean, eYMin, eYMax);
    pDC->SetTextAlign(TA_RIGHT | TA_BOTTOM);
    pDC->TextOut(rcStats.right-1, rcStats.bottom-1, szPosition);
}

}

// delete the fonts.
DeleteObject(&fonScale);
DeleteObject(&fonHorz);
DeleteObject(&fonVert);

// return the area we have to draw the plot in.
return rcPlotArea;
UpdateData(FALSE);
}

void CLinePlotTestDlg::SizeControls()
{
    CRect rcClient;
    GetClientRect(&rcClient);
    rcClient.left += 500;

    if (m_LinePlot.m_hWnd!=NULL)
        m_LinePlot.MoveWindow(512, 200, rcClient.Width()*0.966, rcClient.Height()*0.58);

    // if (m_lstReceived.m_hWnd!=NULL)
    // {
    //     CRect rcArea;
    //     m_lstReceived.GetWindowRect(&rcArea);
    //     ScreenToClient(&rcArea);
    //     rcArea.bottom = rcClient.bottom;
    //     m_lstReceived.MoveWindow(rcArea.left, rcArea.top, rcArea.Width(), rcArea.Height());
    // }
    // }

    //////////////////////////////////////

void CLinePlotTestDlg::OnSize(UINT nType, int cx, int cy)
{
    CDialog::OnSize(nType, cx, cy);

    SizeControls();
}
////////////////////////////////////

BOOL CLinePlotTestDlg::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Add your specialized code here and/or call the base class
    //cs.style |= WS_CLIPCHILDREN;

    return CDialog::PreCreateWindow(cs);
}
////////////////////////////////////

```

```

void CLinePlotTestDlg::OnBtnCurrColor()
{
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();

    if (nIndex<0 || nIndex>=m_LinePlot.Count())
        AfxMessageBox("There is no plot selected.");
    else
    {
        // get the color.
        COLORREF crCurr = m_LinePlot.GetColor(nIndex);
        // show the color dialog.
        CColorDialog dlgColor(crCurr, CC_FULLOPEN);

        if (dlgColor.DoModal()==IDOK)
        {
            // update the current color.
            m_LinePlot.SetColor(nIndex, dlgColor.GetColor());
        }
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnBtnCurrUpdate()
{
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();

    if (nIndex<0 || nIndex>=m_LinePlot.Count())
        AfxMessageBox("There is no plot selected.");
    else
    {
        // get the number of points (this must match for an update).
        UINT uiPointCount = m_LinePlot.GetPointCount(nIndex);
        // is it valid?
        if (uiPointCount>0)
        {
            // declare a new array which will be copied.
            FLOATPOINT *pData = new FLOATPOINT[uiPointCount];
            srand((unsigned)time(NULL));
            for (UINT ii=0; ii<uiPointCount; ii++)
            {
                pData[ii].x = (float)(ii);
                pData[ii].y = (float)(rand() * 20.0f / RAND_MAX) - 10.0f;
            }
            m_LinePlot.SetData(nIndex, pData);
            delete pData;
        }
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnBtnCurrRemove()
{
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();

    if (nIndex<0 || nIndex>=m_LinePlot.Count())
        AfxMessageBox("There is no plot selected.");
    else
    {
        if (m_LinePlot.Remove(nIndex)!=true)
        {
            AfxMessageBox("Could not remove current plot.");
        }
    }
    if (m_cmbSystem.GetCurSel() == 0 ){
    }
    else {
        UINT ii;
        CString m_szXCaption, m_szYCaption;

        // add a plot to the control.
    }
}

```

```

    srand((unsigned)time(NULL));

    // add a plot to the control.
    FLOATPOINT pData3[100];
    for (ii=0; ii<100; ii++)
    {
        pData3[ii].x = (float)(ii);
        pData3[ii].y = (float)(-cos((float)(ii/0.7)/16.0f)*10+10);
    }
    COLORREF crColor3 = RGB (255, 0, 0);
    m_LinePlot.Add("Beam", crColor3, CLinePlot::LpLine, pData3, 1);
}

/////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnPlotMouseMove(NMHDR* pNMHDR, LRESULT* pResult)
{
    // add this message to the notification list.
    // m_lstReceived.AddString(_T("NM_PLOT_MOUSE_MOVE"));
}
/////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnPlotLimitsChange(NMHDR* pNMHDR, LRESULT* pResult)
{
    // add this message to the notification list.
    // m_lstReceived.AddString(_T("NM_PLOT_LIMITS_CHANGE"));
}
/////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnPlotSelChange(NMHDR* pNMHDR, LRESULT* pResult)
{
    // add this message to the notification list.
    //m_lstReceived.AddString(_T("NM_PLOT_SEL_CHANGE"));

    // get the current plot index.

    int nIndex = m_LinePlot.Selected();

    if (nIndex<0 || nIndex>=m_LinePlot.Count())
        ; //AfxMessageBox("There is no plot selected.");
    else
    {
        // change enum type to index.
        int nComboIndex = (int)(m_LinePlot.GetStyle(nIndex));

        if (nComboIndex>=0 && nComboIndex<m_cmbStyle.GetCount())
        {
            // set the combo selection.
            m_cmbStyle.SetCurSel(nComboIndex); // default style
        }

        if (nComboIndex>=0 && nComboIndex<m_cmbSystem.GetCount())
        {
            // set the combo selection.
            m_cmbSystem.SetCurSel(nComboIndex); // default System
        }

        if (nComboIndex>=0 && nComboIndex<m_cmbTarget.GetCount())
        {
            // set the combo selection.
            m_cmbTarget.SetCurSel(nComboIndex); // default Target
        }
    }
}
/////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnCloseupCmbStyle()
{
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();

```

```

if (nIndex<0 || nIndex>=m_LinePlot.Count())
    AfxMessageBox("There is no plot selected.");
else
{
    int nComboIndex = m_cmbStyle.GetCurSel();

    if (nComboIndex>=0 && nComboIndex<m_cmbStyle.GetCount())
    {
        // change to the enum type.
        CLinePlot::enumPlotStyle enumStyle = (CLinePlot::enumPlotStyle)(nComboIndex);
        // set the plot type.
        m_LinePlot.SetStyle(nIndex, enumStyle);
    }
}
}
}
//void CLinePlotTestDlg::OnBtnClear()
//{
//    m_lstReceived.ResetContent();
//}
//void CLinePlotTestDlg::OnBtnPrint()
//{
//    m_LinePlot.Print();
//}
//void CLinePlotTestDlg::OnBtnLoadFromFile()
//{
//    CString szTitle = "Load Plot Data From....";
//    CString szSelFilename = "TestPlot.txt";
//    CString szDefExtension = ".txt";

//    long nFilterIndex = 0;
//    long nNumFilters = 0;
//    CString szFilter = "";
//    // do for certain file types.
//    szFilter += "Text File (*.txt)|*.txt|";
//    nNumFilters++;
//    szFilter += "|";

//    // create the dialog.
//    CFileDialog dlgFile(TRUE, NULL, NULL, OFN_HIDEREADONLY, szFilter);

//    // set the title
//    dlgFile.m_ofn.lpstrTitle = szTitle.GetBuffer(MAX_PATH);
//    // set the flags.
//    dlgFile.m_ofn.Flags |= OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT;
//    // set the default filename.
//    dlgFile.m_ofn.lpstrFile = szSelFilename.GetBuffer(MAX_PATH);
//    // set the default extension.
//    dlgFile.m_ofn.lpstrDefExt = szDefExtension.GetBuffer(MAX_PATH);
//    // set the number of filters
//    dlgFile.m_ofn.nMaxCustFilter = nNumFilters;
//    // set the default filter.
//    dlgFile.m_ofn.nFilterIndex = 0;

//    // show the dialog.
//    if (dlgFile.DoModal()==IDOK)
//    {
//        // release the buffer used for the filename.
//        szSelFilename.ReleaseBuffer();

//        // pull out the name minus the extension.
//        CString szName = szSelFilename;
//        int nPos = szSelFilename.ReverseFind('.');
//        if (nPos>=0)
//            szName = szSelFilename.Left(nPos);
//        nPos = szName.ReverseFind('\\');
//        if (nPos>=0)
//            szName = szName.Right(szName.GetLength() - nPos - 1);
//    }
//}

```

```

// get a random color.
srand((unsigned)time(NULL));
BYTE yRed = (BYTE)(rand() / (RAND_MAX / 256) + 1);
BYTE yGreen = (BYTE)(rand() / (RAND_MAX / 256) + 1);
BYTE yBlue = (BYTE)(rand() / (RAND_MAX / 256) + 1);
COLORREF crPlot = RGB(yRed, yGreen, yBlue);

