

**HIGH ACCEPTANCE ANGLE OPTICAL FIBER BASED SOLAR DAY-
LIGHTING SYSTEM USING TWO-STAGE REFLECTIVE NON-IMAGING
DISH CONCENTRATOR**

WONG CHONG CHIAT

**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Engineering
(Hons.) Electrical and Electronic Engineering**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

January 2016

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : Wong Chong Chiat

ID No. : 11UEB01769

Date : 16/4/2016

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**HIGH ACCEPTANCE ANGLE OPTICAL FIBER BASED SOLAR DAY-LIGHTING SYSTEM USING TWO-STAGE REFLECTIVE NON-IMAGING DISH CONCENTRATOR**” was prepared by **WONG CHONG CHIAT** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : Prof. Dr Chong Kok Keong

Date : _____

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2016, Wong Chong Chiat. All right reserved.

ACKNOWLEDGEMENTS

First of all, I would like to thank everyone who had contributed to the successful completion of my Final Year Project. I would like to express my deep sense of gratitude to my research supervisor, Prof. Chong Kok Keong for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would like to thank my advisor and co-supervisors for their guidance, keen interest and encouragement at various stages of my project development.

Last but not least, I would like to express my sincere gratitude to my beloved parents and friends who always give me encouragements when I need it. They become my motivation in the progress of completing this project with full of passion.

HIGH ACCEPTANCE ANGLE OPTICAL FIBER BASED SOLAR DAY-LIGHTING SYSTEM USING TWO-STAGE REFLECTIVE NON-IMAGING DISH CONCENTRATOR

ABSTRACT

Solar energy is the most abundant and reliable renewable energy that can help to overcome the problem of increasing demand of energy around the world, especially in Malaysia, which has high amount of average solar radiation throughout the year. Solar energy is greatly used in generating electricity however the process of converting solar energy into electricity inevitably causes power losses. The concept of solar day-lighting system is to gather sunlight during the day and transfer it into indoor building to save energy for lighting. Solar day-lighting is a very common application in green building because lighting is one of the main energy consumption in high rise building even during day time. In this project a prototype of daylighting system was constructed by using high acceptance angle optical fiber coupled to a two-stage reflective non-imaging dish concentrator. Solar tracking system is one of the important feature of the prototype to increase efficiency and the main focus of study in this report. Combining daylighting system with solar concentrator can have better efficiency as compared to typical day-lighting system. The finding and tracking accuracy of the prototype were discussed in this report.

TABLE OF CONTENTS

DECLARATION	ii
APPROVAL FOR SUBMISSION	iii
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF SYMBOLS / ABBREVIATIONS	xiv
LIST OF APPENDICES	xv

CHAPTER

1	INTRODUCTION	1
1.1	Background	1
1.2	Problem Statements	2
1.3	Aims and Objectives	2
2	LITERATURE REVIEW	3
2.1	Basic Concept of Daylighting	3
2.2	Daylighting Systems	3
2.2.1	Passive Daylighting	4
2.2.2	Active Daylighting	5
2.3	Types of Sun Trackers	7
2.3.1	One-axis Tracking System	7
2.3.2	Two-axis Tracking System	8

2.4	Comparison of Solar Tracking System	9
2.5	Solar Concentrator	12
3	METHODOLOGY	14
3.1	Solar Concentrator Design	14
3.1.1	Simulation Results	15
3.1.2	Prototype Construction	17
3.1.3	Experimental Results	18
3.2	Solar Tracking Mechanism	20
3.2.1	Azimuth-axis Tracking	21
3.2.2	Elevation Tracking	22
3.3	Equation for Solar Tracking	23
3.4	Derivation of Equation	31
3.4.1	Derivation of Elevation Tracking Equation	31
3.4.2	Derivation of Azimuth Tracking Equation	35
3.5	Software Development	36
3.5.1	Timer Interrupt	36
3.5.2	Motor Pulse	38
3.5.3	Measuring Backlash	40
3.5.4	Measuring Tracking Accuracy	46
3.5.5	Overall Solar Tracking Program	48
3.6	Circuit Design	52
4	RESULTS AND DISCUSSION	53
4.1	Backlash Measurement Result	53
4.2	Tracking Accuracy	54
4.3	Solar Tracking Improvement	59
5	CONCLUSION AND RECOMMENDATIONS	64
5.1	Conclusion	64
5.2	Recommendations	65
	REFERENCES	66

APPENDICES**68**

LIST OF TABLES

TABLE	TITLE	PAGE
4.1	Backlash Error Results	53
4.2	6 th April 2016 Tracking Results	56
4.3	7 th April 2016 Tracking Results	57
4.4	8th April 2016 Tracking Results	58

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	Solatube Daylighting System (solatube.com, N.D.)	4
2.2	Top View of Simple Closed Loop Tracking (Geo Bruce, N.D.)	6
2.3	Side View of Simple Closed Loop Tracking (Geo Bruce, N.D.)	6
2.4	Types of One-Axis Tracker (Chuck Juda, 2013)	8
2.5	Two-Axis Tracker Designs (Chuck Juda, 2013)	8
2.6	The Daily Energy Captured by DAST (Dual-axe solar system) and SSS (Static Solar System) for a month (Lee and Nasrudin, 2013)	9
2.7	Electrical Power Generation in a Sunny Day (Lee and Nasrudin, 2013)	10
2.8	Electrical Power Generation in a Cloudy Day (Lee and Nasrudin, 2013)	10
2.9	Comparison of Fixed Mount and One-Axis Tracker System (Deepthi, Ponni, Ranjitha and Dhanabal, 2013)	11
2.10	Comparison of Fixed Mount and Two-Axis Tracker System (Deepthi, Ponni, Ranjitha and Dhanabal, 2013)	11
2.11	Setup of the Sunlight Collectng System for (a) Parabolic Mirror and (b) Fresnel Lens (Irfan and Seoyong, 2012)	13
2.12	Proposed Daylighting System (Irfan and Seoyong, 2013)	13

3.1	Solar Concentrator Initial Design	14
3.2	Cassegrain Reflector (a-levelphysics.com, N.D.)	15
3.3	Simulated Cassegrain Reflector using OpticalRayTracer	16
3.4	Two Stage Reflector Secondary Design	17
3.5	SolidWorks Drawing Prototype for Primary Reflector	18
3.6	Constructed Primary Reflector Prototype	18
3.7	Experiment Setup	19
3.8	Focused Solar Image after Secondary Reflection	19
3.9	Solar Image after Primary Reflection	20
3.10	Final Mechanical Tracking System Design	20
3.11	Azimuth-axis Rotation Mechanism	21
3.12	Encoder Using IR Sensors	22
3.13	Places to Measure the Elevation Angle	22
3.14	Illustration of Sun and Equatorial Plane	24
3.15	Illustration of Sun Location to Observer Q Location	27
3.16	Overall Illustration of Sun Position Translation	29
3.17	Flowchart for Azimuth and Altitude calculation	30
3.18	Linear Actuator extend and form Triangular Shape	32
3.19	Illustration of Linear Actuator Shaft, A' and Elevation angle, θ_E	33
3.20	Illustration of Incoming Sunlight forms Right Angle Triangle	33
3.21	Linear actuator Model EAC4R-E30-ARMKD-3	35
3.22	Motor Properties taken from Datasheet	35
3.23	Stepper Motor for Azimuth-axis Tracking	36

3.24	Flow Chart of Timer Interrupt	37
3.25	Flow Chart to Control Azimuth Stepper Motor	39
3.26	Flow Chart to Control Linear Actuator Stepper Motor	40
3.27	Illustration of Backlash	41
3.28	U-shaped Microphotoelectric Sensor	41
3.29	SolidWorks Drawing of Sensor, Sensor plate and Secondary Gear	42
3.30	Mounting of Sensor, Sensor plate and Secondary Gear	42
3.31	Programming Flow Chart to Measure Backlash	45
3.32	Bar Mounted on Solar Concentrator	46
3.33	Flow Chart to Measure Tracking Accuracy	47
3.34	Flow Chart to reset The Solar Concentrator to Initial Position	48
3.35	Flow Chart of Solar Tracking	50
3.36	Flow Chart for Overall Tracking System	51
3.37	Overall Circuit Design	52
4.1	Digital Compass in iPhone 6	56
4.2	Light-dependant Resistors fixed on Sensor Holder	59
4.3	Top View and Labelling of LDRs	60
4.4	Flow Chart of Closed-loop Tracking	61
4.5	Overall Hybrid Tracking System	62

LIST OF SYMBOLS / ABBREVIATIONS

δ	declination angle, degrees
ω	hour angles, degrees
t_s	solar time, hours
x	function of the day number, degrees
A	azimuth angle, degrees
α	altitude angle, degrees
θ_z	zenith angle, degrees
Φ	latitude, degrees
θ_E	elevation angle, degrees
A'	linear shaft length, mm
S	arc length of a secondary gear, mm
r	radius of secondary gear, mm
θ	angle of arc, radians
LC	longitude correction, hours
LCT	local clock time, 24-hour
EOT	equation of time, minutes
N	day number, days
D	daylight saving

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Source Code for Backlash Measurement	68
B	Overall Source Code	70

CHAPTER 1

INTRODUCTION

1.1 Background

Due to fast growing of energy demand and depletion of fossil fuels, many alternative sources of energy has been used and utilize. Solar energy is one of the most abundant renewable energy that can be found on earth. This renewable energy can ultimately replace all the non-renewable energy such as natural gas if the efficiency of solar conversion can be maximize.

As Malaysia is steadily becoming a developed country, the demand of energy increases as well. This has force authorities to find more energy sources to meet the demand. Since Malaysia is located at the equatorial line, there is more than 10 hours sunlight available per day. It is possible to get around 6 hours of direct sunlight irradiation of a range from 800W/m^2 to 1000W/m^2 as well as highly predictable weather condition (Amin, Lung and Sopian, 2009). Therefore solar energy is an obvious choice for energy source replacement in Malaysia. Solar energy is widely used in photovoltaic system to convert it to electrical energy. Besides of this application, solar energy can also use in daylighting which collect daylight directly and use in indoor buildings. This method is believe to be more efficient than converting light energy to electrical energy and back to light energy. There will be a lot of energy losses during electrical energy transmission and conversion process.

1.2 Problem Statements

Solar daylighting provide a onetime cost for energy crisis problem, however it is in low progress in the market due to several issues such as high initial cost, utilization difficulties, application limitation, technology challenges etc. (Mayhoub, 2014). There is a need of a best design that has high efficiency and simple architecture. The purpose of this project is to design and develop a lowest possible cost and highest possible efficiency daylighting system by using solar concentrator, optical fibre and solar tracker. The main focus on this report will be solar tracking.

1.3 Aims and Objectives

The objectives of the thesis are stated as following:

- (i) To design two-stage reflective non-imaging dish concentrator and its associated daylighting transmission system.
- (ii) To analyse the performance of an optical-fiber based day-lighting transmission system using non-imaging dish concentrator.
- (iii) To design and develop high accuracy solar tracker.
- (iv) To evaluate the feasibility of newly proposed day-lighting system to be integrated into the roof of building.

CHAPTER 2

LITERATURE REVIEW

2.1 Basic Concept of Daylighting

Solar daylighting is an application that collect outdoor sunlight and transmit it into indoor building for visual purpose. Since there is outdoor lighting, the indoor artificial lighting such as light bulbs and light emitting diode can be reduced to save power consumption. Furthermore, comparing to artificial lighting, solar daylighting does not has conversion process which is much more efficient than artificial lighting. Solar daylighting concept had been widely used in building design for pass few decades, the common practice is to place windows facing the sun as more time as possible during the day. Reflective surfaces are also used in indoor building so that light can be reflected as deep as possible. Solar daylighting is common application in green building. Solar daylighting is important to reduce energy usage because lighting one of the main energy consumption in building. Besides of energy saving, research has also shown that sunlight has positive effects on human body. It can reduce seasonal affective disorder and other illnesses (A. Dunne, 1989).

2.2 Daylighting Systems

There are two major types of daylighting system which is an active daylighting system and passive daylighting system. In general, both daylighting systems have the same objective that is to collect outdoor daylight and use for indoor lighting. However both

systems use different kind of approach to accomplish the objective. Several research had been carried out to maximize the efficiency for both systems.

2.2.1 Passive Daylighting

Passive daylighting is a system that collect sunlight using non-mobile or static system. Since there is no mechanical movement involves, therefore it does not consist of any tracking system. The efficiency of light transmission into indoor building is greatly dependant on the design of collecting device, material used and position of the system. One of the example that used to reflect the collected sunlight deep into building is a light shelf. Another example of passive daylighting system is a Solatube Daylighting System shown in figure 1 which use a dome-shaped collector and tubular daylighting device to transmit light at minimal heat transfer. The advantage of passive daylighting is that it is less complicated and cheaper compared to active daylighting system.



Figure 2.1: Solatube Daylighting System (solatube.com, N.D.)

2.2.2 Active Daylighting

Different from passive daylighting, active daylighting involves of mechanical mechanism to track and follow the sunlight and improve efficiency of light collection for lighting purpose. Active daylighting is not stationary and works together with solar tracker to line the normal of daylighting system perpendicular with the direction of sunlight eliminating the cosine effect of the system. Cosine effect is defined as the angle between the normal of daylighting system to and the incoming direction of sunlight. There are two tracking method of active daylighting systems which is a closed loop system and an open loop system.

Closed loop system involves of active sensors that capture sunlight illumination at all times. When the sun move to away from the sensor, a feedback signal is generated to readjust the mechanical device such as motors to align the daylighting system facing perpendicularly back to the direction of sunlight. In 2006, Luque-Heredia et al. (2006) developed a sun-tracking error monitoring system that uses a monolithic optoelectronic sensor for a concentrator photovoltaic system that achieved tracking accuracy of better than 0.1° according to the result from the case study. However since sensors read the sunlight directly, the accuracy of this system will be greatly affected the change of weather. Furthermore the sensors used might be expensive which in turns increase the overall cost of the system. For example, Kribus et al. (2004) designed a system that used four CCD cameras to improve the pointing error of solar image up to 0.1 mrad. CCD camera are rather expensive and complicated to use. Figure 2 and figure 3 shows an example of simple closed loop daylighting system using 4 light dependant resistors.

Open loop system track the sun without physically using sensors like closed loop system. Instead, mathematical formula or algorithm is used to calculate the position of sun base on daylighting system geographical information, date and time. Open loop system includes of encoder that reads the actuators positions or angle so that correct adjustment to the daylighting system can be performed. This system is normally used together with a programmable logic controller to compute the algorithm and controls the sun tracker.

In general, a closed loop system provides a better accuracy compare to open loop system but weather change such as cloud seriously affects its accuracy. Closed loop system could be more expensive than open loop due to the sensor used. Open loop system has lower accuracy but sun tracking accuracy is affected by weather. Open loop system is a better choice in this project as it will be placed outdoor to gather sunlight during day time.

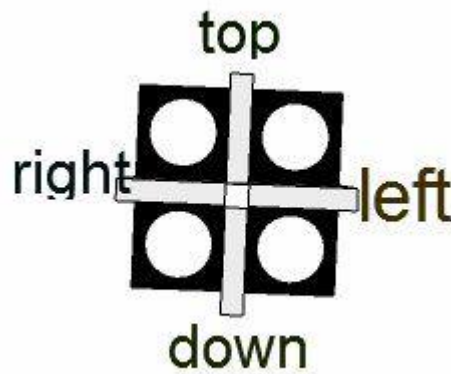


Figure 2.2: Top View of Simple Closed Loop Tracking (Geo Bruce, N.D.)

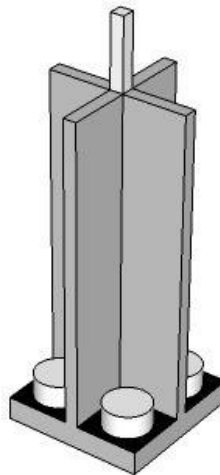


Figure 2.3: Side View of Simple Closed Loop Tracking (Geo Bruce, N.D.)

2.3 Types of Sun Trackers

Typical solar daylighting system using windows has low efficiency and the light output greatly affected by position of sun. To overcome this issue, a solar tracker is included in the system. As discussed in previous section, open loop tracking system is generally better for this project as weather plays a crucial part in the application. To utilize maximum sunlight energy collection during the day, the position of sun in respect to position of daylighting on earth must be well understood and studied. There are two types of sun tracker. The first is the one-axis sun tracker and the second is a two-axis sun tracker.

2.3.1 One-axis Tracking System

One-axis tracker can only tilt itself in a single axis following the movement of sunlight. One-axis tracker is separated into three different category which is a vertical axis tracker, horizontal axis tracker and tilted axis tracker. Vertical axis tracker always line its axis vertical to the earth surface or normal of the earth surface. It can rotate according to the zenith axis and it is known as azimuth sun tracker. Horizontal axis tracker means that it axis is always horizontal and parallel with the surface of the earth, it tilts itself along East-West or North-South direction following the sun. For tilted axis tracker, the tilting method is close to horizontal axis tracker, just that the tracking axis is adjusted to a certain pre-set angle along North-South direction before tilting. Figure 4 show illustration of three different types of one-axis tracker. The advantage of one-axis tracker is that it uses one less motor compare to two-axis tracker as it only have to rotate at single axis. However the cosine effect of one-axis tracker could be large due to annual motion of the earth and sun in north-south direction depending on seasons. This cause solar daylighting system to not always perpendicular to the sun.



