DUAL AXIS SOLAR TRACKER STUDY

LIM ZI CHAO

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **DUAL AXIS SOLAR TRACKER STUDY**

SESI PENGAJIAN: 2010/2011

Saya **LIM ZI CHAO (870509-04-5287)**

(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)

Alamat Tetap:

LOT 870 KM-8 JLN ENDAU-ROMPIN
KG JANGLAU 26820 KUALA ROMPIN
PAHANG

HAMDAN BIN DANIYAL
(Nama Penyelia)

Tarikh: **20 AUGUST 2010**

Tarikh: : **20 AUGUST 2010**

CATATAN:

* Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.

♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature :____

Name : OMAR BIN ALIMAN

Date : 20 AUGUST 2010

DUAL AXIS SOLAR TRACKER STUDY

LIM ZI CHAO

This thesis is submitted as partial fulfilment of the requirements for the award of the
Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

AUGUST, 2010

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : ______________________________

Author : LIM ZI CHAO

Date : 20 AUGUST 2010

.

To my beloved family and friends,

ACKNOWLEDGEMENT

First of all, I would like to thank my project supervisor, En. Omar Bin Aliman, who has given me much strong logistic support while implementing the project given. He has always assisted me when I handling my project. Besides, I would like to express my sincere appreciation for his valuable advices, guidance and encouragement. This has inspired me to be more confident in trying new things.

Secondly, I would like to thank my family members for giving me their loves and supports throughout my four years of studies in Universiti Malaysia Pahang (UMP), Gambang, Pahang.

Special thanks to staff FKEE, who have given me a great help in accomplishing this project.

Last but not least, I would like to say millions of thanks to all my coursemates and those who has lending me their helping hand.

Thank you

ABSTRAK

Analisis untuk pengesan solar dua hala akan dilaksanakan. Dalam pengembangan teknologi yang tinggi pada masa kini, teknologi pengesan solar dua hala tidak diimplementasikan dengan sepenuhnya. Satu lagi solar sistem yang dipanggil pengesan solar satu hala pula jauh lebih popular dan digunakan secara luas. Namun begitu, pengesan solar satu hala mempunyai keberkesanan yang rendah. Justeru, analisi untuk pengesan solar dua hala akan dilaksanakan untuk mengambil alih penggunaan pengesan solar satu hala yang tidak berksan. Cara untuk menguji prestasi sistem pengesan yang dianalisis adalah dengan menggunakan satu alat pengesan solar untuk menguji samada pancaran dari matahari adalah mengarah kepada permukaan pengesan solar tersebut atau tidak dengan menggunakan pengiraan secara matematik bagi sudut azimuth dan sudut zenith. Satu simulasi menggunakan Borland C++ Builder akan direka untuk menunjukkan proses dan langkah untuk menjalankan satu pengesan solar dua hala. Pada akhir projek ini, ia adalah dijangkakan bahawa sudut azimuth-zenith yang dikira adalah tepat pada mana-mana tempat dan bila-bila masa.

ABSTRACT

An analysis for a dual axis solar tracking study is to be carried out. In the current growth of technology, the technology of this dual axis solar tracking system has not been fully utilized. The other solar power system called single axis solar power system is much more popular and widely used. However, single axis solar tracking system has a very low efficiency, yet it is not cost effective. Therefore, the analysis of dual axis solar tracking system is to be carried out to take over the inefficient single axis solar tracking system. An independent method will be used in order to evaluate the performance of the proposed tracking system. It is by practically using a dummy solar tracker to test whether the reflection of the sun is accurately towards the target point at any given time according to the mathematical calculation of the elevation and azimuth angles of the sun. A simulation using Borland C++ Builder will be created to simulate the processes and the steps to operating a real dual axis solar tracker. At the end of this project, it is expected that the azimuth-elevation angles calculated are precise and accurate at any place and time.

TABLE OF CONTENTS

CHAPTER **TITLE** **PAGE**

1 INTRODUCTION

1.1 Background of study 1
1.2 Design Objectives 2
1.3 Scope of Project 3
1.4 Significance of study 4
1.5 Thesis Overview 5

2 LITERATURE REVIEW

2.1 Introduction 6
2.2 Photovoltaic systems 7
2.3 Time 14
  2.3.1 Sundial 16
  2.3.2 Elliptical orbitx 17
  2.3.3 Inclination of the ecliptic 19
  2.3.4 Equation of time 21
  2.3.5 Sunrise and sunset 23
  2.3.6 Exercises 25

2.4 Tracking systems 26
  2.4.1 Closed-loop types of sun tracking systems 29
  2.4.2 Open-loop types of sun tracking systems 37

3 METHODOLOGY
3.1 Introduction 42
3.2 Mathematical calculations 43
3.3 Dummy solar tracker 45
3.4 Simulation of dual axis solar tracker 51
  3.4.1 The initialization process of the tracker 52
  3.4.2 The operation of the tracker during run-time 56

4 RESULTS AND ANALYSIS
4.1 Introduction 64
4.2 Simulation 64
  4.2.1 Table of calculations for azimuth-elevation angle 65
  4.2.2 Analysis of azimuth-elevation angles using different parameters 67
4.3 Dummy Solar Tracker 71
  4.3.1 Rotation of Dummy Solar Tracker 72
  4.3.2 Dummy Solar Tracking Results 73

5 CONCLUSION

5.1 Conclusion 75
5.2 Future recommendations 76

REFERENCES 77
APPENDICES 84-108

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	Comparison of electricity production and cost between different tracking systems	14
2.2	Performance of sun tracking systems [48-85]	28
2.3	Performance comparison of PSA and Michalsky algorithms used to predict sun's position over the period 1999-2015 [74,75]	38
4.1	Calculations for azimuth-elevation angles	66
4.2	Azimuth-elevation angles for different latitudes	67
4.3	Azimuth-elevation angles for different time of the day	68
4.4	Azimuth-elevation angles for different dates	71

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	One and two axis tracking PV array	10
2.2	Interval of sun’s altitude in a year	11
2.3	Example power with PV static and moving	13
2.4	Comparison Diagram of Dual-Axis tracker & Fixed-Angle installation [31]	13
2.5	The Earth must rotate 360 degrees plus a very small angle, for observer at A to return to the same position relative to the Sun at B	18
2.6(top)	The Earth moves slowest at A and fastest at B	19
2.6(bottom)	Equation of Time component due to the eccentricity of the Earth's orbit	19
2.7	Inclination of the Earth's axis with respect to the plane of its orbit	20
2.8	The annual apparent path of the Sun, the ecliptic, reaches 23½ o north and south of the celestial equator	20
2.9	Equation of Time component due to the obliquity of the ecliptic. (obliquity = 23½ o)	21
2.10	The Equation of Time	22
2.11	The daily or diurnal paths of the Sun during the solstices (21 December and 21 June) and the equinoxes (21 March and 21 September) as seen by an observer at the equator. Solid lines are daytime, dashed lines are night-time.	24

2.12(a)	Collector acceptance angle	31
2.12(b)	Illustration of sun tracking mechanism. Reproduced with permission from Elsevier [55]	31
2.13	Block diagram of sun-tracking system. Reproduced with permission from Elsevier [55]	33
2.14(a)	Four-quadrant sensor and	33
2.14(b)	Transfer functions for both axes. Reproduced with permission from Elsevier [73]	33
2.15(a)	Schematic illustration of vision-based heliostat control system	34
2.15(b)	Photographs showing different shapes of sun images projected by heliostats onto target plane	35
2.15(b)(i)	Centred ellipsoids (the shape of the ellipsoid changes during the day), and	35
2.15(b)(ii) ,(iii)	Ellipsoids outside of target boundaries due to aiming errors. Reproduced with permission from Elsevier [65]	35
2.16	Uncertainties in solar zenith and azimuth angles. Reproduced with permission from Elsevier [78]	39
2.17	Distribution of errors in solar vector (i.e. solar vector: angular position of the sun). Reproduced with permission from Elsevier [81]	40
3.1	Azimuth and elevation angles	43
3.2	Program GUI for manual control of motors	46
3.3	Layout of the dummy solar tracker	47
3.4	Rotating gears for the azimuth axis	47
3.5	Rotating gears for the elevation axis	48
3.6	Protractor used to read the elevation angle of the dummy solar tracker	48
3.7	Shadow indicator on dummy solar tracker	49
3.8	Flow chart of dummy solar panel	50

3.9	Flow chart of simulation	51
3.10	Program GUI for the initialization of the tracker	53
3.11	Example of initializing the tracker	55
3.12	Program GUI for the run-time operation of the tracker	56
3.13	An example of tracker operation during run-time	63
4.1	Graph of azimuth angles-latitude angles	67
4.2	Graph of elevation angles-latitude angles	67
4.3	Graph of elevation angles-time of day	69
4.4	Graph of azimuth angles-time of day	70
4.5	Example of shadow from shadow indicator without active tracking	73
4.6	Example of shadow from shadow indicator during active tracking	74

LIST OF SYMBOLS

δ - The declination of the Earth

ϕ - Latitude

N - The day number, i.e. counted from January first

H - The hour angle

α - Elevation angle

θ - Azimuth angle

X - Deviation angle

λ - Longitude

t_s - Solar time

t_c - Civil time

ΔT - Difference between civil and solar time

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Program to link with Advantech DAQ driver	84
B	Program for dual axis solar tracking operation	90
C	Program for initialization of the solar tracker	101

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

In this technologically based world, electric power plays the sole role of providing energy for the operations of machines, either commercial based or industrial based. However, the generation of electricity often leads to environmental destruction. It is related with aspects as deforestation control, protection of ozone layer, reduction of CO_2 emissions and others. The environmental issues are now being regarded as the most crucial problem. Therefore, the goal of obtaining good energy supplies of electric power is to be emphasized. According to that goal and considering additionally the exhaustion of the energy reserves and the global heating of the planet to promote and improve systems sourced with renewable energies is a must. A good energy source prospect for industrial continuous processes needs to be: more or less constant energy throughout the year; highly reliable and needs little maintenance; low cost to build and operate; virtually no environmental impact; modular and thus flexible in terms of size and applications; landscape friendly. A solar electric system-also known as photovoltaic (PV) systems—fulfils this entire characteristic.

1.2 DESIGN OBJECTIVE

The objectives of this project are to prove the azimuth-elevation concept in dual axis solar tracker and to develop a simulation of the dual axis solar tracker using Borland C++ Builder.

The aim of this dual axis solar tracker study is to prove that the accuracy and consistency of the sun tracking is much more efficient compared to other solar tracking systems such as single axis tracker.

The accuracy of the dual axis solar tracker will be proved by using a dummy solar panel. The dummy solar panel will be controlled according to the azimuth-elevation angles calculated.

1.3 SCOPE OF PROJECT

In this project, the dual axis solar tracker is based on an opened-loop system where the tracker operates according to the sun's geometry and does not have any feedback from the output to the input. The main goal is to prove the operation and efficiency of the solar tracker which operates according to mathematics calculation on the sun geometry.

The scope of this project is:

I. Construct a dummy panel to prove the perpendicularity between the sunlight and the panel.
II. Develop an simulation program by using Borland C++ Builder software.

The limitation of this project is the dummy panel will not be interfaced with the software which controls the rotation and elevation of the solar tracker. The significance of the dummy panel is merely to prove the perpendicularity of the solar tracker with the sunlight. It does not prioritize on design or does not involve any controller. It is merely constructed to obtain the desired angle according to the mathematical equations of the azimuth-elevation angles for the dual axis solar tracker.

The software developed is merely to simulate the dual axis solar tracker's operation. It does not prioritize on design or algorithm of the software. It is not a new program and does not concern about the creativity or feasibility of the algorithm other than to simulate the dual axis solar tracker's operation.

1.4 SIGNIFICANCE OF STUDY

This project is carried out to prove the mathematics equations on the sun's geometry where the solar tracker's movement based on the sun's geometry has a high efficiency which completely eliminates the cosine effects which is the main deficiency in photovoltaic system. The solar power system, especially photovoltaic in Malaysia is still under used because of low efficiency. This is because more than 90% of the photovoltaic systems in Malaysia are standalone systems which use standalone solar panels. Some of the photovoltaic systems use single axis tracker, but still does not emit high output efficiency. The dual axis solar tracker is a powerful alternative for the photovoltaic system. It completely eliminates the cosine effect of the photovoltaic system and significantly raised the efficiency of the photovoltaic system. It is better than a single axis solar tracker since the sun geometry involves two axes, which are azimuth and elevation angles. Furthermore, dual axis solar tracking system increases the average working hours for photovoltaic system. For a normal standalone photovoltaic, the average working hours are 4-5 hours per day. With dual axis solar tracking system, the average working hours will be increased to at least 7-8 hours per day. Therefore, it is absolutely necessary to carry out this project to promote the application and utilize the dual axis solar tracker in Malaysia.

1.5 THESIS OVERVIEW

This thesis is primarily concerned with the analysis and simulation of the dual axis solar tracker. All the work done in this project is presented in 7 chapters:

Chapter 2 outlines the literature review studied in order to understand the difference between dual axis solar tracking systems compared to other photovoltaic systems such as standalone system. This is important in order to make sure that this project is worth-while to be carried out.

Chapter 3 outlines the methodology used in the implementation of the project. This chapter includes the flow of the project development and the flow of the programming used in the project. This is one of the most essential part of the project as it determines the whether the flow of the project is smooth or otherwise.

Chapter 4 outlines the results of the output for the dummy solar axis tracker. An analysis on the efficiency of the azimuth-elevation will be implemented. This is important to determine whether the objective of this project is achieved or not.

Chapter 5 summarizes the overall project design and its future development.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Human beings are faced with oil and coal depletion of fossil fuels such as a serious threat that these fossil fuels is a one-time non-renewable resources, limited reserves and a large amount of combustion of carbon dioxide, causing the Earth’s warming, deterioration of the ecological environment. With the development of society, energy saving and environmental protection has become a topical issue.

The green energy also called the regeneration energy, has gained much attention nowadays. Green energy can be recycled, much like solar energy, water power, wind power, biomass energy, terrestrial heat, temperature difference of sea, sea waves, morning and evening tides, etc [1, 2]. Among these, solar energy is the most powerful resource that can be used to generate power. A good energy source prospect for industrial continuous processes needs to be:

- More or less constant energy throughout the year;

- Highly reliable and needs little maintenance;
- Low cost to build and operate;
- Virtually no environmental impact;
- Modular and thus flexible in terms of size and applications;
- Landscape friendly.

2.2 PHOTOVOLTAIC SYSTEMS

Solar electric systems—also known as photovoltaic (PV) systems—fulfils all of this characteristics [3], so far the efficiency of generating power from solar energy is relatively low. Thus, increasing the efficiency of generating power of solar energy is very important.

As a new type of solar energy, which has unlimited reserves, widespread in the world, the use of economic advantages, so the sun energy using is in the rapid development and application. However, solar energy has some disadvantage, such as: the existence of low-density, intermittent, changing the spatial distribution, so all the defects make the current series of solar energy equipment, such as solar water heaters, solar cells, in the low utilization rate. The sun is moving all the time, no matter what kind of solar energy equipment, if it's part of the energy conversion to always keep vertical with the light, it can be used in a limited area to collect more solar energy. In order to maintain the equipment, solar energy conversion and some of the vertical sunlight, it is necessary to track the sun, and the sun automatic tracking device maybe resolve this issue. The professor of KPCheung and SCMHui form University of Hong Kong's Department of Architecture study the relation between the sun angle and receiving rate. The theoretical analysis shows that: the sun-tracking or not, energy receiving rate difference is about 37.7%[4, 5, 6, 7, 8].

In the past, solar cells have been hooked with fixed elevating angles. They do not track the sun and therefore, the efficiency of power generation is low. For example, the elevating angle of a solar cell for the largest volume of illumination in daytime is 23.5° in southern Taiwan. Since the fixed-type solar panel cannot obtain the optimal solar energy, the transformation efficiency of solar energy is limited. In order to have an approximately constant energy production throughout the day, it is necessary that the photovoltaic panels changes its orientation throughout the day following the path of the sun in the sky, this is possible by means of an automatic solar tracker system. A solar tracker improves the efficiency of solar electric o thermal energy conversion system [9]. Many scholars have proposed different methods for tracking the sun [10, 11, 12, 13, 14, 15]. Many different light source sensors, light intensity sensors, intelligent vision techniques, and CCD equipments were applied to compute the absorbed time of the sun radiation in everyday for measuring the volume of solar energy. So far the majority of solar cell panels worldwide are hooked with fixed angles. Thus, it is clear that the method of tracking the sun is a technique worthy of being developed.

Most PV installations in Malaysia so far are standalone systems. Grid-connected PV started with some test installations recently. The “Prototype Solar House” located near Kuala Lumpur was built in 2002 as part of an Industry Research and development Grant Scheme (IGS) project. It is unique, since in a typical residential house 3 different grid-connected roof installation with different PV generators (polycrystalline, monocrystalline, amorphous silicon; standard mounting, integration, roof tiles) have been built with emphasis also on architecture and thermal aspects. An innovative monitoring system allows assessing all relevant operating parameters for the PV systems. The installations run very reliable with high efficiency, but some grid specific aspects have been found relevant for proper functioning of grid-connected PV systems in

Malaysia. Until now, most applications of PV technology in Malaysia are concentrating on stand-alone systems. Because Malaysia is located just north of the Equator, there is a high annual irradiation and simulation calculations lead to an expected AC energy output for ideally orientated grid-connected PV systems of around 1200 ... 1300 kWh/kWp [16].

The high one-axis and two-axis tracker penetration in large scale PV power plants is due to the fact that sun tracking systems substantially improve the daily energy production of flat plate modules. Indeed, the energy gain is nearly 28 % in middle Europe [17] and even more than 50 % in north Europe [18]. Therefore, if keeping the mechanical costs acceptable, solar trackers represent an interesting possibility to increase the energy yield of conventional solar power plants without increasing the installed power capacity. As it is shown in Figure 1, one or two axis solar trackers already represent 27 % of the total power plant capacity worldwide in 2008 [19]. This part is increasing constantly since the additional costs for the mechanics are still inexpensive compared to PV.

The solar cell is composed of the semiconductors of the P-N junctions [20-21]. It can convert light into electric energy. Therefore we can assume that electricity produced using sunlight shining on the solar cell can be used like common electricity.

Solar trackers main classification is: passive and active. Passive trackers use a low boiling point compressed gas fluid heated by solar energy that is driven to one side or the other by differential gas pressure which causes the PV array to rotate in response to its imbalance. Active trackers use electric actuators (motors and gear trains) to move the PV array commanded by a controller responding to the solar orientation. Solar trackers may be single axis (passive or active) or dual axis (only active). Single axis trackers usually have a manual elevation (axis tilt) adjustment on a second axis which is tuning throughout the year; both are shown in Figure 2.1 and 2.2 [22].

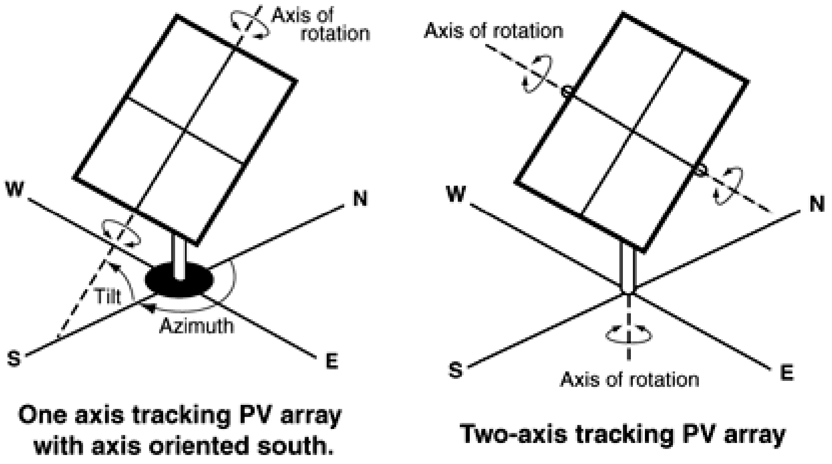

Axis of rotation
Tilt
Azimuth
W
N
S
E
One axis tracking PV array with axis oriented south.
Axis of rotation
W
N
S
E
Axis of rotation
Two-axis tracking PV array

Figure 2.1: One and two axis tracking PV array

The design of tracker was based on the following criterions:

- Low cost;
- Easy maintenance;
- Modular;
- Fulfil technical specifications;
- Easy adjustment in case of different location.