FILE *fPlot;
if ((fPlot = fopen(szSelFilename, "r"))!=NULL)
{
    // get some temp storage.
    std::vector<FLOATPOINT> vecData;
    FLOATPOINT ptTemp = {0.0f, 0.0f};
    int nErr = 0;
    while (!feof(fPlot))
    {
        nErr = fscanf(fPlot, "%f\t%f\n",&(ptTemp.x), &(ptTemp.y));
        if (nErr==EOF)
            break;
        else
        {
            vecData.push_back(ptTemp);
        }
    }
    fclose(fPlot);
    // add to the plot
    m_LinePlot.Add(szName, crPlot, CLinePlot::LpLine, &vecData);
}
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnBtnSaveToFile()
{
    // which plot is the current one?
    int nIndex = m_LinePlot.Selected();
    if (nIndex < 0)
        AfxMessageBox("No plot selected - select one and try again.");
    else
    {
        // get a pointer to the data.
        FLOATPOINT *pptData = m_LinePlot.GetData(nIndex);
        // get the number of points.
        UINT uiPointCount = m_LinePlot.GetPointCount(nIndex);
        // get the name of the plot.
        CString szName = m_LinePlot.GetName(nIndex);
        if (pptData == NULL || uiPointCount == 0)
            AfxMessageBox("Selected plot has no data - try another.");
        else
        {
            CString szTitle = "Save Plot Data To...";
            CString szSelFilename = szName + ".txt";
            CString szDefExtension = "txt";

            long nFilterIndex = 0;
            long nNumFilters = 0;
            CString szFilter = "";
            // do for certain file types.
            szFilter += "Text File (*.txt)|*.txt|";
            nNumFilters++;
            szFilter += "|";

            // create the dialog.
            CFileDialog dlgFile(FALSE, NULL, NULL, 0, szFilter);

            // set the title
            dlgFile.m_ofn.lpstrTitle = szTitle.GetBuffer(MAX_PATH);
            // set the flags.
            dlgFile.m_ofn.Flags |= OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT;
            // set the default filename.
            dlgFile.m_ofn.lpstrFile = szSelFilename.GetBuffer(MAX_PATH);
            // set the default extension.

```



```

dlgFile.m_ofn.lpstrDefExt = szDefExtension.GetBuffer(MAX_PATH);
// set the number of filters
dlgFile.m_ofn.nMaxCustFilter = nNumFilters;
// set the default filter.
dlgFile.m_ofn.nFilterIndex = 0;

// show the dialog.
if (dlgFile.DoModal()==IDOK)
{
    // release the buffer used for the filename.
    szSelFilename.ReleaseBuffer();

    FILE *fPlot;
    if ((fPlot = fopen(szSelFilename, "w"))!=NULL)
    {
        int nErr = 0;
        for (UINT ii=0; ii<uiPointCount; ii++)
        {
            nErr = fprintf(fPlot, "%f\t%f\n",pptData[ii].x, pptData[ii].y);
        }
        fclose(fPlot);
    }
}
}
}
}
}
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void CLinePlotTestDlg::OnCbnSelchangeCmbStyle3()
{
    // add this message to the notification list.
    //m_lstReceived.ResetContent(); // Reset message received content
    // m_lstReceived.AddString(_T("Transmitted Data.....  "));

    UpdateData();
    double transmitter_gain = 0.00,
           transmit_optics = 0.00,
           receive_optics  = 0.00;

    double Diameter = 0.00,
           Energy    = 0.00,
           Width     = 0.00,
           RepRate   = 0.00,
           Power     = Energy*Width ,
           Duration  = 0.00,
           Calculate  = 0.00,
           Wavelength = 532e-9,
           PI        = 4.0*atan(1.0),
           Planck    = 6.626e-34,
           Light     = 2.998e8,
           Divergence = 0.00,
           PhotonsTx,
           Pointing,
           TxGain,
           Temperature,
           Pressure,
           QuantumDetector,
           Humidity = 0.00,
           Cloud,
           Aperture = 0.00,
           Cross_section,
           Range    = 0.5*m_strTime_Of_Flight*Light,
           Atmosphere;

    // SYSTEMTIME st;
    // GetLocalTime(&st);
    // m_Static_Time.Format("%H:%M");
    // CString t = CTime::GetCurrentTime().Format("%H:%M:%S");

    //CString t = ColeDateTme::GetCurrentTime().Format("%H:%M:%S");
    //m Static Time = timeDest;

```

```

if( m_cmbSystem.GetCurSel() == 0 ){
    Diameter = 0.000;
    Energy    = 0.00,
    Width     = 0.00,
    Power     = Energy*Width    ,
    Duration  = 0.00,
    Wavelength = 0,
    transmit_optics = 0.00,
    receive_optics = 0.00,
    Divergence = 0.00,
    m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2),
    Pointing   = 0.00,
    m_pointing_error = Pointing,
    PhotonsTx  = Energy*Wavelength/(Planck*Light);
    m_Intensity = 0.00,
    Temperature = 0.00,
    Pressure    = 0.00,
    Humidity    = 0.00,
    QuantumDetector = 0.00,
    Atmosphere  = 0.00;
    Cloud       = 0.00;
    Range       = 0.00;
    m_transmitter_gain = 0.00;
    m_strTxPhotons = PhotonsTx;
    m_strDiameter = Diameter;
    m_strRepetition = Width;
    m_strPower = Power;
    m_strWidth = Duration;
    m_strEnergy = Energy;
    m_wavelength = Wavelength;
    m_transmit_optics = transmit_optics;
    m_receive_optics = receive_optics;
    m_Quantum_Dect = QuantumDetector/100;
    m_Atmosphere = Atmosphere;
    m_Cloud = Cloud;
    m_Aperture = Aperture;
    m_Cross_Section = 0.00;
    m_Rx_Photon = 0.00;
    SetWindowText(_T("No System selected"));
}

else if( m_cmbSystem.GetCurSel() == 1){
    Diameter = 100.00e-3;
    Energy    = 0.45e-3,
    Width     = 1000,
    Power     = Energy*Width,
    Wavelength = 532e-9,
    Duration  = 20.00,
    transmit_optics = 0.40,
    receive_optics = 0.40,
    m_strDiameter = Diameter;
    Temperature = 268;
    Pressure    = 853;
    Humidity    = 0.00;
    QuantumDetector = 0.30;
    Atmosphere  = 0.02;
    Cloud       = 0.1;
    Aperture    = 0.714712328691678;
    Range       = 0.5*m_strTime_Of_Flight*Light,
    m_strRepetition = Width;
    m_strPower = Power;
    m_strWidth = Duration;
    m_strEnergy = Energy;
    m_wavelength = Wavelength;
    m_transmit_optics = transmit_optics;
    m_receive_optics = receive_optics;
    Divergence = 4e6*Wavelength/(PI*m_strDiameter/2),
    m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2),
    m_pointing.SetRange(0,488),
    Pointing    = 4.88e-6,
    m_pointing_error = Pointing,
    PhotonsTx   = Energy*Wavelength/(Planck*Light);
    m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
    m_strTxPhotons = PhotonsTx;

```

```

        m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp (-2*(m_divergence/
m_pointing_error)*(m_divergence/m_pointing_error));
m_Quantum_Dect = QuantumDetector;
m_Atmosphere    = Atmosphere;
m_Cloud        = Cloud;
m_Aperture     = Aperture;
SetWindowText(_T("Satellite Laser Ranger System is selected, now select the Satellite of target"));
}

else {
    Diameter = 900e-3;
    Energy    = 100.00e-3,
    Width     = 20,
    Power     = Energy*Width,
    Wavelength = 532e-9,
    Duration  = 80.00,
    transmit_optics = 0.40,
    receive_optics  = 0.40,
    m_strDiameter = Diameter;
    Temperature = 268;
    Pressure    = 853;
    Humidity    = 0.00;
    QuantumDetector = 0.40;
    Atmosphere  = 0.02;
    Cloud       = 0.1;
    Aperture    = 0.714712328691678;
m_strRepetition = Width;
m_strPower      = Power;
m_strWidth      = Duration;
m_strEnergy     = Energy;
m_wavelength    = Wavelength;
m_transmit_optics = transmit_optics;
m_receive_optics  = receive_optics;
    Divergence = 4e6*Wavelength/(PI*Diameter),
    m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2),
    m_pointing.SetRange(0,488),
    Pointing    = 4.88e-6,
    m_pointing_error = Pointing,
    PhotonsTx   = Energy*Wavelength/(Planck*Light);
    m_Intensity    = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
    m_strTxPhotons = PhotonsTx;
m_Quantum_Dect = QuantumDetector;
m_Atmosphere    = Atmosphere;
m_Cloud        = Cloud;
m_Aperture     = Aperture;
m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp(-2*(m_divergence/m_pointing_error)*
(m_divergence/m_pointing_error));
//m_transmitter_gain = (4*PI*m_Aperture*2.9241)/(m_wavelength*m_wavelength);
Range = 0.5*m_strTime_Of_Flight*Light;
m_SlantRange = Range;
m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
*1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
m_Photon_Second = m_Rx_Photon*m_strRepetition;
SetWindowText(_T("Lunar Laser Ranger System is selected, now select the Satellite of target"));
}
//m_strTxPhotons.Format(_T("%.3f"));