Figure 2.4: Types of One-Axis Tracker (Chuck Juda, 2013)

2.3.2 Two-axis Tracking System

For two-axis tracker, it is divided into two types, azimuth-elevation tracker and polar tracker. Two-axis tracker basically can rotate in two different axes to align the solar system perpendicular to the sun's position for maximum light collection no matter where the solar system is located. In other words, two-axis tracker has a higher degree of flexibility and accuracy in directional pointing. It is basically a combination of horizontal-axis tracker and vertical axis tracker. The cosine effect is completely eliminated in ideal case. However, two-axis tracker requires two actuators for each axis, thus the cost and power consumption of two-axis tracker is higher compared to one-axis tracker. The control for this system is also much more complicated than one-axis tracker. Figure 5 illustrates a typical design of two-axis tracker.

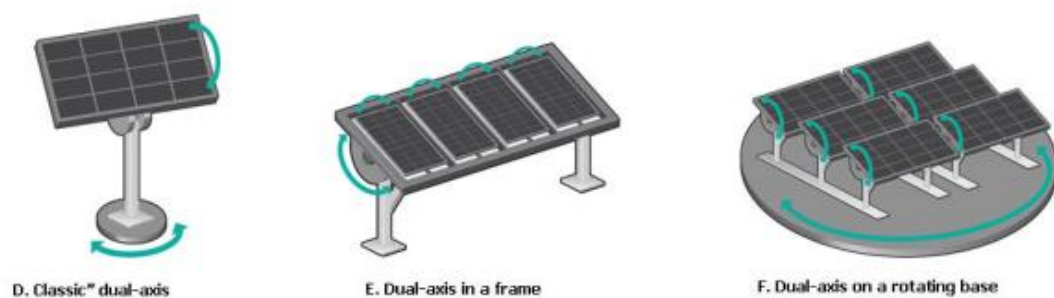


Figure 2.5: Two-Axis Tracker Designs (Chuck Juda, 2013)

2.4 Comparison of Solar Tracking System

Many researches had been carried out to study and enhance the accuracy of different solar tracking systems. In year 2013, Lee et al. (2013) carried out an experiment to study the performance of dual-axis solar tracker and static solar system in Malaysia. The dual axis tracker is able to rotate from 70° east to 70° west and also from 30° north to 30° south. An open loop tracking system is used for their experiment since power out of solar panel during cloudy day was also studied. As a result, the efficiency of dual-axis solar tracker 82.12% and 24.91% higher than static solar tracking system during sunny day and cloudy day respectively. Figure 6 below indicates the testing data throughout a month. Figure 7 indicates electrical power generation in a sunny and figure 8 indicates electrical power generation during cloudy day.

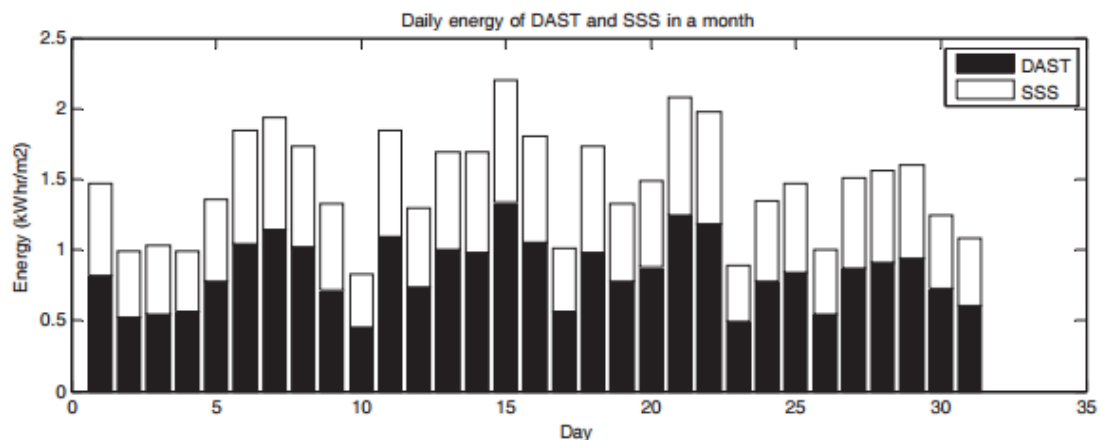


Figure 2.6: The Daily Energy Captured by DAST (Dual-axe solar system) and SSS (Static Solar System) for a month (Lee and Nasrudin, 2013)

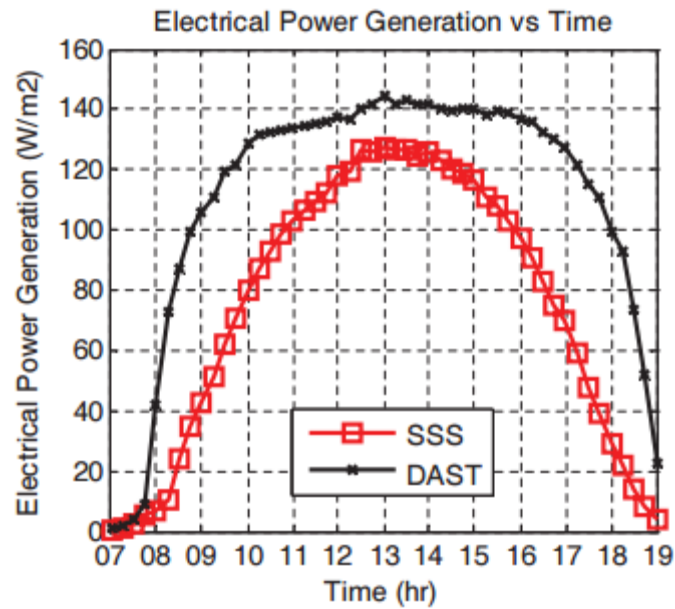


Figure 2.7: Electrical Power Generation in a Sunny Day (Lee and Nasrudin, 2013)

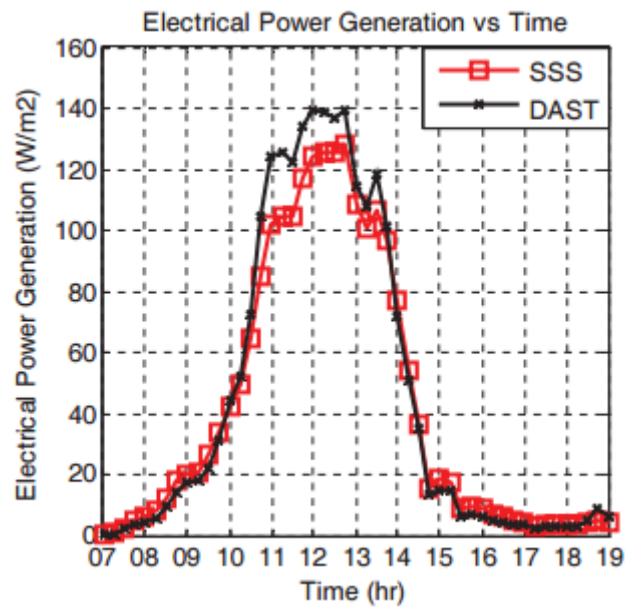


Figure 2.8: Electrical Power Generation in a Cloudy Day (Lee and Nasrudin, 2013)

In year 2013, Deepthi et al. (2013) has conducted a study to compare the efficiency of single-axis and dual-axis with fixed mount. Both solar tracking use an

open loop system where one-axis system used light dependant resistor and two-axis system used photo transistor sensors. Deepthi et al. (2013) conclude that one-axis tracker had 13% better efficiency than fixed mount while two-axis tracker 25% better efficiency than fixed mount. Base on Deepthi et al. results, it can also be conclude that two-axis tracker has better efficiency than one-axis tracker even during cloudy days. Figure 9 shows the comparison result between one-axis tracker and fixed mount. Figure 10 shows the comparison result between two-axis tracker and fixed mount.

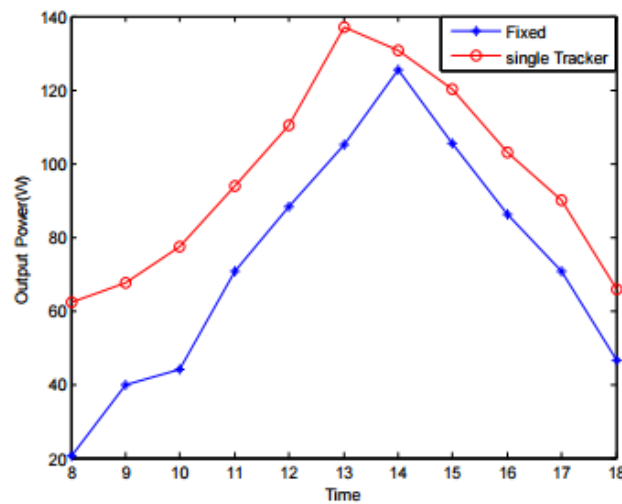


Figure 2.9: Comparison of Fixed Mount and One-Axis Tracker System
(Deepthi, Ponni, Ranjitha and Dhanabal, 2013)

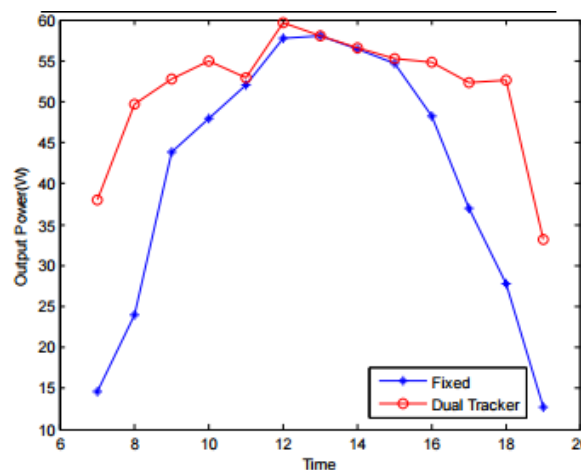


Figure 2.10: Comparison of Fixed Mount and Two-Axis Tracker System
(Deepthi, Ponni, Ranjitha and Dhanabal, 2013)

Base on the comparison studies carried out by two different groups of researchers, the best solar tracking system in terms of efficiency is two-axis solar tracker follow by one-axis tracker. Although static solar system is the least preferred due to its low power output, the price is cheapest and the operation is least complicated.

2.5 Solar Concentrator

Besides of utilizing solar tracking system to maximize light collection in daylighting system, a solar concentrator can also improves overall lighting efficiency. Most of the applications for solar concentrator is in solar heating and photovoltaic system. Research on designing of solar concentrator for daylighting purpose is also growing.

One of the common solar concentrator is the parabolic mirror concentrator. In photovoltaic system, only one parabolic contractor is required however in daylighting two parabolic dish is required due to the blockage of light by optics fiber. Two stage parabolic mirror also face an issue of light absorption by the supporting rod causing some light energy loss. Another solar concentrator for daylighting system is use of two stage Fresnel lens that concentrate the sunlight into optical fibre. Fresnel lens solar concentrator eliminates the shadow effect in two stage parabolic mirror concentrator. Comparison study was carried out by Irfan et al. and Seoyong et al. (2012) on these two different concentrators. The average illuminance for Fresnel lens obtained by them is 705 lx and 675lx for parabolic mirror. The difference between these two was only 4.2%. Therefore the light efficiency and performance for both systems is close. Figure 11 illustrate two different collecting system used by Irfan and Seoyong et al.

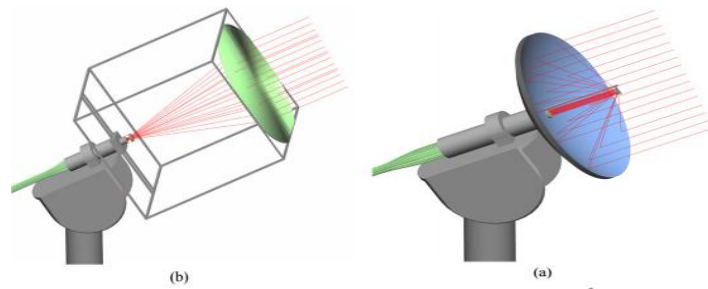


Figure 2.11: Setup of the Sunlight Collecting System for (a) Parabolic Mirror and (b) Fresnel Lens (Irfan and Seoyong, 2012)

Another approach of concentrating sunlight is by using multiple flat reflecting surface, focusing multiple sun rays into single reflector and reflect back into the light transmission system. Flat reflecting surface such as mirror is generally cheaper to manufacture compare to Fresnel lens and parabolic mirror. For example Irfan et al. (2013) had developed a daylighting system using multiple heliostat to direct sunlight towards a focusing mirror, the focusing mirror then reflect the light into mirror light pipe (MLP) that transmit the light into buildings. Figure 12 shows the layout of their design. An average illuminance of 850 lx was achieved by this design. The illuminance output is higher than using parabolic mirrors and Fresnel lens.

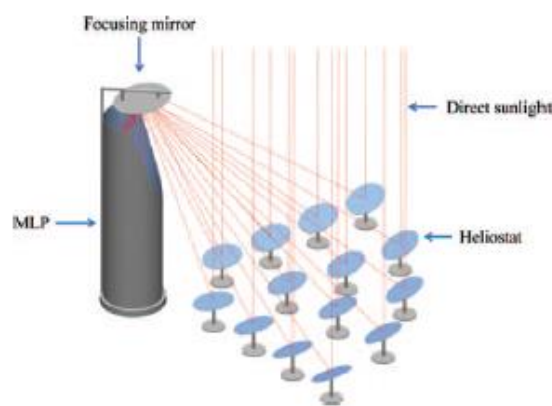


Figure 2.12: Proposed Daylighting System (Irfan and Seoyong, 2013)

CHAPTER 3

METHODOLOGY

3.1 Solar Concentrator Design

The proposed solar concentrator design is a two stage multi-facet mirrors to reflect the sunlight to optical fibres. It is basically a combination concept of parabolic mirror and multiple flat mirrors design discussed in previous section. The mirrors were aligned to meet the approximated parabolic shape so that reflective effect of parabolic shape can be achieved. The illustration of initial design idea is shown in figure 3.1. In the initial idea, the primary reflector and secondary mirror was planned to be approximate parabolic shape which is made up of multiple flat mirrors. The concept is similar to Cassegrain reflector that is widely used in reflecting telescope illustrated in figure 3.2.

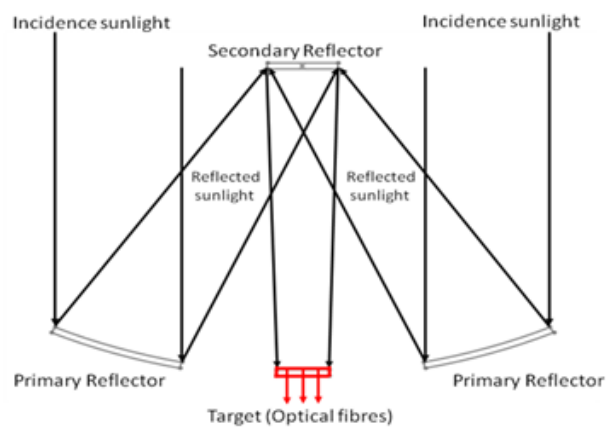


Figure 3.1: Solar Concentrator Initial Design

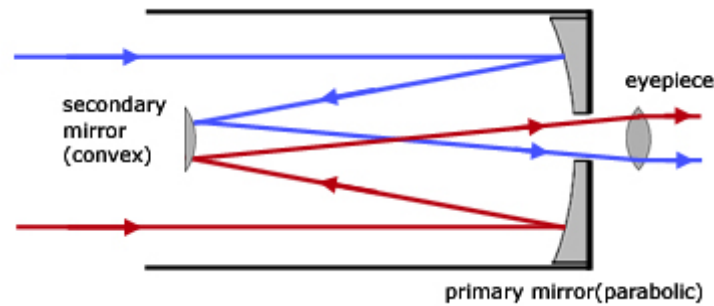


Figure 3.2: Cassegrain Reflector (a-levelphysics.com, N.D.)

3.1.1 Simulation Results

For the first step, a simple simulation was carried out using OpticalRayTracer program to estimate the size and number of mirrors to be used and also the focal point. The primary reflector was set to be using multiple 5cm x 5cm mirrors and the total size will be 60cm which made up of 12 rows of mirrors. The secondary reflector was unsure. The 2D view of Cassegrain reflector was simulated as in figure 3.3, the secondary reflector was placed at 30cm away from the primary mirror while the focal point is around 5.7cm from the centre of primary reflector. The optical fibre will be placed at the focal point. After that, multiple flat mirrors was placed along the curved mirrors, estimating the size of secondary flat mirrors to be used. It was found that to obtain approximate Cassegrain effect, the secondary flat mirrors had to be smaller than 1cm^2 , which is hard and expensive to manufacture in real life. Therefore the first design was concluded to be not practical and changes were made to the design.

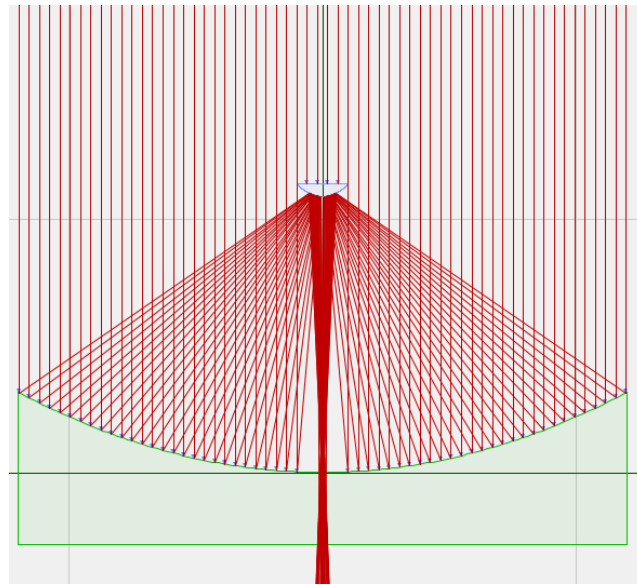


Figure 3.3: Simulated Cassegrain Reflector using OpticalRayTracer

For the second design, the Cassegrain reflector concept was abandoned. The plan was to use multiple reflections in the secondary design. Each set consists of four primary mirrors that were used to couple with one secondary mirror, then the secondary mirror reflect the ray back into the optical fibre. The dimension for primary was same as default which is 5cm x 5cm, the spacing between mirrors is 0.5cm. Therefore the whole dimension for primary mirrors is 10.05cm x 10.05cm. Four primary mirrors was suggested to be attached on a tilting plane, providing easier angle adjustment. Simulation was also made for this design to estimate the minimum possible size of the secondary mirror. The result was shown in figure 3.4 using OpticalRayTracer program. Since the program only give two dimension simulation, only two mirrors in each set can be simulated. Two sets of reflections were simulated and the minimum possible size of secondary mirror was estimated to be 8cm x 8cm. The distance between primary and secondary reflector was 35cm. The angle between the Mirror A and the flat surface was 10° and 8° for Mirror B. For the secondary reflector, the angle is -2° . All the results was obtained from OpticalRayTracer.