Technical specifications were obtained considering similar commercial equipment [23] and that the installation of the PV array experimental facilities is in Cuernavaca, Mor., they are:

- Automatic interval of daily sun tracking rotation angle of 130º;
- Interval of year tilting angle of 47º (Figure 2);
- Secure operation and maintain position in winds of until of 120 km/h;

- Difference between orthogonal sun position an PV array orientation lower than 10 °
  for at least 8 labour hours every day;
- Energy consumption lower than 5% of the PV array generation.

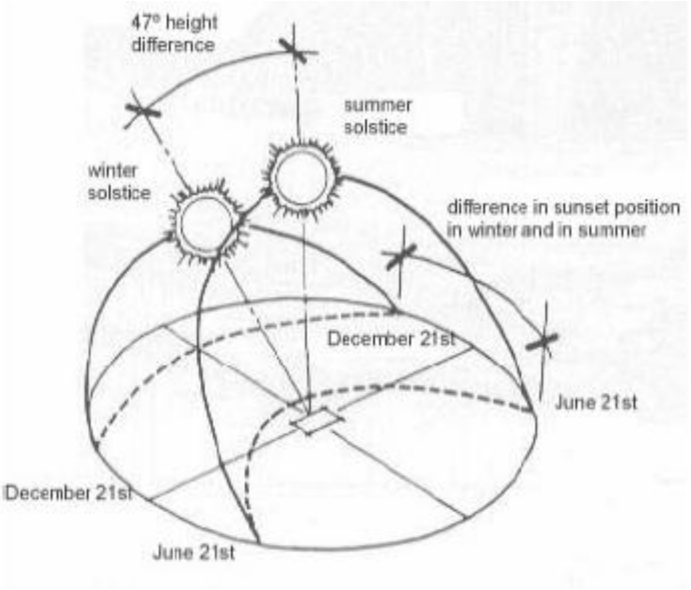


Figure 2.2: Interval of sun's altitude in a year

The technical solution for converting the solar energy in electricity is well known: the photovoltaic (PV) systems. The efficiency of the PV system depends on the degree of use and conversion of the solar radiation. The energy balance refers to the surface that absorbs the incoming radiation and to the balance between energy inflow and energy outflow. The rate of useful energy leaving the absorber is given by the difference between the rate of incident radiation on absorber and the rate of energy loss from the absorber [24]. There are two ways for maximizing the rate of useful energy: optimizing the conversion to the absorber level by properly choosing the absorber materials, and increasing the incident radiation rate by using tracking systems [25]. Having in view the operating principle, there are two fundamentals types of tracking systems: passive and active trackers. The passive trackers are based on thermal expansion of a Freon-based liquid from one edge of the tracker to another because of the heat sensitive working fluid [26]. The active trackers are based on electrically operated positioning drives, which need motors, gearboxes, mechanisms, couplings etc. Usually, the nowadays active tracking systems are based on planar or spatial linkages, and gear mechanisms. Basically, the active tracking systems are mechatronic systems, which integrate mechanics, electronics and information technology. The orientation of the photovoltaic panels may increase the efficiency of the conversion system from 20% up to 50% [27, 28, 29, 30]. The photovoltaic system with tracking is efficient if the following condition is achieved:

$$\varepsilon = (EPT - EPF) - EC >> 0,$$

where EPT is the electric energy produced by the photovoltaic panel with tracking, EPF - the energy produced by the same panel without tracking (fixed), and EC - the energy consumption for orienting the panel.

In general, the power developed in solar energy systems depends fundamentally upon the amount of solar energy captured by the collector, and thus the problem of developing tracking schemes capable of following the trajectory of the sun throughout the course of the day on a year-round basis has received significant coverage in the literature. For example, various schemes have been proposed for optimizing the tilt angle and orientation of solar collectors designed for different geographical latitudes or possible utilization periods [7-8]. In general, the results showed that by using mathematical models to optimize the tilt angle and orientation of the solar collector, a yearly gain of more than 5% could be obtained in the captured solar radiation compared to the case in which the collector was fixed on a horizontal surface. In Aden city (Yemen), the improvement in the performance of a solar cooker during summer was found to be as much as 40% for higher elevation angle and 70% for lower elevation angle, based on the developed tracking algorithms in [5]. Moreover, it was shown in [9] that the amount of solar energy captured by a tilted collector could be increased by more than 40% by adjusting the tilt angle on a seasonal basis.

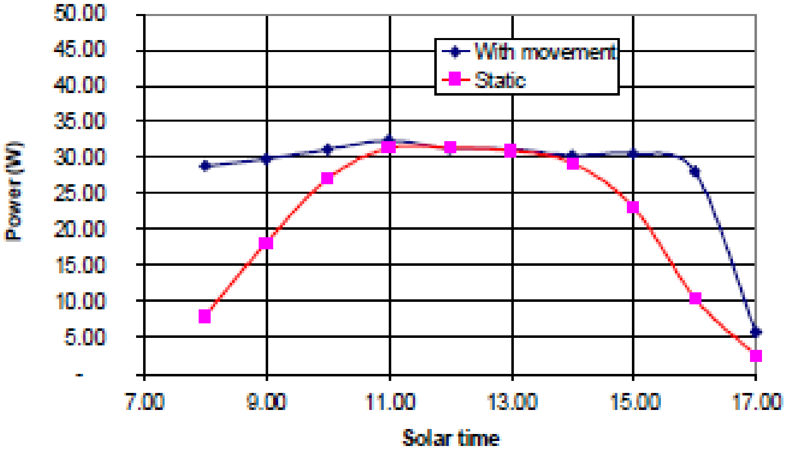


Figure 2.3: Example power with PV static and moving

In Figure 2.3 is shown the example of energy generated with PV static and with tracker movement, the increase of energy is the high difference between both curves. The result was with the pattern and magnitude that was expected.

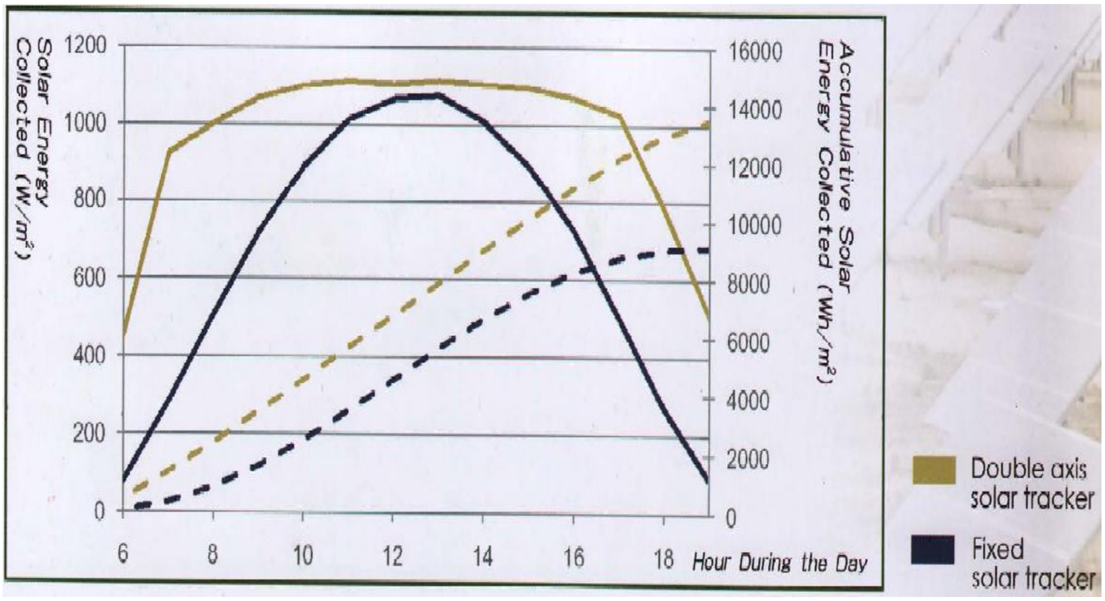


Figure 2.4: Comparison Diagram of Dual-Axis tracker & Fixed-Angle installation [31]

Table 2.1: Comparison of electricity production and cost between different tracking systems.

Installation	Fixed	Single-Axis	Dual-Axis
Demo			
Electricity Production	100%	Increases 20% - 30% relative to fixed installation	Increases 30% - 40% relative to fixed installation
Cost	100%	Only 10% - 15% higher than fixed installation	Only 10% - 15% higher than fixed installation

From the above table, it is clear that the benefits of a dual axis solar tracker is significantly enormous compared to both fixed solar panels and single axis solar tracker. Therefore, the dual axis solar tracker is chosen to be the subject to be studied in this project.

2.3 TIME

In order to obtain the time for the operating hours, there are many factors to be taken into account. The facts and information about time is studied.

The period called the day is the time for the Earth to rotate once upon its axis with respect to the Sun. It has arbitrarily been divided into twenty-four hours. The twenty-four hour division has led to a very practical geographical division of the Earth into twenty-four zones, east and west, called time zones. The rotating Earth causes the sun to appear to move westward, covering one time zone each hour. North-south lines, called standard meridians, are described as passing centrally through each time zone. The standard meridians do not form the boundaries of the zones, but the boundaries lay half-way between these meridians. For political convenience, the boundaries of the time zones are often distorted east or west to include neighboring communities within the same time zone.

During the period of British maritime dominance, the zero, or prime meridian was designated as passing through the observatory at Greenwich, England. The standard meridians then are numbered east and west from there to the one on the opposite side of the Earth, called the International Date Line.

The system is fundamentally sound, for it permits travel throughout the world under standardized time conditions. All clocks within a time zone read the same under these standardized time conditions and that time is called the standard time of that zone. When one moves from one time zone to an adjacent one, the time must be changed by a full hour. If there were no time zones, as was once the case, each community would set its own time based on the passage of the Sun at that particular location. This, of course, led to a great deal of confusion until time zones were agreed upon. The need for such standardization occurred with the development of the railroads which quickly spanned the continents

2.3.1 SUNDIAL

A sundial is a simple instrument for reading time during daylight hours from the position of the Sun. However, comparison of sundial time, also called apparent time, with standard time usually shows a considerable difference, and the difference increases and decreases during the year. The rate of rotation of the Earth is remarkably uniform, varying only a tiny fraction of a second per year, but there are other factors in the Earth's motion that influence sundial time. When placed on a standard meridian, the sundial will agree with a standard time clock on only four days of the year, running alternately ahead of and behind the standard clock. Since the standard clock is assumed to run at a uniform rate, the conclusion must be that the Sun does not move across the sky at the same rate throughout the year.

The standard clock keeps the time of an imaginary Sun, called the mean Sun, which is assumed to move uniformly throughout the year. Standard time is sometimes called mean time, referring to the mean Sun. The difference between the standard clock and the sundial is the same from year to year on the same date. The difference is tabulated as the equation of time. The equation of time is usually defined (and will be so used in this article) as:

EQUATION OF TIME = LOCAL APPARENT SOLAR TIME (sundial time) minus LOCAL MEAN SOLAR TIME (standard time).

Thus, if the sundial is ahead of the clock (reading later), the equation of time is positive, and if behind (reading earlier), it is negative. The equation of time is often displayed graphically on terrestrial globes as the narrow figure-8 curve, called the analemma.

2.3.2 ELLIPTICAL ORBIT

The reasons for this yearly variation in the apparent motion of the Sun are twofold. The first reason has to do with the fact that the Earth's orbit is not a perfect circle, but is elliptical with the Sun being nearer one end of the ellipse. The speed of the Earth in this elliptical orbit varies from a minimum at the farthest distance to a maximum at the closest distance of the Earth to the Sun. The second reason for the yearly variation has to do with the fact that the Earth's equator is inclined to the plane of the Earth's orbit around the Sun. These two effects are explained in the following paragraphs.

1. **Elliptical Orbit**. While the Earth is rotating upon its axis, it is also moving around the Sun in the same sense, or direction, as its rotation. If we select a spot on the Earth where the Sun is directly overhead, in order for that spot to rotate with the Earth and come back so that the Sun is overhead again, it must turn a little extra because of the Earth's motion around the Sun. The Earth turns a little more than once with respect to the stars in order to complete one rotation with respect to the Sun. The "little extra" is just the angle through which the Earth has moved around the Sun in a day's time. On the average, this angle amounts to a little less than one degree per day (360 degrees/ 365 ¼ days) and is illustrated in Figure 2.5.

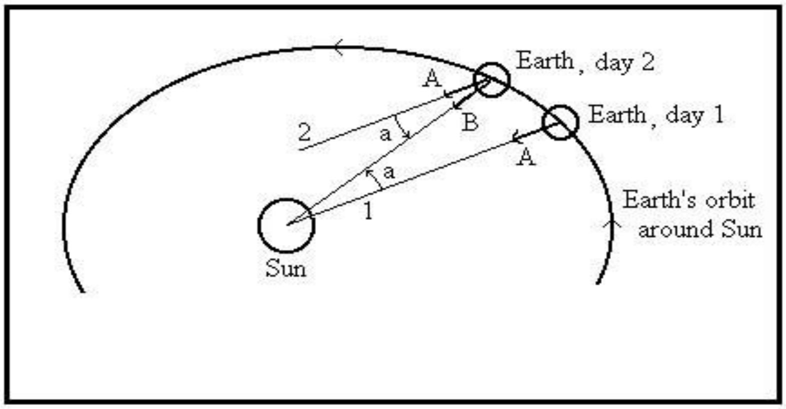


Figure 2.5: The Earth must rotate 360 degrees plus a very small angle, for observer at A to return to the same position relative to the Sun at B.

The time for the Earth to turn this small angle is about four minutes. This little difference would cause no concern if it was always the same, but it is not! Recalling that the Earth moves in an elliptical path around the Sun, rather than a circular path, it turns out that the Earth is nearer to the Sun in January than in July. The difference is about three million miles (out of an average distance of ninety-three million miles). The speed of the Earth in its orbit increases as it gets nearer to the Sun. Since the Earth is closest to the Sun in January and furthest in July, it follows that the Earth is moving more rapidly in its orbit in January than in July! Thus, the Earth must rotate a little more each day from October to April to return to a chosen spot to face the Sun again. This small amount each day accumulates until it amounts to a difference of 7.7 minutes on April.

2. Having to turn a little more each day means the sundial lags behind the standard clock and so the sundial time minus standard time on April 2 is -7.7 minutes. From April 2 on, the Earth rotates a little less each day to return to a chosen spot to face the Sun again, and this decrease accumulates from April to October until it amounts to a difference of +7.7 minutes on October 2. The difference between sundial time and clock time

resulting from the varying speed of the Earth in its orbit is graphically illustrated in Figure 2.6.

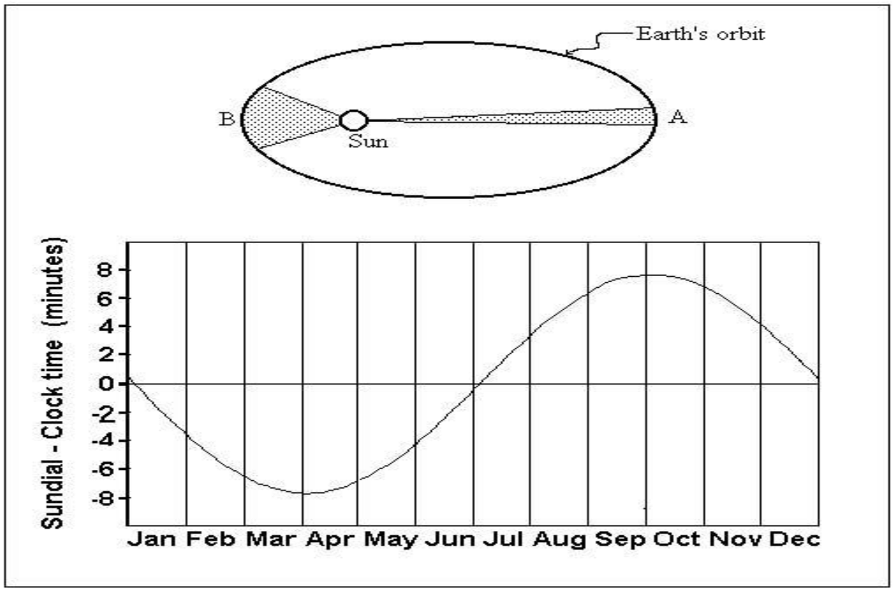


Figure 2.6: (top) The Earth moves slowest at A and fastest at B.
(bottom) Equation of Time component due to the eccentricity of the Earth's orbit.

2.3.3 INCLINATION OF THE ECLIPTIC

The second reason for the yearly variation of the Equation of Time has to do with the fact that the Earth's equator is inclined to the plane of the Earth's orbit around the Sun.

2. **Inclination of the Ecliptic**. Another element enters the scene, causing the sundial to vary from the clock. This effect is purely a geometrical one. The axis of rotation of the

Earth is not perpendicular to the plane of its orbit around the Sun, but is tilted by an angle of 23½ ^{o}. So, as the Earth revolves around the Sun, the North Pole is tilted 23½ ^{o} toward the Sun on June 21, and 23½ ^{o} away from the Sun on December 21, as illustrated in Fig. 12. These are the dates of the summer and winter solstices as recognized in the northern hemisphere. The result, as seen from the northern hemisphere, is that the Sun crosses the sky at noon much higher in June than in December, and if one were to plot the path of the Sun during the year, as seen against the background of the stars, it would appear as a line crossing over the celestial equator on March 21 and September 21 the vernal and autumnal equinoxes. The annual apparent path of the Sun against the background of the stars, called the ecliptic, is shown in Fig.2.7, along with the celestial equator. The celestial equator is an imaginary line in the sky directly above the Earth's equator. Thus, when the Sun is on the celestial equator, it stands directly above the Earth's equator we see that the path extends north and south > of the equator by 23½ ^{o}.

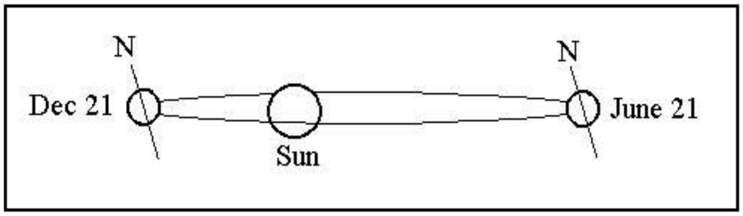


Figure 2.7: Inclination of the Earth's axis with respect to the plane of its orbit.

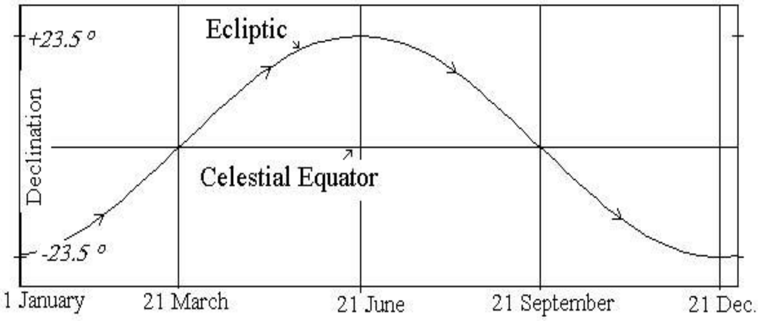


Figure 2.8: The annual apparent path of the Sun, the ecliptic, reaches 23½ ^{o} north and south of the celestial equator.

Ignoring the change of speed of the Earth in its elliptical orbit (effect number one above), the true eastward motion of the Sun is greatest when all of its motion is due eastward. This occurs in June and December. In March and September, part of the Sun's motion is northward or southward, and the eastward part of its motion is reduced. This makes the sundial fall behind the standard clock at the solstices and move ahead of the clock at the equinoxes. Fig. 2.9 illustrates this geometrical effect upon the equation of time.

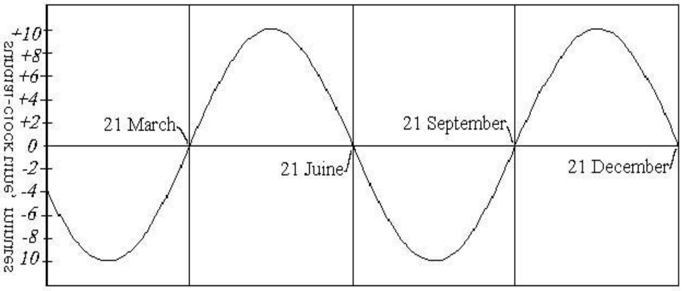


Figure 2.9: Equation of Time component due to the obliquity of the ecliptic. (obliquity = 23½ ^{o}).