CString Diameters, Energies, Repetitions, Powers, Durations, Photons;
Diameters.Format(_T("%.2f"), Diameter);
Energies.Format(_T("%.5f"), Energy);
Repetitions.Format(_T("%.2f"), Width);
Powers.Format(_T("%.2f"), Power);
Durations.Format(_T("%.f"), Duration);
Photons.Format(_T("%.f"), PhotonsTx);

UpdateData(FALSE);
}

void CLinePlotTestDlg::OnCbnSelchangeCmbStyle2()
{
    //m_1stReceived.ResetContent(); // Reset message received content

```

```

//m_lstReceived.AddString(_T("Transmitted Data.....  "));
UpdateData();
double Time_Of_Flight = 0.00,
       Light = 2.99792458e8,
       Planck = 6.626e-34,
       Range      = 0.50*m_strTime_Of_Flight*Light,
       Ideal_Time  = 2.5,
       PI          = 4.0*atan(1.0),
       Time_Delay   = Ideal_Time-m_strTime_Of_Flight,
       Cross_section = 0.00;

if( m_cmbTarget.GetCurSel() == 0 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        Time_Of_Flight = 0;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay   = Ideal_Time-Time_Of_Flight;
        Cross_section = 0;
        m_Min = 0;
        m_Max = 0;
        SetWindowText(_T("No sattellite selected!"));
    }
}

else if( m_cmbTarget.GetCurSel() == 1 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        Time_Of_Flight = 1.274e-1;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay   = Ideal_Time-Time_Of_Flight;
        Cross_section = 60e6;
        m_Min = 60e6;
        m_Max = 60e6;
        SetWindowText(_T("ETALON selected, now Adjust the Varying Parameters (Transmitted and Additional)"));
    }
}

else if( m_cmbTarget.GetCurSel() == 2 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        // m_HSIlder.SetRange(0, 2000, TRUE);
        Time_Of_Flight = 1.276e-1;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay   = Ideal_Time-Time_Of_Flight;
        Cross_section = 360e6;
        m_Min = 360e6;
        m_Max = 360e6;
        SetWindowText(_T("GLONASS selected, now Adjust the Varying Parameters (Transmitted and Additional)"));
    }
}

else if( m_cmbTarget.GetCurSel() == 3 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        //m_HSIlder.SetRange(0, 2000, TRUE);
        Time_Of_Flight = 1.35e-1;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay   = Ideal_Time-Time_Of_Flight;
        m_Min = 40e6;
        m_Max = 40e6;
        Cross_section = 40e6;
        SetWindowText(_T("GPS selected, now Adjust the Varying Parameters (Transmitted and Additional)"));
    }
}

else if( m_cmbTarget.GetCurSel() == 4 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        // m_HSIlder.SetRange(0, 20000, TRUE);
        Time_Of_Flight = 4e-2;
        Range = 0.50*Time_Of_Flight*Light;
    }
}

```

```

        Time_Delay    = Ideal_Time-Time_Of_Flight;
        m_Min = 7e6;
        m_Max = 7e6;
        Cross_section = 7e6;
        SetWindowText(_T("LAGEOS selected, now Adjust the Varying Parameters (Transmitted and Additional)"));
    });
}

}

else if( m_cmbTarget.GetCurSel() == 5 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        //m_HSlider.SetRange(0, 20000, TRUE);
        Time_Of_Flight = 5.43333e-3;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay    = Ideal_Time-Time_Of_Flight;
        m_Min = 0.65e6;
        m_Max = 0.65e6;
        Cross_section = 0.65e6;
        SetWindowText(_T("STELLA selected, now Adjust the Varying Parameters (Transmitted and Additional)"));
    });
}

}

else if( m_cmbTarget.GetCurSel() == 6 ){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        //m_HSlider.SetRange(0, 20000, TRUE);
        Time_Of_Flight = 2.522305137047;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay    = Ideal_Time-Time_Of_Flight;
        if (m_Difference == -30) {
            m_Min = 9.6e3;
            m_Max = 9.56e8;
            Cross_section = 3.32e8;
        }

        else if (m_Difference == 0) {
            m_Min = 7.34e3;
            m_Max = 7.96e8;
            Cross_section = 2.20e8;
        }

        else {
            m_Min = 7.34e3;
            m_Max = 7.96e8;
            Cross_section = 2.20e8;
        }
        SetWindowText(_T("Apollo 11 is selected, now Adjust the Varying Parameters (Transmitted and Additional)"));
    }
}

}

else if( m_cmbTarget.GetCurSel() == 7){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        //m_HSlider.SetRange(0, 20000, TRUE);
        Time_Of_Flight = 2.542947552386;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay    = Ideal_Time-Time_Of_Flight;
        if (m_Difference == -30) {
            m_Min = 9.6e3;
            m_Max = 9.56e8;
            Cross_section = 3.32e8;
        }

        else if (m_Difference == 0) {
            m_Min = 7.34e3;
            m_Max = 7.96e8;
            Cross_section = 2.20e8;
        }

        else {
            m_Min = 7.34e3;

```

```

        m_Max = 7.96e8;
        Cross_section = 2.20e8;
    }
    SetWindowText(_T("Apollo 14 is selected, now Adjust the Varying Parameters (Transmitted and
Additional)"));
}
}
else if( m_cmbTarget.GetCurSel() == 8){
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        //m_HSlider.SetRange(0, 20000, TRUE);
        Time_Of_Flight = 2.447802471428 ;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay = Ideal_Time-Time_Of_Flight;
        if (m_Difference == -30) {
            m_Min = 2.88e4;
            m_Max = 2.87e9;
            Cross_section = 9.97e8;
        }

        else if (m_Difference == 0) {
            m_Min = 2.46e4;
            m_Max = 2.39e9;
            Cross_section = 6.59e8;
        }

        else {
            m_Min = 1.47e4;
            m_Max = 2.87e9;
            Cross_section = 5.26e8;
        }
        SetWindowText(_T("Apollo 15 is selected, now Adjust the Varying Parameters (Transmitted and
Additional)"));
    }
}

else {
    if (m_cmbSystem.GetCurSel() == 0) SetWindowText(_T("First Select the System to be used!"));
    else {
        //m_HSlider.SetRange(0, 20000, TRUE);
        Time_Of_Flight = 0.00535 ;
        Range = 0.50*Time_Of_Flight*Light;
        Time_Delay = Ideal_Time-Time_Of_Flight;
        Cross_section = 3;
        SetWindowText(_T("Space Debris Object is selected, now Adjust the Varying Parameters (Transmitted
and Additional)"));
    }
}
m_Cross_Section = Cross_section;
m_SlantRange = Range*1e-3;
m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
m_transmitter_gain*Cross_section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange*
1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
m_Photon_Second = m_Rx_Photon*m_strRepetition;
m_strTime_Of_Flight = Time_Of_Flight;
UpdateData(FALSE);
}

void CLinePlotTestDlg::OnDeltaposTxoopsicseff(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    UpdateData();

    double transmit_optics,
        Planck = 6.626e-34,
        Light = 2.998e8,
        PI = 4.0*atan(1.0);

    if( m_cmbSystem.GetCurSel() == 0 ){
        transmit_optics = 0.00;
        SetWindowText(_T("Transmit Optics Efficiency Adjustment is not Activated, First Select the System to be

```

```

used"));
}

else if( m_cmbSystem.GetCurSel() == 1 ){
transmit_optics = m_Txoptics.GetPos();
transmit_optics = transmit_optics/10 + 0.40;
SetWindowText(_T("Transmit Optics Efficiency Adjustment is Activated"));

}

else {transmit_optics = m_Txoptics.GetPos();
transmit_optics = transmit_optics/10 + 0.40;
SetWindowText(_T("Transmit Optics Efficiency Adjustment is Activated"));
}

// Get the current value of each spin control

m_transmit_optics = transmit_optics;
m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange*
*1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
m_Photon_Second = m_Rx_Photon*m_strRepetition;
UpdateData(FALSE);
*pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposdivergence(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltapospointingerror(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    UpdateData();

    double Pointing,
        Light = 2.998e8,
        Planck = 6.626e-34,
        PI = 4.0*atan(1.0);

    if( m_cmbSystem.GetCurSel() == 0 ){
        Pointing = 0.00,
        m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp (-2*(m_divergence/m_pointing_error)*
(m_divergence/m_pointing_error));

        SetWindowText(_T("Beam Divergence Adjustment is not Activated, First Select the System to be used!"));

    }

    else if( m_cmbSystem.GetCurSel() == 1 ){
        Pointing = m_pointing.GetPos(),
        m_pointing_error = (Pointing - 65536)/10000000 + 4.88e-6;
        //m_pointing_error = Pointing,
        m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp (-2*(m_divergence/m_pointing_error)*
(m_divergence/m_pointing_error));
        SetWindowText(_T("Beam Divergence Adjustment is Activated"));
    }

    else {
        Pointing = m_pointing.GetPos(),
        m_pointing_error = (Pointing - 65536)/10000000 + 4.88e-6;
        //m_pointing_error = Pointing,
        m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp(-2*(m_divergence/m_pointing_error)*
(m_divergence/m_pointing_error));
        //m_transmitter_gain = (4*PI*(PI*(m_strDiameter/2)*(m_strDiameter/2))*(2.5636))/(m_wavelength*
m_wavelength);
        SetWindowText(_T("Beam Divergence Adjustment is Activated"));
    }

    m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*

```

```

    m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
    *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
    m_Photon_Second = m_Rx_Photon*m_strRepetition;
    UpdateData(FALSE);
    *pResult = 0;
}

```