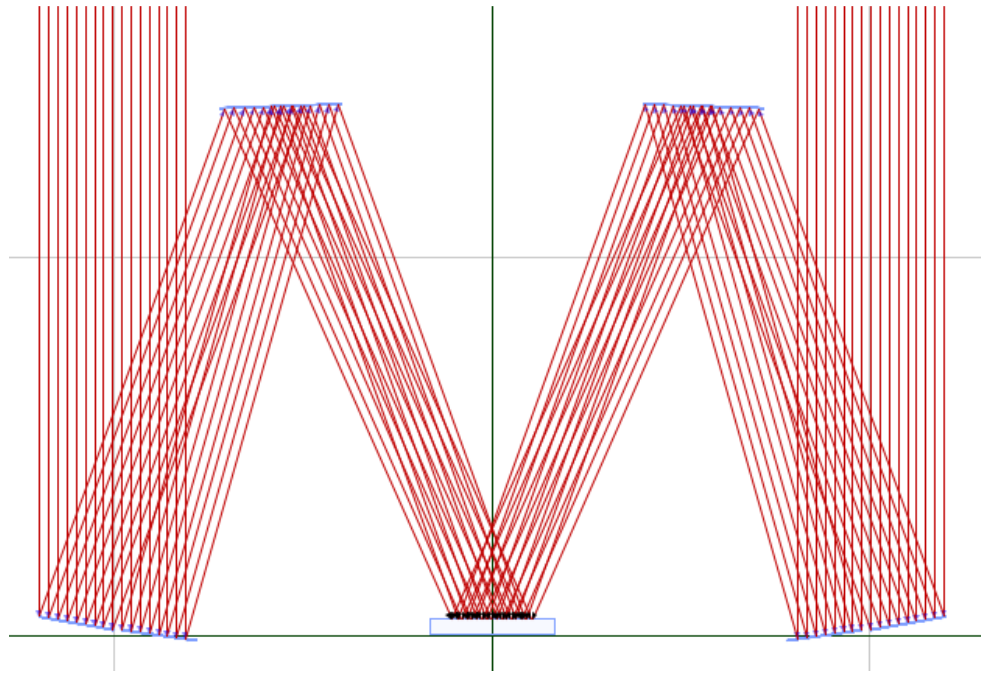


Figure 3.4: Two Stage Reflector Secondary Design

3.1.2 Prototype Construction

To further verify the simulation result, a prototype was made to estimate the size of secondary mirror. The design was drawn into SolidWorks before construction of hardware. Each primary mirror has three contact points that were attached to three machine screws using silicone paste. Both pins are adjustable to obtain desired angular position for the mirror. Four of these structures were then attached to a base plane that will be used to adjust the angle of four mirrors all together at once. In between of the mirrors and base plane, a spring was used to prevent the mirror from falling down at the same time provide flexibility for angle adjustment. Detail SolidWorks drawing is shown in figure 3.5. Since the purpose of this prototype is to evaluate the possibility of second design and also estimate the size of secondary mirror, recycle materials were used as much as possible. Figure 3.6 shows the constructed prototype. Experiment was carried out and discussed in next section.

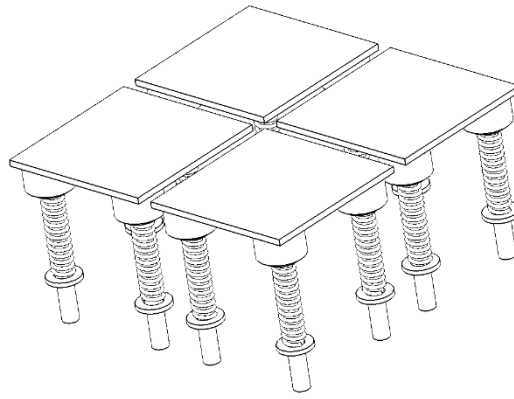


Figure 3.5: SolidWorks Drawing Prototype for Primary Reflector

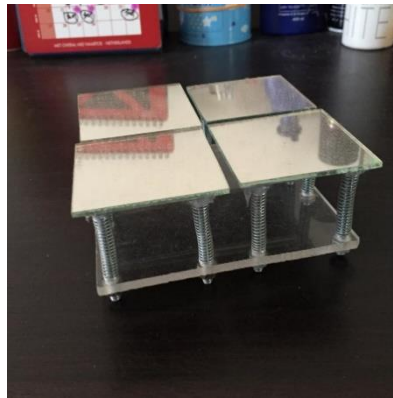


Figure 3.6: Constructed Primary Reflector Prototype

3.1.3 Experimental Results

A simple experiment was carried out to determine the feasibility of the design and also estimate the size for secondary mirror. The experiment setup was shown in figure 3.7. To obtain the primary mirror angle for four 25cm^2 solar rays to be reflected into a single point, each primary mirror was carefully tilted until single point secondary reflection is obtained as shown in figure 3.8. The secondary mirror used was larger than $8\text{cm} \times 8\text{cm}$ to ensure all light is reflected. Then the secondary mirror was removed and the solar image was recorded and measure. As shown in figure 3.9, the solar image is approximately the size of $8\text{cm} \times 8\text{cm}$ which is same as the size of secondary mirror in simulation. Therefore it can be concluded that the second design is viable. However there is one weakness about this experiment which is the incoming sun ray is not

perpendicular to the surface as there is no sun tracking system. This might cause inaccuracy of result, thus simulation was done in section 3.2.1 to compensate this error.



Figure 3.7: Experiment Setup



Figure 3.8: Focused Solar Image after Secondary Reflection



Figure 3.9: Solar Image after Primary Reflection

3.2 Solar Tracking Mechanism

Figure 3.10 is the final design for solar tracking mechanical mechanism done by Wai Cheng Ooi. It is a two-axis tracking system which combine the use of linear actuator to control the elevation and stepper motor to control the azimuth axis rotation.

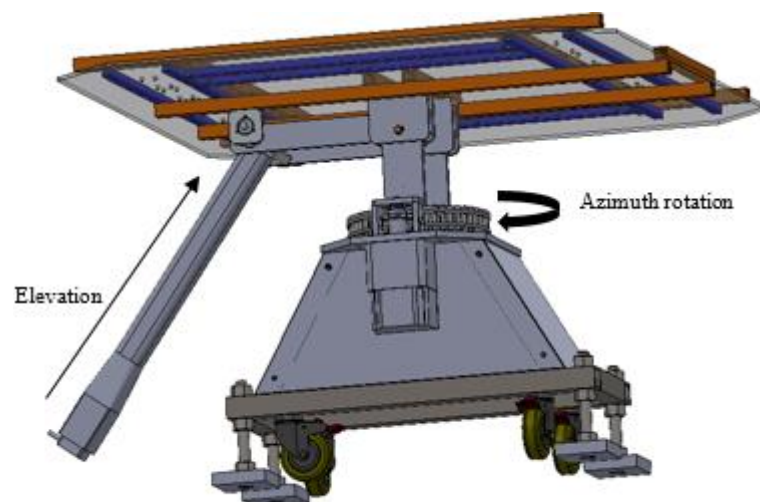


Figure 3.10: Final Mechanical Tracking System Design

3.2.1 Azimuth-axis Tracking

Detail design on azimuth rotation mechanism was shown in figure 3.11. The structure was constructed by using a primary gear attached to the stepper motor shaft and couple with a secondary gear. To save cost, instead of purchasing a big secondary gear, it was constructed using multiple screws on two rotating plates. Primary gear will be slotted into the space between the screws and turn the solar concentrator above. Another advantage of this system is that secondary gear can act as an encoder to save cost on extra encoder. Encoder was planned to be constructed using infrared sensors to count the screws, the design is illustrated in figure 3.12. This hardware construction will be carry out in FYP 2. However this system believed to has high backlash error due to the empty space between primary and secondary gear teeth. To compensate this error, pulsing control of stepper motor is used together with the infrared sensor, increasing resolution of the rotation and to detect whether the system is rotating or not when the direction changes.

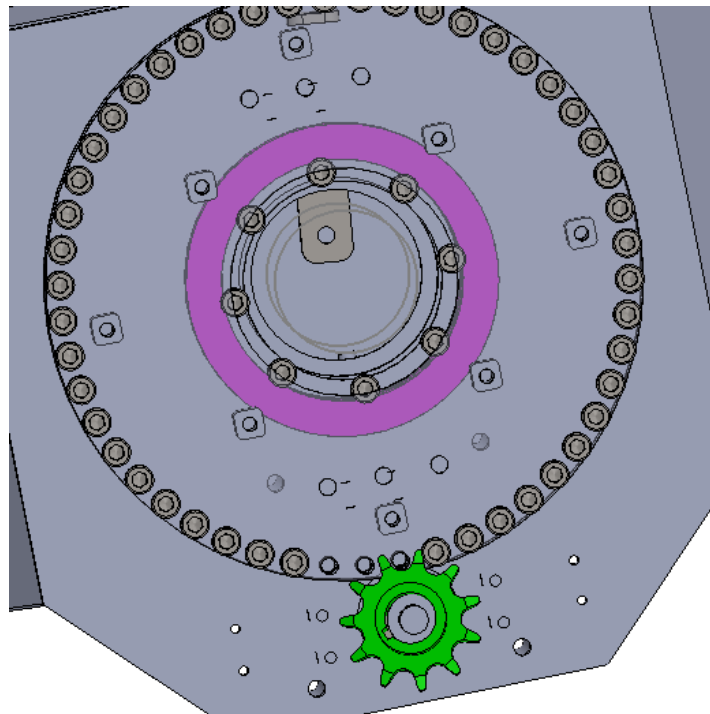


Figure 3.11: Azimuth-axis Rotation Mechanism

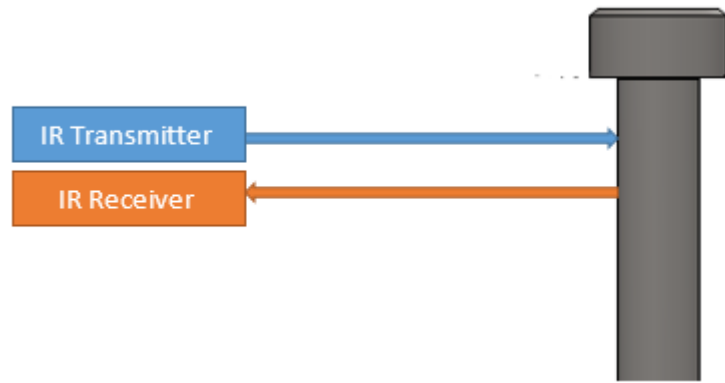


Figure 3.12: Encoder Using IR Sensors

3.2.2 Elevation Tracking

The elevation is rather simple compare to Azimuth-axis tracking. The elevation of the solar concentrator was control by a linear actuator. The linear actuator used is an EAC6 model with build in encoder from Oriental motor. However there is an issue as the elevation angle of the system is not certain to have a linear relationship with the linear actuator. An experiment should be conducted to measure the every elevation angle change at each linear actuator step during FYP 2. Graph of elevation angle versus motor steps should be plotted to examine their relationship. The elevation angle will be measure at two rotating bearing label in figure 3.13 when the stepping increases and both results will be compared.

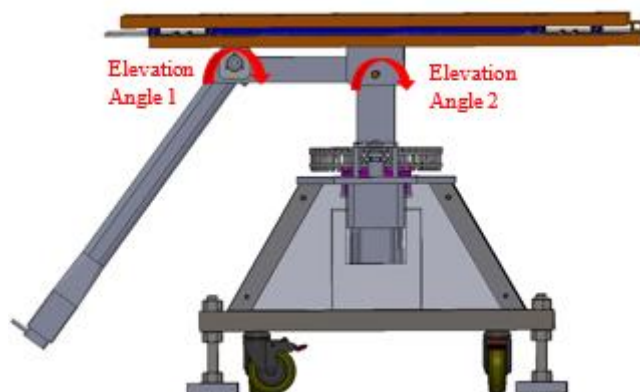


Figure 3.13: Places to Measure the Elevation Angle

3.3 Equation for Solar Tracking

In order to track the sun in closed-loop system, it is important to derive an equation that determines the position of the sun in terms of azimuth angle and elevation angle relative to the observer position. Once all the formulas are derived, it will be integrated into the programmer to move the solar concentrator facing directly towards the incoming solar irradiation to maximise the collection of solar energy. For the above mechanical design, azimuth rotation can move 360° which means there are no limit for the rotation. However there is limit for the elevation mechanical design, the limit of the design determines how long the solar tracker can track the sun.

Few translations are needed to obtain the solar tracking equation. First of all the declination angle, δ need to be obtained. The declination angle is defined as the angle between the equatorial planes of the earth to the perpendicular line of sun to the centre of earth. When the sun is directly at top of the observer, the declination angle is equal to zero. Figure 3.13 illustrate the declination angle of sun and earth, Q is the observer position. The declination angle can be calculated using the following equation.

$$\delta = \sin^{-1} 0.39795 \cos[0.98563(N - 173)] \quad (3.1)$$

where

δ = declination angle, degrees

N = day number, days

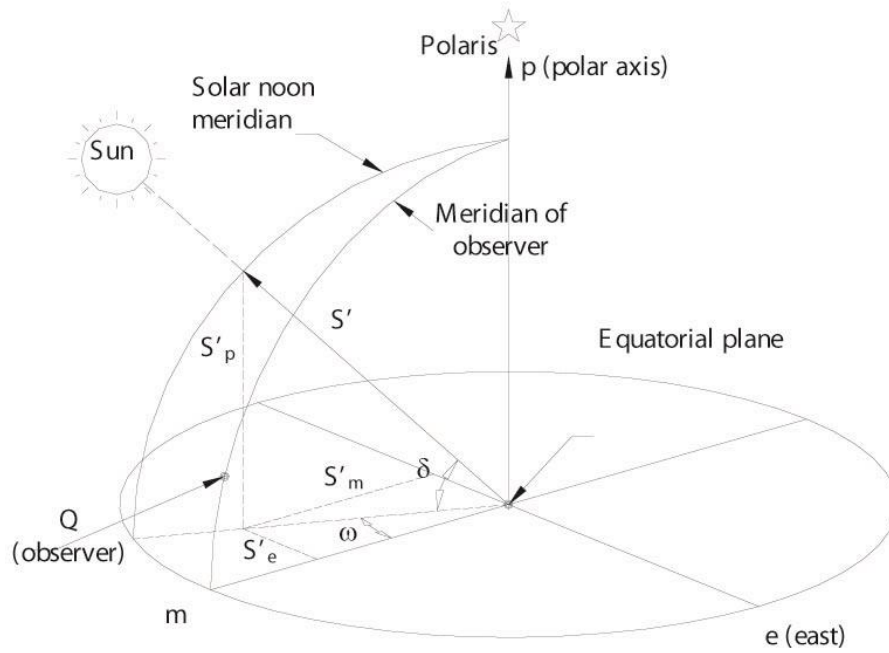


Figure 3.14: Illustration of Sun and Equatorial Plane

Secondly is the hour angle which is represented as ω in Figure 3.14. As shown in Figure 3.14, the hour angle is the angle between the meridian where the observer lies and the meridian where the incoming perpendicular solar irradiation strikes. The hour angle can be calculated using the equation below:

$$\omega = 15(t_s - 12) \quad (3.2)$$

where

ω = hour angle, degrees

t_s = solar time, hours

Solar time is the 24-hour clock time of depending on the observer location. For every one hour of solar time, hour angle increase by 15 degrees. When the sun is at highest point in the sky, the hour angle is zero because the solar time is equal to 12. Solar time cannot be measured, it must be calculated using following equation:

$$t_s = LC - LCT - \frac{EOT}{60} + LC + D \quad (3.3)$$

where

LC = longitude correction, hours

LCT = local clock time, 24-hour

EOT = equation of time, minutes

D = daylight saving

Daylight saving is the practice of advancing clocks by one hour during summer so that daylight in the evening is one hour longer. D equals to 1 if there is daylight saving in observer location and 0 if there is no daylight saving. In Malaysia, no daylight saving is practiced therefore D should be 0.