2.3.4 EQUATION OF TIME

In the two previous sections, we have seen how the difference between sundial time and standard time depends on two effects: the eccentricity of the Earth's orbit and the inclination of the Earth's orbit. The combination of these two effects, which is the true equation of time, is plotted in Fig.15. In December and January these two effects are both working to slow the sundial time, while in June and July the two effects are opposed to each other. The sundial lags only six minutes during June when the two effects are opposed, but lags 13½ minutes during December. The equation of time expresses the relationship between the sundial and standard time, and the standard time is then available from the sundial by applying the proper value, plus or minus, from the equation of time. But such conversion yields true standard time only if the sundial is on the standard meridian. One must know one's distance east or west of the standard meridian in order to make the remaining correction to the sundial time.

The Earth turns through one time zone in an hour. The time zone is 15 degrees wide (one twenty-fourth of 360 degrees), so each degree of longitude within the time zone is equivalent to four minutes of time (60 min./15^o). This then is the correction to make for each degree of longitude away from the standard meridian: minus if east or plus if west of the standard meridian

As an example, suppose that you are located at longitude 155 degrees west. What is the correction to arrive at standard time for your time zone? The standard meridian is the 150 degree west meridian, so you are located 5 degrees west of that. Every degree is 4 minutes of time, so the sun passes overhead at your longitude 4 x 5 = 20 minutes later than at the standard meridian. Thus, you must add 20 minutes from your sundial time to get the standard zone time. This, of course, is in addition to the time that must be added or subtracted according to the equation of time. See Appendix B for additional examples.

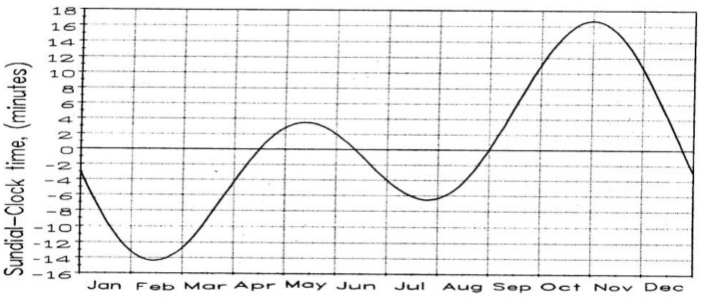


Figure 2.10: The Equation of Time

2.3.5 SUNRISE AND SUNSET

When the clock is gaining on the sundial, the Sun rises and sets later each day, and when the sundial is gaining on the clock, the Sun rises and sets earlier each day. If the two effects which give us the equation of time were solely responsible for sunrise and sunset times, these times would be late in summer and winter and early in spring and fall. Most of us would say at once that, of course this is not true. But it is true for anyone living on the equator!

On a standard meridian at the equator one might expect the Sun to rise at 6:00 A.M. and set at 6:00 P.M., but the Sun rises at 6:03 A.M. in July, a summer month, and also rises late, at 6:11 A.M. in February, a winter month. It rises seven minutes before 6:00 A.M. in mid-May, and 20 minutes before 6:00 A.M. at the end of October. At the equator these effects are entirely accounted for by the equation of time.

The daily path of the Sun as seen at the equator on the first day of spring, summer, fall, and winter is illustrated in Figures 2.11 and 2.11a. At the equator the Sun rises perpendicularly from the horizon and sets perpendicularly, regardless of the season. Also, the total path of the Sun, day and night, is divided equally by the horizon. There are always twelve hours of daytime and twelve hours of night-time at the equator, except for two minor effects that increase daytime by about eight minutes. First, since we mark the instant of sunrise as the time the Sun's upper edge or "limb" just touches the horizon, the actual center of the Sun is still below the horizon by half the diameter of the Sun, 16 arc minutes or ¼ degree. It will take an additional minute for the Sun's center to be on the horizon. At sunset the same thing happens and so an additional two minutes are gained for daytime. Second, when the Sun's limb appears at the horizon, it is actually still 43 arc minutes below the horizon but only appears to be at the horizon due to the refraction or bending of the Sun's rays by the Earth's atmosphere. This effect causes the sunrise to appear about three minutes early and sunset late by the same amount. Taking both effects together, the length of daytime is about 8 minutes more than 12 hours, and so, of course, night-time will be 8 minutes less than 12 hours, resulting in daytime being 16 minutes more than night-time at the equator, or for that matter, anywhere during the equinoxes (March 21 and September 21).

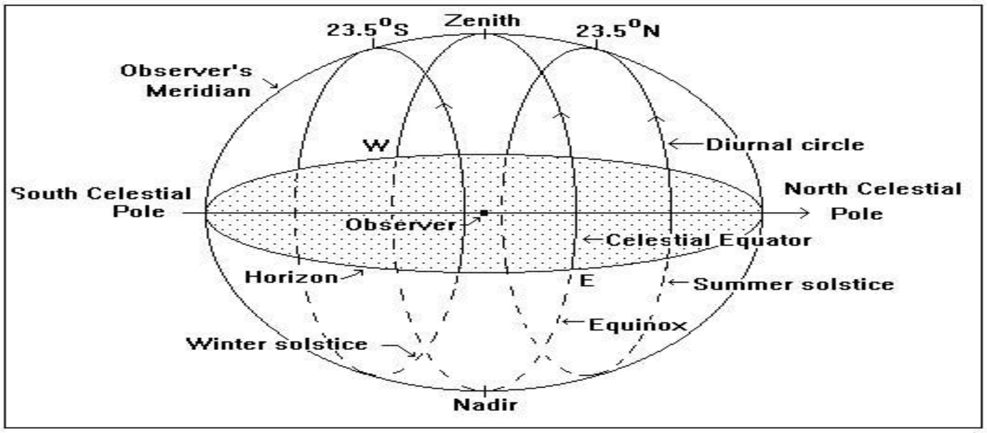


Figure 2.11: The daily or diurnal paths of the Sun during the solstices (21 December and 21 June) and the equinoxes (21 March and 21 September) as seen by an observer at the equator. Solid lines are daytime, dashed lines are night-time. At all seasons on the equator, the daily paths of the Sun are divided equally above and below the horizon.

The same information shown in Figure 2.11 is presented in Figure 2.11a, below, in the form of a polar plot of the position of the Sun in the coordinates of the azimuth and altitude of the Sun as seen by an observer at that latitude.

2.3.6 EXERCISES

The Equation of Time (ET) allows one to calculate the Standard Time at which the Sun will be found in a particular part of the sky. For example, if it is asked at what time the Sun rises in Hilo, Hawaii, on March 21, we must know the ET on March 21 and the longitude of Hilo. The ET can be read from the graph and is approximately -8 min. The longitude of Hilo is 155o W, corresponding to the example under Equation of Time. First, let us determine the time of sunrise at the Standard Meridian of 150° W, and then make the correction for Hilo's longitude. The apparent time at sunrise must be 4 minutes before 6 a.m. (The explanation of the 4 minutes is found under Sunrise and Sunset). From the definition of ET = Apparent time - Mean (or Standard) time, we see that the Standard time in this case is the App. time - ET, or, 5h 56min - (-8 min) = 6h 4min. This is now corrected for Hilo's longitude by adding 20 minutes, giving sunrise as 6h 24min or 6:24 a.m., HST (Hawaii Standard Time).

Another example: at what time will the Sun be on the local meridian to an observer in Boston on October 31? Remember that the local meridian is the great circle from north to south passing through the zenith. The ET on October 31 is approximately +16 minutes, and Boston is situated at longitude 71° W. The Apparent time when the Sun is on the meridian is 12 noon. Thus, Standard time = 12h 00min - (+16 min) = 11h 44min. This is the correct time at the Standard Meridian of 75°W, but Boston is 4° east of this, so the Sun crosses the Boston meridian 16 minutes before the 75th meridian. Hence, the answer is that the Sun crosses the meridian of Boston on October 31 at 11h 28min or 11:28 a.m., EST (Eastern Standard Time).

An interesting problem for the Hawaii student is to determine on what day and at what time the Sun passes directly overhead, i.e., through the local zenith. (Note that Hawaii is the only place in the United States where this can happen!) This requires the use of a table of the declination of the Sun for every day of the year and for the particular year in question, and such a table can be found in the American Ephemeris and Nautical Almanac (see REFERENCES). The declination of the Sun corresponds to latitude on the Earth, so that, when the declination of the Sun equals the latitude of the observer, the Sun will pass through the observer's zenith. Since the latitude of Hilo is 19^o 43', we need to look for the day on which the Sun has a declination of 19^o 43'. We then need to find the ET on that day and proceed to determine, as above, the HST at which that zenith crossing occurs. We like to call this a "shadow less" noon, for indeed, a flagpole has no shadow at that instant!

In the design process of the tracking systems, the solar radiation represents the main input data. Interacting with atmospheric phenomena involving reflection, scattering, and absorption of radiation, the quantity of solar energy that reaches the earth's surface is reduced in intensity. The total solar radiation received at ground level includes two main components: direct solar radiation and diffuse radiation [32]. The solar radiation can be measured using traditional instruments, or can be digitally recorded with a data acquisition system. Within an EU funded project, a solar radiation atlas was realized for Europe [33]. At the same time, there were developed large meteorological databases, such as Meteonorm - Global Meteorological Database for Engineers, Planners and Education.

2.4 TRACKING SYSTEMS

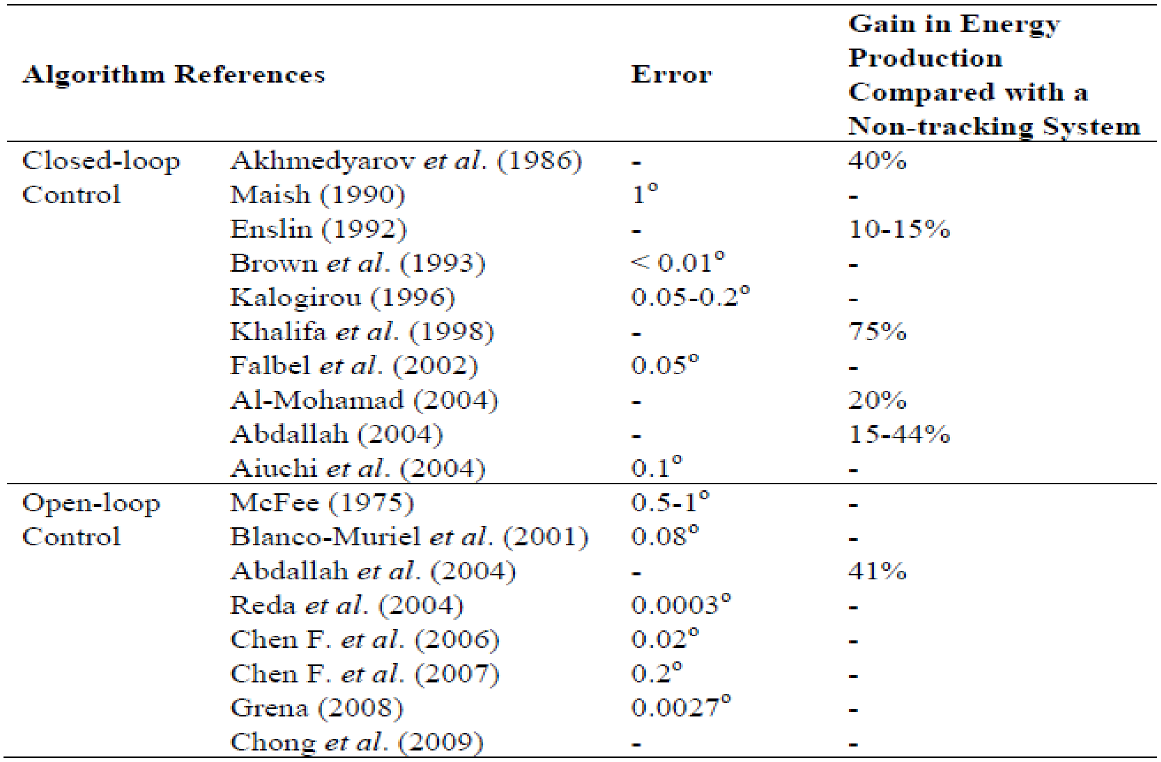
In addition, different models were developed for estimating the solar radiation. The traditional Angstrom’s linear approach is based on measurements of sunshine duration, while relatively new methods are based on artificial neural networks – ANN

[34]. In reference [35], there are studied four models for estimating the monthly mean solar radiation, including linear Angstrom-Prescot variation, quadratic equation, logarithmic variation, and exponential function; the root mean square error is the principal elements of this comparative analysis. A step by step procedure was developed for implementing an algorithm to calculate the solar irradiation, using both zenith and azimuth angles to describe sky elements position [36].

Regarding the control process of the tracking systems, in literature, closed loop systems with photo sensors are traditionally used. The photo sensors are responsible for discrimination of the sun position and for sending electrical signals, proportional with the error, to the controller, which actuates the motors to track the sun. Many authors have adopted this method as a basis in construction and design of such systems [37, 38, 39]. Although, the orientation based on the sun detecting sensors, may introduce errors in detection of real sun position for variable weather conditions.

Other solutions are the opened loop systems, which are based on mathematic algorithms/programs that may provide predefined parameters for the motors, depending on the sun positions on the sky dome. These positions can be precisely determined because they are functions of the solar angles that can be calculated for any local area [27, 40, 41]. By using this control technique, based on predefined parameters, the errors introduced by the use of the sensors may be avoided.

Table 2.2: Performance of sun tracking systems [48-85].



Algorithm References		**Error**	**Gain in Energy Production Compared with a Non-tracking System**
Closed-loop Control	Akhmedyarov *et al.* (1986)	-	40%
	Maish (1990)	1°	-
	Enslin (1992)	-	10-15%
	Brown *et al.* (1993)	< 0.01°	-
	Kalogirou (1996)	0.05-0.2°	-
	Khalifa *et al.* (1998)	-	75%
	Falbel *et al.* (2002)	0.05°	-
	Al-Mohamad (2004)	-	20%
	Abdallah (2004)	-	15-44%
	Aiuchi *et al.* (2004)	0.1°	-
Open-loop Control	McFee (1975)	0.5-1°	-
	Blanco-Muriel *et al.* (2001)	0.08°	-
	Abdallah *et al.* (2004)	-	41%
	Reda *et al.* (2004)	0.0003°	-
	Chen F. *et al.* (2006)	0.02°	-
	Chen F. *et al.* (2007)	0.2°	-
	Grena (2008)	0.0027°	-
	Chong *et al.* (2009)	-	-

In 1975, one of the first automatic solar tracking systems [42-45] was presented by McFee, in which an algorithm was developed to compute total received power and flux density distribution in a central receiver solar power system [42]. By subdividing each mirror into 484 elements and summing the contributions of all elements, the sun position could be determined with a tracking error tolerance of $0.5_{o} – 1_{o}$. Several years later, Semma and Imamru used a simple microprocessor to adaptively adjust the positions of the solar collectors in a photovoltaic concentrator such that they pointed toward the sun at all times [45]. Mathematical theories of tracking error distributions were also developed to improve the algorithms of determining sun position [46,47]. With rapid advances in the computer technology and systems control fields in recent decades, the literature now contains many sophisticated sun tracking systems designed to maximize the efficiency of solar thermal and photovoltaic systems. Broadly speaking, these systems can be classified as either closed-loop or open-loop types, depending on their mode of signal operation (Table 1).

2.4.1 Closed-loop Types of Sun Tracking Systems

Closed-loop types of sun tracking systems are based on feedback control principles. In these systems, a number of inputs are transferred to a controller from sensors which detect relevant parameters induced by the sun, manipulated in the controller and then yield outputs (i.e. sensor-based). In 1986, Akhmedyarov *et al*. [48] first increased the output power of a solar photoelectric station in Kazakhstan from 357 W to 500 W by integrating the station with an automatic sun tracking system. Several years later, Maish [49] developed a control system called SolarTrak to provide sun tracking, night and emergency storage, communication, and manual drive control functions for one and two axis solar trackers in a low-cost, user-friendly package. The control algorithm used a six-degree self alignment routine and a self-adjusting motor actuation time in order to improve both the pointing accuracy and the system reliability. The experimental results showed that the control system enabled a full-day pointing accuracy of better than $+0.1_{o}$ to be achieved. In 1992, Agarwal [50] presented a twoaxis tracking system consisting of worm gear drives and four bar-type kinematic linkages to facilitate the accurate focusing of the reflectors in a solar concentrator system. In the same year, Enslin [51] applied the principles of maximum power point tracking (MPPT) to realize a power electronic converter for transforming the output voltage of a solar panel to the required DC battery bus voltage. An MPPT system consists of two basic components: a switch mode converter and a control/tracking section. The switch mode converter is the core of the entire system and allows energy at one potential to be drawn, stored as magnetic energy in an inductor, and then released at a different potential. By setting up the switch mode section in various different topologies, either high-to-low or low-to-high voltage converters can be constructed. The goal of an MPPT system is to provide a fixed input voltage and/or current, such that the solar panel is held at the maximum power point, while allowing the output to match the battery voltage. In [52], the converter was controlled to track the maximum power point of the input source under varying input and output parameters and was shown to provide a minimum input source saving of 15% for 3-5 kWh/day systems. Brown and Stone [53] developed a tracking system for solar concentrators in which a neural network was applied to an error model in order to compensate for tracking errors. The test data showed that the resulting system was capable of reducing the tracking error to a value of less than 0.01$_o$ (0.2 mrad). Kalogirou [54] presented a one-axis sun*Sensors* tracking system utilizing three light-dependent resistors (LDRs). The first LDR detected the focus state of the collector, while the second and third LDRs were designed to establish the presence (or absence) of cloud cover and to discriminate between day and night, respectively. The output signals from the three LDRs were fed to an electronic control system which actuated a low-speed 12 - V DC motor in such a way as to rotate the collector such that it remained pointed toward the sun. In 1997, Stone and Sutherland [55] presented a multiple tracking measurement system comprising more than 100 heliostats for tracking the sun's position on an hourly basis from early morning to late evening. Hua and Shen [56] compared the solar tracking efficiencies of various MPPT algorithms and implemented a simple control method which combined a discrete time control scheme and a proportional-integral (PI) controller to track the maximum power points (MPPs) of a solar array.

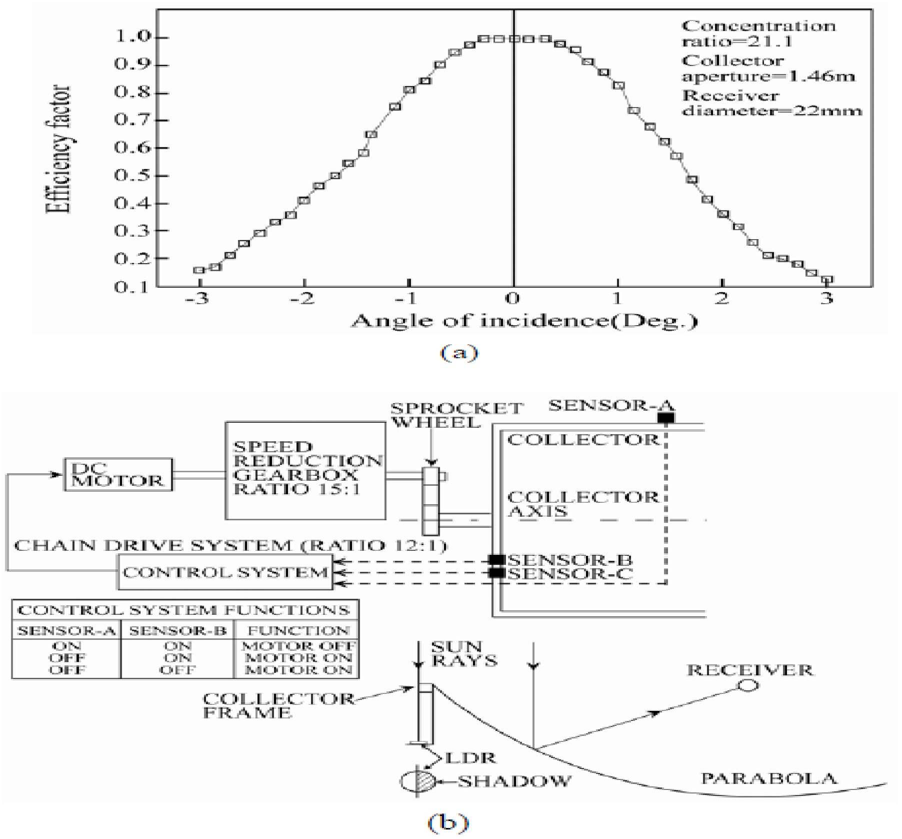


Figure 2.12: (a) Collector acceptance angle. (b) illustration of sun tracking mechanism. Reproduced with permission from Elsevier [55].