```

void CLinePlotTestDlg::OnDeltaposdiameter(NMHDR *pNMHDR, LRESULT *pResult)
{

```

```

    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    UpdateData();

```

```

    double Diameter,
           Divergence,
           Wavelength = 532e-9,
           PI = 4.0*atan(1.0),
           TxGain,
           Light = 2.998e8,
           Planck = 6.626e-34,
           Range = 0.5*m_strTime_Of_Flight*Light,
           Pointing;

```

```

    if( m_cmbSystem.GetCurSel() == 0 ){
        Diameter = 0.00;
        Divergence = 0.00,
        m_strDiameter = Diameter;
        m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2);
        m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
        SetWindowText(_T("Beam Diameter Adjustment is not Activated, First Select the System to be used!"));
        m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp (-2*(m_divergence/m_pointing_error)*
        (m_divergence/m_pointing_error));
    }

```

```

    else if( m_cmbSystem.GetCurSel() == 1 ){

        Diameter = m_output_beam.GetPos();
        m_strDiameter = 100e-3+Diameter/100;
        Divergence = 4*Wavelength/(PI*m_strDiameter/2),
        m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2);
        m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
        SetWindowText(_T("Beam Diameter Adjustment is Activated"));
        m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp (-2*(m_divergence/
        m_pointing_error)*(m_divergence/m_pointing_error));
        int nIndex = m_LinePlot.Selected();

```

```

//if (nIndex<0 || nIndex>=m_LinePlot.Count())
//AfxMessageBox("There is no plot selected.");

```

```

// else

```

```

{
    // get the number of points (this must match for an update).
    UINT uiPointCount = m_LinePlot.GetPointCount(nIndex) + Diameter;
    // is it valid?
    if (uiPointCount>0)
    {
        // declare a new array which will be copied.

```

```

        if (Beam) {
            FLOATPOINT *pDataB = new FLOATPOINT[uiPointCount];
            srand((unsigned)time(NULL));
            for (UINT ii=0; ii<uiPointCount; ii++)
            {
                pDataB[ii].x = (float)(ii);
                pDataB[ii].y = (float)(-cos((float)((ii)/(m_strDiameter))/16.0f)*10+10);
            }
            m_LinePlot.SetData(nIndex, pDataB);
            delete pDataB;
        }
    }
}

```



```

else if (Turbulence){
    FLOATPOINT *pDataT = new FLOATPOINT[uiPointCount+30];
    srand((unsigned)time(NULL));
    for (UINT ii=0; ii<uiPointCount; ii++)

    {
        pDataT[ii].x = (float)(ii);
        pDataT[ii].y = (float)((rand() / (RAND_MAX / 13)+m_direction));
    }
    m_LinePlot.SetData(nIndex, pDataT);
    delete pDataT;}

else if (Photons){
    FLOATPOINT *pDataP = new FLOATPOINT[uiPointCount+30];
    srand((unsigned)time(NULL));
    for (UINT ii=0; ii<uiPointCount; ii++)

    {
        pDataP[ii].x = (float)(ii);
        pDataP[ii].y = (float)((rand() / (RAND_MAX / (m_Photon_Second)))));
    }
    m_LinePlot.SetData(nIndex, pDataP);
    delete pDataP;}

else {
    AfxMessageBox("Choose visualisation");
}
}
}

else if ( m_cmbSystem.GetCurSel() == 2 ){

Diameter = m_output_beam.GetPos();
m_strDiameter = 900e-3+Diameter/100;
Divergence = (4*Wavelength/(PI*m_strDiameter/2));
m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2);
m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp(-2*(m_divergence/m_pointing_error)*
(m_divergence/m_pointing_error));
//m_transmitter_gain = (4*PI*(PI*(m_strDiameter/2)*(m_strDiameter/2))*(2.5636))/(m_wavelength*
m_wavelength);
SetWindowText(_T("Beam Diameter Adjustment is Activated"));

int nIndex = m_LinePlot.Selected();

{
    // get the number of points (this must match for an update).

    UINT uiPointCount = m_LinePlot.GetPointCount(nIndex);
    // is it valid?
    if (uiPointCount>0)
    {
        // declare a new array which will be copied.

        if (Beam) {
            Photons = false;
            Turbulence = false;
            FLOATPOINT *pDataB = new FLOATPOINT[uiPointCount+30];
            srand((unsigned)time(NULL));
            for (UINT ii=0; ii<uiPointCount+Diameter; ii++)

            {
                pDataB[ii].x = (float)(ii);
                pDataB[ii].y = (float)(-cos((float)((ii)/(m_strDiameter))/16.0f)*10+10);
                //pDataB[ii].y = (float)(10*0.3989*(cos((float)(-ii*ii/2)+sin((float)(-ii*ii/2)))));
            }
            m_LinePlot.SetData(nIndex, pDataB);
            delete pDataB;}

        else if (Turbulence){

```

```

    Beam = false;
    Photons = false;
    FLOATPOINT *pDataT = new FLOATPOINT[uiPointCount+30];
    srand((unsigned)time(NULL));
    for (UINT ii=0; ii<uiPointCount; ii++)

    {
        pDataT[ii].x = (float)(ii+Diameter);
        pDataT[ii].y = (float)((rand() / (RAND_MAX / 13)));
    }
    m_LinePlot.SetData(nIndex, pDataT);
    delete pDataT;}

    else if (Photons){
        Beam = false;
        Turbulence = false;
        FLOATPOINT *pDataP = new FLOATPOINT[uiPointCount];
        srand((unsigned)time(NULL));
        for (UINT ii=0; ii<uiPointCount; ii++)

        {
            pDataP[ii].x = (float)(ii);
            pDataP[ii].y = (float)((rand() / (RAND_MAX / (m_Photon_Second))));
        }
        m_LinePlot.SetData(nIndex, pDataP);
        delete pDataP;}

    else {
        AfxMessageBox("Choose visualisation");
    }
}

}

else {

Diameter = m_output_beam.GetPos();
m_strDiameter = 200e-3+Diameter/100;
Divergence = (4*Wavelength/(PI*m_strDiameter/2));
m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2);
m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
m_transmitter_gain = (8/(m_pointing_error*m_pointing_error))*exp(-2*(m_divergence/m_pointing_error)*
(m_divergence/m_pointing_error));
//m_transmitter_gain = (4*PI*(PI*(m_strDiameter/2)*(m_strDiameter/2))*(2.5636))/(m_wavelength*
m_wavelength);
SetWindowText(_T("Beam Diameter Adjustment is Activated"));

int nIndex = m_LinePlot.Selected();

{
    // get the number of points (this must match for an update).

    UINT uiPointCount = m_LinePlot.GetPointCount(nIndex);
    // is it valid?
    if (uiPointCount>0)
    {
        // declare a new array which will be copied.

        if (Beam) {
            Photons = false;
            Turbulence = false;
            FLOATPOINT *pDataB = new FLOATPOINT[uiPointCount+30];
            srand((unsigned)time(NULL));
            for (UINT ii=0; ii<uiPointCount+Diameter; ii++)

            {
                pDataB[ii].x = (float)(ii);
                pDataB[ii].y = (float)(-cos((float)((ii)/(m_strDiameter))/16.0f)*10+10);
            }
            m_LinePlot.SetData(nIndex, pDataB);

```

```

    delete pDataB;}

    else if (Turbulence){
        Beam = false;
        Photons = false;
        FLOATPOINT *pDataT = new FLOATPOINT[uiPointCount+30];
        srand((unsigned)time(NULL));
        for (UINT ii=0; ii<uiPointCount; ii++)

        {
            pDataT[ii].x = (float)(ii+Diameter);
            pDataT[ii].y = (float)((rand() / (RAND_MAX / 13)));
        }
        m_LinePlot.SetData(nIndex, pDataT);
        delete pDataT;}

    else if (Photons){
        Beam = false;
        Turbulence = false;
        FLOATPOINT *pDataP = new FLOATPOINT[uiPointCount];
        srand((unsigned)time(NULL));
        for (UINT ii=0; ii<uiPointCount; ii++)

        {
            pDataP[ii].x = (float)(ii);
            pDataP[ii].y = (float)((rand() / (RAND_MAX / (m_Photon_Second))));
        }
        m_LinePlot.SetData(nIndex, pDataP);
        delete pDataP;}

    else {
        AfxMessageBox("Choose visualisation");
    }
}
}

}

m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
*1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
m_Photon_Second = m_Rx_Photon*m_strRepetition;
UpdateData(FALSE);
*pResult = 0;