Equation of time (EOT) is the difference between the mean time and solar time. Mean time is define as the average length of a day. Mean second is calculated as $1/86,400$ of the time for earth to complete one orbit around the sun. During 1968, Woolf had derived an equation to calculate EOT which the accuracy lies within 30 seconds during daylight hours. Therefore angles of sun is expected to be updated every 30 seconds for this daylighting project. The equation is stated below:

$$EOT = 0.258 \cos x - 7.416 \sin x - 3.648 \cos 2x - 9.228 \sin 2x \quad (3.4)$$

where

EOT = equation of time, minutes

x = function of the day number, degrees

Angle x is calculated using the equation below:

$$x = 360(N - 1)/365.242 \quad (3.5)$$

where

x = function of the day number, degrees

N = day number, days

N is the number of days passed starting from 1st of January. After obtaining the EOT, the next step is to calculate the longitude correction (LC). LC is calculated using equation (3.6). Local longitude is the longitude reference to observer location. Longitude of standard time zone meridian depends on the time zone of observer location. In case of Malaysia, it is UTC+8.00.

$$LC = (\text{longitude of standard time zone meridian} - \text{local longitude})/15 \quad (3.6)$$

After getting the EOT, LCT, LC and D, t_s can be calculated using equation (3.3). In turn the hour angle can be calculated using equation (3.2). Hour angle and declination angle determines the location of sun in relatives to earth. Direction of vector S' pointing to the sun from the centre of the earth may be described in terms of S'_m , S'_e and S'_p as shown in Figure 3.14. The direction vectors are written as,

$$\begin{aligned} S'_m &= \cos \delta \cos \omega \\ S'_e &= -\cos \delta \sin \omega \\ S'_p &= \sin \delta \end{aligned} \quad (3.7)$$

Equations above are only for sun position in relative to centre of the earth. For relation of observer position to sun position, the azimuth angle, A and the zenith angle, θ_z are illustrated in Figure 3.15. Azimuth angle is the angle between the straight line pointing to North and the Central rays from sun, therefore when the sun is at east, azimuth angle equals to 90° , 180° when the sun at south and so on. Zenith angle describe the elevation angle of the sun to the zenith. To reduce difficulties in solar tracking later, altitude angle, α is used instead. Altitude angle is just 90° deduct the zenith angle.

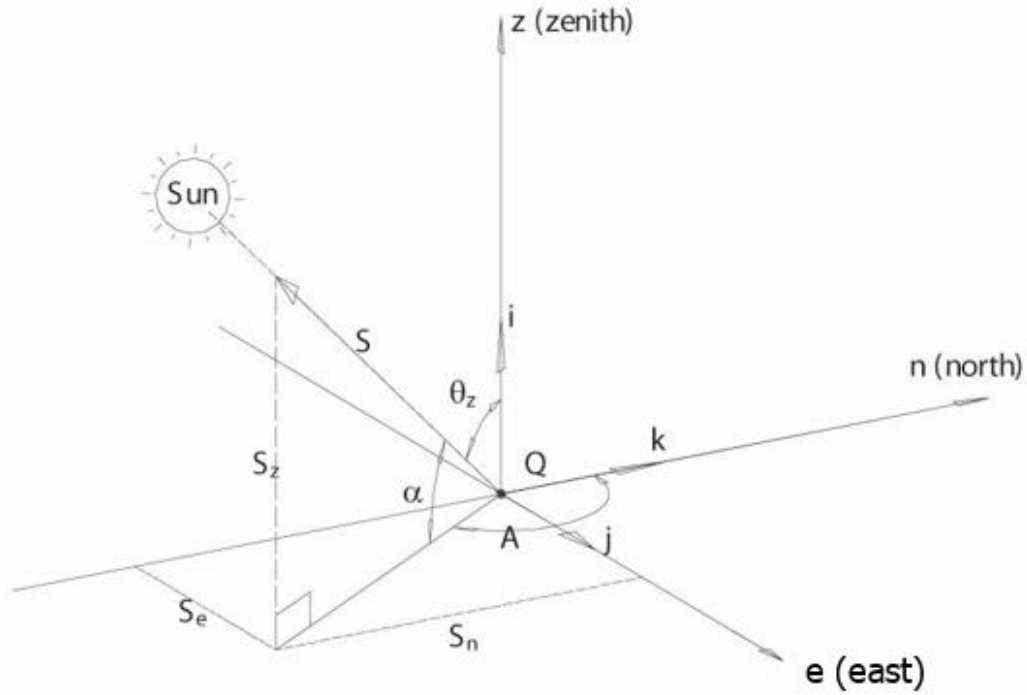


Figure 3.15: Illustration of Sun Location to Observer Q Location

Similar to equation (3.7), vector direction, S of observer position point to the sun can be written as,

$$\begin{aligned} S_z &= \sin \alpha \\ S_e &= \cos \alpha \sin A \\ S_n &= \cos \alpha \cos A \end{aligned} \quad (3.8)$$

where

α = altitude angle, degrees

By rotating the equatorial axis based on right hand rule through the latitude angle, Φ and the translation along the radius of the earth, vectors S and S' can be combined in matrix form below:

$$\begin{bmatrix} S_z \\ S_e \\ S_n \end{bmatrix} = \begin{bmatrix} \cos \Phi & 0 & \sin \Phi \\ 0 & 1 & 0 \\ -\sin \Phi & 0 & \cos \Phi \end{bmatrix} \begin{bmatrix} S'_m \\ S'_e \\ S'_p \end{bmatrix} \quad (3.9)$$

where

Φ = local latitude, degrees

Solving the matrix, we will have

$$\begin{aligned} S_z &= S'_m \cos \Phi + S'_p \sin \Phi \\ S_e &= S'_e \\ S_n &= S'_p \cos \Phi - S'_m \sin \Phi \end{aligned} \quad (3.10)$$

The direction vectors for both S and S' has already been solved in equation (3.7) and (3.8). Therefore substitute (3.7) and (3.8) into equation (3.10) yields,

$$\sin \alpha = \sin \delta \sin \Phi + \cos \delta \cos \Phi \cos \omega \quad (3.11)$$

$$\cos \alpha \sin A = -\cos \delta \sin \omega \quad (3.12)$$

$$\cos \alpha \cos A = \sin \delta \cos \Phi - \cos \delta \cos \omega \sin \Phi \quad (3.13)$$

Using equation (3.11), the altitude angle can be calculated based on the latitude, declination angle and hour angle. Latitude based on observer location while declination angle and hour angle can be calculated using equation (3.1) and (3.2) respectively. However for azimuth angle, both equation (3.12) and (3.13) can be used. Equation (3.13) is rearranged to equation (3.14) test which quadrant azimuth lies in.

$$\cos A = \frac{\sin \delta \cos \Phi - \cos \delta \cos \omega \sin \Phi}{\cos \alpha} \quad (3.14)$$

If $\cos A$ is more than 0, then azimuth angle is,

$$A = \sin^{-1}(-\cos \delta \sin \omega / \cos \alpha) \quad (3.15)$$

If $\cos A$ is less than 0, then azimuth angle is,

$$A = 180 - \sin^{-1}(-\cos \delta \sin \omega / \cos \alpha) \quad (3.16)$$

Therefore the equation for azimuth angle and altitude angle are solved. In general it is based on translation from centre of the earth to surface of earth where observer is

located in relative to the sun. Figure 3.16 is an overall illustration and Figure 3.17 is the summarise flow chart to calculate the azimuth and altitude angle which will be integrated into the program later. This sub function is named solar().

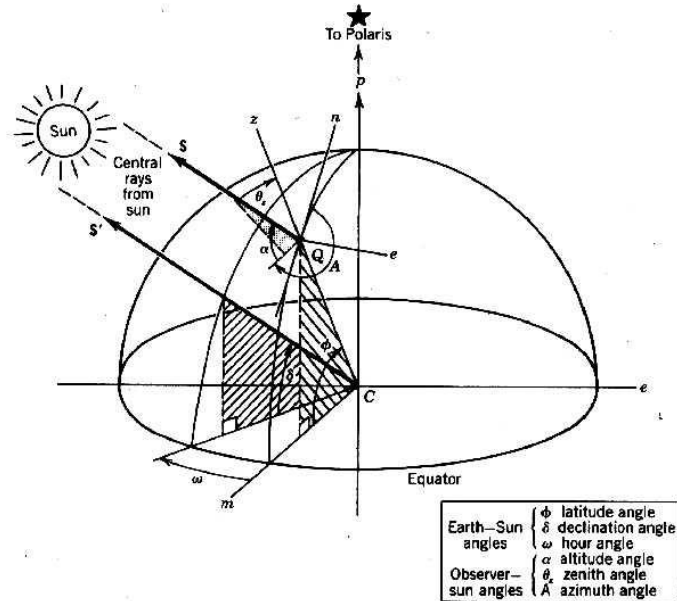
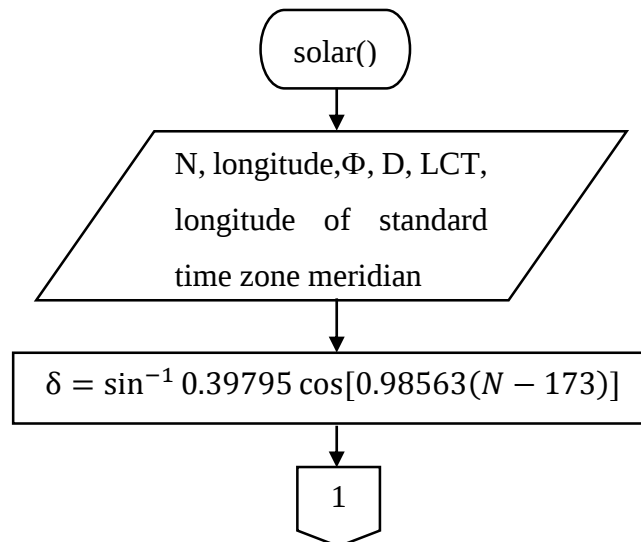


Figure 3.16: Overall Illustration of Sun Position Translation



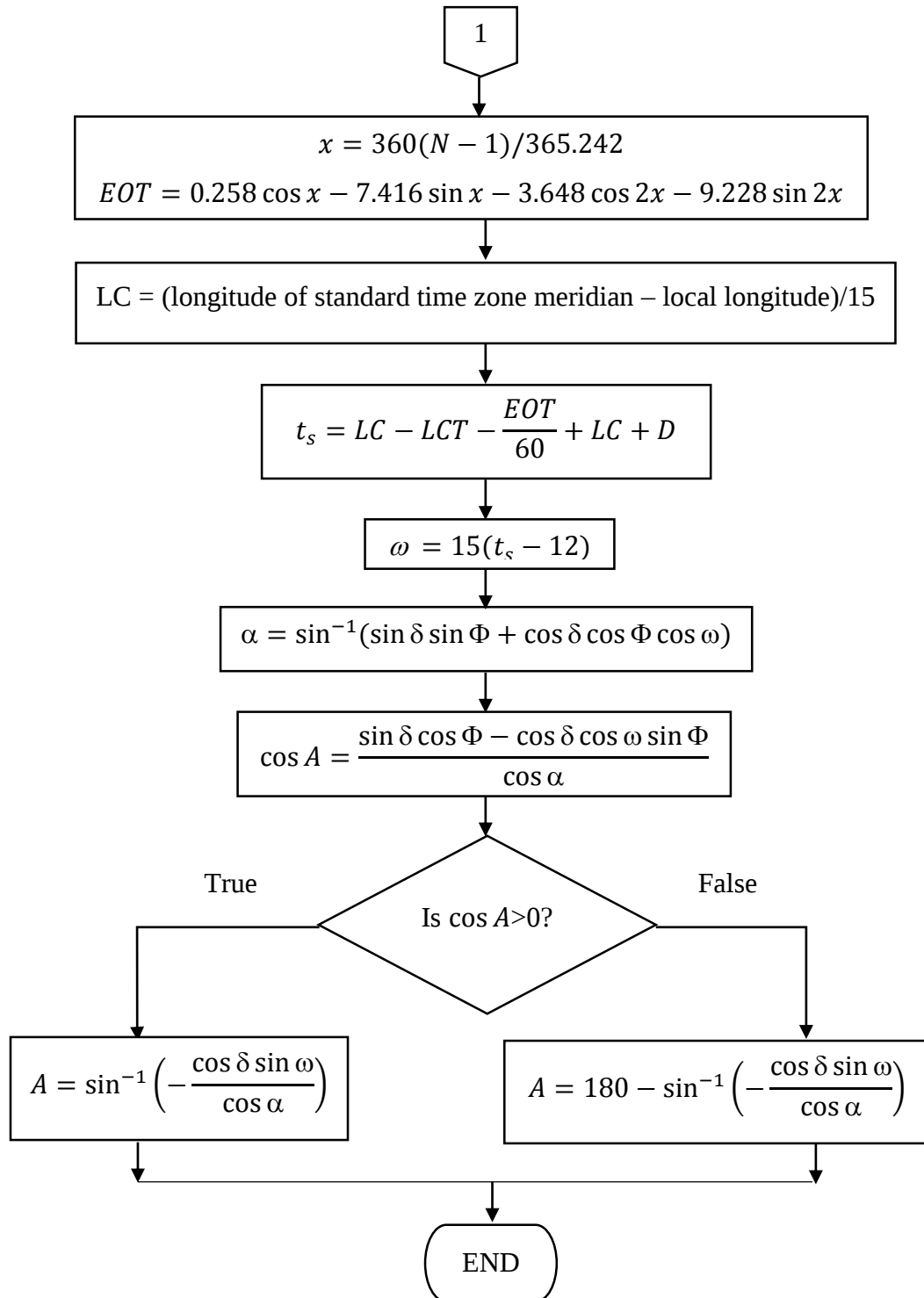


Figure 3.17: Flowchart for Azimuth and Altitude calculation

3.4 Derivation of Equation

In order to track the sun position, necessary equation need to be derived to convert the altitude angle and azimuth angle into controllable stepper motor value. In this design, the altitude angle is control by linear actuator while the azimuth angle is control by the rotating stepper motor.

3.4.1 Derivation of Elevation Tracking Equation

In section 3.3, the altitude angle can be calculated but it cannot be used directly into the constructed hard shown in Figure 3.13. The altitude angle need to be converted into elevation angle by certain equation so that the linear actuator elevates the solar concentrator facing perpendicularly to the sun. Derivation of this equation will be discussed in this section.

When the linear actuator pushes up the solar concentrator, it will form a triangular shape shown in Figure 3.18. The black arrow represent the extended length of linear actuator shaft, yellow line is the length from ending of linear actuator shaft to pivot point of solar concentrator, red line is the length from starting of linear actuator shaft to the pivot point of solar concentrator. θ_E is the elevation angle of the system.

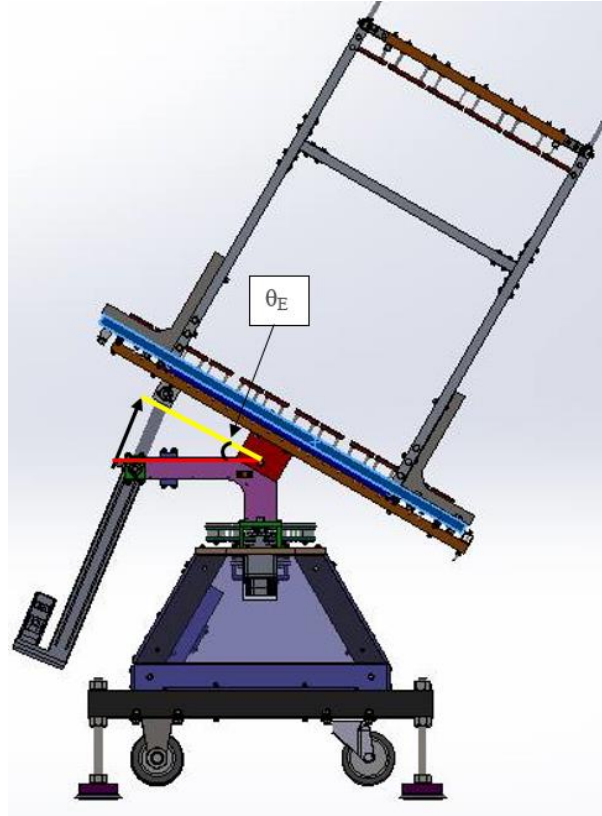


Figure 3.18: Linear Actuator extend and form Triangular Shape

For easy derivation purpose, the triangle lines are extract out in Figure 3.19. The arrow, yellow line and red line are labelled A', B and C respectively. Linear actuator shaft labelled as A' is a manipulated variable, θ_E is a responding variable while B and C is a constant number. From the constructed hardware, B and C are measured to be 210mm and 240mm respectively. In order to derive equation of A' in terms of altitude angle, equation of A' in terms of θ_E must be derived first. To do this, equation (3.17) of cosine law below is used,

$$A'^2 = B^2 + C^2 - 2BC \cos \theta_E \quad (3.17)$$

where

θ_E = elevation angle, degrees

A' = linear shaft length, mm

Given that B and C is 210mm and 240mm respectively, equation (3.17) can be further simplified into,

$$A' = \sqrt{101700 - 100800 \cos \theta_E} \quad (3.18)$$

Equation of A' in terms of θ_E is obtained.

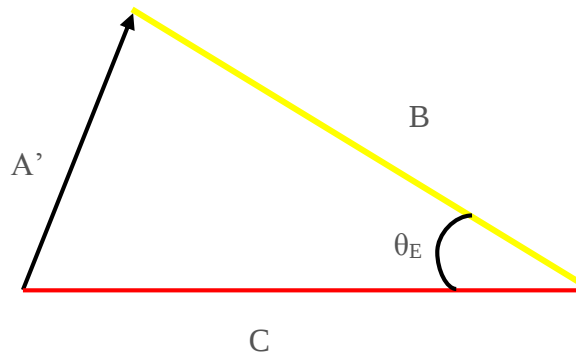


Figure 3.19: Illustration of Linear Actuator Shaft, A' and Elevation angle, θ_E

After obtaining equation (3.18), the next step is to derive equation that relates the altitude angle, elevation angle and also the linear actuator shaft length. Altitude angle is defined as the angle between the horizontal plane and the incoming solar irradiation. In this case, the horizontal plane is C plane which is always parallel to the ground surface therefore altitude angle can be redefined as the angle between the C plane and incoming solar irradiation. Assuming ideal case, the solar concentrator surface which is B plane always track the sun accurately, the incoming solar irradiation will always be perpendicular to B plane. Thus the incoming solar irradiation will form a right angle triangle inside of the triangle. The illustration is shown in Figure 3.18.