In 1998, Khalifa and Al-Mutawalli [56] developed a two-axis sun tracking system to enhance the thermal performance of a compound parabolic concentrator. The system was designed to track the sun's position every three to four minutes in the horizontal plane and every four to five minutes in the vertical plane. As shown in Figure 2.12, the tracking system was comprised of two identical sub-systems, one for each axis, with each sub-system consisting of two adjacent photo-transistors separated by a partition of a certain height. In the tracking operation, the difference in the voltage signals of the two photo-transistors was amplified and used as a command signal to drive the collector around the corresponding axis until the voltage difference reduced to zero, indicating that the sun's rays were once again normal to the collector surface. It was shown that the tracking system had a powerconsumption of just 0.5 Whr and yielded an improvement of around 75% in the collected solar energy, compared to a fixed collector of equivalent dimensions. Yousef [57] developed a sun tracking system in which the nonlinear dynamics of the tracking mechanism were controlled using a fuzzy logic control algorithm implemented on a PC and supported by an interfacing card consisting of a sensor data acquisition function, motor driving circuits, signal conditioning circuits and serial communications. Kim *et al*. [58] proposed an enhanced incremental conductance (IncCond) MPPT control algorithm for determining the maximum power operation point (MPOP) of a photovoltaic power system subject to rapidly changing levels of solar radiation. It was shown that the decision regarding the MPOP could be rendered robust to short-term fluctuations in the photocurrent by inserting a test signal in the control input. Falbel *et al.* [59] presented a sun-oriented attitude-control system combined with a concentrating solar panel for use in a satellite (CUBESAT). The solar sensor had the form of a two-axis analog device, which measured the sun’s location relative to its optical axis based on the differential signal obtained from a quadrant silicon detector upon which a circular spot generated by the sun’s irradiance was imaged. The calibration results showed that the sensor was capable of locating the position of the sun with an accuracy of +0.05$_{o}$. Urbano *et al.* [60] presented a 5 Watt-PV module for a stand-alone solar tracking system with a capacity of 2.6 kW.

The tracking system was designed to follow the position of the sun autonomously in the altitude and azimuth directions and was driven by two 12 V DC motors, each with a power consumption of 36 W and both fed by a single electrolytic condenser charged by the PV module. Jiang and Cao [61] constructed an emulated sunflower based on a spherical four-quadrant photoelectric sensor for solar tracking purposes. The sunflower was designed in such a way that when the sun’s rays were aligned with the normal direction of the detector surface, the photocurrents produced by the ray’s incident in each quadrant were equal to one another. However, any changes in the sun’s position produced a differential change in the output signals of each quadrant. It was shown that through an appropriate manipulation of the four output signals, a control signal could be produced to drive the position of the detector such that the difference between the output signals was once again restored to zero. Luque- Heredia *et al.* [62] presented a sub-degree precision sun tracker for 1,000X micro-concentrator modules. The tracking system comprised a lightweight structure designed to remain operative for high wind speeds, yielded 95% of the available direct solar radiation and featured an electronic tracking control unit which relied on an adaptive algorithm for absorbing unforeseen or time varying errors with automatic calibration of cheap sun pointing sensors against array power output.

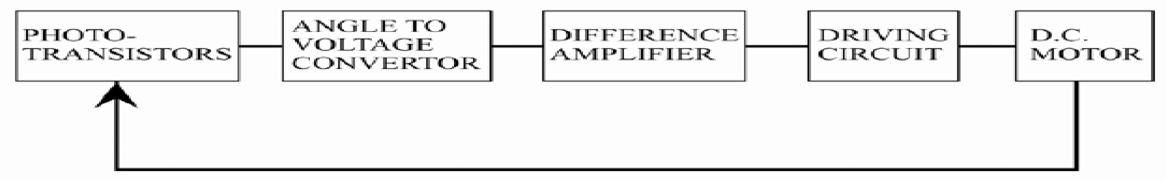


Figure 2.13: Block diagram of sun-tracking system. Reproduced with permission from Elsevier [66].

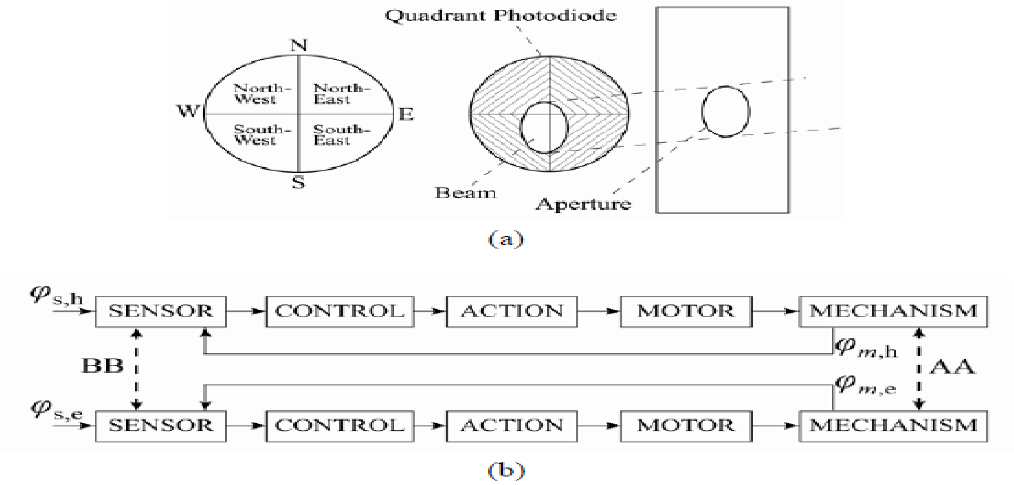
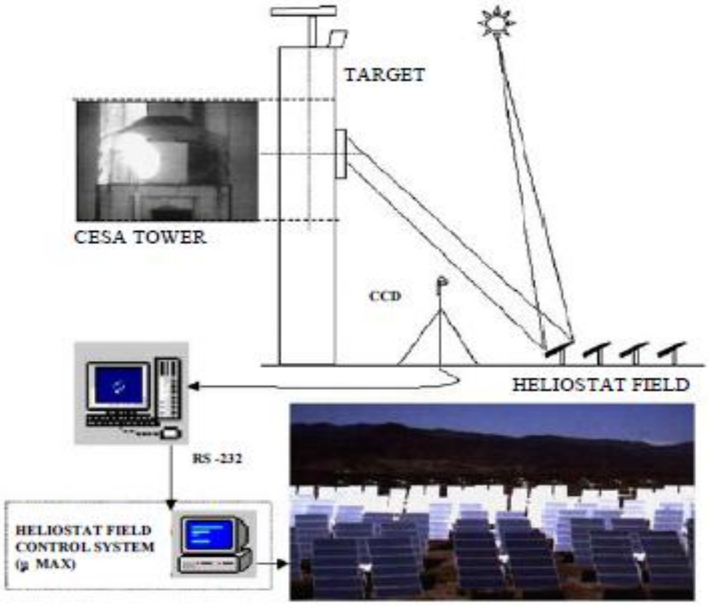


Figure 2.14: (a) four-quadrant sensor and (b) transfer functions for both axes. Reproduced with permission from Elsevier [73].

In 2004, Roth *et al.* [63, 64] designed and constructed a sun tracking system in which a pyrheliometer was used to measure the direct solar radiation. The system was controlled by a closed loop servo system consisting of a four-quadrant photodetector (Figure 2.14(a)) to sense the sun's position and two small DC motors to drive the instrument platform in such a way that the sun's image remained at the centre of the four-quadrant detector at all times. Note that the cross coupling of AA and BB in Figure 2.14(b) is virtually zero due to the orthogonal disposition of the axes and the parallel mounting of the sensor. In the same year, Berenguel *et al*. [65] developed an automatic heliostat offset correction control system based on an artificial vision technique and common charge-coupled device (CCD) equipment (Figure 2.15(a)). In the proposed approach, a B/W CCD camera captured images of the sun projected from the heliostats with a resolution of 640 x 380 or 768 x 576 (Figure 2.12(b)) and supplied the images in real time to a computer via a frame-grabber with a PCI bus. The images were then compared with reference images in which the sun's rays were incident in a normal direction to the heliostat surface. The difference between the two images was used to compute a command signal, which was then passed to the heliostat control system and used to actuate the heliostat servomotors in such a way that the heliostat surfaces were restored to an angle of 90 degrees to the sun's rays.



(a)

Figure 2.15(a): Schematic illustration of vision-based heliostat control system.

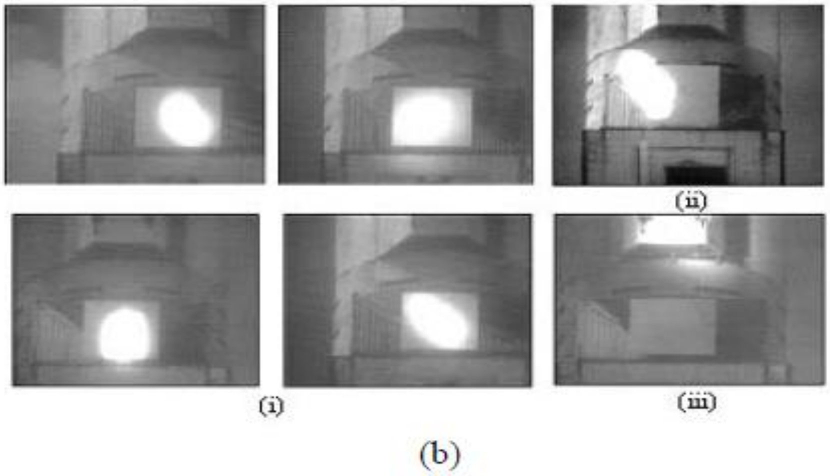


Figure 2.15(b): Photographs showing different shapes of sun images projected by heliostats onto target plane: (i) centered ellipsoids (the shape of the ellipsoid changes during the day), and (ii), (iii) ellipsoids outside of target boundaries due to aiming errors. Reproduced with permission from Elsevier [65].

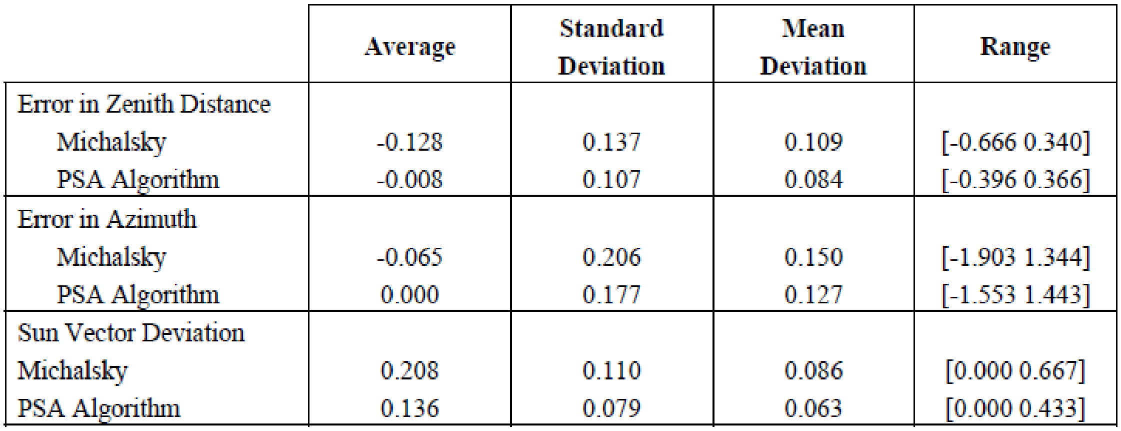
Abdallah [66] investigated the respective effects of four different electro-mechanical sun-tracking systems on the current, voltage and power characteristics of a flat-plate photovoltaic system. The results showed that tracking systems comprising two axes, one vertical axis, one east-west axis and one north-south axis, and one north-south axis, increased the electrical output powers of the photovoltaic system by around 43.87%, 37.53%, 34.43% and 15.69%, respectively, compared to that obtained from a photovoltaic system with a fixed surface inclined at 32_{o} to the north. Al-Mohamad [67] used a programmable logic controller (PLC) to control a photovoltaic module for following the sun's radiation. It was shown that by collecting and storing the data relating to the sun's radiation, and using this information to control the photovoltaic module, the daily output power of the photovoltaic system could be improved by more than 20% relative to that obtained from a fixed module. Aiuchi *et al.* [68] presented a simple sun tracking photo-sensor designed to ensure a constant direction of the reflected solar radiation. In the proposed device, two photo-cells were placed side by side at the bottom of a box with an aperture. When the reflected solar radiation passed through the aperture, the photo-cells were fractionally illuminated and produced an electric current proportional to the size of the illuminated area. A constant direction of the reflected solar radiation was maintained simply by monitoring the output signals of the two photo-cells and adjusting the angle of the reflection mirror as required ensuring that the two signals remained equal at all times. It was shown that the resulting system achieved a tracking error of less than 0.6 mrad on a sunny day. In 2005, Alata *et al.* [69] designed and simulated three sun tracking systems, namely: (1) one-axis sun tracking with the tilted aperture equal to the latitude angle, (2) equatorial two-axis sun tracking, and (3) azimuth/ elevation sun tracking. For each tracking system, the modelling and controller design tasks were accomplished using the first-order Sugeno fuzzy inference system. In addition, the insulation incident on the two-axis sun tracking system was determined in accordance with fuzzy IF- THEN rules. Having generated the input/output data, a subtractive clustering algorithm and a Least Square Estimation (LSE) scheme were applied to generate a set of fuzzy rules with which to predict the solar angles given the local time. Finally, the fuzzy rules were tuned by an Adaptive Neuro-Fuzzy Inference System (ANFIS) and implemented in an open-loop control system. In 2007, Kim [70] presented a robust MPPT system based on a sliding mode controller (SMC) for a three-phase grid-connected photovoltaic system. The proposed system comprised both a MPPT controller and a current controller (Figure 5). The MPPT controller generated a current reference directly from the solar array power information, while the current controller used an integral sliding mode scheme to ensure a tight control of the current. The proposed system demonstrated a robust tracking performance in the presence of both modelling uncertainties and parameter variations.

2.4.2 Open-loop Types of Sun Tracking Systems

An open-loop type of controller computes its input into a system using only the current state and the algorithm of the system and without using feedback to determine if its input has achieved the desired goal (i.e. algorithm-based). The system is simpler and cheaper than the closed-loop type of sun tracking systems. It does not observe the output of the processes that it is controlling. Consequently, an open-loop system cannot correct any errors so that it could make and may not compensate for disturbances in the system. Open-loop control algorithms of sun tracking systems utilize some form of solar irradiation geometry model [71]. In 1983, Al-Naima and Yaghobian [72] developed a solar tracking system featuring a two-axis equatorial mount and a microprocessor, in which the tracking operation was performed on the basis of the astronomical coordinates of the sun. The experimental results demonstrated that the proposed system yielded a significantly better tracking performance than that obtained by a conventional sensor*Sensors* controlled system. Several years later, Lorenz [73] proposed a set of design guidelines for a window glazing which rejected solar radiation during the summer, but accepted it during the winter. The design featured a purely passive control algorithm based on seasonal changes in the incident angle of the solar rays.

Blanco-Muriel *et al.* [74] argued that sun-tracking systems in which open-loop controllers are used to compute the direction of the solar vector should be both highly accurate (in order to enhance the solar concentration efficiency) and computationally straightforward (to minimize the price of the tracking system). Having reviewed existing solar vector prediction algorithms, the authors developed a new algorithm for predicting the solar vector given knowledge of the time (given as the date and the Universal Time) and the location (given as the longitude and latitude of the observer in degrees). The performance of the proposed algorithm was verified by evaluating the direction of the sun vector for 447,048 reference values of the true horizontal coordinates of the sun over the period 1999~2015. It was shown that the algorithm enabled the direction of the solar vector to be determined with an error of less than 0.5 minutes of arc. Table 2.3 compares the sun vectors generated by the proposed algorithm with those computed using the algorithm proposed by Michalsky in 1988 [75]. Overall, the results show that the estimates obtained from the proposed algorithm for the azimuth and zenith angle of the sun are approximately 15% and 22%, respectively, better than those obtained from the algorithm presented in [75]. In 2003, Beshears *et al.* [76] presented a micro controller-based sun positioning system for hybrid lighting applications in which the celestial bearing of the sun with respect to the earth was computed directly from knowledge of the local time, date, latitude, longitude and time zone information.

Table 2.3: Performance comparison of PSA and Michalsky algorithms used to predict sun's position over the period 1999-2015 [74,75].



	Average	**Standard Deviation**	**Mean Deviation**	**Range**
Error in Zenith Distance				
Michalsky	-0.128	0.137	0.109	[-0.666 0.340]
PSA Algorithm	-0.008	0.107	0.084	[-0.396 0.366]
Error in Azimuth				
Michalsky	-0.065	0.206	0.150	[-1.903 1.344]
PSA Algorithm	0.000	0.177	0.127	[-1.553 1.443]
Sun Vector Deviation				
Michalsky	0.208	0.110	0.086	[0.000 0.667]
PSA Algorithm	0.136	0.079	0.063	[0.000 0.433]

In 2004, Abdallah and Nijmeh [77] developed an electro-mechanical, two-axis tracking system in which the motion of the sun tracking surface was controlled by an open-loop control algorithm implemented using a PLC unit. The proposed system incorporated two separate tracking motors, namely one motor to rotate the sun tracking surface about the horizontal north-south axis, i.e. to adjust the slope of the surface (Figure 6(a)) and the other to rotate the sun tracking surface about the vertical axis, i.e.

to adjust the azimuth angle of the surface (Figure 6(b)). The experimental results indicated that the two-axis tracking system increased the total daily energy collection by approximately 41.34% compared with that obtained from a fixed surface tilted at 32_o towards the south. In the same year, Reda and Andreas [78] presented a simple step-by-step procedure for implementing a solar position algorithm. In the proposed algorithm, the solar zenith, azimuth and incidence angles were derived using the following main parameters: ecliptic longitude and latitude for mean Equinox of date, apparent right ascension and apparent declination, together with the following correction parameters: notation in longitude, notation in obliquity, obliquity of ecliptic and true geometric distance. The results showed that the solar zenith and azimuth angles could be calculated with uncertainties of $+0.0003_o$ (Figure 16).

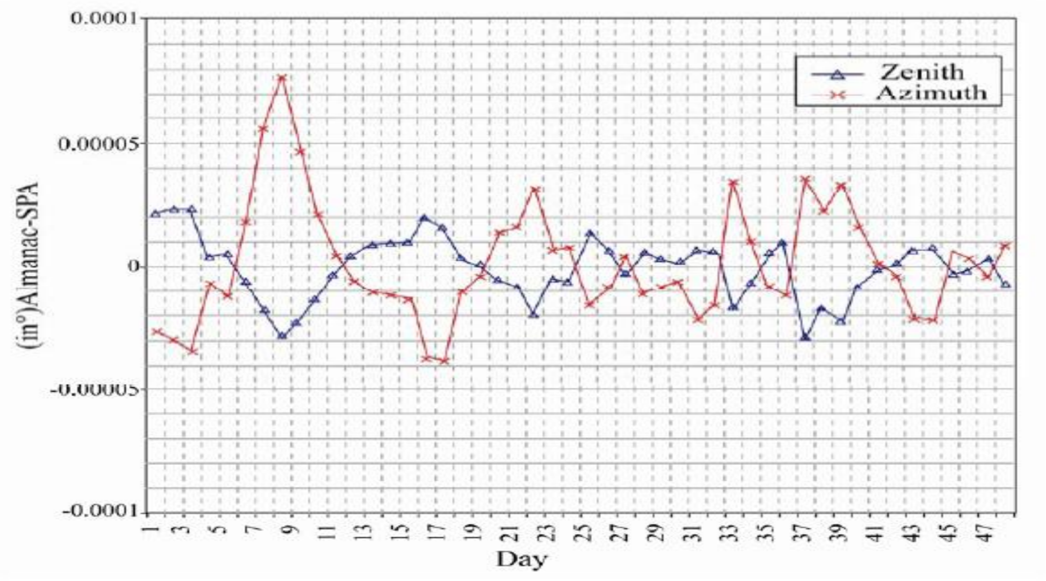


Figure 2.16. Uncertainties in solar zenith and azimuth angles. Reproduced with permission from Elsevier [78].