}

void CLinePlotTestDlg::OnDeltaposenergy(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    //UpdateData();
    double Energy,
        Diameter,
        Width,
        PI = 4.0*atan(1.0),
        Intensity = 0.00,
        Power = 0.00,
        Light = 2.998e8,
        Planck = 6.626e-34,
        Wavelength = 532e-9,
        PhotonsTx = m_strEnergy*Wavelength/(Planck*Light);

    if( m_cmbSystem.GetCurSel() == 0 ){
        Diameter = 0.00,
        Energy = 0.00,
        // Width = 0.00,
        // Power = 0.00,
        Light = 0.00,
        Planck = 6.626e-34,
        Wavelength = 532e-9,
        //m_strPower = m_strEnergy*m_strRepetition;
        //Intensity = 2*m_strPower/(PI*(Diameter/2)*(Diameter/2)),

```

```

        PhotonsTx = Energy*Wavelength/(Planck*Light);
        m_strEnergy = Energy;
        //m_strRepetition = Width;
        //m_strTxPhotons = m_strPower/(Light*Planck/m_wavelength);
        m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
        SetWindowText(_T("Laser Pulse Energy Adjustment is not Activated, First Select the System to be used!"));
    }

    else if( m_cmbSystem.GetCurSel() == 1 ){
        Energy = m_total_energy.GetPos();
        m_strEnergy = (Energy - 65536)/100000 + 0.45e-3;
        //m_strRepetition = 1000e-3;
        m_strPower = m_strEnergy*m_strRepetition;
        m_strTxPhotons = m_strPower/(Light*Planck/Wavelength);
        m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
        SetWindowText(_T("Laser Pulse Energy Adjustment Activated"));
    }

    else if( m_cmbSystem.GetCurSel() == 2 ){//m_strDiameter = 1000e-3,
        Energy = m_total_energy.GetPos();
        m_strEnergy = (Energy - 65536)/1000 + 100e-3;
        m_strPower = m_strEnergy*m_strRepetition;
        m_strTxPhotons = m_strPower/(Light*Planck/Wavelength);
        m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
        SetWindowText(_T("Laser Pulse Energy Adjustment Activated"));
    }

    else { //m_strDiameter = 1000e-3,
        Energy = m_total_energy.GetPos();
        m_strEnergy = (Energy - 65536)/1000 + 20e-3;
        m_strPower = m_strEnergy*m_strRepetition;
        m_strTxPhotons = m_strPower/(Light*Planck/Wavelength);
        m_Intensity = 2*m_strEnergy*m_strRepetition/(PI*(m_strDiameter/2)*(m_strDiameter/2));
        SetWindowText(_T("Laser Pulse Energy Adjustment Activated"));
    }

    m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
    m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
    *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
    m_Photon_Second = m_Rx_Photon*m_strRepetition;
    UpdateData(FALSE);
    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposlaserpower(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);

    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposrepreate(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here

    *pResult = 0;
}

void CLinePlotTestDlg::OnEnChangeLaserenergyEdit2()
{
    // TODO: If this is a RICHEDIT control, the control will not
    // send this notification unless you override the CDialog::OnInitDialog()
    // function and call CRichEditCtrl().SetEventMask()
    // with the ENM_CHANGE flag ORed into the mask.

    // TODO: Add your control notification handler code here
}

```

```

void CLinePlotTestDlg::OnDeltaposRxopticseff(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    UpdateData();

    double recieve_optics,
        Planck = 6.626e-34,
        Light = 2.998e8,
        PI = 4.0*atan(1.0);

    if( m_cmbSystem.GetCurSel() == 0 ){
        recieve_optics = 0.00;
        m_receive_optics = recieve_optics;
        SetWindowText(_T("First Select the System to be used!"));
    }

    else if( m_cmbSystem.GetCurSel() == 1 ){
        recieve_optics = m_Rx_Optics.GetPos();
        m_receive_optics = (recieve_optics - 65536)/10 + 40e-2;
        SetWindowText(_T("Receive Optics Efficiency Adjustment is Activated"));
    }

    else if ( m_cmbSystem.GetCurSel() == 2 ){
        recieve_optics = m_Rx_Optics.GetPos();
        m_receive_optics = (recieve_optics - 65536)/10 + 40e-2;
        SetWindowText(_T("Receive Optics Efficiency Adjustment is Activated"));
    }

    else {
        recieve_optics = m_Rx_Optics.GetPos();
        m_receive_optics = (recieve_optics - 65536)/10 + 40e-2;
        SetWindowText(_T("Receive Optics Efficiency Adjustment is Activated"));
    }

    m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
    m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
    *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
    m_Photon_Second = m_Rx_Photon*m_strRepetition;
    UpdateData(FALSE);
    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposQuantumDectetor(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    UpdateData();

    double QuantumDetector,
        Planck = 6.626e-34,
        Light = 2.998e8,
        PI = 4.0*atan(1.0);

    if( m_cmbSystem.GetCurSel() == 0 ){
        QuantumDetector = 0.00;
        m_Quantum_Dect = (QuantumDetector - 65536)/100 + 0.40;
        SetWindowText(_T("First Select the System to be used!"));
    }

    else if( m_cmbSystem.GetCurSel() == 1 ){
        QuantumDetector = m_quantum_detector.GetPos();
        m_Quantum_Dect = (QuantumDetector - 65536)/100 + 0.30;
        SetWindowText(_T("Quantum Detector Efficiency Adjustment is Activated"));
    }

    else if ( m_cmbSystem.GetCurSel() == 2 ){
        QuantumDetector = m_quantum_detector.GetPos();
        m_Quantum_Dect = (QuantumDetector - 65536)/100 + 0.40;
        SetWindowText(_T("Quantum Detector Efficiency Adjustment is Activated"));
    }
}

```

```

    }

    else {
        QuantumDetector = m_quantum_detector.GetPos();
        m_Quantum_Dect = (QuantumDetector - 65536)/100 + 0.40;
        SetWindowText(_T("Quantum Detector Efficiency Adjustment is Activated"));
    }

    m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
    m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
    *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
    m_Photon_Second = m_Rx_Photon*m_strRepetition;
    UpdateData(FALSE);
    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposAtmTrans(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    UpdateData();
    double Atmosphere,
        Planck = 6.626e-34,
        Light = 2.998e8,
        PI = 4.0*atan(1.0);
    Atmosphere = m_Atm.GetPos();
    m_Atmosphere = (Atmosphere-65536)/100 + 0.02;
    m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
    m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
    *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
    m_Photon_Second = m_Rx_Photon*m_strRepetition;
    UpdateData(FALSE);
    SetWindowText(_T("Atmospheric Transmission Adjustment is Activated"));
    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposCloudcover(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    UpdateData();
    double Cloud,
        Planck = 6.626e-34,
        Light = 2.998e8,
        PI = 4.0*atan(1.0);
    Cloud = m_cloud.GetPos();
    m_Cloud = Cloud/10 + 0.1;
    m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
    m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
    *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
    m_Photon_Second = m_Rx_Photon*m_strRepetition;
    UpdateData(FALSE);
    SetWindowText(_T("Cirrus Transmission is Activated"));
    *pResult = 0;
}

void CLinePlotTestDlg::OnDatetimechangeDatetimerpicker1(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMDATETIMECHANGE pDTChange = reinterpret_cast<LPNMDATETIMECHANGE>(pNMHDR);
    // TODO: Add your control notification handler code here
    *pResult = 0;
}

void CLinePlotTestDlg::OnDtnDatetimechangeDatetimerpicker2(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMDATETIMECHANGE pDTChange = reinterpret_cast<LPNMDATETIMECHANGE>(pNMHDR);
    //UpdateData();
    CTime TimeNow;
    TimeNow = CTime::GetCurrentTime();
    UpdateData(FALSE);
}

```

```

    *pResult = 0;
}

void CLinePlotTestDlg::OnDtnDatetimechangeDatetimetypepicker3(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMDATETIMECHANGE pDTChange = reinterpret_cast<LPNMDATETIMECHANGE>(pNMHDR);
    //UpdateData();
    //CString t = CTime::GetCurrentTime().Format("%H:%M:%S");
    // m_Static_Time = t;
    //UpdateData(FALSE);
    *pResult = 0;
}

void CLinePlotTestDlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    int CurPos = m_HScroll.GetScrollPos();

    // Determine the new position of scroll box.
    switch (nSBCode)
    {
    case SB_LEFT:        // Scroll to far left.
        CurPos = 0;
        break;

    case SB_RIGHT:       // Scroll to far right.
        CurPos = 405000;
        break;

    case SB_ENDSCROLL:   // End scroll.
        break;

    case SB_LINELEFT:    // Scroll left.
        if (CurPos > 0)
            CurPos--;
        break;

    case SB_LINERIGHT:   // Scroll right.
        if (CurPos < 405000)
            CurPos++;
        break;

    case SB_PAGELEFT:    // Scroll one page left.
        {
            // Get the page size.
            SCROLLINFO info;
            m_HScroll.GetScrollInfo(&info, SIF_ALL);

            if (CurPos > 0)
                CurPos = max(0, CurPos - (int) info.nPage);
        }
        break;

    case SB_PAGERIGHT:   // Scroll one page right
        {
            // Get the page size.
            SCROLLINFO info;
            m_HScroll.GetScrollInfo(&info, SIF_ALL);

            if (CurPos < 405000)
                CurPos = min(405000, CurPos + (int) info.nPage);
        }
        break;

    case SB_THUMBPOSITION: // Scroll to absolute position. nPos is the position
        CurPos = nPos;      // of the scroll box at the end of the drag operation.
        break;

    case SB_THUMBTRACK:   // Drag scroll box to specified position. nPos is the
        CurPos = nPos;      // position that the scroll box has been dragged to.
        break;
    }
}

```