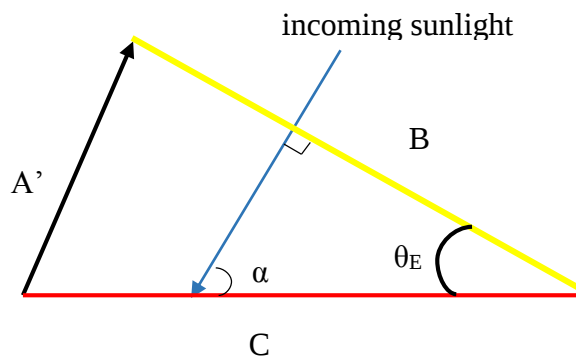


Figure 3.20: Illustration of Incoming Sunlight forms Right Angle Triangle

With the right angle triangle form by incoming sunlight, the elevation can simply be calculated using,

$$\theta_E = 90^\circ - \alpha \quad (3.19)$$

Substituting equation (3.19) into equation (3.18) yields,

$$A' = \sqrt{101700 - 100800 \cos(90^\circ - \alpha)} \quad (3.20)$$

Equation of linear shaft length in terms of altitude angle is obtained, relationship between linear actuator and solar altitude angle is known, and therefore tracking can be done using equation (3.20).

Linear actuator model EAC4R-E30-ARMKD-3 from Oriental Motor is used in this project shown in Figure 3.21. It is a stepper motor attached to actuator which converts motor rotation into linear motion. Figure 3.22 is a table taken from datasheet, elements in the table are necessary for further derivation. Linear actuator used in this project falls under row number four. The “Lead” category in table signify how long the shaft moves for every one rotation, in this case is 6mm. The pulse per resolution (P/R) of the stepper motor can be set within 100 to 10 000, 1000 is set for our system. Stepper motor is controlled by sending pulse into the driver, each pulse increase the stepper motor by one step. For 1000 P/R, each pulse will increase shaft length by 0.006mm. Equation (3.21) enable the programmer to calculate how many pulses to be sent to increase the length of linear shaft.

$$no. of pulse = \frac{A'}{0.006} \quad (3.21)$$

where

A' = linear shaft length, mm



Figure 3.21: Linear actuator Model EAC4R-E30-ARMKD-3

Drive System	Ball Screw	Repetitive Positioning Accuracy [mm]		±0.02	Resolution[P/R]		100~10000
Product Name	Lead [mm]	Transportable Mass [kg]		Thrust [N]	Pushing Force [N]	Holding Force [N]	Maximum Speed [mm/s]
		Horizontal	Vertical				
EAC4R-D□□-ARA□□-◇	12	~15	—	~70	100	70	600
EAC4R-D□□-ARM□□-◇			~7				
EAC4R-E□□-ARA□□-◇	6	~30	—	~125	200	125	300
EAC4R-E□□-ARM□□-◇			~12.5				

Figure 3.22: Motor Properties taken from Datasheet

3.4.2 Derivation of Azimuth Tracking Equation

Compare to elevation tracking, the derivation for azimuth tracking motor equation is much simpler. Model of stepper motor used is PK264A2-SG50 from Oriental Motor which has resolution of 0.036° per step as shown in Figure 3.23. Similar to linear actuator, each pulse sent into the driver rotates the stepper motor by one step. Refer to Figure 3.11, shaft of stepper motor is attached to a primary gear. The primary gear is coupled to secondary gear and secondary gear is attached to the solar concentrator. The primary gear to secondary gear ratio is 11 to 50. Rotation of secondary gear rotates the solar concentrator and the rotated angle is just simply equal to azimuth angle. Therefore, the programmer just need to calculate the azimuth angle of the sun and rotates the solar concentrator to the calculated azimuth angle. Equation (3.22) will be

used to calculate number of pulse needed to rotate the concentrator based on solar azimuth angle. This equation also taken care of the gear ratio.

$$\text{no. of pulse} = \frac{50}{11} \times \frac{A}{0.036^\circ} \quad (3.22)$$

where

A = azimuth angle, degrees



Figure 3.23: Stepper Motor for Azimuth-axis Tracking

3.5 Software Development

In this section, the development of program will be discussed mostly in terms of flow chart. Arduino Mega will be served as the computer to operate the whole system. Its function includes solar angles calculation, pulse generation, interrupts and so on.

3.5.1 Timer Interrupt

Before tracking the sun, programmer will ask user for several inputs including date and time. In order to the track the time accurately, timer interrupt is used. Timer

interrupt will be triggered at every 1 second and check whether the time has passed 24 hours. If 24 hours have passed, then the day number will increase by 1 and time1 will reset to 0. If day number is more than 365, then it will be reset to 1. Since time1 is in 24-hour format, every time the timer is triggered, time1 will be increased by $\frac{1}{3600}$ hour. Variable time1 will be stored as LCT later on. Flow chart of timer interrupt is shown in Figure 3.24.

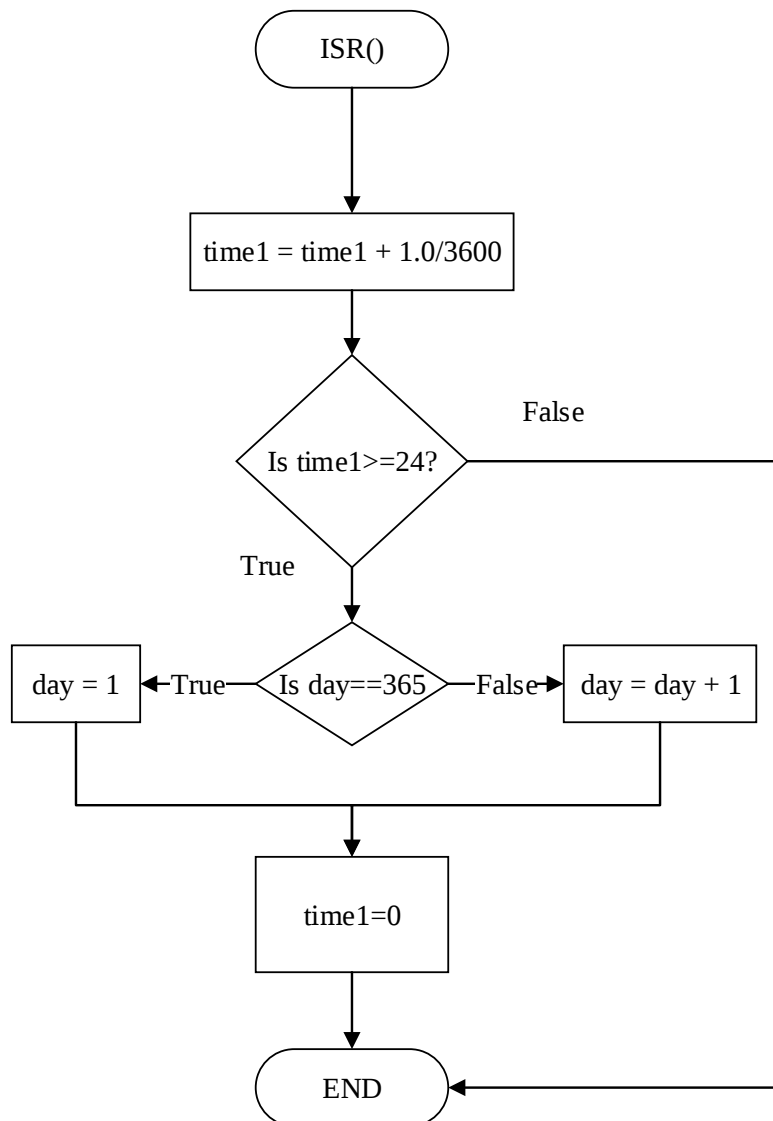


Figure 3.24: Flow Chart of Timer Interrupt

3.5.2 Motor Pulse

Both linear actuator motor and azimuth motor are controlled by sending pulses. Each pulse sent to the driver will increase the motor by one step. In order to write a tidier program, it is better to create sub function that generate pulses. Both motors have 1-pulse mode or 2-pulse mode. 1-pulse mode means that pulses are only sent to one input, and the direction of rotation is controlled by another ON and OFF only input. 2-pulse mode means that two different pulses are sent into two different pins, one pin control clockwise rotation and the other one control counter clockwise rotation. In case of linear actuator, one pin extend the linear shaft and the other retract it. 2-pulse mode is used for both type of motors used in this project. Method of generating pulse is just setting the pin high, delay for certain period and then setting the pin to low again.

Arduino pin 52 and pin 53 will generate pulses into azimuth motor driver pin 1 and pin 3 respectively. Pin 52 will controls clockwise rotation while pin 53 will controls counter clockwise rotation. Flow chart for azimuth pulse control is shown in Figure 3.25. Arduino pin 22 and pin 24 will generate pulse into linear actuator motor driver pin 31 and pin 35. Pin 22 will controls extension of shaft and pin 24 controls the retraction. Flow chart for azimuth pulse control is shown in Figure 3.26. Input value to both sub functions labelled as numTimes is the number of pulses to be generated. If numTimes is more than 0, then clockwise rotation or extension will be carried out, if numTimes less than 0, then counter clockwise rotation or retraction will be carried out.

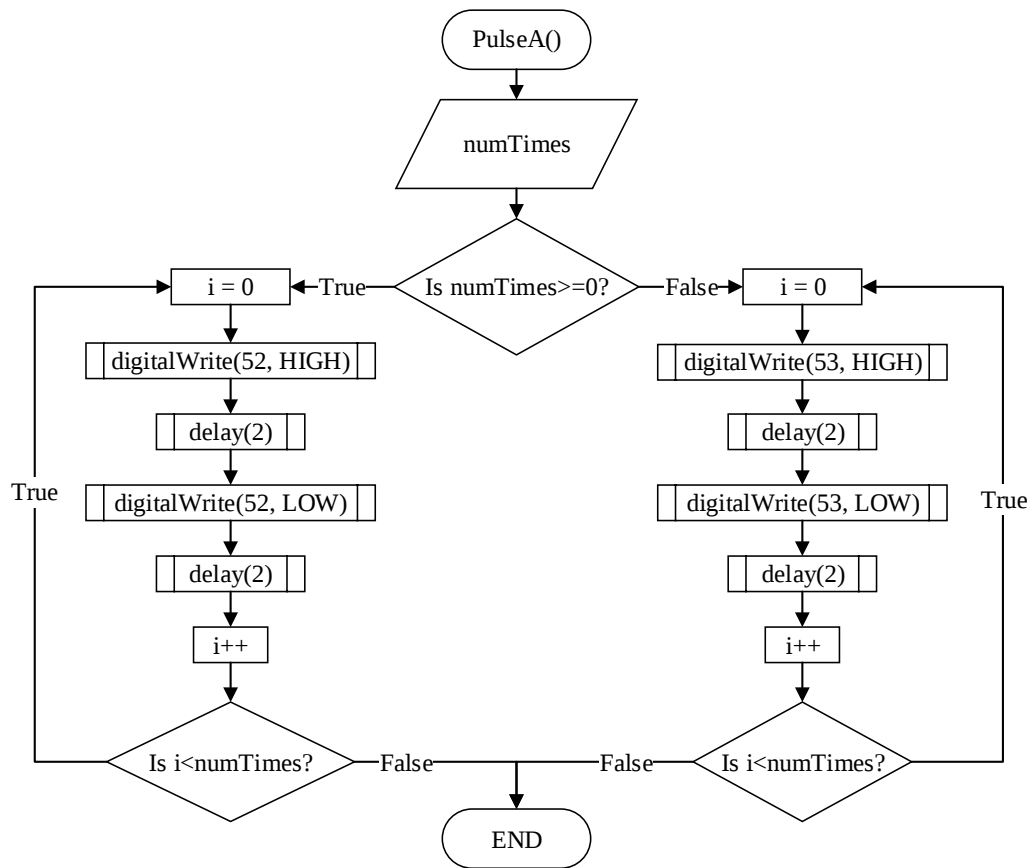


Figure 3.25: Flow Chart to Control Azimuth Stepper Motor

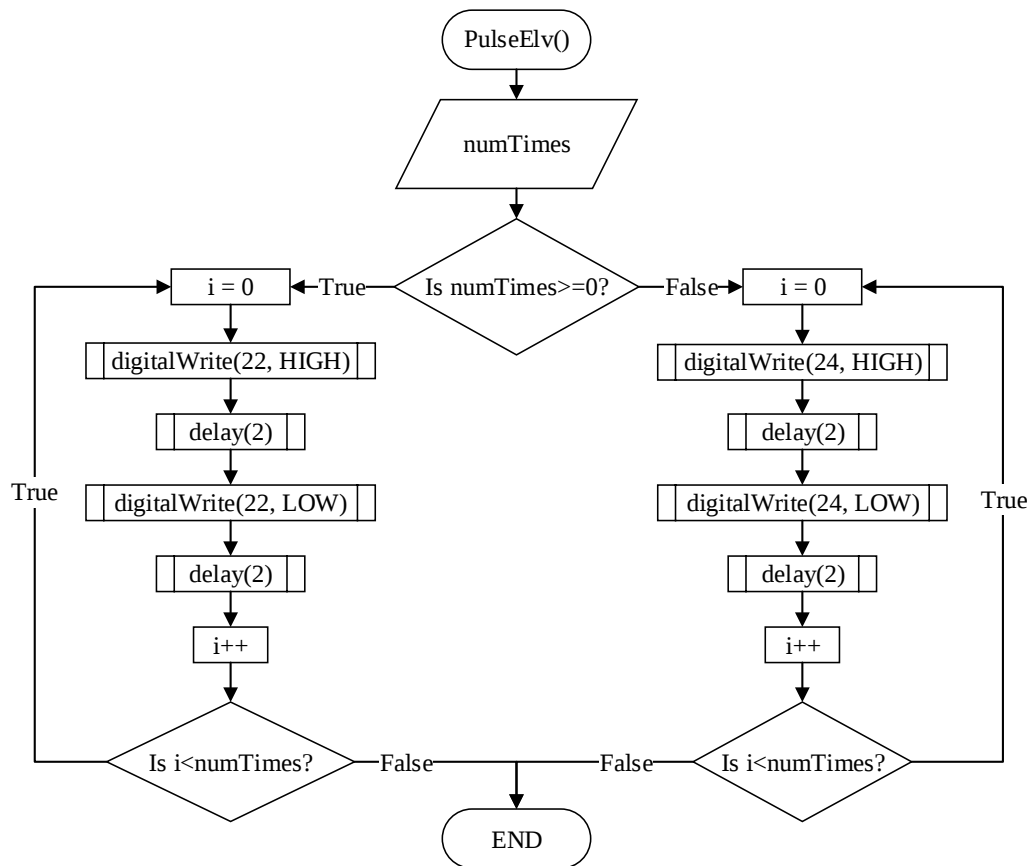


Figure 3.26: Flow Chart to Control Linear Actuator Stepper Motor

3.5.3 Measuring Backlash

As discussed previously in section 3.2.1, azimuth-axis tracking consist of primary gear and secondary gear coupling each other. Since secondary gear is custom made, there is gap between the contact of primary gear and secondary gear. This gap will cause backlash problem when the motor tries to rotate different direction from previous rotation. For example, a motor first rotates in clockwise direction, after that it needs to rotate in counter clockwise direction however there is a gap between the gears. The primary gear is rotating but the secondary gear is not because there is no contact between them, the programmer will thinks that it is rotating the solar concentrator but in fact it is not moving at all, therefore there will be loss of motion. Illustration of backlash is shown in Figure 3.27.

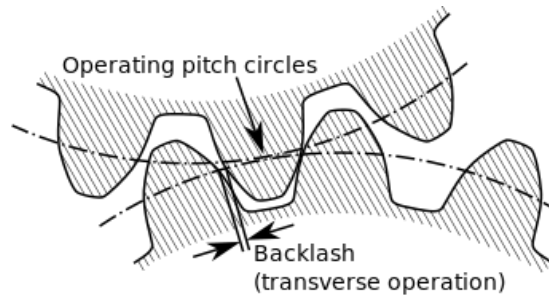


Figure 3.27: Illustration of Backlash

The solution approached to solve this problem is to compensate the backlash error whenever the programmer wants the motor to rotate in different direction, but first the backlash have to be measured. An experiment is designed to measure the backlash error and will be discussed later. Infrared sensor mentioned in section 3.2.1 can be used to measure but infrared sensor accuracy is greatly affected under sunlight. Therefore another sensor named U-shaped micro photoelectric sensor is used. U-shaped micro photoelectric sensor is a type of sensor that is in U-shaped that detects object that pass through it. When an object passes through the U-shaped gap, it will block the transmitter from the receiver then the output signal will changed. Figure 3.28 is a sample of photoelectric sensor taken from data sheet. A plate was attached below of secondary gear. The purpose of this plate is to block the sensor transmitter and receiver when it enters the sensor gap. The mounting is shown in Figure 3.29 and Figure 3.30. The plate is also served as initial reference for solar concentrator, therefore the sensor should always pointing towards North direction.