In a recent study, Grena [81] presented an algorithm for obtaining highly precise values of the solar position. Taking the fractional Universal Time (UT), the date, and the difference between UT and Terrestrial Time (TT) as inputs, the algorithm computed the angular position of the earth with respect to the sun in the ecliptic plane and then used this angle and the inclination angle of the earth's rotational axis to calculate the position of the sun. In the previous algorithms, the maximum error was $\pm 0.0003_{o}$ [78]. It was shown that the maximum error of the proposed algorithm, i.e. 0.0027_{o} (Figure 17), was higher than that of the algorithm presented by Reda and Andreas [78], i.e. ± 0.0003_{o}, but was sufficient for most solar engineering applications and could be obtained at a fraction of the computational cost.

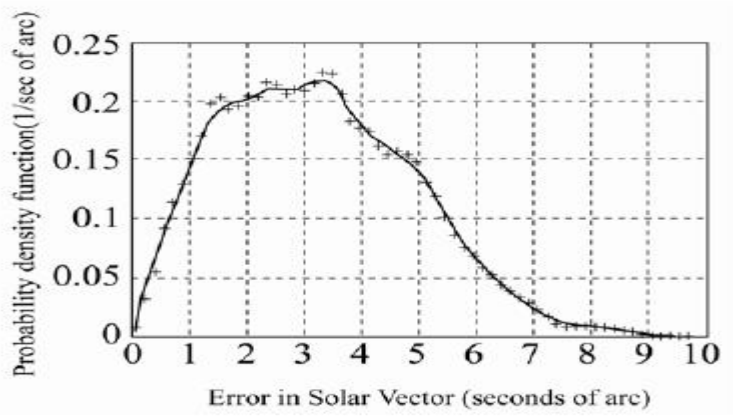


Figure 2.17: Distribution of errors in solar vector (i.e. solar vector: angular position of the sun). Reproduced with permission from Elsevier [81].

Recently, Chen *et al.* [82-83] and Chong *et al.* [84-85] presented general sun tracking formulas for open-loop type of sun tracking systems to solve the problem of any arbitrarily oriented sun-tracking axes for off-axis and on-axis solar collector respectively.

Reference [42] presents a hybrid control systems that consists of a combination of opened loop tracking strategies based on solar movement models and closed loop strategies using a dynamic feedback controller. Energy saving is also considered, which implies that the sun is not constantly tracked with the same accuracy, to prevent energy over-consumption by the motors.

From all the systems given above, opened loop system, because it has a simpler design and more cost effective compared to both closed loop system and hybrid control system which requires high technology components such as sensors and embedded microcontrollers. Despite having a simpler design, the dual axis solar tracker system has very high output efficiency where the tracker is moving according to the sun's geometry which is very precise. This can ensure the tracker will always synchronize with the movement of the sun, thus consistently tracking it.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter will cover the process involved in the development of the dual axis solar tracker study project. The processes involved are under constant changes due to unexpected changes or complications. The processes of the development of this project will be divided into 3 parts: Mathematical calculations, flow chart of dummy solar panel, and flow chart of simulation of the dual axis solar tracker.

3.2 MATHEMATICAL CALCULATION

In figure below is shown the trajectory of the Sun since dawn until the decline, that trajectory define a curve in the space which is the apparent trajectory of the sun with respect to the observer located in a specific place and date.

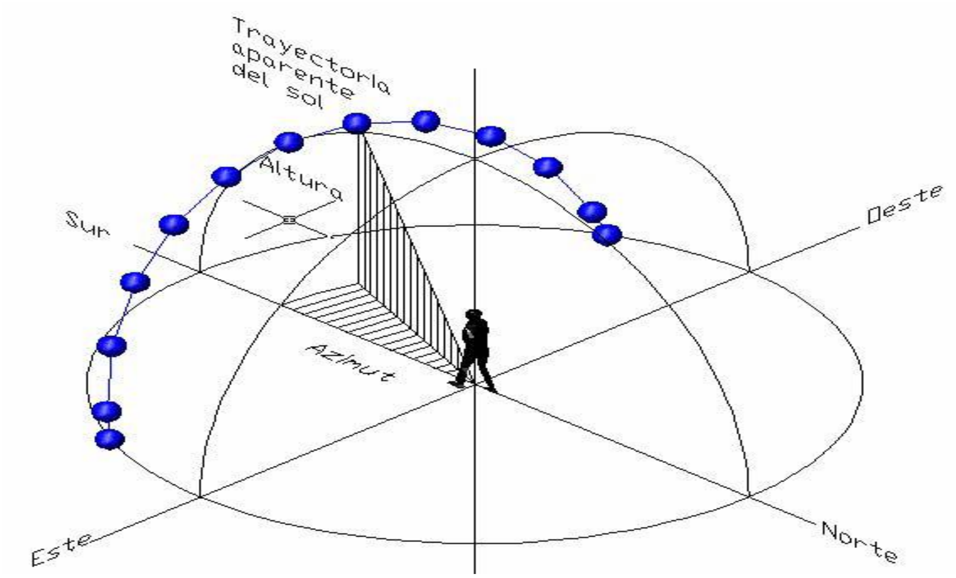


Figure 3.1: Azimuth and elevation angles

In order to know the position the sun in a date, time and place, it is necessary to know the angle of elevation or inclination of the sun h with respect to the plane and the angle of daily apparent movement of the sun or angle of azimuth. The measure of the last one is made from south toward the west (i.e., clockwise) as in surveying. With both angles, azimuth and the one of height, any point in the space is defined. As in satellite dish installation, azimuth is one of the two coordinates and the other is altitude, sometimes called elevation above the horizon. In order to determine these angles, the declination of the Earth δ with respect to the equator is first calculated. This angle is formed by the latitude of the place and a variable value product movement of translation of the Earth, and is worth:

$$\delta = 23.45 \sin \left[\left(284 + N \right) 360^{\circ}/365 \right] \tag{1}$$

Where N the day number, i.e. counted from January first.

The hour angle H is the angle between the local meridian and the hour of study, that is to say, is the angle that forms noon between paving and the hour of reference, that is obtained by:

$$H = (\textit{hour of reference} - 12)*15^{\circ} \tag{2}$$

With these data the angle of the height of the sun α is obtained:

$$\sin \alpha = \cos \phi \cos \delta \cos H + \sin \phi \sin \delta \tag{3}$$

Where ϕ is local place latitude. And the azimuth angle θ is:

$$\sin \theta = -\cos \delta \sin H / \cos \alpha \tag{4}$$

However, the declination axis of the place defines the duration of the day and the night, varying in time during the year. This basic cycle is what it is known as solar time Ts. There is a difference between the civil time, the one used by everybody and the solar time, this difference between both times ΔT can be calculated by the following equation of time:

$$\Delta T = 12 + (0.1236 \sin X - 0.0043 \cos X) + (0.1583 \sin 2X + 0.0608 \cos 2X) \tag{5}$$

Where X is the deviation angle which is calculated by the following equation:

$$X = 360^{\circ}(N - 81) / 365.2422 \tag{6}$$

The longitude λ, from a prime or international meridian, of standard time of the location is:

$$\lambda = (\textit{standard meridian of local time} - \textit{local longitude}) / 15^{\circ} \tag{7}$$

The international or prime meridian (Greenwich) is 90 º West of Mexico's central meridians, therefore the solar time t_s is given by:

$$t_s = t_c + \Delta T/60 - \lambda \tag{8}$$

Where t_c is civil time.

3.3 DUMMY SOLAR TRACKER

The dummy solar tracker is used to prove the accuracy of the solar tracking. The panel for the dummy solar tracker is a flat plastic panel, which does not produce any electrical output from the solar tracking. The control of the dummy solar tracker is by using Borland C++ Builder.

Dummy solar tracker construction:

i) Design the dummy solar tracker

The first step of this part of the project is to design the dummy solar tracker which fulfils the basic construction of an actual solar tracker. The design of the dummy solar tracker should include a shadow indicator to prove the perpendicularity of the dummy to the sunlight, and also azimuth angle indicator and elevation angle indicator for manual setup of the dummy during evaluation since the dummy is not interfaced with the controller. The core of the dummy solar tracker will be 2 power window motors motor which will be used as the rotator for both elevation and azimuth axis.

The main rotaters for both axes of the solar dummy tracker are power window motors. These motors are chosen for the high torque and the economical factor. The power window motors will be controlled using digital outputs from the DAQ card of Borland C++ Builder. There will be a program to manually control the power window motors according to the azimuth-elevation angles calculated to check whether the angles acquired are accurate or not.

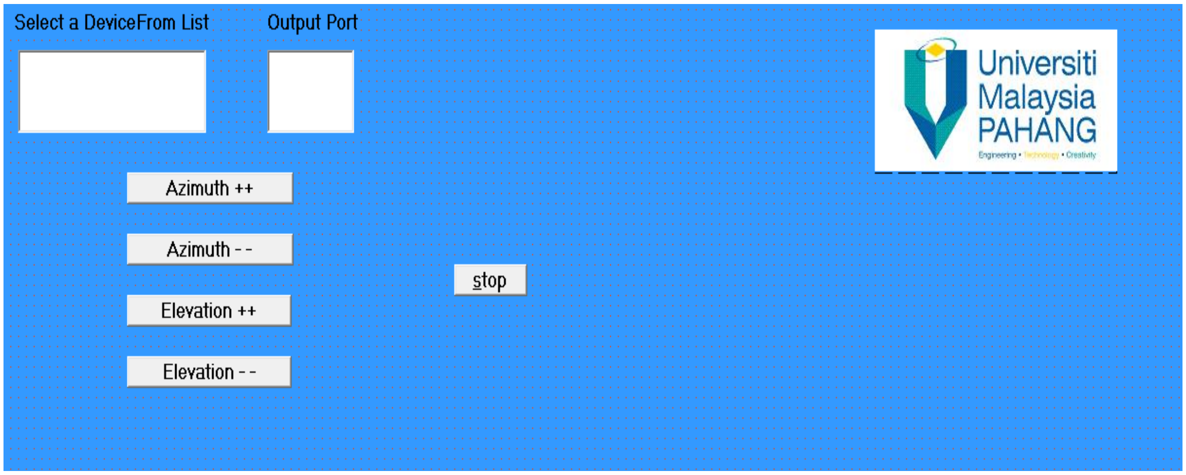


Figure 3.2: GUI for manual control of motors.

ii) Construction of the dummy solar tracker

The second step is to construct the dummy solar tracker. The dummy solar tracker will be made of a flat surface at the top of the solar tracker, supported by two rotatable axes, which are the azimuth axis and the elevation axis. A simple contractor is placed right beside the panel of the dummy solar tracker to enable user to read the elevation angle. Bicycle gears and chains are used as the rotating tools for both axes to minimize the torque and to improve the accuracy of the rotation. Figures below show some examples of the construction of the dummy solar tracker.

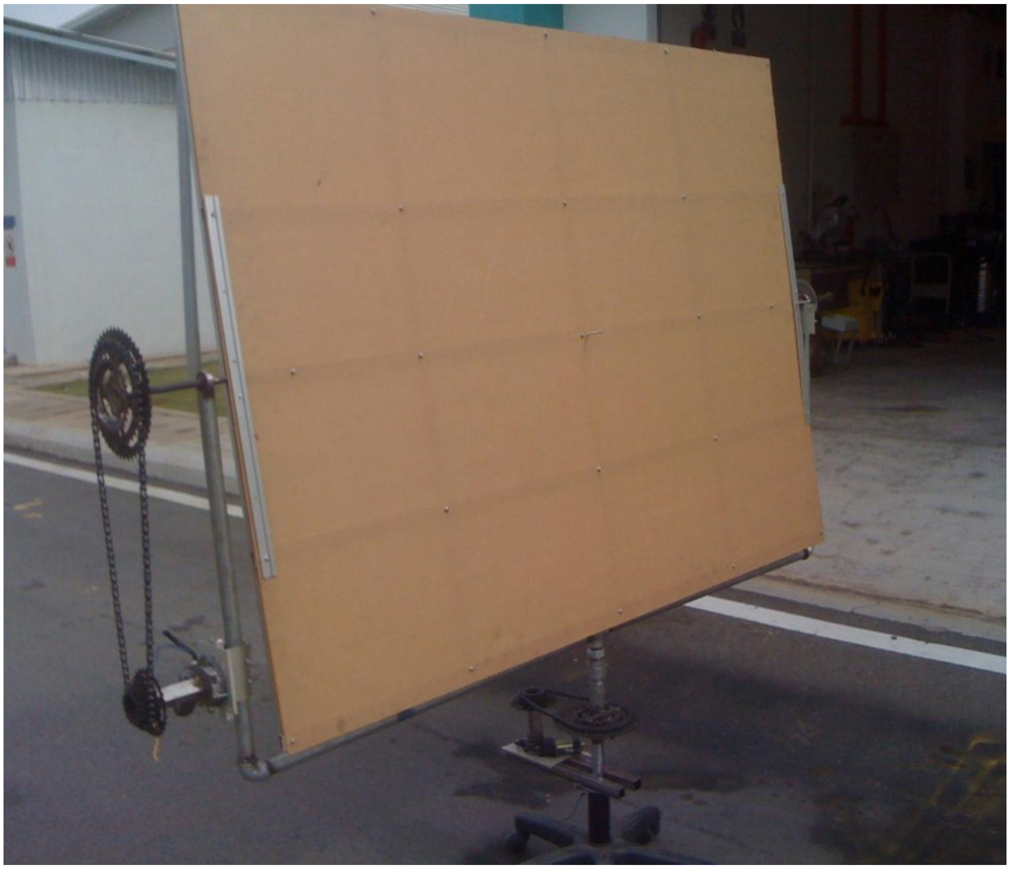


Figure 3.3: Layout of the dummy solar tracker

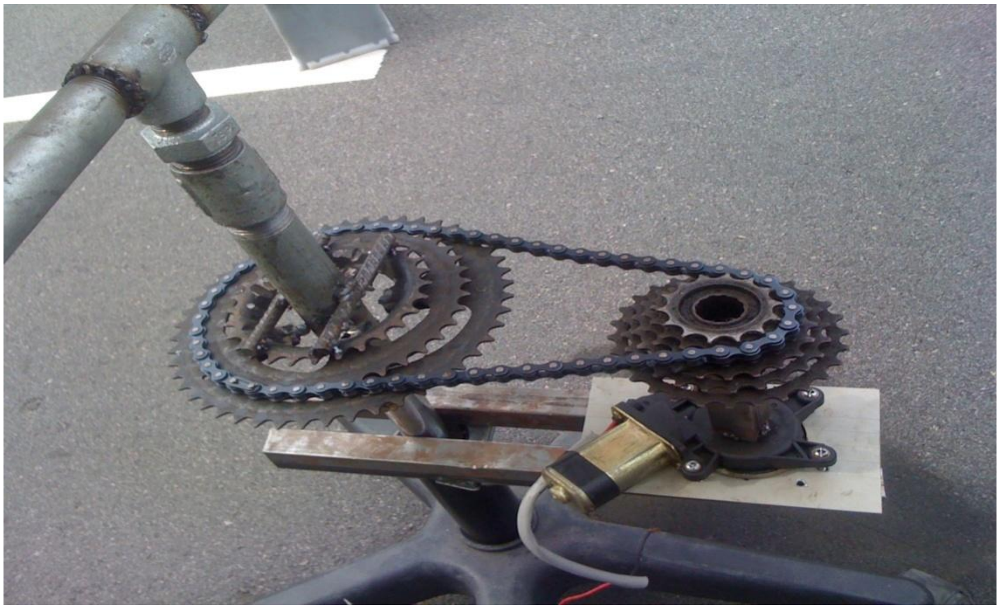


Figure 3.4 Rotating gears for the azimuth axis.

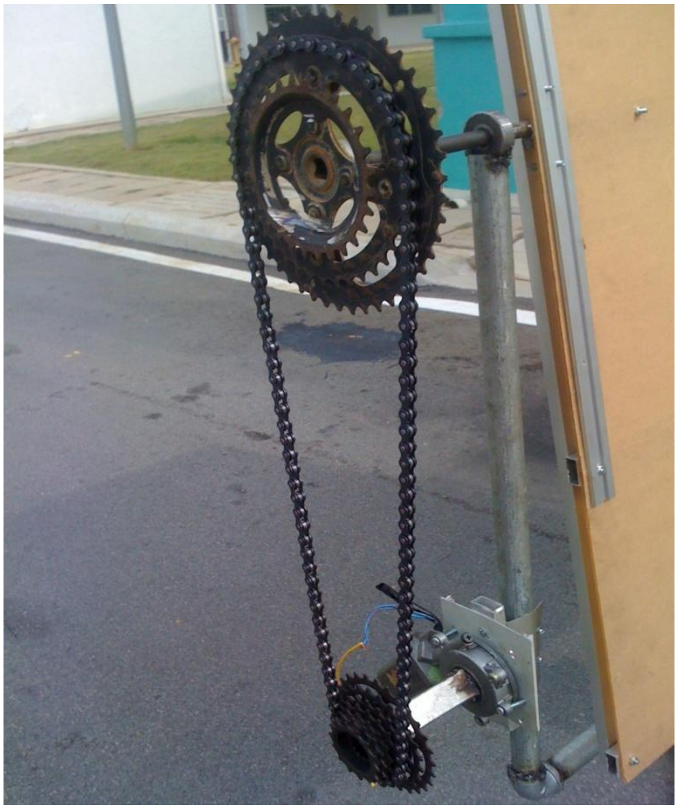


Figure 3.5: Rotating gears for the elevation axis.

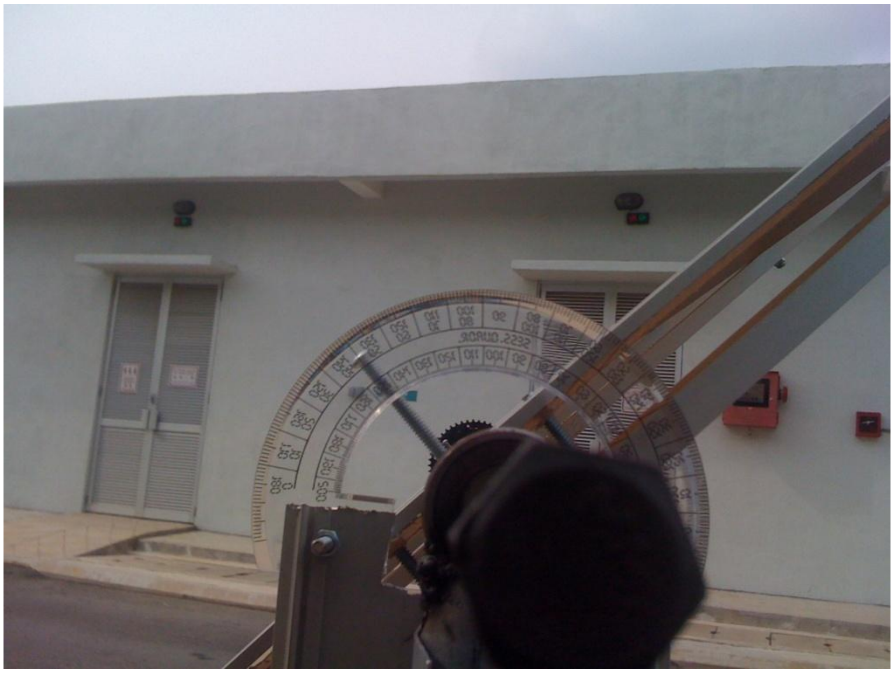


Figure 3.6: Protractor used to read the elevation angle of the dummy solar tracker.

iii) Test and evaluation of the dummy solar tracker

The last step is to test and evaluate the function ability of the dummy solar tracker, which whether it fulfills the condition where the tracker is perpendicular with the sunlight at the given time or not. There will be a straight rod placed perpendicularly on the surface of the solar tracker. The accuracy of the tracking will be shown by the length of the shadow. Figure 3.7 below shows the shadow indicator on the dummy solar tracker.