```

    // Set the new position of the thumb (scroll box).
    m_HScroll.SetScrollPos(CurPos);

    //m_ScrollBarInfo.Format("%d", m_distance);
    //SetDlgItemText(IDC_ScrollBar_Info, m_ScrollBarInfo);

    CDiallog::OnVScroll(nSBCode, nPos, pScrollBar);
}

void CLinePlotTestDlg::OnDeltaposduration(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    *pResult = 0;
}

void CLinePlotTestDlg::OnDeltaposSpin14(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    // TODO: Add your control notification handler code here
    *pResult = 0;
}

void CLinePlotTestDlg::OnNMCustdrawHslider(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMCUSTOMDRAW pNMCD = reinterpret_cast<LPNMCUSTOMDRAW>(pNMHDR);
    // TODO: Add your control notification handler code here
    *pResult = 0;
}

void CLinePlotTestDlg::OnCbnSelchangeCmbStation()
{
    // TODO: Add your control notification handler code here
    UpdateData();

    UpdateData(FALSE);
}

void CLinePlotTestDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    UpdateData();
    CTime CurTime = CTime::GetCurrentTime();
    double ValHours = CurTime.GetHour() - 2;
    double ValMinutes = CurTime.GetMinute();
    double ValSeconds = CurTime.GetSecond();
    int ValDays = CurTime.GetDay();
    int ValMonths = CurTime.GetMonth();
    int ValYears = CurTime.GetYear();
    //double ValUtc = CurTime.GetGmtTm();

    m_Clock_Time = CTime::GetCurrentTime();

    double JD, JDN;
    int A, B, Y, m, D2;
    double Lamb, Bett, Pie, E;

    Y = ValYears;
    int M = ValMonths;
    double D = ValDays + ValHours/24 + ValMinutes/1440 + ValSeconds/86400;

    A = int(Y/100);
    B = 2 - A + int(A/4);

    JD = int(365.25*Y) + int(30.6001*(M+1)) + D + 1720994.5 + B;
    m_T = (JD - 2415020.0)/36525 - 1;

```



```

    m_Lp = (270.434164 + 481267.8831*m_T) - int(int(270.434164 + 481267.8831*m_T)/360)*360;    //Moon's mean longitude ✓
    m_M = (358.475833 + 35999.0498*m_T) - int(int(358.475833 + 35999.0498*m_T)/360)*360;    //Sun's mean anomaly ✓
    m_Mp = (296.104608 + 477198.8491*m_T) - int(int(296.104608 + 477198.8491*m_T)/360)*360;    //Moon's mean anomaly ✓
    m_D = (350.737486 + 445267.1142*m_T) - int(int(350.737486 + 445267.1142*m_T)/360)*360;    //Moon's mean elongation ✓
    m_F = (11.250889 + 483202.0251*m_T) - int(int(11.250889 + 483202.0251*m_T)/360)*360;    //Mean distance of Moon from its ascending node ✓
    E = 23.452294 - 0.0130125*m_T - 0.00000164*m_T*m_T + 0.000000503*m_T*m_T*m_T;

    Lamb = m_Lp + 6.288750*sin(m_Mp) + 1.274018*sin(2*m_D - m_Mp) + 0.658309*sin(2*m_D) + 0.213616*sin(2* m_Mp) - 0.185596*sin(m_M) - 0.114336*sin(2*m_F); ✓
    Bett = 5.128189*sin(m_F) + 0.280606*sin(m_Mp+m_F) + 0.277693*sin(m_Mp - m_F) + 0.173238*sin(2*m_D - m_F) + 0.055413*sin(2*m_D+m_F-m_Mp) + 0.046272*sin(2*m_D-m_F-m_Mp); ✓
    Pie = 0.950724 + 0.051818*cos(m_Mp + 0.009531*cos(2*m_D-m_Mp) + 0.007843*cos(2*m_D) + 0.002824*cos(2* m_Mp) + 0.000857*cos(2*m_D+m_Mp)); ✓

    D2 = 6378.14/sin(Pie/57);
    m_Julian_Day = JD;
    m_Julian_Date = JD - 2400000.5;
    m_direction = m_T;
    //m_distance = int (D2);
    m_HSlider.SetPos(D2);
    m_HScroll.SetScrollPos(D2);

    //m_ScrollBarInfo.Format("%d", D2);
    //SetDlgItemText(IDC_ScrollBar_Info, m_ScrollBarInfo);

    m_ScrollBarInfo.Format(_T("%d"), m_HSlider.GetPos());

    UpdateData(FALSE);

    CDialog::OnTimer(nIDEvent);
}

void CLinePlotTestDlg::OnDtnDatetimechangeClock(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMDATETIMECHANGE pDTChange = reinterpret_cast<LPNMDATETIMECHANGE>(pNMHDR);
    // TODO: Add your control notification handler code here
    *pResult = 0;
}

void CLinePlotTestDlg::OnEnChangeTime()
{
    // TODO: If this is a RICHEDIT control, the control will not
    // send this notification unless you override the CDialog::OnInitDialog()
    // function and call CRichEditCtrl().SetEventMask()
    // with the ENM_CHANGE flag ORed into the mask.
}

void CLinePlotTestDlg::OnEnChangeDistance()
{
    // TODO: If this is a RICHEDIT control, the control will not
    // send this notification unless you override the CDialog::OnInitDialog()
    // function and call CRichEditCtrl().SetEventMask()
    // with the ENM_CHANGE flag ORed into the mask.

    // TODO: Add your control notification handler code here
}

void CLinePlotTestDlg::OnEnChangeDirection()
{
    // TODO: If this is a RICHEDIT control, the control will not
    // send this notification unless you override the CDialog::OnInitDialog()
    // function and call CRichEditCtrl().SetEventMask()
    // with the ENM_CHANGE flag ORed into the mask.
}

```

```
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnLoadFromFile2()
```

```
{
    CHomeDataDlg Dlg;
    Dlg.m_Home_Temp = m_Local_Temp;
    Dlg.m_Home_Press = m_Local_Pressure;
    Dlg.m_Home_Humi = m_Local_Humidity;
    Dlg.m_Home_WindS = m_Local_WindS;
    Dlg.m_Home_WindD = m_Local_WindD;
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnSaveToFile2()
```

```
{
    MoonDataDlg Dlg(AfxGetMainWnd());
    m_Clock_Time = CTime::GetCurrentTime();
    Dlg.m_Local_Time = m_Clock_Time;
    Dlg.m_Moon_Rise = (_T(""));
    Dlg.m_Moon_Set = (_T(""));
    Dlg.m_Moon_Rise_Date = (_T(""));
    Dlg.m_Moon_Set_Date = (_T(""));
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnCurrUpdate2()
```

```
{
    MoonMirrorsDlg Dlg;
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnCurrRemove2()
```

```
{
    DayNightDlg Dlg;
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnPrint2()
```

```
{
    MoonPhaseDlg Dlg;
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnSaveToFile3()
```

```
{
    LaserPathDlg Dlg;
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnPrint3()
```

```
{
    TelescopeDlg Dlg;
    Dlg.DoModal();
}
```

```
void CLinePlotTestDlg::OnBnClickedBtnCurrUpdate3()
```

```
{
    BeamDiameterDlg Dlg;
    Dlg.DoModal();
}
```

```

void CLinePlotTestDlg::OnBnClickedBtnCurrColor2()
{
    CloudCoverDlg Dlg;
    Dlg.DoModal();
}

void CLinePlotTestDlg::OnBnClickedPlotStyle()
{
    // TODO: Add your control notification handler code here
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();

    if (nIndex<0 || nIndex>=m_LinePlot.Count())
        AfxMessageBox("There is no plot selected.");
    else
    {
        int nIndex = m_cmbStyle.GetCurSel();

        if (nIndex>=0 && nIndex<m_cmbStyle.GetCount())
        {
            // change to the enum type.
            CLinePlot::enumPlotStyle enumStyle = (CLinePlot::enumPlotStyle)(nIndex);
            // set the plot type.
            m_LinePlot.SetStyle(nIndex, enumStyle);
        }
    }
}

void CLinePlotTestDlg::OnBnClickedButton1()
{
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();
    double PI = 4.0*atan(1.0);
    double Diameter = m_output_beam.GetPos();
    Beam = true;
    Visuals = true;
    Photons = false;
    Turbulence = false;
    BandPandT = false;