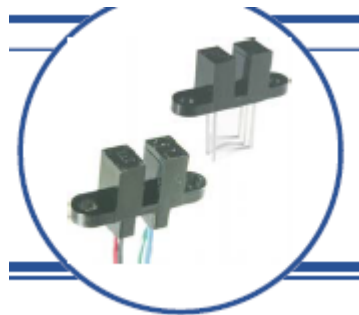


Figure 3.28: U-shaped Microphotoelectric Sensor

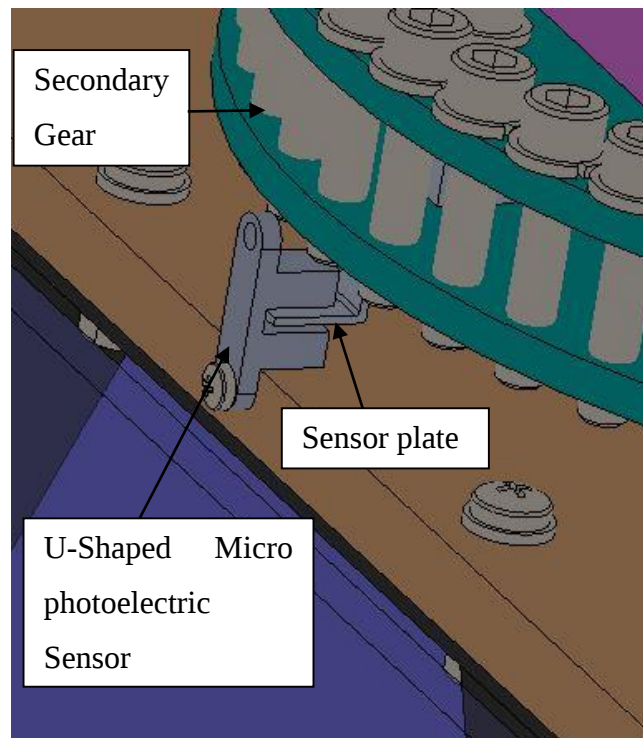


Figure 3.29: SolidWorks Drawing of Sensor, Sensor plate and Secondary Gear

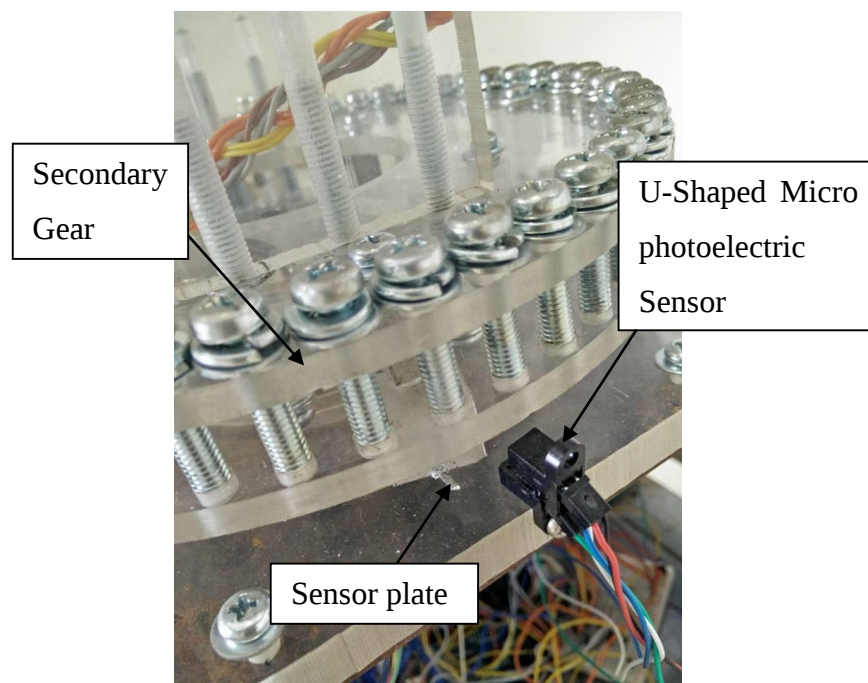


Figure 3.30: Mounting of Senor, Sensor plate and Secondary Gear

The sensor output is HIGH when no object is between the gaps and LOW when object is in the gap. At first the azimuth stepper motor will rotate counter clockwise until sensor output is LOW, which means the solar concentrator has been reset to initial position. It is important to make sure that there is no backlash gap after the counter clockwise rotation else the result will be inaccurate. After that, the motor then rotates in clockwise direction until the sensor output is LOW again. While rotating, the programmer will record total pulses have been sent to the motor driver. Given that resolution of azimuth stepper motor is $0.036^\circ/\text{step}$, the equation to calculate backlash error is,

$$\text{backlash error } ^\circ = (\text{total pulses} - \text{pulses for one rotation}) \times 0.036 \times \frac{11}{50} \quad (3.23)$$

Given that one rotation is 360° , the pulses for one rotation can be calculated using equation (3.22). Substituting $A = 360^\circ$, no. of pulse is equal to 45455. Therefore equation 3.23 can be further simplified into equation (3.24).

$$\text{backlash error } ^\circ = (\text{total pulses} - 45455) \times 0.036 \times \frac{11}{50} \quad (3.24)$$

The width of the sensor plate should also be taken into consideration in measuring the backlash because the photoelectric sensor only senses the edge of the sensor plate, the area within the sensor plate will be a loss. To eliminate this error, the width of the sensor plate is measured. The width is measured to be 1.5mm. Assuming the width of is equal to the arc length of a secondary gear, the loss in degrees can be calculated using equation,

$$S = r\theta \quad (3.25)$$

where

S = arc length of a secondary gear, mm

r = radius of secondary gear, mm

θ = angle of arc, radians

Given that the radius of secondary gear is 210mm, the angle of arc which is the angle loss when measuring the backlash is calculated to be 0.4° . Using equation (3.22), to move $A = 0.4^\circ$, 51 pulses are needed. Equation (3.24) can be further improved into equation (3.26) below. In addition, photoelectric sensor is also used to reset the solar concentrator back to initial position but it can only sense the edge of the sensor plate. Therefore half of the angle loss 0.2° should be added so that the sensor plate is accurately position in the middle of the sensor.

$$\text{backlash error } ^\circ = (\text{total pulses} - 45455 + 51) \times 0.036 \times \frac{11}{50} \quad (3.26)$$

The sensor output pin is connected to Arduino input pin 26. Programming flowchart to measure backlash is shown in Figure 3.31. The result of the experiment will be discussed in next chapter.

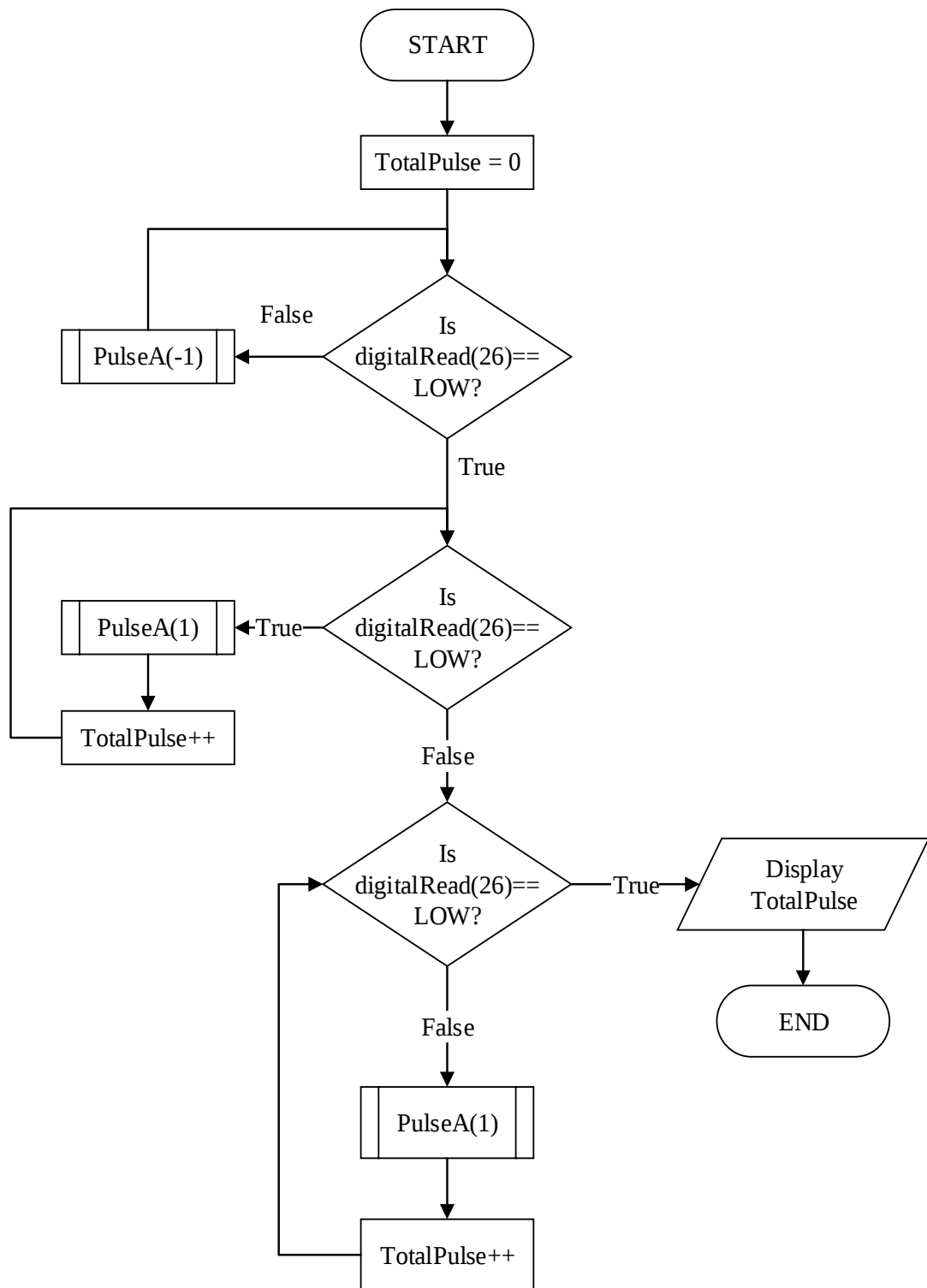


Figure 3.31: Programming Flow Chart to Measure Backlash

3.5.4 Measuring Tracking Accuracy

The tracking accuracy of the system is measured by observing the shadow of the solar concentrator. Tracking code was also developed to measure the tracking accuracy. A bar shown in Figure 3.32 is mounted parallel above the solar concentrator for user to observe the shadow. When tracking code is designed as a button interrupt in Arduino. When tracking code runs, user needs to operate azimuth stepper motor and linear actuator manually until the shadow is eliminated. While operating, the programmer will calculate how much azimuth angle and altitude angle the solar concentrator has moved using equation (3.22) and (3.20). The angles will be the tracking error of overall system. Figure 3.33 below is the flow chart of tracking code. Input pin 49 and 51 will control the linear actuator motor retraction and extension while pin 47 and 45 will control the azimuth stepper motor clockwise and counter clockwise rotation. Pin 32 is used to keep the tracking code running while measuring tracking accuracy. DirA variable is to record the rotating direction. Backlash error is added when moving in different direction.



Figure 3.32: Bar Mounted on Solar Concentrator

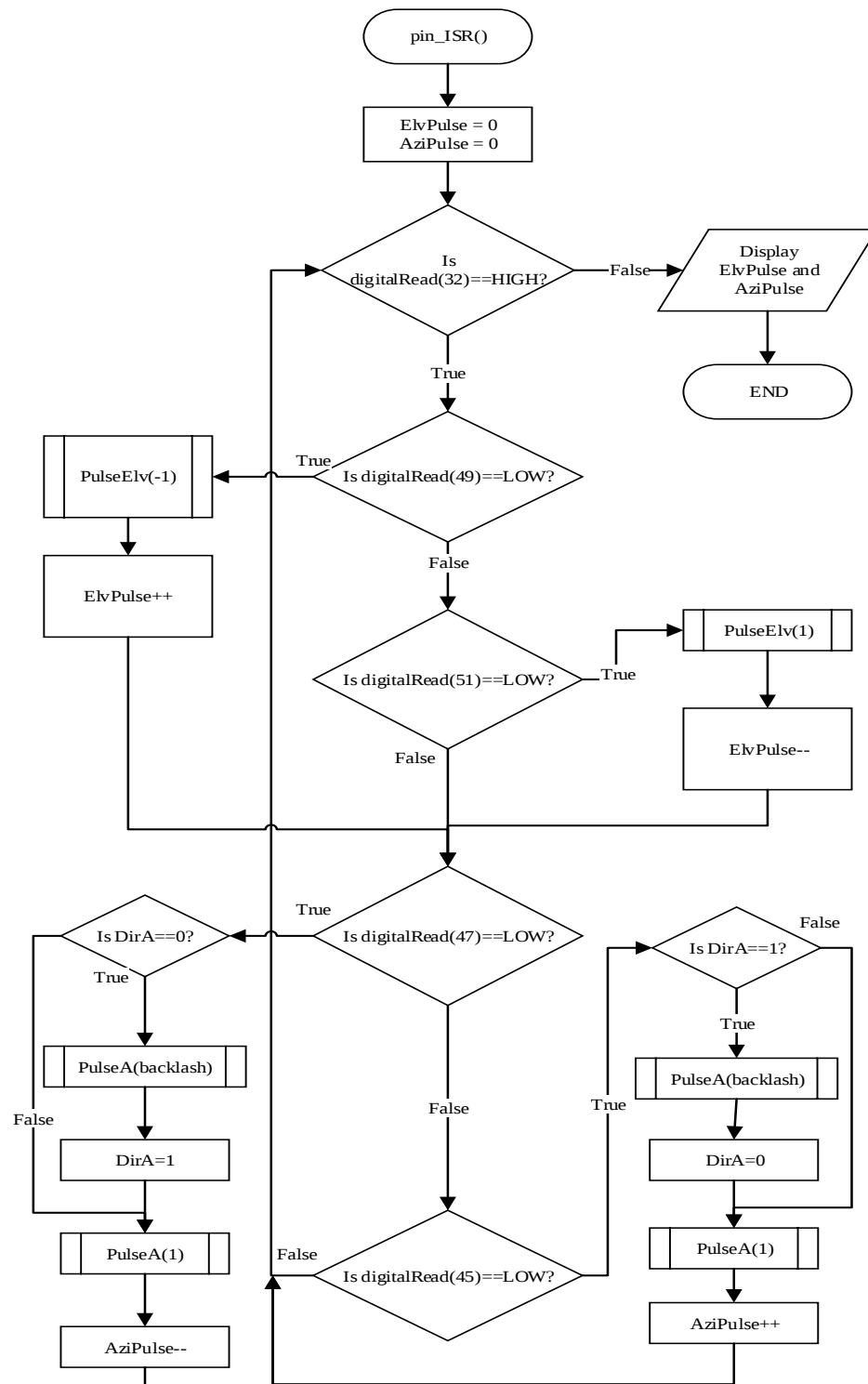


Figure 3.33: Flow Chart to Measure Tracking Accuracy

3.5.5 Overall Solar Tracking Program

First of all, input of current time zone, date, time, and position is required. Since encoder is not used in this project, before the tracking starts, the solar concentrator need to be reset to its initial position. Before running the code, the sensor need to position as such it is pointing to North direction. Linear actuator motor driver purchased has a built-in encoder and EEPROM memory inside, therefore the initial position can be stored inside the motor driver. Linear actuator driver pin 29 is the position reset pin, a simple pulse to this pin will send reset command to the motor driver. This pin is connected to Arduino output pin 28. For azimuth stepper motor, the U-shaped micro photoelectric is used to reset solar concentrator back to initial position. Figure 3.34 below is the flow chart for resetting.

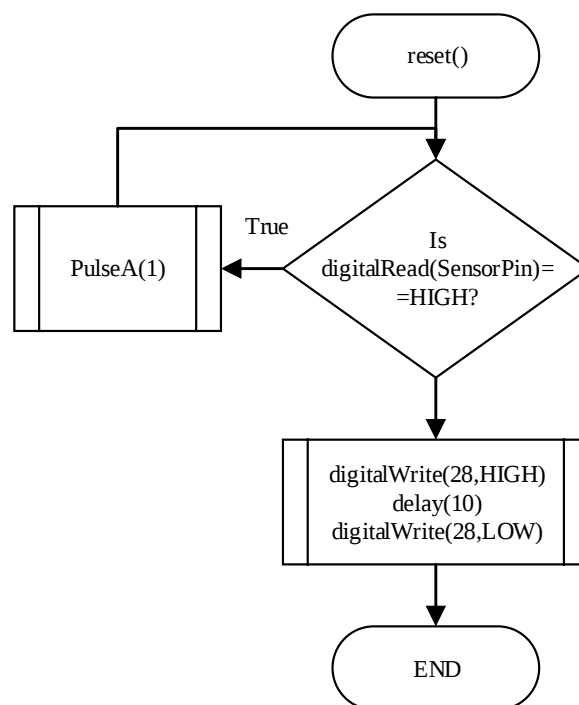


Figure 3.34: Flow Chart to reset The Solar Concentrator to Initial Position

Tracking starts after the solar concentrator is reset to its initial position. First of all programmer will run solar() function in Figure 3.17 to update the solar altitude and azimuth angle. Altitude angle is stored as 'elv' variable while azimuth angle is stored as 'A' variable. As mentioned previously, the mechanical limit for altitude angle is within 40 °to 90 °, therefore if the calculated altitude angle is out of range, the system

will sleep and keep updating the altitude angle every 30 seconds until it is within the limit. When the altitude angle is within the limit, it will be converted into length of linear actuator shaft using equation (3.20).

Moving on, the length and altitude angle are stored in a temporary variable “elvP” and “AP” for next cycle calculation and converted into number of steps using equation (3.21) and equation (3.22) respectively. The calculation of steps will have some floating points and the motor drivers only receive integer number of pulses. If these floating points are ignored, it will caused loss of pulse in the future. Therefore the floating points are stored as variables ‘errorE’ and ‘errorA’. The floating points will be added into the calculated pulse number in next cycle of calculation. With this error compensation algorithm, the maximum error for azimuth motor steps will be $\pm 0.018^\circ$ and $\pm 0.003\text{mm}$ for linear actuator steps.

Before sending the pulse into the motor driver, backlash is compensated if azimuth motor is rotating in different direction and then the linear actuator and azimuth stepper motor move the solar concentrator to desired position. Azimuth angle and altitude are set to be updated every 30 seconds, therefore the system will sleep for 30 seconds after every positioning. During the next cycle of calculation, new calculated angles will deduct the previous angles stored in temporary variable, the calculated results will be the angles to be moved. Figure 3.35 is a detailed algorithm for solar tracking and Figure 3.36 is an overall tracking program flow chart.