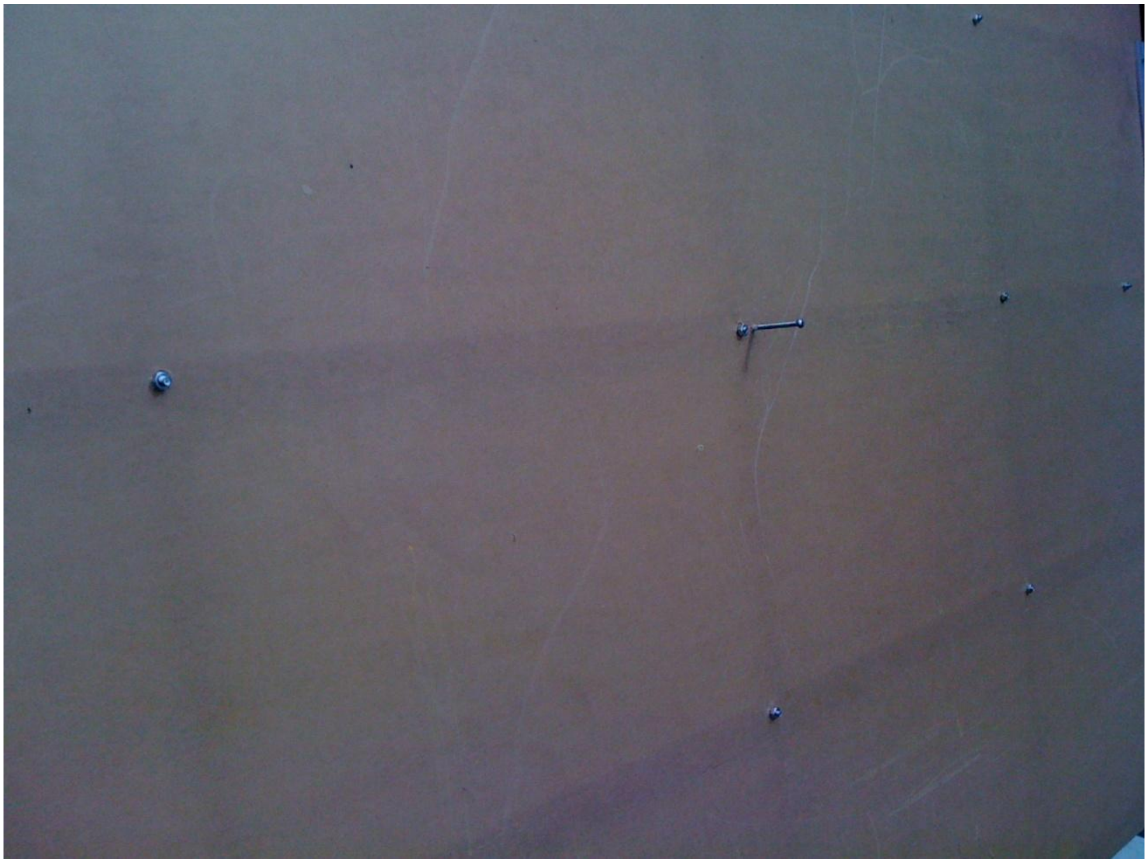


Figure 3.7: Shadow indicator on dummy solar tracker.

Flow chart of dummy solar panel

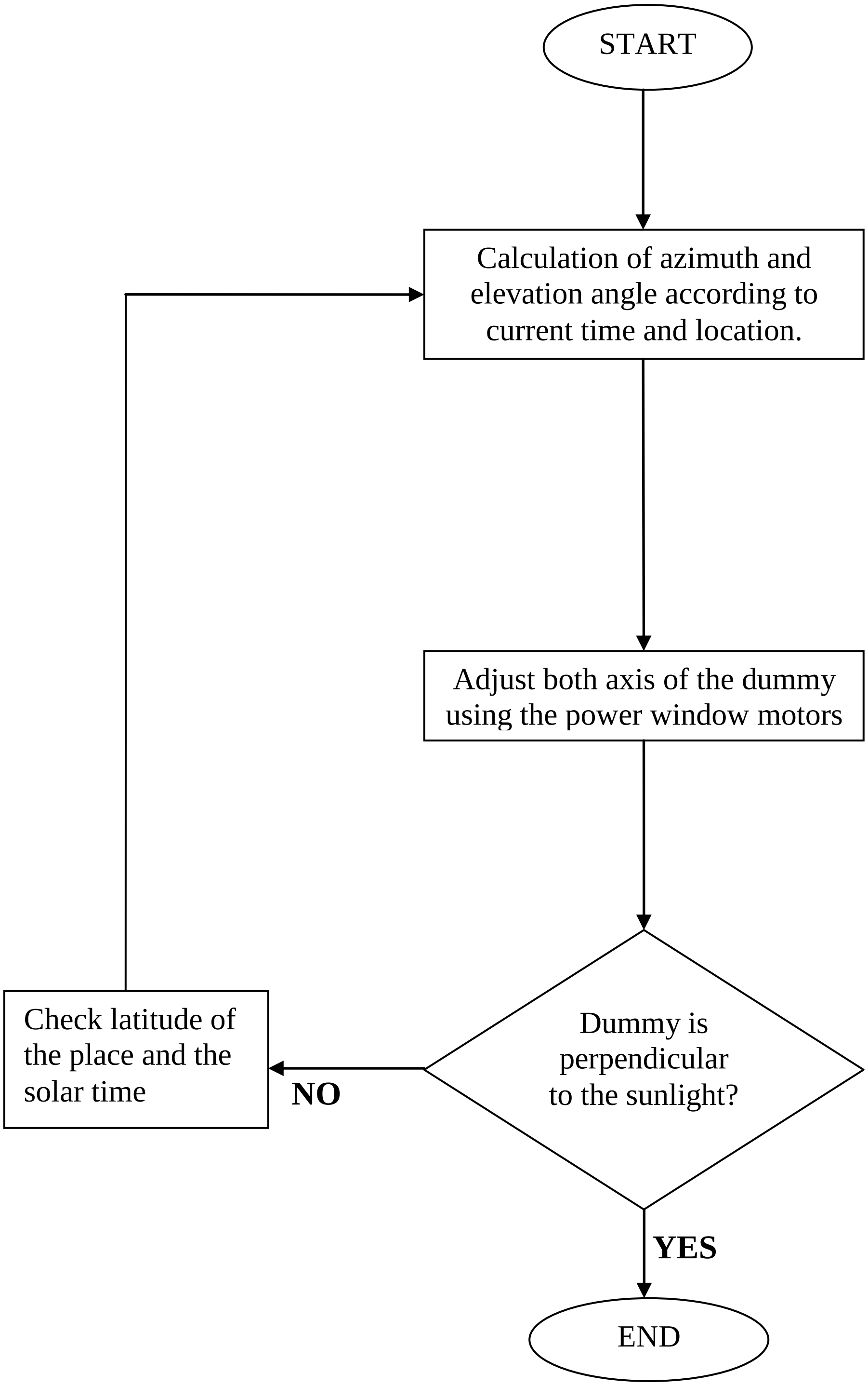

START
Calculation of azimuth and elevation angle according to current time and location.
Adjust both axis of the dummy using the power window motors
Dummy is perpendicular to the sunlight?
NO
Check latitude of the place and the solar time
YES
END

Figure 3.8: Flow chart of dummy solar tracker

3.4 SIMULATION OF DUAL AXIS SOLAR TRACKER

Flow chart of simulation using Borland C++ Builder.

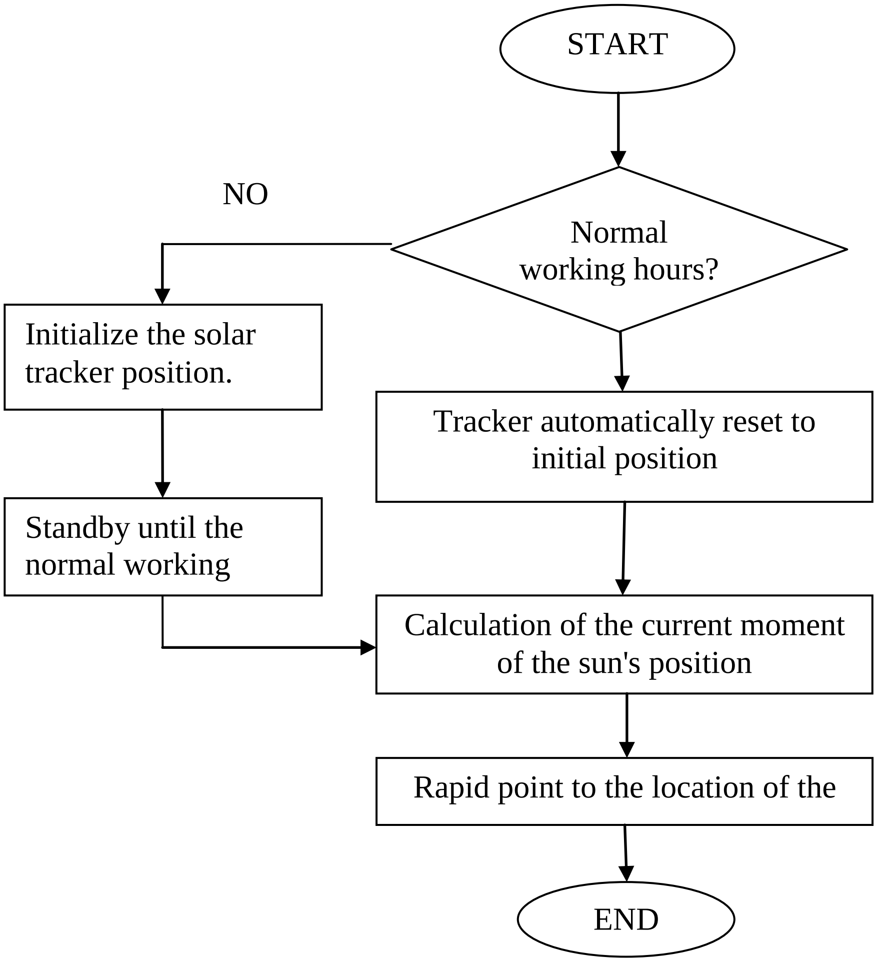

START
Normal working hours?
NO
Initialize the solar tracker position.
Standby until the normal working
Tracker automatically reset to initial position
Calculation of the current moment of the sun's position
Rapid point to the location of the
END

Figure 3.9: Flow chart of simulation.

Borland C++ Builder is chosen to be the software used for simulation of the dual axis solar tracker because it contains the required features which includes the application of the mathematical formulas of the azimuth and elevation angles which controls the simulation of the tracker, and also the necessary components for the interface between stepper motor with the computer using a USB DAQ card.

The simulation of the solar tracker involves 2 stages, which is initialization of the tracker and operation during run-time of the tracker. The initialization stage of the tracker is essential to set the position of the tracker for the starting operation time. This is necessary when the solar tracker is to be used for the first time, and also when the solar tracker is being shut down for maintenance purpose. The operation during run-time of the tracker is where the simulation integrates with the dummy solar tracker and operates automatically to track the sun.

3.4.1 The initialization process of the tracker

The initialization for the tracker is important because this project emphasis on the open-loop system, which does not involve sensors. Therefore, the tracker will not act according to the real position from the sun; instead, it will operate according to the mathematic calculations alone. Hence, when maintenance is to be carried out, the position of the tracker will be incorrect and irrelevant to the operation time since there is no sensor involved in the system. Now, the initialization for the tracker should be implemented to manipulate the position of the tracker to specific angle which corresponds to the time where the tracker will start operates. This can ensure that the starting position of the tracker is correct.

The operation for the solar tracker only takes place during the time when the sun is above sea level, which means from sunrise until sunset. Hence, the default starting position can be set for the tracker to be returned to. For example, in Pekan, Pahang, the average time for sunrise is 7.15am and the average time for sunset is 7.20pm. Therefore, the default time for the tracker to start operating can be set as 7.00am and the default time for the tracker to return to the initial position can be set as 7.45pm. This means that the initialization process should be carried out only after maintenance took place, so that the tracker can be operated right after the maintenance is done, instead of having to standby until the next day.

The program GUI for the initialization of the tracker:

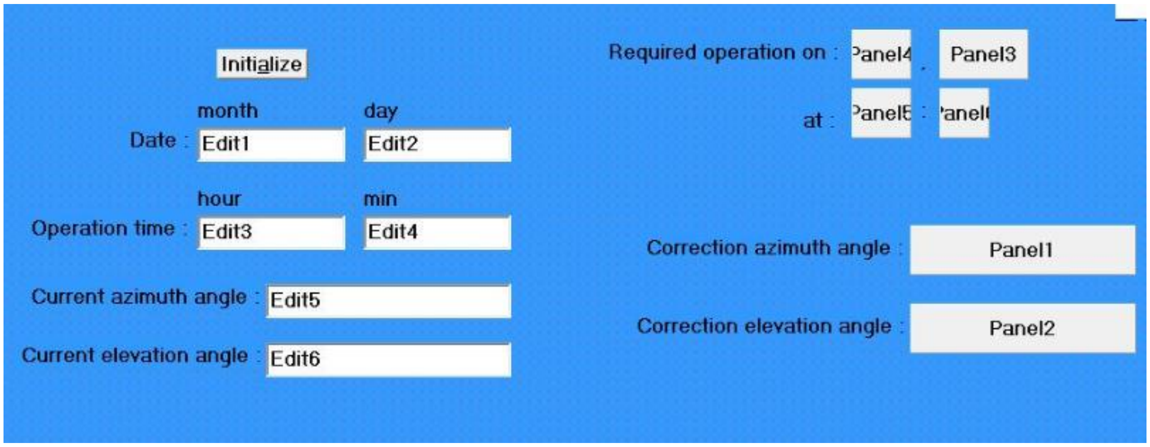


Figure 3.10: Program GUI for the initialization of the tracker.

The program coding for the initialization of the tracker:

i) Coding for user to set the desired time and day for the solar tracker to start tracking.

```
{
double a,X,N,decl,decl2,lati2,X2,T,lon,Tc,Ts,H,H2,elev,elev2,azim,azim2,azim3,
pi = 3.1416,lati,longi;
int month,day,hour,min;
hour = Edit3->Text.ToInt();
min = Edit4->Text.ToInt();
day = Edit2->Text.ToInt();
month = Edit1->Text.ToInt();
}
```

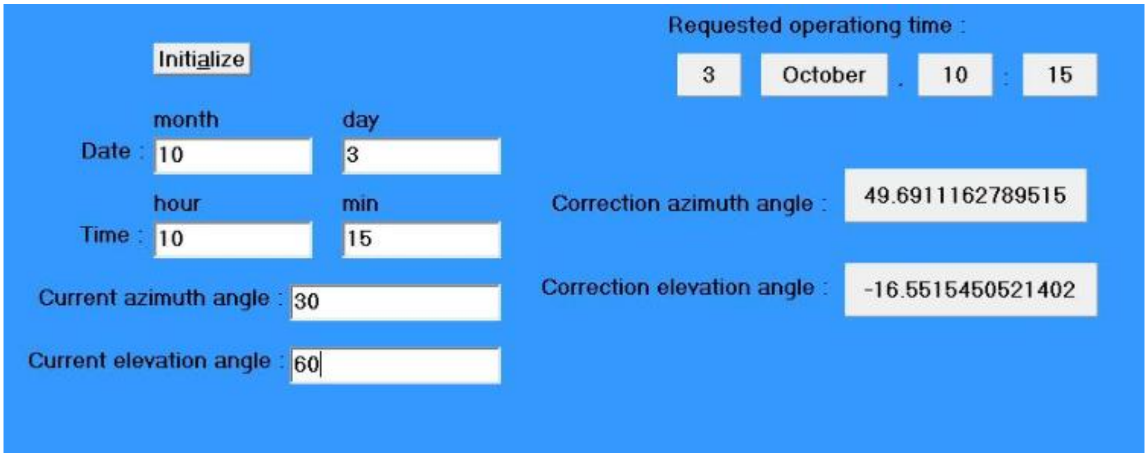
ii) Coding for calculation of correction angles of azimuth and elevation.

```
{
azim_ex=Edit5->Text.ToDouble();
elev_ex=Edit6->Text.ToDouble();
a = (284 + N)*360/365;
X = 360*(N - 81)/365.2422;
decl = 23.45*sin(a*pi/180);
decl2 = decl*pi/180;
lati2 = lati*pi/180;
X2 = X*pi/180;
T = 12 + (0.1236*sin(X2)- 0.0043*cos(X2)) + (0.1583*sin(2*X2) + 0.0608*cos(2*X2));
lon = (120 - longi)/15;
Tc = hour+0.016666666666667*min;
Ts = Tc - (T/60) - lon;
H = (Ts - 12)*15;
H2 = H*pi/180;
elev = asin(cos(lati2)*cos(decl2)*cos(H2)+sin(lati2)*sin(decl2));
elev2 = elev*180/pi;
azim = -sin(H2)*cos(decl2)/cos(elev);
if (azim>1)      {
      azim = 2 - azim;
                 }
else if (azim<-1)
```

```
{
        azim = -(2+azim);
}
azim2 = asin(azim);
azim3 = azim2*180/pi;
azim4 = azim3 - azim_ex;
elev3 = elev2 - elev_ex;
Panel1->Caption = azim4;
Panel2->Caption = elev3;
```

*note: azim4 is the correction azimuth angle and elev3 is the correction elevation angle.

An example of initializing the tracker:



Requested operationg time : 3 October . 10 : 15

Date : month 10 day 3

Time : hour 10 min 15

Current azimuth angle : 30

Current elevation angle : 60

Correction azimuth angle : 49.6911162789515

Correction elevation angle : -16.5515450521402

Figure 3.11: Example of initializing the tracker.

The Azimuth-Elevation angles for the operation time requested will be obtained. The difference between the current azimuth-elevation of the tracker will be subtracted to the calculated azimuth-elevation angle at the requested operation time to calculate the required angles correction to be subjected to the tracker.

After the tracker is pointed towards the azimuth-elevation angles for the requested operation time, it means that the tracker is initialized. The operation of the tracker will be started as soon as the real time approaches the requested operation time.

3.4.2 The operation of the tracker during run-time

This section illustrates the operation of the solar tracker during the run-time. Since the power window motors are used as the rotators, the speed of the motors can be manipulated by controlling the voltage input from power source. The frequency or rotating time will be controlled by using the Borland C++ Builder. The output from digital output on DAQ card will act as a switch to turn on/off the power window motors. The minimum time for switching the digital output is 1ms.

The program GUI for the run-time operation of the tracker:

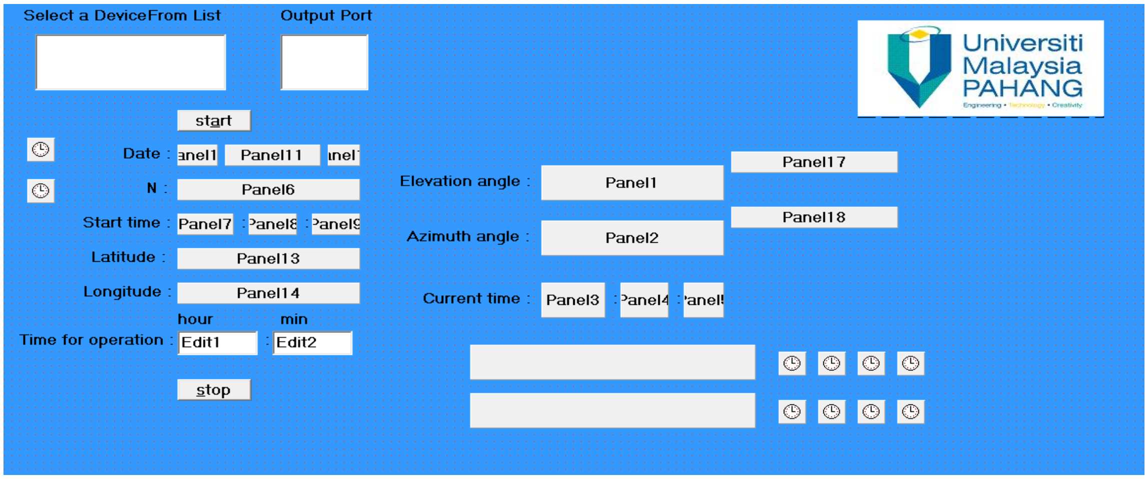


Figure 3.12: Program GUI for the run-time operation of the tracker.