    //if (nIndex<0 || nIndex>=m_LinePlot.Count())
    // AfxMessageBox("First select the system to use!");
    // else
    {
        if (m_LinePlot.Remove(nIndex)!=true)
        {
            // AfxMessageBox("First select the system to use!");
        }
    }

    if (m_cmbSystem.GetCurSel() == 0 ){
        AfxMessageBox("First select the system to use!");
    }

    else if (m_cmbSystem.GetCurSel() == 1 ){
        UINT ii;
        // add a plot to the control.
        srand((unsigned)time(NULL));

        // add a plot to the control.
        FLOATPOINT pData3[101];
        for (ii=0; ii<100; ii++)
        {
            pData3[ii].x = (float)(ii);
            pData3[ii].y = (float)(-cos((float)(ii/m_strDiameter)/16.0f)*10+10);
        }
        COLORREF crColor3 = RGB (255, 0, 0);
        m_LinePlot.Add("Beam", crColor3, CLinePlot::LpLine, pData3, 100);
        m_cmbSystem.SetCurSel(1);
    }
}

```

```

    m_cmbTarget.SetCurSel(1);

}
else {
    UINT ii;
    // CString m_szXCaption, m_szYCaption;
    // m_nSelected = 0;
    // add a plot to the control.
    srand((unsigned)time(NULL));

    // add a plot to the control.
    FLOATPOINT pData3[101];
    for (ii=0; ii<100; ii++)
    {
        pData3[ii].x = (float)(ii);
        //int a = m_strDiameter*100;
        pData3[ii].y = (float)(-cos((float)(ii/m_strDiameter)/16.0f)*10+10);

        //pData3[ii].y = (float)((1000/sqrt(2*PI*a*a))*exp((float)(-(ii - 50)*(ii - 50)/(2*a*a))));
    }

    COLORREF crColor3 = RGB (255, 0, 0);
    m_LinePlot.Add("Beam", crColor3, CLinePlot::LpLine, pData3, 100);
    m_cmbSystem.SetCurSel(2);
    m_cmbTarget.SetCurSel(6);
}
UpdateData(FALSE);

}

void CLinePlotTestDlg::OnBnClickedButton3()
{
    UpdateData();

    // get the current plot index.
    int nIndex = m_LinePlot.Selected();
    Turbulence = true;
    Visuals = true;
    Beam = false;
    Photons = false;
    BandPandT = false;

    //if (nIndex<0 || nIndex>=m_LinePlot.Count())
    // AfxMessageBox("There is no plot selected.");
    // else
    {
        if (m_LinePlot.Remove(nIndex)!=true)
        {
            // AfxMessageBox("Could not remove current plot.");
        }
    }

    if (m_cmbSystem.GetCurSel() == 0 ){
        AfxMessageBox("First select the system to use!");
    }
    else if (m_cmbSystem.GetCurSel() == 1 )
    {
        UINT ii;
        // CString m_szXCaption, m_szYCaption;
        //m_nSelected = 1;
        // add a plot to the control.
        FLOATPOINT pData2[101];
        srand((unsigned)time(NULL));
        // add a plot to the control.
        for (ii=0; ii<101; ii++)
        {
            pData2[ii].x = (float)(ii);
            //pData2[ii].y = (float)((rand() / (RAND_MAX / 2)));
            pData2[ii].y = (float)(exp((float)(ii)));
        }
        COLORREF crColor2 = RGB (0, 0, 255);
        m_LinePlot.Add("Turbulence", crColor2, CLinePlot::LpLine, pData2, 100);
        m_cmbSystem.SetCurSel(1);
    }
}

```

```

    m_cmbTarget.SetCurSel(0);
}

else if (m_cmbSystem.GetCurSel() == 2 ) {
    UINT ii;
// CString m_szXCaption, m_szYCaption;
//m_nSelected = 1;
// add a plot to the control.
FLOATPOINT pData2[101];
srand((unsigned)time(NULL));
// add a plot to the control.
for (ii=0; ii<101; ii++)
{
    pData2[ii].x = (float)(ii);
    // pData2[ii].y = (float)((rand() / (RAND_MAX / 2)));
    pData2[ii].y = (float)(exp((float)(ii)));
}
COLORREF crColor2 = RGB (0, 0, 255);
m_LinePlot.Add("Turbulence", crColor2, CLinePlot::LpLine, pData2, 100);
m_cmbSystem.SetCurSel(3);
//m_cmbTarget.SetCurSel(6);
}

else
{
    UINT ii;
// CString m_szXCaption, m_szYCaption;
//m_nSelected = 1;
// add a plot to the control.
FLOATPOINT pData2[101];
srand((unsigned)time(NULL));
// add a plot to the control.
for (ii=0; ii<101; ii++)
{
    pData2[ii].x = (float)(ii);
    // pData2[ii].y = (float)((rand() / (RAND_MAX / 2)));
    pData2[ii].y = (float)(exp((float)(ii)));
}
COLORREF crColor2 = RGB (0, 0, 255);
m_LinePlot.Add("Turbulence", crColor2, CLinePlot::LpLine, pData2, 100);
m_cmbSystem.SetCurSel(2);
m_cmbTarget.SetCurSel(6);
}

UpdateData(FALSE);
}

void CLinePlotTestDlg::OnBnClickedButton4()
{
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();
    Photons = true;
    Visuals = true;
    Beam = false;
    Turbulence = false;
    BandPandT = false;

    // if (nIndex<0 || nIndex>=m_LinePlot.Count())
    // AfxMessageBox("There is no plot selected.");
    // else
    {
        if (m_LinePlot.Remove(nIndex)!=true)
        {
            // AfxMessageBox("Could not remove current plot.");
        }
    }

    if (m_cmbSystem.GetCurSel() == 0 ){
        AfxMessageBox("First select the system to use!");
    }
}

```

```

else if (m_cmbSystem.GetCurSel() == 1 )
{
    // TODO: Add your control notification handler code here
    UINT ii;

    // add a plot to the control.
    srand((unsigned)time(NULL));
    m_nSelected = 2;
    FLOATPOINT pData1[100];
    for (ii=0; ii<100; ii++)
    {
        pData1[ii].x = (float)(ii);
        pData1[ii].y = (float)(m_Rx_Photon + m_Rx_Photon*(ii/100));
    }
    COLORREF crColor1 = RGB (0, 255, 0);
    m_LinePlot.Add("Photons", crColor1, CLinePlot::LpLine, pData1, 100);
    m_cmbSystem.SetCurSel(1);
    m_cmbTarget.SetCurSel(0);
}

else if (m_cmbSystem.GetCurSel() == 2 ) {
    // TODO: Add your control notification handler code here
    UINT ii;
    CString m_szXCaption, m_szYCaption;
    // add a plot to the control.
    srand((unsigned)time(NULL));
    m_nSelected = 2;
    FLOATPOINT pData1[100];
    for (ii=0; ii<100; ii++)
    {
        pData1[ii].x = (float)(ii);
        pData1[ii].y = (float)(m_Rx_Photon + m_Rx_Photon*(ii/100));
        //pData1[ii].y = (float)(sin((float)(ii-50))/((float)ii-50));
    }
    COLORREF crColor1 = RGB (0, 255, 0);
    m_LinePlot.Add("Photons", crColor1, CLinePlot::LpLine, pData1, 100);
    m_cmbSystem.SetCurSel(2);
    m_cmbTarget.SetCurSel(6);
}

else
{
    // TODO: Add your control notification handler code here
    UINT ii;
    CString m_szXCaption, m_szYCaption;
    // add a plot to the control.
    srand((unsigned)time(NULL));
    m_nSelected = 2;
    FLOATPOINT pData1[100];
    for (ii=0; ii<100; ii++)
    {
        pData1[ii].x = (float)(ii);
        pData1[ii].y = (float)(m_Rx_Photon + m_Rx_Photon*(ii/100));
        //pData1[ii].y = (float)(sin((float)(ii-50))/((float)ii-50));
    }
    COLORREF crColor1 = RGB (0, 255, 0);
    m_LinePlot.Add("Photons", crColor1, CLinePlot::LpLine, pData1, 100);
    m_cmbSystem.SetCurSel(3);
    m_cmbTarget.SetCurSel(11);
}

m_szXCaption = "Atmospheric transmission (%)";
m_szYCaption = "Number of Photons";
UpdateData(FALSE);
}

void CLinePlotTestDlg::OnBnClickedBandt()
{
}

void CLinePlotTestDlg::OnBnClickedBandp()

```

```

{
}

void CLinePlotTestDlg::OnBnClickedBandpandt()
{
    // TODO: Add your control notification handler code here
    // get the current plot index.
    int nIndex = m_LinePlot.Selected();
    BandPandT = true;
    Visuals = false;

    //if (nIndex<0 || nIndex>=m_LinePlot.Count())
    // AfxMessageBox("There is no plot selected.");
    // else
    {
        if (m_LinePlot.Remove(nIndex)!=true)
        {
            // AfxMessageBox("Could not remove current plot.");
        }
    }

    if (m_cmbSystem.GetCurSel() == 0 ){
        AfxMessageBox("First select the system to use!");
    }

    else {

        UINT ii;
        CString m_szXCaption, m_szYCaption;
        m_nSelected = 0;
        // add a plot to the control.
        srand((unsigned)time(NULL));