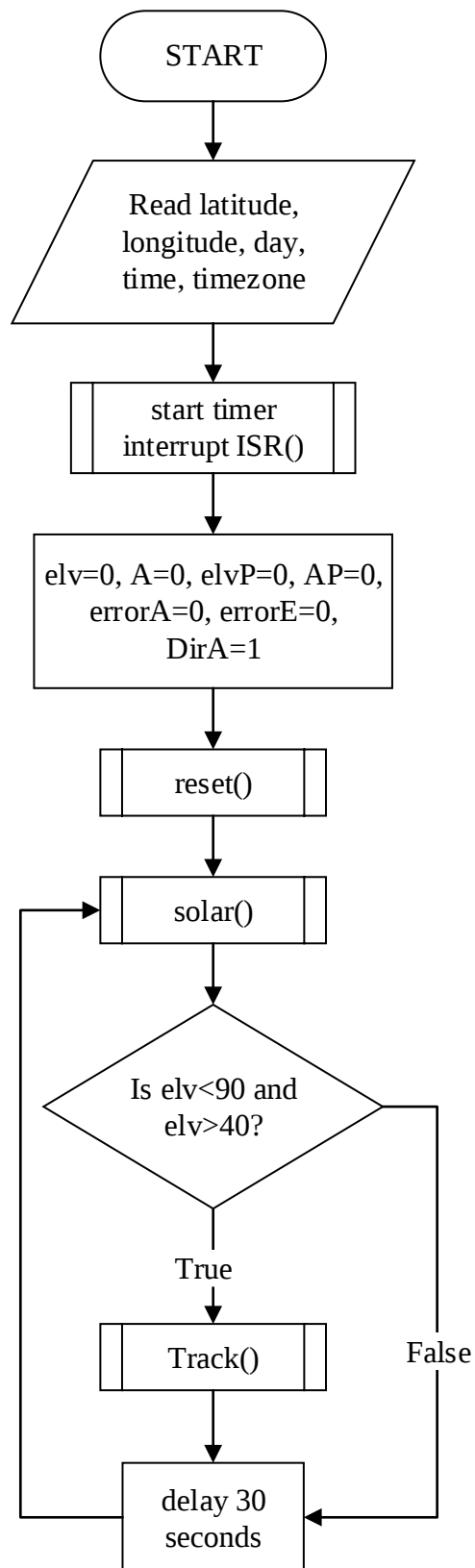


Figure 3.36: Flow Chart for Overall Tracking System

3.6 Circuit Design

Figure 3.37 is the overall schematic diagram including two stepper motors, motor drivers, closed-loop sensors, U-shaped micro photoelectric sensor, button interrupt and manual controller. The schematic diagram was drawn using Multisim software. Note that only used pins were drawn into the schematic to avoid confusion.

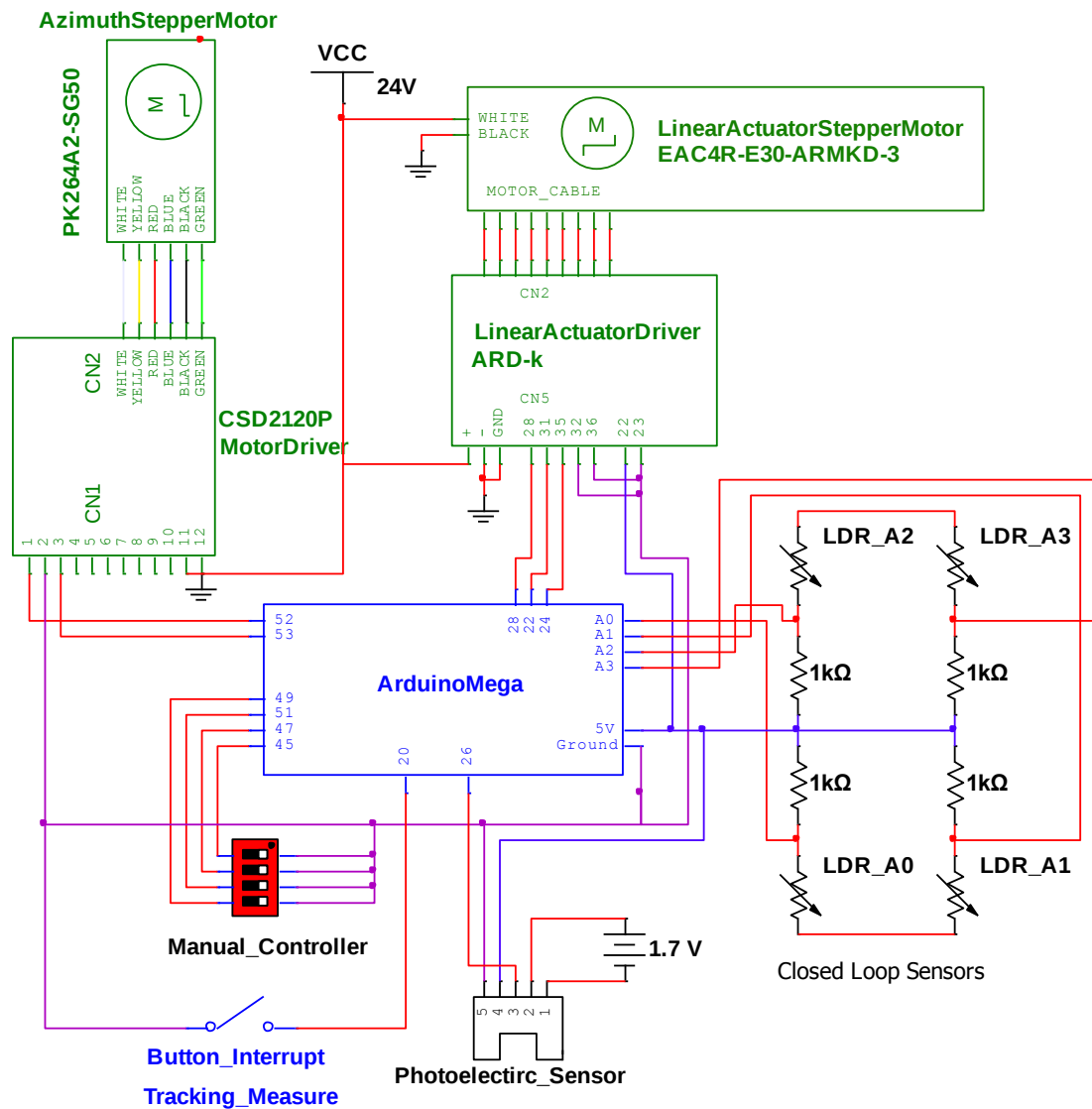


Figure 3.37: Overall Circuit Design

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Backlash Measurement Result

Experiment to measure backlash of the system is designed in section 3.5.3. The experiment was repeated for ten times and the average backlash was used in the programming code. To obtain more accurate result, the backlash not only tested in clockwise rotation but also tested in counter clockwise rotation subsequently. Total pulse display is converted backlash error in degrees using equation (3.26). All the results are recorded in Table 4.1.

Table 4.1: Backlash Error Results

no.	Total Pulse Sent	Backlash (Pulses)	Backlash Error (degrees)
1	45591	187	1.48104
2	45593	189	1.49688
3	45592	188	1.48896
4	45593	189	1.49688
5	45592	188	1.48896
6	45592	188	1.48896
7	45592	188	1.48896
8	45594	190	1.5048
9	45592	188	1.48896
10	45591	187	1.48104
average:		188.2	1.490544

From Table 4.1, the results for backlash error in 10 consecutive experiments varied around $\pm 0.01^\circ$; it can be assumed that the result is quite consistent. Therefore backlash error 1.49° will be the value in backlash compensation. Every time the programmer rotates azimuth stepper motor in different direction, it must rotate for 1.49° first, which is equal to 188 pulses.

4.2 Tracking Accuracy

To measure tracking accuracy, designed programming code in section 3.5.4 is used. Tracking programming code was designed as a button interrupt at Arduino interrupt pin 20. Whenever pin 20 is triggered during tracking, the tracking will stop for measurement. Displayed result ElvPulse and AziPulse are the tracking errors for altitude angle and azimuth angle. The AziPulse is converted into angle using equation (3.22) but for ElvPulse equation (3.20) and equation (3.21) need to be combined and modified before it can be used.

Substitute (3.20) into (3.21),

$$no. of pulse = \frac{\sqrt{101700 - 100800 \cos(90^\circ - \alpha)}}{0.006} \quad (4.1)$$

Rearranging (4.1),

$$\alpha = 90^\circ - \cos^{-1} \left(\frac{101700 - (no. of pulse \times 0.006)^2}{100800} \right) \quad (4.2)$$

Second altitude angle should be number of pulse added by ElvPulse, therefore the tracking error should be $\nabla \alpha$.

$$|\nabla \alpha| = \left[90^\circ - \cos^{-1} \left(\frac{101700 - [(no. of pulse + ElvPulse) \times 0.006]^2}{100800} \right) \right] - \left[90^\circ - \cos^{-1} \left(\frac{101700 - (no. of pulse \times 0.006)^2}{100800} \right) \right] \quad (4.3)$$

Further simplify (4.3),

$$|\nabla\alpha| = -\cos^{-1}\left(\frac{101700 - [(no. of pulse + ElvPulse) \times 0.006]^2}{100800}\right) + \cos^{-1}\left(\frac{101700 - (no. of pulse \times 0.006)^2}{100800}\right) \quad (4.4)$$

Equation (4.4) is the final equation to calculate elevation tracking error. $\nabla\alpha$ is solar altitude tracking error. Number of pulse is the total pulse sent for solar tracking.

Measurement of solar tracking error was carried out in a period of three hours from 1000 morning to 1300 in the afternoon for three consecutive days, 6th, 7th and 8th April 2016. Tracking error was recorded for every 15 minutes interval. Photoelectric sensor mounted at the side of the hardware shown in Figure 3.29 and Figure 3.30 will be set as reference for North point. Before the experiment starts, the whole structure was align manually so that the sensor was pointing to the North. A digital compass in iPhone 6 shown in Figure 4.1 below was used to find the North point from structure position. Since whole alignment was done manually, certain amount of inaccuracy was expected. The Result for three days was recorded in Table 4.2, Table 4.3 and Table 4.4 respectively.

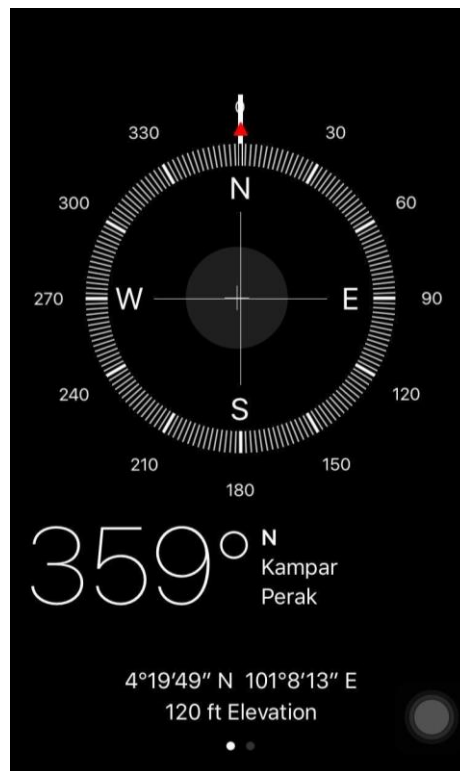


Figure 4.1: Digital Compass in iPhone 6

Table 4.2: 6th April 2016 Tracking Results

6th April

time (24-hour)	Azimuth error (pulses)	Azimuth error (degrees)	Altitude error (pulses)	Altitude error (degrees)
1000	1011	8.00712	453	0.77241
1015	1023	8.10216	521	0.87772
1030	950	7.524	533	0.88863
1045	1107	8.76744	475	0.78519
1100	1005	7.9596	491	0.8067
1115	998	7.90416	531	0.86957
1130	987	7.81704	503	0.82407
1145	1003	7.94376	511	0.84203
1200	1120	8.8704	489	0.81741
1215	1352	10.7078	487	0.8382
1230	1567	12.4106	476	0.86885

1245	1485	11.7612	431	0.89127
1300	1481	11.7295	342	0.93039
average:		9.192683		0.847111

Table 4.3: 7th April 2016 Tracking Results

7th April

time (24-hour)	Azimuth error (pulses)	Azimuth error (degrees)	Altitude error (pulses)	Altitude error (degrees)
1000	1233	9.76536	503	0.85759
1015	1027	8.13384	527	0.88767
1030	1122	8.88624	482	0.80339
1045	1335	10.5732	514	0.84962
1100	1338	10.597	553	0.90857
1115	1191	9.43272	519	0.8499
1130	1034	8.18928	496	0.81263
1145	1267	10.0346	507	0.8355
1200	1219	9.65448	521	0.87086
1215	1452	11.4998	519	0.89302
1230	1599	12.6641	462	0.84302
1245	1570	12.4344	433	0.89194
1300	1556	12.3235	404	1.06524
average:		10.3222		0.87454

Table 4.4: 8th April 2016 Tracking Results**8th April**

time (24-hour)	Azimuth error (pulses)	Azimuth error (degrees)	Altitude error (pulses)	Altitude error (degrees)
1000	1387	10.985	485	0.82668
1015	1269	10.0505	473	0.79646
1030	1355	10.7316	499	0.83164
1045	1311	10.3831	530	0.87601
1100	1283	10.1614	471	0.77374
1115	1297	10.2722	522	0.8548
1130	1302	10.3118	557	0.91256
1145	1196	9.47232	534	0.88
1200	1254	9.93168	463	0.77415
1215	1539	12.1889	481	0.82782
1230	1568	12.4186	506	0.92168
1245	1503	11.9038	454	0.92997
1300	1516	12.0067	408	1.05099
average:		10.8321		0.86588

From the results, the average azimuth tracking error and altitude tracking error for three days are 10.12° and 0.86° respectively. The solar azimuth tracking error is relatively higher compare to solar altitude tracking error. The reason of high error is believed to be caused by inaccuracy when setting the initial position of solar concentrator. Metal around the structure might also affect the accuracy of digital compass in iPhone 6. Inaccurate input of latitude and longitude can also cause tracking error in the system. Besides that, it can also be observed that azimuth tracking error increased significantly after 1200. The reason is because the azimuth tracking angle suddenly change significantly after 1200. Slanted ground surface where the experiment was carried out affecting the initial position and causing error to solar altitude tracking.

4.3 Solar Tracking Improvement

From the experiment results in section 4.2, it can be concluded that solar azimuth tracking was quite inaccurate. The reasons were due to inaccuracy when placing the solar concentrator to its initial position. There are a few methods to solve this problem. The method chosen to be used was to adopt a closed-loop feedback of the sun's location. By including a closed-loop feedback, the system becomes a hybrid system.

Light-dependant resistor (LDR) will be used as solar sensor for closed-loop tracking. When light intensity hitting the LDR increases, its resistant will decrease. By using this properties of the LDR, solar irradiance can be detected. Four LDRs were used to detect light from up, down, left and right side. To do this, the sensors need to be mounted on four walls to block the LDRs from the sun when the incoming sunlight is not perpendicular to the solar concentrator. Figure 4.2 is a picture of how the LDRs are fixed on the sensor holder. The whole sensor structure will be place at top and parallel with the solar concentrator.

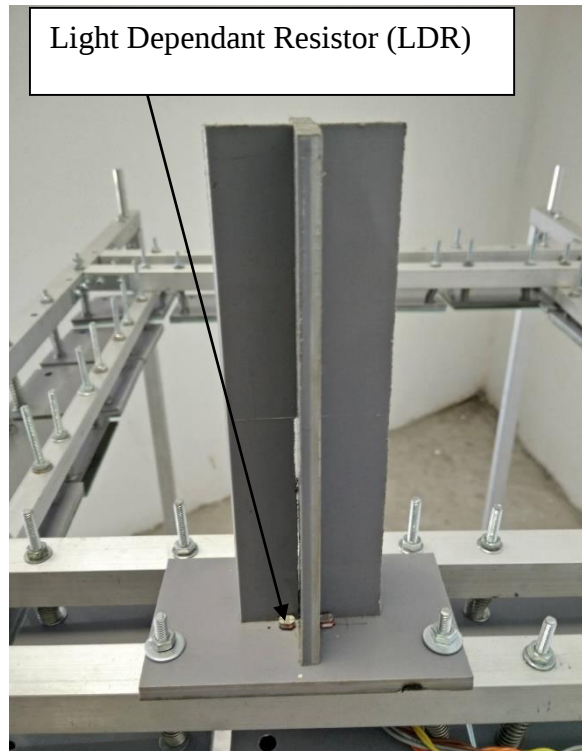


Figure 4.2: Light-dependant Resistors fixed on Sensor Holder

Voltage divider rule was used to read the light intensity of the sunlight in terms of voltage. Four different LDR voltages will be read by Arduino analogue pin A0, A1, A2 and A3. The labelling and the top view of LDRs is shown in Figure 4.3. Sum of A0 and A3 will be the light intensity at the left side, sum of A1 and A2 is the light intensity at the right side. If sum of A0 and A3 is higher compare to sum of A1 and A2, then the motor will rotates clockwise until sum of A1 and A2 is higher or the same. Same vice versa process will be carried out if A1 and A2 is higher. For solar altitude tracking, if sum of A3 and A2 is higher than sum of A0 and A1, means that the sun is closer to Zenith, therefore the linear actuator needs to retract. The detailed flow chart is shown in Figure 4.4.