The time for operation can be set as default. For example, the default time for operation can be set as 7am since the average sunrise time for Malaysia is 7.15am.

The program coding for the tracker operation during run-time:

i) Timer looping for continuous calculation of the azimuth-elevation angle:

```
int StartCount()
{
    for (x=0;x<100000000000;x++)
    {
        Sleep (1000);
    }
    return 0;
}
```

ii) Coding to obtain the azimuth-elevation angles:

```
{
    double N;
    SYSTEMTIME Local_time;
    GetLocalTime(&Local_time);
    day = Local_time.wDay;
    month = Local_time.wMonth;
    start_hour=Edit1->Text.ToInt();
    start_min=Edit2->Text.ToInt();
```

```
Timer1->Enabled=true;
Panel10->Caption=Local_time.wYear;
Panel12->Caption=Local_time.wDay;
if (month==1)
        {
        Panel11->Caption="January";
        N=day;
        }
else if (month==2)
        {
        Panel11->Caption="February";
        N=31+day;
        }
else if (month==3)
        {
        Panel11->Caption="March";
        N=31+28+day;
        }
else if (month==4)
        {
        Panel11->Caption="April";
        N=31+28+31+day;
        }
else if (month==5)
        {
        Panel11->Caption="May";
        N=31+28+31+30+day;
        }
else if (month==6)
        {
        Panel11->Caption="June";
        N=31+28+31+30+31+day;
        }
else if (month==7)
        {
        Panel11->Caption="July";
        N=31+28+31+30+31+30+day;
        }
else if (month==8)
        {
        Panel11->Caption="August";
        N=31+28+31+30+31+30+31+day;
        }
else if (month==9)
        {
        Panel11->Caption="September";
        N=31+28+31+30+31+30+31+31+day;
        }
else if (month==10)
        {
        Panel11->Caption="October";
        N=31+28+31+30+31+30+31+31+30+day;
        }
else if (month==11)
        {
        Panel11->Caption="November";
        N=31+28+31+30+31+30+31+31+30+31+day;
        }
else if (month==12)
        {
        Panel11->Caption="December";
        N=31+28+31+30+31+30+31+31+30+31+30+day;
        }
    Panel6->Caption=N;
    hour = Local_time.wHour;
    min = Local_time.wMinute;
    sec = Local_time.wSecond;
    Panel7->Caption=hour;
    Panel8->Caption=min;
    Panel9->Caption=sec;
    Panel13->Caption=lati;
    Panel14->Caption=longi;
    a = (284 + N)*360/365;
    X = 360*(N - 81)/365.2422;
    CreateThread(0,0,(LPTHREAD_START_ROUTINE)&StartCount,0,0,0);

}
//---------------------------------------------------------------------
{
SYSTEMTIME Local_time;
GetLocalTime(&Local_time);
decl = 23.45*sin(a*pi/180);
decl2 = decl*pi/180;
lati2 = lati*pi/180;
X2 = X*pi/180;
T = 12 + (0.1236*sin(X2)-0.0043*cos(X2)) + (0.1583*sin(2*X2) + 0.0608*cos(2*X2));
lon = (120 - longi)/15;
Tc2 = hour+0.01666666666666667*min+0.000277777777777778*sec;
Tc = Tc2 + (0.00027777777777777778)*x;
Ts = Tc - (T/60) - lon;
H = (Ts - 12)*15;
H2 = H*pi/180;
elev = asin(cos(lati2)*cos(decl2)*cos(H2)+sin(lati2)*sin(decl2));
```

```
elev2 = elev*180/pi;
azim = -sin(H2)*cos(decl2)/cos(elev);
if (azim>1)
{
        azim = 2 - azim;
}
else if (azim<-1)
{
        azim = -(2+azim);
}

azim2 = asin(azim);
azim3 = azim2*180/pi;
Panel1->Caption="Standby position!";
Panel2->Caption="Standby position!";
Panel3->Caption=Local_time.wHour;
Panel4->Caption=Local_time.wMinute;
Panel5->Caption=Local_time.wSecond;
//---------------------------------------------------------------------------
```

The time and date are obtained according to the system clock in the computer, which corresponds to the actual time when this simulation is being tested.

iii) Coding for user to set the starting time of the solar tracker.

```
{
start_hour=Edit1->Text.ToInt();
start_min=Edit2->Text.ToInt();
if (Local_time.wHour>=start_hour && Local_time.wMinute>=start_min)
```

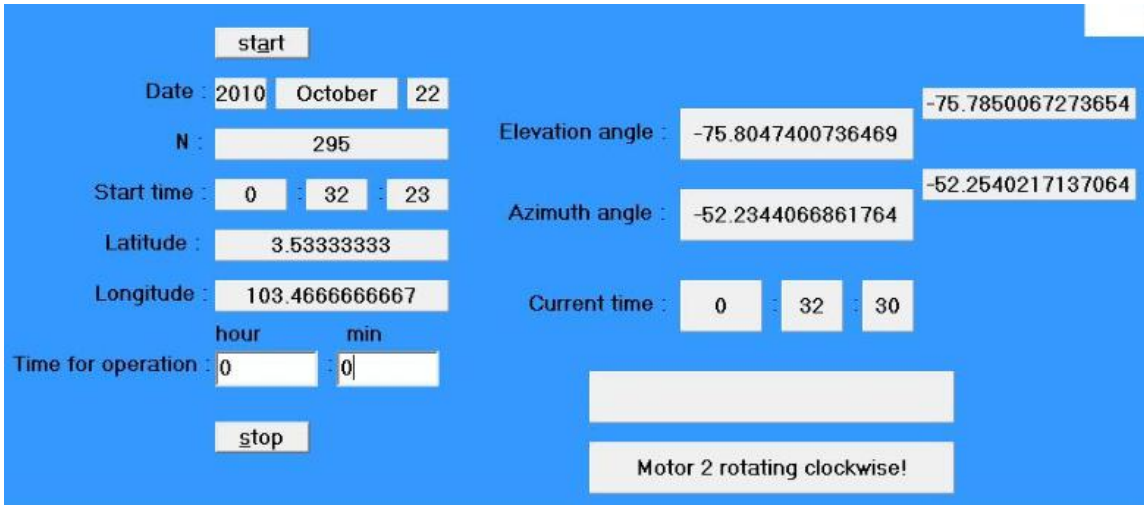
```
        {
        Panel1->Caption=elev2;
        Panel2->Caption=azim3;
        start_hour=0;
        start_min=0;
        }


}
//---------------------------------------------------------------------------
```

iv) Coding to stop the simulation process

```
{
Timer1->Enabled=false;
x=0;
}
//---------------------------------------------------------------------------
```

An example of tracker operation during run-time:



Date : 2010 October 22

N : 295

Start time : 0 : 32 : 23

Latitude : 3.53333333

Longitude : 103.4666666667

Time for operation : hour 0 : min 0

Elevation angle : -75.8047400736469

-75.7850067273654

Azimuth angle : -52.2344066861764

-52.2540217137064

Current time : 0 : 32 : 30

Motor 2 rotating clockwise!

Figure 3.13: An example of tracker operation during run-time.

Depending on the specifications of the motor used for the dual axis solar tracker, a specific value change in the azimuth-elevation angle will trigger the rotation of the motor.

CHAPTER 4

RESULTS AND ANALYSIS

4.1 INTRODUCTION

The results for this project are divided into two sections: software implementation - simulation of dual axis solar tracker using Borland C++ Builder, and dummy solar tracker.

4.2 SIMULATION

The simulation of the system illustrates the operation of a real dual axis tracker. The operation includes initialization of the tracker and the run-time of the dual axis solar tracker.

4.2.1 Table of calculations for azimuth-elevation angles

The table of calculations is done using Microsoft Excel by inserting the required formulas. Results can be obtained by manipulating the variables such as local latitude, date, time and so on. The results obtained from the calculations can be analyzed and to make sure that the results of the azimuth-elevation angles obtained are accurate theoretically.

Besides that, the azimuth-elevation angles shown from Microsoft Excel can be used to compare to the results obtained from the calculations from simulation to ensure that there are no mistakes in the software simulation.

For different place or time, the azimuth-elevation angles will vary. For example, the difference in latitude for different places will affect the results.

Table 4.1: Calculations for azimuth-elevation angles

For working hours in 1 day.

N	ϕ	B	lon	LST	ΔT	λ	t_c	ts	δ	H	α	θ
214	3.53	131	103.5	120	11.9	1.102	7	6.1	17.65	-88.6	2.45	72.46
214	3.53	131	103.5	120	11.9	1.102	8	7.1	17.65	-73.6	16.7	72.62
214	3.53	131	103.5	120	11.9	1.102	9	8.1	17.65	-58.6	31	71.5
214	3.53	131	103.5	120	11.9	1.102	10	9.1	17.65	-43.6	45.1	68.39
214	3.53	131	103.5	120	11.9	1.102	11	10.1	17.65	-28.6	58.7	61.14
214	3.53	131	103.5	120	11.9	1.102	12	11.1	17.65	-13.6	70.6	42.28
214	3.53	131	103.5	120	11.9	1.102	13	12.1	17.65	1.449	75.8	-5.64
214	3.53	131	103.5	120	11.9	1.102	14	13.1	17.65	16.45	68.6	-47.6
214	3.53	131	103.5	120	11.9	1.102	15	14.1	17.65	31.45	56.1	-63.1
214	3.53	131	103.5	120	11.9	1.102	16	15.1	17.65	46.45	42.4	-69.2
214	3.53	131	103.5	120	11.9	1.102	17	16.1	17.65	61.45	28.2	-71.8
214	3.53	131	103.5	120	11.9	1.102	18	17.1	17.65	76.45	14	-72.7
214	3.53	131	103.5	120	11.9	1.102	19	18.1	17.65	91.45	-0.31	-72.3

For similar working hours in different days

N	ϕ	X	lon	LST	ΔT	λ	t_c	ts	δ	H	α	θ
215	3.53	132	103.5	120	11.9	1.102	12	11.1	17.38	-13.6	70.8	42.86
215	3.53	132	103.5	120	11.9	1.102	12	10.7	17.38	-19.5	66.4	52.72
216	3.53	133	103.5	120	11.9	1.102	12	10.7	17.11	-19.5	66.5	53.29
217	3.53	134	103.5	120	11.9	1.102	12	10.7	16.83	-19.5	66.7	53.88
218	3.53	135	103.5	120	11.9	1.102	12	10.7	16.55	-19.5	66.8	54.48
219	3.53	136	103.5	120	11.9	1.102	12	10.7	16.26	-19.5	67	55.11
220	3.53	137	103.5	120	11.9	1.102	12	10.7	15.96	-19.5	67.1	55.75
221	3.53	138	103.5	120	11.9	1.102	12	10.7	15.67	-19.5	67.3	56.42
222	3.53	139	103.5	120	11.9	1.102	12	10.7	15.36	-19.5	67.4	57.1
223	3.53	140	103.5	120	11.9	1.102	12	10.7	15.06	-19.5	67.6	57.8
224	3.53	141	103.5	120	11.9	1.102	12	10.7	14.74	-19.5	67.7	58.52
225	3.53	142	103.5	120	11.9	1.102	12	10.7	14.43	-19.5	67.9	59.26
226	3.53	143	103.5	120	11.9	1.102	12	10.7	14.11	-19.5	68	60.02
227	3.53	144	103.5	120	11.9	1.102	12	10.7	13.78	-19.5	68.2	60.8
228	3.53	145	103.5	120	11.9	1.102	12	10.7	13.45	-19.5	68.3	61.59

4.2.2 Analysis on azimuth-elevation angles with different parameters

An analysis is taken by using different latitude angles to calculate the azimuth-elevation angles. All the other variables such as time, date are consistent. The time is set as 12pm. It is showed in the table below.

Table 4.2: Azimuth-elevation angles for different latitudes.

Latitude	Azimuth	Elevation
0	46.40	63.92
2	49.52	65.26
4	53.00	66.51
6	56.88	67.66
8	61.17	68.69

From the results of the table above, it is observed that the latitude does not affect elevation angles too much, but it drastically affects the azimuth angles. Experiments have proved that the higher the latitude, the lower the irradiation from the sun.

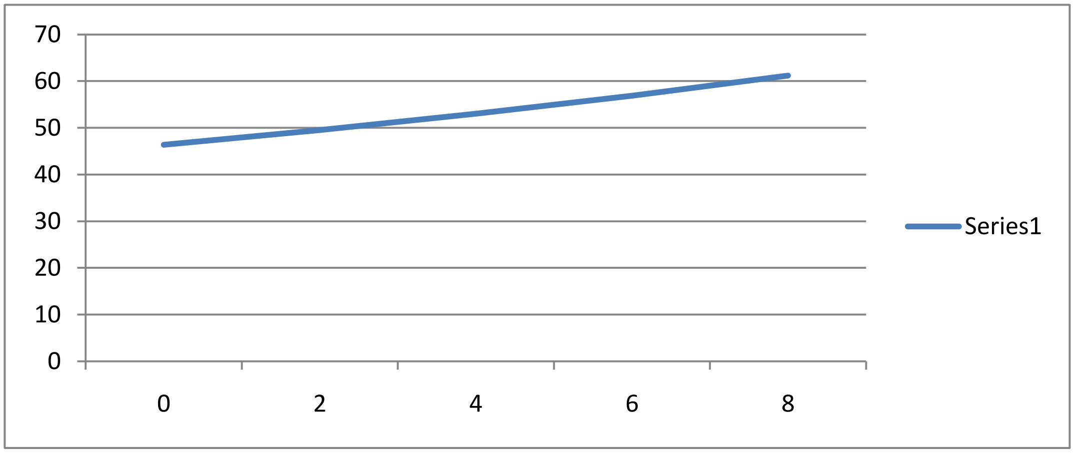


Figure 4.1: Graph of azimuth angles-latitude angles.

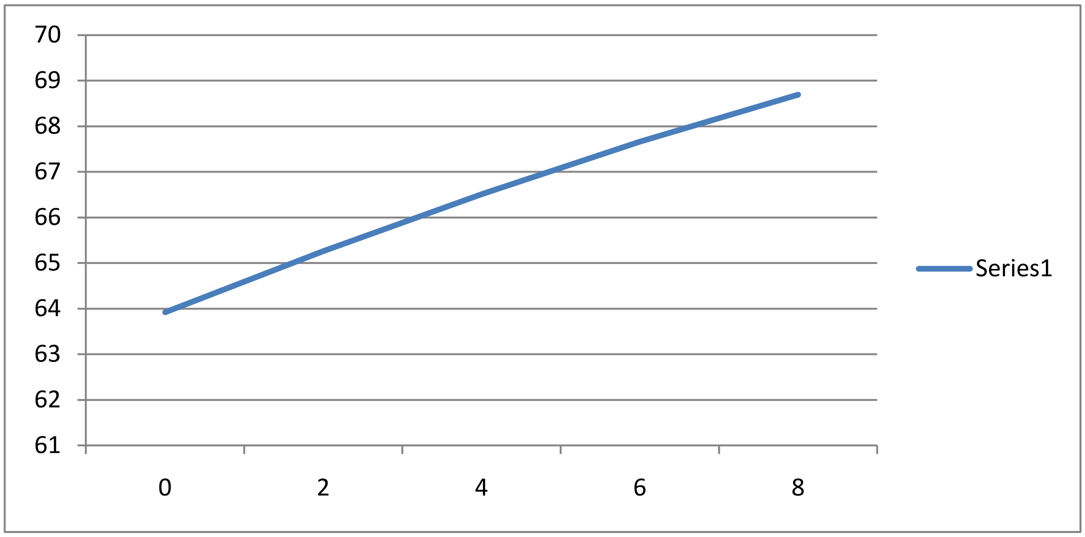


Figure 4.2: Graph of elevation angles-latitude angles.

From the graphs above, it is proved that the azimuth-elevation angles are directly proportional to the latitude angles.

Another analysis is taken by using different time on the same day and same latitude to obtain the results for the azimuth-elevation angles. The latitude is set to 3.0 degree. The results are shown in the table below.

Table 4.3: Azimuth-elevation angles for different time of the day.

Time	Azimuth	Elevation
8	72.60	10.90
10	69.54	39.23
12	51.22	65.90
14	-34.35	72.10
16	-66.82	47.70
18	-72.33	19.50

The table above proved that for the positive value of azimuth angle indicates that the solar tracker is facing east while the negative value of azimuth angle indicates that the solar tracker is facing west. The results above also showed that zero elevation angle means the sun is directly on the sea level. The elevation angles are minimum during sunrise and sunset, and the elevation angles are maximum during the afternoon, which is 12pm at solar time.

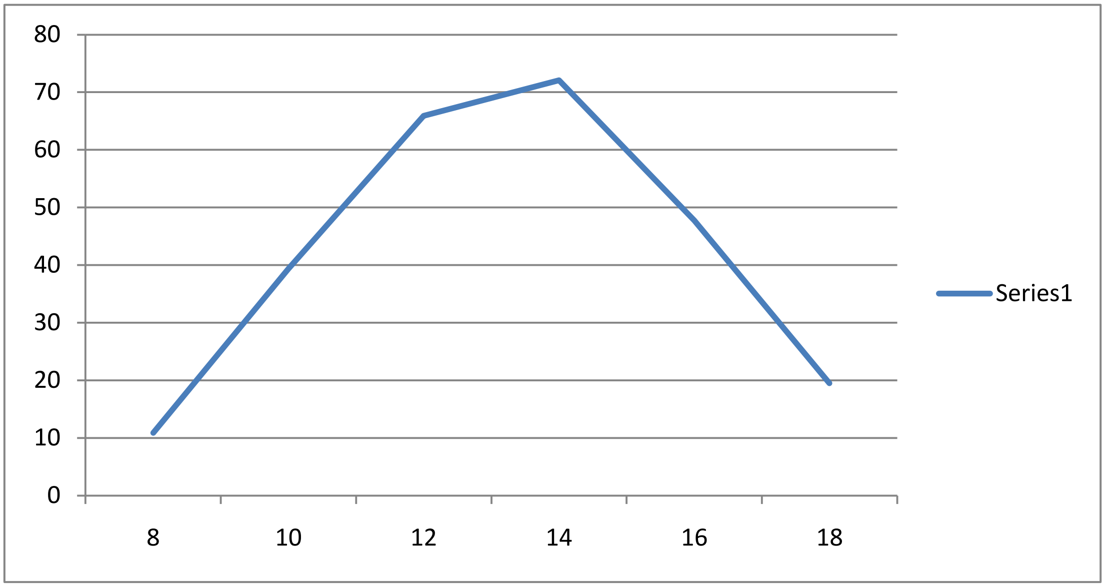


Figure 4.3: Graph of elevation angles-time of day

The graph above showed the vertical movement of the sun from 8am till 6pm. The sun is consistently rising from 8am until 14pm, and started setting after 2pm. The time for the sun to start setting varies depending on the latitude of the place.

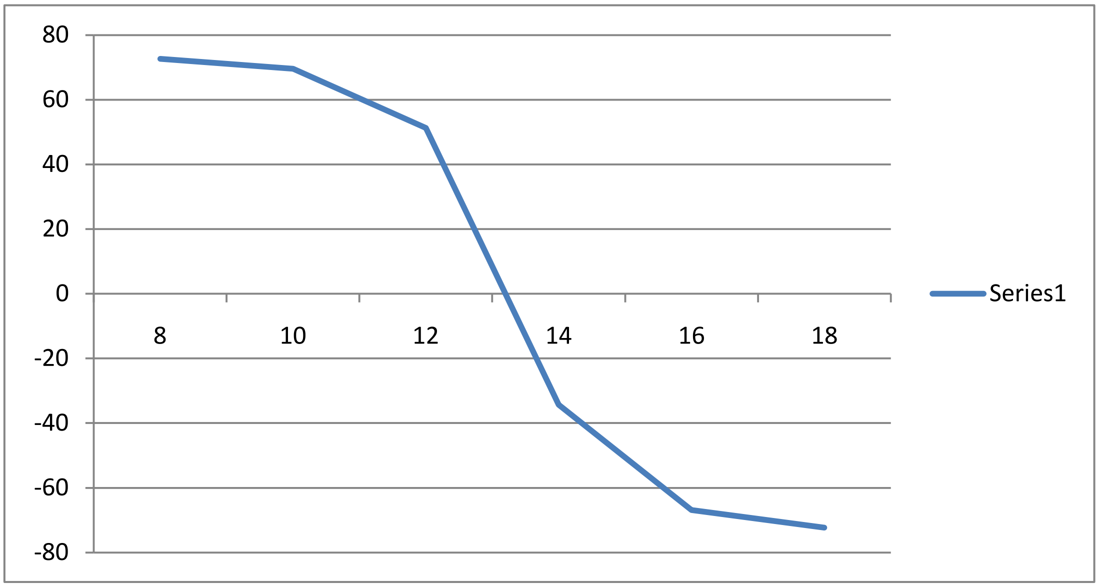


Figure 4.4: Graph of azimuth angles-time of day

The graph above showed the horizontal movement of the sun from 8am till 6pm. The positive values show the sun is positioned at the east while the negative values show the sun is positioned at the west. The time for the sun to shift from east to west varies depending on the latitude of the place.