        // add a plot to the control.
        FLOATPOINT pData3[100];
        for (ii=0; ii<100; ii++)
        {
            pData3[ii].x = (float)(ii);
            pData3[ii].y = (float)(-cos((float)(ii/m_strDiameter)/16.0f)*10+10);
        }
        COLORREF crColor3 = RGB (255, 0, 0);
        m_LinePlot.Add("Beam", crColor3, CLinePlot::LpLine, pData3, 100);

        FLOATPOINT pData1[101];
        for (ii=0; ii<101; ii++)
        {
            pData1[ii].x = (float)(ii);
            pData1[ii].y = (float)((rand() / (RAND_MAX / 13)));
        }
        COLORREF crColor1 = RGB (0, 255, 0);
        m_LinePlot.Add("Photons", crColor1, CLinePlot::LpLine, pData1, 100);

        FLOATPOINT pData2[101];
        for (ii=0; ii<101; ii++)
        {
            pData2[ii].x = (float)(ii);
            pData2[ii].y = (float)((rand() / (RAND_MAX / 13)));
        }
        COLORREF crColor2 = RGB (0, 0, 255);
        m_LinePlot.Add("Turbulence", crColor2, CLinePlot::LpLine, pData2, 100);
        m_cmbSystem.SetCurSel(2);
        m_cmbTarget.SetCurSel(6);
        UpdateData(FALSE);
    }

}

void CLinePlotTestDlg::OnEnChangeBeamdiameterEdit()
{

```

```

// TODO: If this is a RICHEDIT control, the control will not
// send this notification unless you override the CDialog::OnInitDialog()
// function and call CRichEditCtrl().SetEventMask()
// with the ENM_CHANGE flag ORed into the mask.

// TODO: Add your control notification handler code here
}

```

```

void CLinePlotTestDlg::OnDeltaposAtmTrans3(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMUPDOWN pNMUpDown = reinterpret_cast<LPNMUPDOWN>(pNMHDR);
    double Angle, Planck = 6.626e-34, Light = 2.997e8, PI = 4.0*atan(1.0);
    UpdateData();
    Angle = m_IncidentAngle.GetPos();
    m_Difference = (Angle - 65536)*30 - 30;
    if (m_cmbTarget.GetCurSel() == 0 )
    {
        m_Min = 0;
        m_Max = 0;
        m_Cross_Section = 0;
    }

    if (m_cmbTarget.GetCurSel() == 1)
    {
        m_Min = 60e6;
        m_Max = 60e6;
        m_Cross_Section = 60e6;
    }

    if (m_cmbTarget.GetCurSel() == 2)
    {
        m_Min = 360e6;
        m_Max = 360e6;
        m_Cross_Section = 360e6;
    }

    if (m_cmbTarget.GetCurSel() == 3)
    {
        m_Min = 40e6;
        m_Max = 40e6;
        m_Cross_Section = 40e6;
    }

    if (m_cmbTarget.GetCurSel() == 4)
    {
        m_Min = 7e6;
        m_Max = 7e6;
        m_Cross_Section = 7e6;
    }

    if (m_cmbTarget.GetCurSel() == 5)
    {
        m_Min = 0.65e6;
        m_Max = 0.65e6;
        m_Cross_Section = 0.65e6;
    }

    if (m_cmbTarget.GetCurSel() == 6)
    {
        if (m_Difference == -30) {
            m_Min = 9.6e3;
            m_Max = 9.56e8;
            m_Cross_Section = 3.32e8;
        }

        else if (m_Difference == 0) {
            m_Min = 7.34e3;
            m_Max = 7.96e8;
            m_Cross_Section = 2.20e8;
        }
    }
}

```



```

else {
    m_Min = 7.34e3;
    m_Max = 7.96e8;
    m_Cross_Section = 2.20e8;
}

    if (m_cmbTarget.GetCurSel() == 7)
    {
        if (m_Difference == -30) {
            m_Min = 9.6e3;
            m_Max = 9.56e8;
            m_Cross_Section = 3.32e8;
        }

        else if (m_Difference == 0) {
            m_Min = 7.34e3;
            m_Max = 7.96e8;
            m_Cross_Section = 2.20e8;
        }

        else {
            m_Min = 7.34e3;
            m_Max = 7.96e8;
            m_Cross_Section = 2.20e8;
        }

        if (m_cmbTarget.GetCurSel() == 8)
        {
            if (m_Difference == -30) {
                m_Min = 2.88e4;
                m_Max = 2.87e9;
                m_Cross_Section = 9.97e8;
            }

            else if (m_Difference == 0) {
                m_Min = 2.46e4;
                m_Max = 2.39e9;
                m_Cross_Section = 6.59e8;
            }

            else {
                m_Min = 1.47e4;
                m_Max = 2.87e9;
                m_Cross_Section = 5.26e8;
            }
        }

        m_Rx_Photon = m_Quantum_Dect*(m_strEnergy*m_wavelength/(Planck*Light))*m_transmit_optics*
        m_transmitter_gain*m_Cross_Section*((1/((4*PI*m_SlantRange*1e3*m_SlantRange*1e3)))*(1/(4*PI*m_SlantRange
        *1e3*m_SlantRange*1e3)))*m_Aperture*m_receive_optics*m_Atmosphere*m_Cloud;
        m_Photon_Second = m_Rx_Photon*m_strRepetition;
        UpdateData(FALSE);
        SetWindowText(_T("Incident Angle Adjustment is Activated"));
        *pResult = 0;
    }
}

void CLinePlotTestDlg::OnBnClickedReset()
{
    UpdateData();
    int nIndex = m_LinePlot.Selected();
    if (m_LinePlot.Remove(nIndex)!=true)
    {
        // AfxMessageBox("First select the system to use!");
    }
    double Diameter = 0.00,
           Energy    = 0.00,
           Width     = 0.00,
           RepRate   = 0.00,
           Power     = Energy*Width,
           Duration  = 0.00,
           Calculate  = 0.00,

```

```

        Wavelength = 532e-9,
        PI          = 4.0*atan(1.0),
        Planck      = 6.626e-34,
        Light       = 2.998e8,
        Divergence  = 0.00,
        transmitter_gain,
        transmit_optics = 0.00,
        receive_optics = 0.00,
        PhotonsTx,
        Pointing,
        TxGain,
        Temperature,
        Pressure,
        QuantumDetector,
        Humidity = 0.00,
        Cloud,
        Aperture = 0.00,
        Cross_section,
        Range     = 0.5*m_strTime_Of_Flight*Light,
        Atmosphere;

// SYSTEMTIME st;
// GetLocalTime(&st);
// m_Static_Time.Format("%H:%M");
// CString t = CTime::GetCurrentTime().Format("%H:%M:%S");

//CString t = ColeDateTime::GetCurrentTime().Format("%H:%M:%S");
//m_Static_Time = timeDest;
    Diameter = 0.000;
    Energy    = 0.00,
    Width     = 0.00,
    Power     = Energy*Width,
    Duration  = 0.00,
    Wavelength = 0,
    transmit_optics = 0.00,
    receive_optics = 0.00,
    Divergence = 0.00,
    m_divergence = 4*1.3*m_wavelength/(PI*m_strDiameter/2),
    Pointing = 0.00,
    m_pointing_error = Pointing,
    PhotonsTx = Energy*Wavelength/(Planck*Light);
    m_Intensity = 0.00,
        Temperature = 0.00,
        Pressure     = 0.00,
        Humidity     = 0.00,
        QuantumDetector = 0.00,
        Atmosphere   = 0.00;
    Cloud = 0.00;
    Range = 0.00;
    m_transmitter_gain = 0.00;
    m_strTxPhotons = PhotonsTx;
    m_strDiameter = Diameter;
    m_Min = 0.00;
    m_Max = 0.00;
m_strRepetition = Width;
m_strPower = Power;
m_strWidth = Duration;
m_strEnergy = Energy;
m_wavelength = Wavelength;
m_transmit_optics = transmit_optics;
m_receive_optics = receive_optics;
m_Quantum_Dect = QuantumDetector/100;
m_Atmosphere = Atmosphere;
m_Cloud = Cloud;
m_Aperture = Aperture;
m_Cross_Section = 0.00;
m_Rx_Photon = 0.00;
m_divergence = 0.0;
m_Rx_Photon = 0.00;
m_Photon_Second = 0.00;
m_SlantRange = 0.00;
m_cmbSystem.SetCurSel(0);
m_cmbTarget.SetCurSel(0);
SetWindowText(_T("System Reset!"));

```

```
    UpdateData(FALSE);
}

void CLinePlotTestDlg::OnBnClickedPhotonsdata()
{
    PhotonsData Dlg;
    Dlg.DoModal();
}

void CLinePlotTestDlg::OnBnClickedPredictions()
{
    PredictionsDlg Dlg;
    Dlg.DoModal();
}

void CLinePlotTestDlg::OnBnClickedHelp()
{
}
```