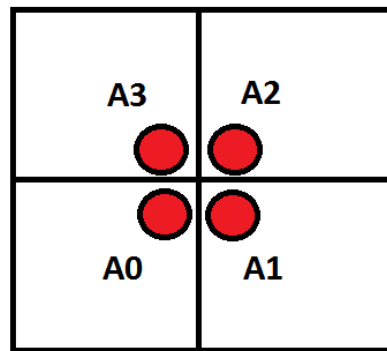


Figure 4.3: Top View and Labelling of LDRs

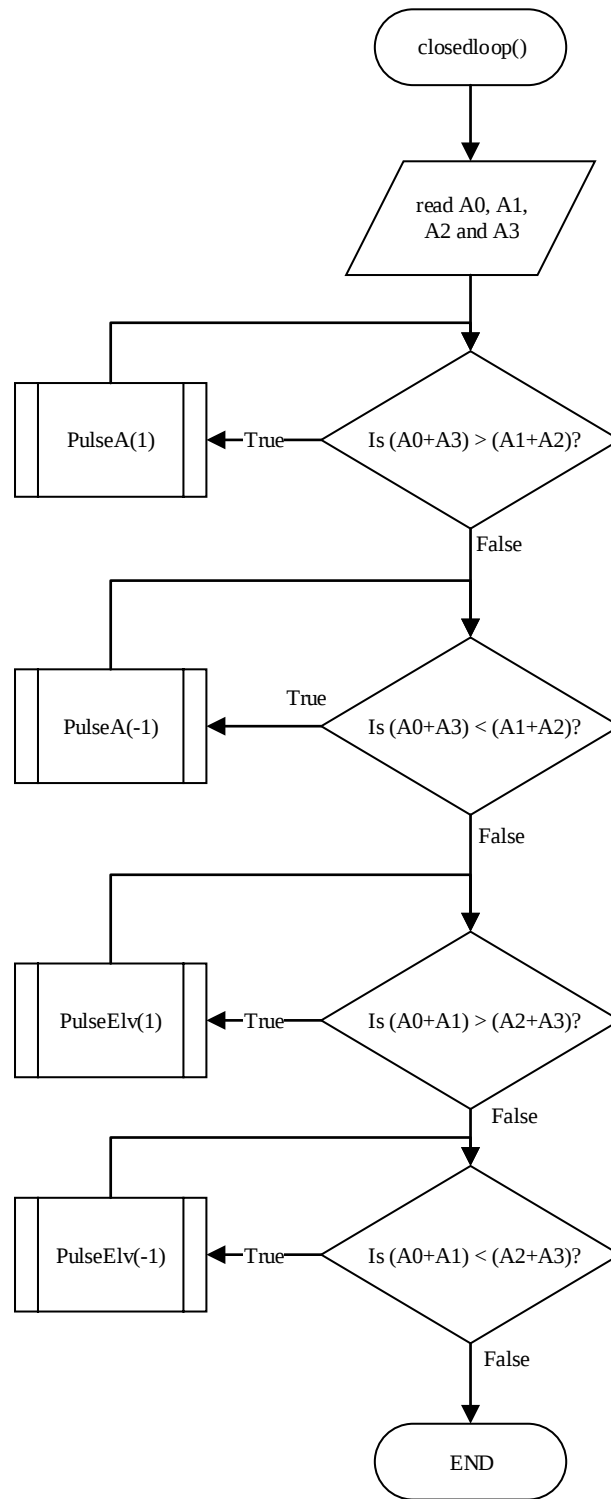


Figure 4.4: Flow Chart of Closed-loop Tracking

Common hybrid tracking method use open-loop tracking to move target close to the sun position and then activate closed-loop tracking for more precise adjustment. The purpose of this method is to compensate the inaccurate open-loop tracking.

However this kind of system will be greatly affected by the weather. When the weather is bad, the system will shut down closed-loop tracking and activate open-loop tracking. The accuracy will drop due to inaccurate open-loop tracking. Normally to increase the accuracy of open-loop tracking, expensive digital compass, GPS module and expensive sensors are used to get accurate initial data for initial positioning. To reduce cost, closed-loop sensors are used to compensate the errors due to inaccurate initial positioning.

First the system will operate open-loop to operate the solar concentrator facing approximately to the sun. After that closed-loop tracking will be activate for to move the concentrator to a more accurate position. When the solar concentrator is set to accurate position, the closed-loop sensor will be turned off and open-loop system will run the next cycle of tracking. This method is explained clearer in the form of flow chart in Figure 4.5.

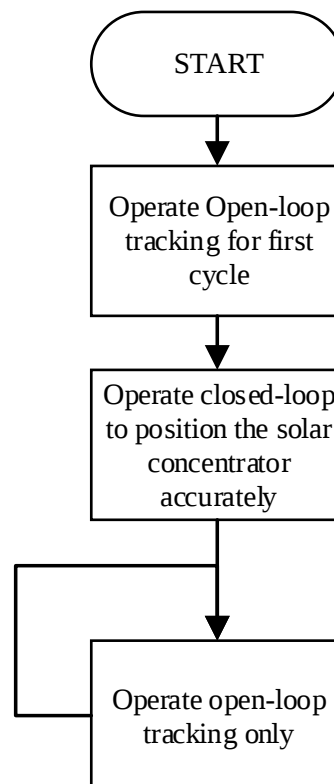


Figure 4.5: Overall Hybrid Tracking System

By using closed-loop system to set the first position, input of initial location, coordinates and direction need not be very accurate. Therefore the cost of using expensive sensors can be saved. Furthermore, since the closed-loop tracking is only run once, the next cycles of tracking will not be affected by weather.

Unfortunately, the LDRs used faces some issues. During indoor experiment, the LDRs can operate normally but when the experiment was taken outdoor, the sunlight intensity is too high, LDRs value is at maximum even though it is shaded. The LDRs were also located too far away from the programmer, longer wires used to connect the LDR to the programmer cause the increase in resistance and voltage loss. These factors cause a large error when reading the voltage. Due to lack of time, the improvement of closed-loop sensors and experiments to obtain tracking accuracy could not manage to carry out before the deadline of FYP2.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the solar tracking general formula was studied and integrated into this project. Several formulas were modified and improved for the application for solar daylighting. In section 3.4, several formulas were derived to convert solar altitude angle into linear length of actuator shaft via the law of cosine and conversion from secondary gear rotations to primary gear rotations.

During construction of the programming code, one of the problem faced is the accuracy in calculating time. Timer interrupt in Arduino Mega was used to count the time accurately for every seconds instead of using delay function. Furthermore, to ease the process on writing codes, the programming codes were written into few major functions and they were explained section by section in the report. Another issue encountered is the backlash problem from the mechanical structure. The problem was also solved by measuring the backlash and compensating it during operation.

After developing the software and hardware, the tracking accuracy measurement was carried out outdoors. The solar azimuth tracking was not so accurate due to inaccurate positioning, coordinates input and also slanted ground surface. These issues were planned to solve by integrating an active tracking system but unfortunately due to unforeseen issues of the sensors and lack of time, the accuracy measurement of the system cannot be carried out.

5.2 Recommendations

One of the recommendations to improve the overall system is to reduce the backlash. Even though the backlash is compensated in software, external factors such as strong wind can still move the solar concentrator when stationary. Secondary gear of our prototype was constructed manually by slotting the screws in a circular plate. The backlash can be reduced if the spacing between the screws is reduced.

Furthermore, the design of solar concentrator can also be improved. The space between the mirrors is not utilised well enough. Further research on the solar concentrator design is suggested to be carried out.

Lastly, improvement on the closed-loop tracking sensors is also recommended. As mentioned in section 4.3, the hybrid system could not be completed due to the inaccuracy of sensors. Changing the LDR to a better sensor such as a photodiode might be able to improve reading light intensity outdoors.

REFERENCES

- A. Dunne, 1998. Some Effects of the Quality of Light on Health. *Journal of Orthomolecular Medicine*, Vol.4, No. 4, 1989, pp. 229-232.
- A-levelphysicstutor.com, N.D. *Telescopes*. [online] Available at: < <http://www.a-levelphysicstutor.com/optics-telescs.php#gal>> [Accessed 20 August 2015].
- Chuck, J., 2013. *5 Ways to Track Your Solar Tracker*. [blog] Available at: <<http://blog.pepperl-fuchs.us/blog/bid/253098/5-Ways-to-Track-Your-Solar-Tracker>> [Accessed 24 October 2015].
- Deepthi, S., Ponni, A., Ranjitha, R. and R. Dhanabal. (2013). Comparison of Efficiencies of Single-Axis Tracking System and Dual-Axis Tracking System with Fixed Mount. *International Journal of Engineering Science and Innovative Technology*, 2(2).
- Geo, B., N.D.. *Arduino Solar Tracker*. [online] Available at: <<http://www.instructables.com/id/Arduino-Solar-Tracker/>> [Accessed 20 August 2015].
- Irfan, U. and Seoyong, S. (2013). Concept of Solar Tower for Daylighting in Multi-Floor Buildings. *Journal of Green Science and Technology*, 1, pp. 1-6.
- Kribus, A.; Vishnevetsky, I.; Yogeve, A. & Rubinov, T. (2004). Closed loop control of heliostats. *Energy*, Vol 29, No. 5-6, (April-May 2004) pp. 905-913, ISSN 0360-5442.
- Lee, J.F. and Nasrudin, A.R. (2013). Performance Comparison of Dual-Axis Solar Tracker vs Static Solar System in Malaysia. 2013 IEE Conference on Clean Energy and Technology (CEAT), pp. 102-107.
- Luque-Heredia, I.; Cervantes, R. & Quesmere, G. (2006). A sun tracking error monitor for photovoltaic concentrators. *Proceedings of the IEEE 4th World Conference on Photovoltaic Energy Conversion*, pp. 706-709, isbn 1-4244-0017-1, Waikoloa Village, USA, May 7-12 2006.
- Mayhoub, M.S., 2014. Innovative daylighting systems challenges: A critical study. *Energy and Buildings*. DOI: 10.1016/j.enbuild.2014.04.019.

Solatube.com, (N.D.). 290 DS/ 290 DSe. [online] Available at: <<http://www.solatube.com/residential/daylighting#models>> [Accessed 24 August 2015].

APPENDICES

APPENDIX A: Source Code for Backlash Measurement

```

int TotalPulse = 0;

int ACWPin = 52;
int ACCWPin = 53;

int SensorPin = 26;

void setup() {
    Serial.begin(9600);

    pinMode(ACWPin, OUTPUT);
    pinMode(ACCWPin, OUTPUT);

    pinMode(SensorPin, INPUT);

    //reset to initial position
    while(digitalRead(26) != LOW) {
        PulseA(-1);
    }

    //Move away from the sensor, but still counting the pulse
    while(digitalRead(26) == LOW) {
        PulseA(1);
        TotalPulse++;
    }

    //Move until one rotation is complete
    while(digitalRead(26) != LOW) {
        PulseA(1);
        TotalPulse++;
    }

    Serial.println(TotalPulse);
}

void loop() {
}

void PulseA(double numTimes) {
    if (numTimes >= 0) {
        for (double i = 0; i < numTimes; i++) {

```

```
        digitalWrite(ACWPin, HIGH);  
        delay(2);  
        digitalWrite(ACWPin, LOW);  
        delay(2);  
    }  
}  
  
if (numTimes<0){  
    numTimes=numTimes*-1;  
    for(double i = 0;i<numTimes;i++){  
        digitalWrite(ACCWPin, HIGH);  
        delay(2);  
        digitalWrite(ACCWPin, LOW);  
        delay(2);  
    }  
}  
}
```

APPENDIX B: Overall Source Code

```

float LO = radians(101.79);
float LA = radians(3.12);
float day=91;
float hour=11.0;
float minute=51.0;
float time1 = hour + (minute/60.0);
float D =0;
float TZ = 8;

float elapse_time;
float elv=0;
float A=0;
float tempElv, tempA;
float stepsElv, stepsA;
float elvP=0, AP=0, errorE=0, errorA=0, steptotalElv=0,
steptotalA=0, DirA=1, DirE=1;
float elvdownlimit = 50;

int ACWPin = 52;
int ACCWPin = 53;
int ECWPin = 22;
int ECCWPin = 24;
int emergencyPin = 20;
int SensorPin = 26;
int UP=49;
int DOWN=51;
int LEFT=47;
int RIGHT=45;

int sensor1, sensor2, sensor3, sensor4;

void setup() {

    Serial.begin(9600);

    // initialize timer1
    noInterrupts();           // disable all interrupts
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1  = 0;

    OCR1A = 62500;           // compare match register 16MHz/256/2Hz
    TCCR1B |= (1 << WGM12);  // CTC mode
    TCCR1B |= (1 << CS12);   // 256 prescaler
    TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt

    interrupts();           // enable all interrupts

    //This section of code reset the Azimuth and Elevation to initial
    pinMode(30, INPUT_PULLUP); //start pin
    pinMode(28, OUTPUT); //reset linear actuator pin
    pinMode(32, INPUT_PULLUP); //Manual mode or tracking mode

```

```

pinMode(ACWPin, OUTPUT);
pinMode(ACCWPin, OUTPUT);
pinMode(ECWPin, OUTPUT);
pinMode(ECCWPin, OUTPUT);
pinMode(SensorPin, INPUT);
pinMode(emergencyPin, INPUT_PULLUP);

pinMode(UP, INPUT_PULLUP);
pinMode(DOWN, INPUT_PULLUP);
pinMode(LEFT, INPUT_PULLUP);
pinMode(RIGHT, INPUT_PULLUP);

attachInterrupt(digitalPinToInterrupt(emergencyPin), pin_ISR, RISING); //Emergency pin

while(digitalRead(30)==HIGH){}

reset();
}

ISR(TIMER1_COMPA_vect)          // timer compare interrupt service
routine
{
    time1=time1+1.0/3600;
    if (time1>=24){
        if (day==365){
            day =1;
        }
        else{
            day=day+1;
        }
        time1=0;
    }
    //Serial.print(time1,8);
    //Serial.print(" ");
    //Serial.print(day);
    //Serial.print("\n");
}

void loop() {

    solar();

    if (elv<90 && elv>elvdownlimit){

        Track(); // start tracking

    }

    delay(30000); //delay 30 seconds
}

void solar()
{
    float N = day;
    float LCT = time1;
    float d, x, LC, EOT, ts, w, cosA;

    LCT = time1;

    d = asin(0.39795 * cos (radians(0.98563 * (N-173))));

```

```

x = 360 * (N-1) / 365.242;
x = radians(x);

EOT = 0.258 * cos(x) - 7.416*sin(x) - 3.648*cos(2*x) -
9.228*sin(2*x);

LC = ((TZ*15) - degrees(LO))/15;

ts = LCT + EOT/60 - LC - D;

w = 15 * (ts - 12);
w = radians(w);

elv = asin(sin(d) * sin(LA) + cos(d) * cos(w) * cos(LA));

cosA = (sin(d)*cos(LA) - cos(d)*cos(w)*sin(LA))/cos(elv);

if (cosA > 0)
{
    A = asin(-cos(d)*sin(w)/cos(elv));
    A = degrees(A);
}
if (cosA < 0)
{
    A = 180 - degrees(asin(-cos(d)*sin(w)/cos(elv)));
}

elv = degrees(elv);
}

void PulseA(double numTimes){
    if (numTimes>=0){
        for(double i = 0;i<numTimes;i++){
            digitalWrite(ACWPin, HIGH);
            delay(2);
            digitalWrite(ACWPin, LOW);
            delay(2);
        }
    }

    if (numTimes<0){
        numTimes=numTimes*-1;
        for(double i = 0;i<numTimes;i++){
            digitalWrite(ACCWPin, HIGH);
            delay(2);
            digitalWrite(ACCWPin, LOW);
            delay(2);
        }
    }
}

void PulseElv(double numTimes){
    if (numTimes>=0){
        for(double i = 0;i<numTimes;i++){
            digitalWrite(ECWPin, HIGH);
            delay(2.5);
            digitalWrite(ECWPin, LOW);
            delay(2.5);
        }
    }
}

```

```

    if (numTimes<0){
        numTimes=numTimes*-1;
        for(double i = 0;i<numTimes;i++){
            digitalWrite(ECCWPin, HIGH);
            delay(2.5);
            digitalWrite(ECCWPin, LOW);
            delay(2.5);
        }
    }
}

void reset(){

    PulseA(-1000);

    if (digitalRead(26)==LOW) {
        Serial.print("pulsing5000 ");
        PulseA(-5000);
    }

    while (digitalRead(SensorPin)==HIGH) {
        digitalWrite(ACWPin, HIGH);
        delay(2);
        digitalWrite(ACWPin, LOW);
        delay(2);
    }

    PulseA(52);

    errorA=-0.3263;

    PulseElv(-32019);

    digitalWrite(28,HIGH);
    delay(10);
    digitalWrite(28,LOW);
}

void Track(){
    elv = sqrt((21*21+24*24-2*21*24*cos(radians(90-elv)))); //Covert
    to length

    tempElv = elv;
    tempA = A;

    elv = elv - elvP;
    A = A - AP;

    elvP = tempElv;
    AP = tempA;

    stepsElv = (elv/0.0006) + errorE;
    stepsA = ((A*50.0/11.0)/0.036) + errorA;

    errorE = stepsElv - round(stepsElv);
    errorA = stepsA - round(stepsA);

    stepsElv = round(stepsElv);
    stepsA = round(stepsA);
}

```

```
steptotalElv = steptotalElv + stepsElv;
steptotalA = steptotalA + stepsA;

if(stepsA>0 && DirA==0){
    PulseA(190); //backlash
    DirA=1;
}
else if(stepsA<0 && DirA==1){
    PulseA(-190); //backlash
    DirA=0;
}

PulseA(stepsA);
PulseElv(stepsElv);
}

void pin_ISR() {

    while(1){
        Serial.print("stop\n");}
}
```