The analysis on the effects of date on azimuth-elevation angles is also being carried out. The time is set as 12pm and the latitude is set as 3.0 degree. The results are tabulated in the table below.

Table 4.4: Azimuth-elevation angles for different dates.

Date	Azimuth	Elevation
1st day of January	35.15	57.79
1st day of March	52.04	65.54
1st day of May	65.15	68.78
1st day of July	41.08	62.15
1st day of September	64.63	68.77
1st day of November	48.68	64.35

The table above showed that the date does affect the azimuth-elevation angles. The change in date does not has a constant effect on the azimuth-elevation angles. So we can conclude that the date randomly effects the azimuth-elevation angles.

4.3 Dummy Solar Tracker

The purpose of the dummy solar tracker is to prove the perpendicularity between the solar panel and the sun. The integration of the dummy solar tracker and the simulation using Borland C++ proves the feasibility of dual axis solar tracking system in practical. The concept of this dummy solar tracker is that the dummy solar tracker will automatically track the sun when it is initialized and activated. The initialization process is to set the initial position of the dummy solar tracker according to the starting time of operation for the solar tracking. In practical, this initialization process is only necessary during the installation of the tracker and during maintenance.

4.3.1 Rotation of Dummy Solar Tracker

The mechanism of the rotation for both the axis of the solar tracker is controlled by using manual calculation of the speed of the power window motor according to the value of voltage injected. In this project, the voltage is set to be 12V, and the speed of the power window motor is to be calculated by manually testing the rotation. The time of rotation for 180 ° is taken, and then the time required for the power window to rotate for 1 ° can be obtained by dividing the total rotation time of 1 full cycle with 180 °.

Calculation of rotation time:

Time taken for the elevation axis to rotate 180 °: 90s.
Time required for the elevation axis to rotate 1 °: $\frac{60}{180} = 0.5$s
Time taken for the azimuth axis to rotate 180 °: 120s.
Time required for the azimuth axis to rotate 1 °: $\frac{120}{180} = 0.667s$

This method is not the most precise method for accurate results, however, due to the limitations of this project, this method is chosen. In a practical dual axis solar tracker, stepper motors can be chosen to replace the power window motors in this dummy solar tracker. This is because the angle of rotation for the stepper motor can be precisely obtained by referring to the data sheet of that motor. Then, the desired angle of rotation can be achieved precisely and accurately.

4.3.2 Tracking Results

The straight aluminium rod in the center of the panel is used to prove the accuracy of the solar tracking of the system. When the rod has minimum length of shadow, it proves that the panel of the dummy solar tracker is in a perpendicular to the sun, which means the solar tracking is accurate. Figures below shows some examples of the performance of the dummy solar tracker based on the shadow indicator.

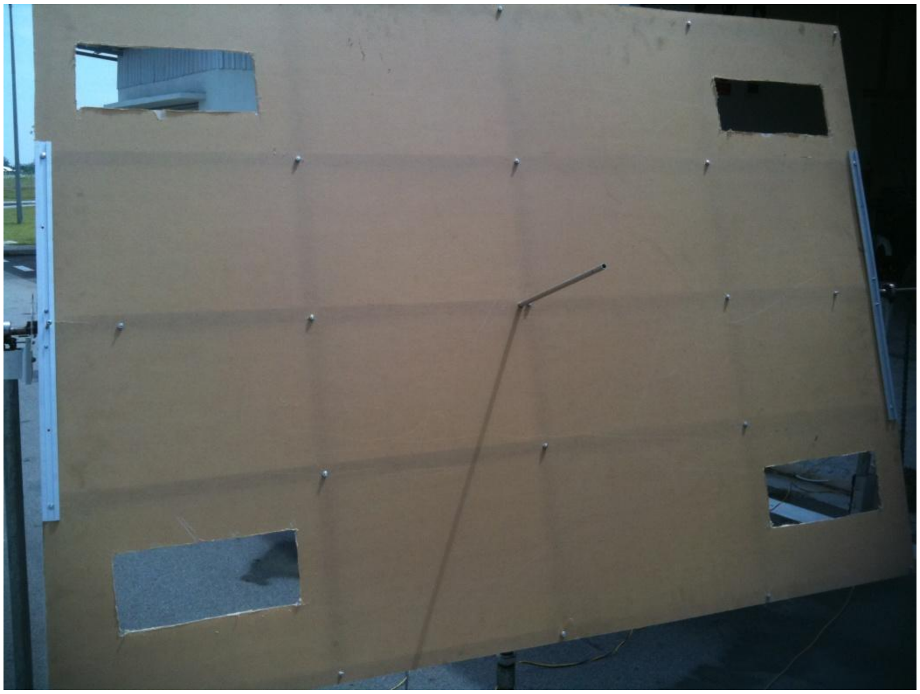


Figure 4.5: Example of shadow from shadow indicator without active tracking.

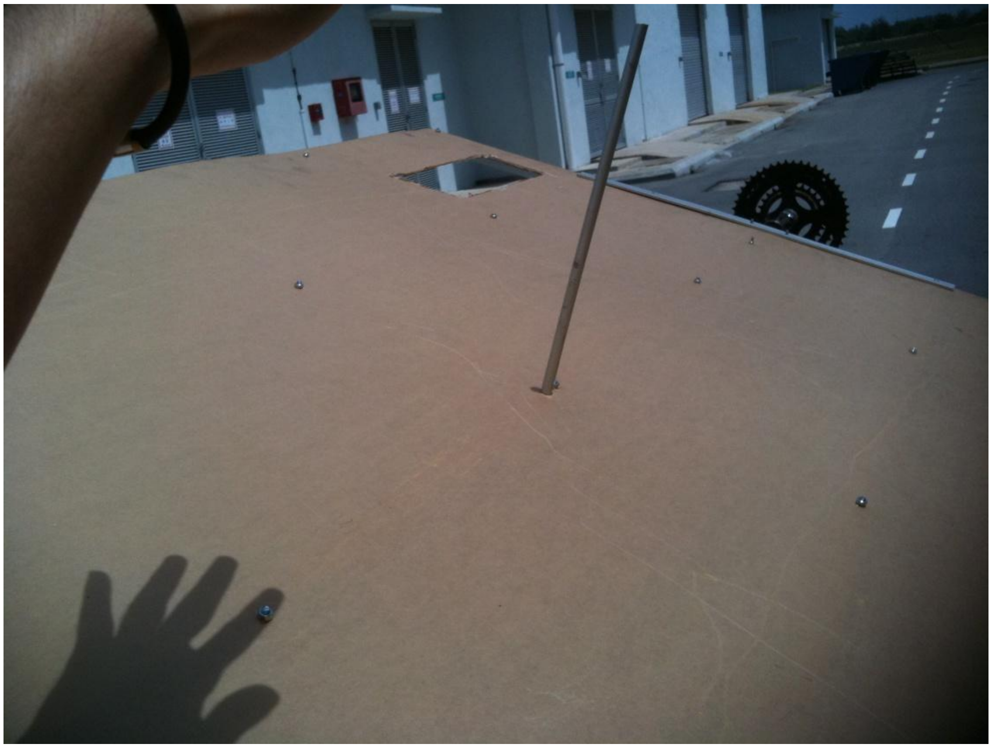


Figure 4.6: Example of shadow from shadow indicator during active tracking.

From figure 4.5 above, the shadow indicator produces a long shadow and proves that the position of the panel is not directly perpendicular to the sun. Figure 4.6 above shows that there’s no shadow emitted from the shadow indicator, which proves that the panel of the dummy solar tracker is directly perpendicular to the sun.

Note*: The purpose of the hand is to show that the sun is actually present.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

This project has successfully developed a simulation of a dual axis solar tracker. To prove the accuracy of the azimuth-elevation angles, a dummy solar tracker is constructed to with a shadow indicator to show the perpendicularity between the surface of the tracker and the sun.

From the evaluation of the dummy solar tracker, it is proved that the open-loop system is the most cost efficient among all of the solar tracking systems because the sun is tracked accurately and consistently without needing to rely on any sensors.

5.2 FUTURE RECOMMENDATION

This project has successfully demonstrated the simulation of a dual axis solar tracker with the aid of a dummy solar tracker. Future work on this project may include:

- A real photovoltaic panel to replace the dummy solar tracker. This will require a much higher cost. Besides that, the construction of the dual axis solar tracker must emphasis on the torque and larger motors must be chosen in order to support the amount of weight from the photovoltaic panel.

- Feedback from the solar tracker to the system. A data transfer between the system and the panel of the dummy solar tracker can be manifested. This will increase the accuracy of the whole solar tracking system and reducing the need of maintenance. Whenever the position of the panel is not accurate, there will be a signal sent to the system to correct the solar tracker. A linear position sensor can be used as the input signal for the data exchange.

- Analysis of power output from the dual axis solar tracker. The power output from the dual axis solar tracker from can be collected and compared to the standalone photovoltaic panel. The standalone photovoltaic panel can be represented by a non operating dual axis solar tracker. This will further prove the efficiency of the dual axis solar tracking system.

REFERENCES

1. S. R. Bull, “Renewable energy today and tomorrow,” IEEE Proc., vol. 89, no. 8, pp. 1216-1226, 2001.
2. S. Rahman, “Green power: what is it and where can we find it?” IEEE Power and Energy Magazine, vol. 1, no. 1, pp. 30-37, 2003.
3. Solar Energy International, Photovoltaics: Design and Installation Manual, New Society Publishers, 2004.
4. Barakat B, Rahab H, Mohmedi B, Naiit N. Design of a tracking system to be used with photovoltaic panels (Arabic). In: Proceedings of the Fourth Jordanian International Mechanical Engineering Conference JIMEC 2001.p. 471–488.
5. Abdalah S, Nijmeh S. Design, construction and operation of one axis suntracking system with PLC control. Jord J Appl Sci Univer 2002;4(2):45–53.
6. P.A.Davies.SUN-TRACHING MECHANISM USING EQUATORIAL AND ECLIPTICAXES[J],Solar Energy，1993,Vol.50，No.6，pp.487-489
7. Willian A.Lynch and Ziyad M.Salameh.SIMPLE ELECTROOPTICALLY CONTROLLED DUAL-AXIS SUN TRACKER[J] Solar Energy，1990.Vol.45，No.22，pp.65-69.
8. Marlett Wentzel and Anastassios Pouris.The development impact of solar cookers:A review of solar cookingimpact research in South Africa[J]，Energy Policy，2007.35(3)，1909-1919.
9. T. Markvart, Solar Electricity, Wiley, 2007
10. D. A. Pritchard, “Sun tracking by peak power positioning for photovoltaic concentrator arrays,” IEEE Contr. Syst. Mag., vol. 3, no. 3, pp. 2-8, 1983.
11. A. Konar and A. K. Mandal, “Microprocessor based automatic sun tracker,” IEE Proc. Sci., Meas. Technol., vol. 138, no. 4, pp. 237-241, 1991.
12. B. Koyuncu and K. Balasubramanian, “A microprocessor controlled automatic sun tracker,” IEEE Trans. Consumer Electron., vol. 37, no. 4, pp. 913-917, 1991.
13. J. D. Garrison, “A program for calculation of solar energy collection by fixed and tracking collectors,” Sol. Energy, vol. 72, no. 4, pp. 241-255, 2002.

14. P. P. Popat "Autonomous, low-cost, automatic window covering system for daylighting Applications," Renew. Energ., vol. 13, no. 1, pp. 146, 1998.
15. M. Berenguel, F. R. Rubio, A. Valverde, P. J. Lara, M. R. Arahal, E. F. Camacho, and M. López, "An artificial vision-based control system for automatic heliostat positioning offset correction in a central receiver solar power plant," Sol. Energy, vol. 76, no. 5, pp.563-575, 2004.
16. *C. Reise, T. Erge, „Solar Irradiation and Energy Yields for Photovoltaic Systems in Kuala Lumpur", Fraunhofer ISE, January 2002 (on behalf of NLCC Architects)*
17. J. Wen and T. F. Smith, "Absorption of solar energy in a room," Sol. Energy, vol. 72, no. 4, pp. 283-297, 2002.
18. H. Gabler, F.H. Klotz, H.-D. Mohring, *Ertragspotenzial nachgeführter PV in Europa: Anspruch und Wirklichkeit*, 20. Symposium PV Solarenergie, Bad Staffelstein, March 2005.
19. T. Huld, M. Súri, T. Cebecauer, E. Dunlop, *Optimal Mounting Strategy for Single-Axis Tracking Non- Concentrating PV in Europe*, 23rd PVSEC, Valencia, September 2008.
20. Solarserver, Solarreport January 2008, http://www.solarserver.de
21. T. F. Wu, Y. K. Chen, and C. H. Chang, Power Provision and Illumination of Solar Light, Chuan Hwa Science & Technology Book CO., LTD, 2007.
22. C. C. Chuang, Solar Energy Engineering-Solar Cells, Chuan Hwa Science & Technology Book CO., LTD, 2007.
23. R. A. Messenger, J. Ventre, *Photovoltaic Systems Engineering*, CRC Press, 2003.
24. G.N. Tiwari, *Solar Energy, chapter 1: Fundamentals*, Alpha Science Int. Ltd., Pangbourne, 2002.
25. Y. Goswami, F. Kreith, and J. Kreder, *Principles of Solar Engineering, chapter 2: Fundamentals of Solar Radiation*, Taylor & Francis, Philadelphia, 1999.
26. M.J. Clifford and D. Eastwood, "Design of a Novel Passive Tracker", in *Solar Energy*, vol. 77 (3), 2004, pp. 269-280.
27. S. Abdallah and S. Nijmeh, "Two-Axis Sun Tracking with PLC Control", in *Energy Conversion and Management*, vol. 45, 2004, pp. 1931-1939.

28. A. Canova, L. Giaccone, and F. Spertino, “Sun Tracking for Capture Improvement”, *22nd European Photovoltaic Solar Energy Conference - EUPVSEC,* WIP Renewable Energies, Milano, pp. 3053-3058, September 2007.
29. H.D. Mohring, F. Klotz, and H. Gabler, “Energy Yield of PV Tracking Systems”, *21st European Photovoltaic Solar Energy Conference - EUPVSEC,* WIP Renewable Energies, Dresden, pp. 2691-2694, September 2006.
30. L. Narvarte and E. Lorenzo, “Tracking Gains and Ground Cover Ratio", *22nd European Photovoltaic Solar Energy Conference - EUPVSEC,* WIP Renewable Energies, Milano, pp. 3153-3156, September 2007.
31. M. Meliß, *Regenerative Energiequellen: Praktikum, chapter 1: Berechnung von Sonnenstand und Strahlung*, Springer - Verlag, Berlin, 1997.
32. www.rimlifegreentech.com
33. K. Scharmer and J. Greif, *The European Solar Radiation Atlas*, l’Ecole des Mines, Paris, 2000.
34. F.S. Tymvios, C.P. Jacovides, and C. Scouteli, “Comparative Study of Angstrom’s and Artificial Neural Networks’ Methodologies in Estimating Global Solar Radiation”, in *Solar Energy*, vol. 78 (6), 2005, pp. 752-762.
35. M. Salmi, M. Chegaar, and P. Mialhe, “Modèles d'Estimation de l'Irradiation Solaire Globale au Sol”, in *Revue Internationale d'Heliotechnique Energie - Environment*, no. 35, 2007, pp. 19-24.
36. I. Reda and A. Andreas, “Solar Position Algorithm for Solar Radiation Applications”, in *Solar Energy*, vol. 76 (5), 2004, pp. 577-589.
37. P. Baltas, M. Tortoreli, and P. Russel, “Evaluation of Power Output for Fixed and Step Tracking Photovoltaics”, in *Solar Energy*, vol. 37 (2), 1986, pp. 147-163.
38. F. Dobon, A. Lugo, and J. Monedero, “First Results of the Tetra-Track System and Control Electronics”, *3rd World Conference on Photovoltaic Energy Conversion,* IEEE & WCPEC-3, Osaka, pp. 2050-2053, 2003.
39. K. Karimov, M. Saqib, and P. Akhter, “A Simple Photovoltaic Tracking System”, in *Solar Energy: Materials and Solar Cells*, vol. 87 (1-4), 2005, pp. 49-59.

40. C. Alexandru and M. Comsit, “Virtual Prototyping of the Solar Tracking Systems”, *International Conference on Renewable Energy and Power Quality - ICREPQ*, European Association for the Development of Renewable Energy and Power Quality, Sevilla, pp. 41-42, March 2007.
41. P. Roth, A. Georgiev, and H. Boudinov, “Design and Construction of a System for Sun-Tracking”, in *Renewable Energy*, vol. 29 (3), 2004, pp. 393-402.
42. McFee, R.H. Power collection reduction by mirror surface nonflatness and tracking error for a central receiver solar power system. *Appl. Opt.* 1975, *14*, 1493-1502.
43. Carden, P.O. Steering a field of mirrors using a shared computerbased controller. *Sol. Energ.* 1977, *20*, 343-355.
44. Hughes, R.O. Effects of tracking errors on the performance of point focusing solar collectors. *Sol. Energ.* 1980, *24*, 83-92.
45. Semma, R. P.; Imamura, M. S. Sun tracking controller for multi-kW photovoltaic concentrator system. In *Proceedings of the 3rd International Photovoltaic Sol Energy Conf,* Cannes, France, Oct. 27-31, 1980.
46. Badescu, V. Une evaluation probabiliste pour l’erreur d’orientation des heliostats. *Rev. Phys. Appl.* 1982, *17*, 421-434.
47. Badescu, V. Theoretical derivation of heliostat tracking error distribution. *Sol. Energ.* 2008, *82*, 1192-1197.
48. Akhmedyarov, K.A.; Bazarov, B.A.; Ishankuliev. B.; Karshenas, K.E.; Schaimerdangulyev, G. Economic efficiency of the FV-500 solar photoelectric station with automatic tracking of the sun. *Appl. Solar Energ.* 1986, *22*, 44-47.
49. Maish, A. B. Performance of a self-aligning solar array tracking controller. In *Proceedings of the IEEE Photovoltaic Specialists Conference,* Kissimimee, FL, USA, May 21-25, 1990.
50. Agarwal, A.K. Two axis tracking system for solar concentrators. *Renew. Energ.* 1992, *2*, 181-182.
51. Enslin, J.H.R. Maximum power point tracking: a cost saving necessity in solar systems. *Renew. Energ.* 1992, *2*, 543-549.

52. Brown, D.G.; Stone, K.W. High accuracy/low cost tracking system for solar concentrators using a neural network. In *Proceedings of the 28th Intersociety Energy Conversion EngineeringConference*, Atlanta, GA, USA, Aug. 8-13, 1993.
53. Kalogirou, S.A. Design and construction of a one-axis sun-tracking system. *Sol. Energ.* 1996, *57,* 465-469.
54. Stone, K.W.; Sutherland J.P. Solar two heliostat tracking performance. In *Proceedings of International Solar Energy Conference,* Washington DC, USA, Apr. 27-30, 1997.
55. Hua, C; Shen, C. Comparative study of peak power tracking techniques for solar storage system. In *Proceedings of IEEE Applied Power Electronics Conference and Exposition,* Anaheim, CA, *USA*, Feb. 15-19, 1998.
56. Khalifa A.N.; Al-Mutawalli S.S. Effect of two-axis sun tracking on the performance of compound parabolic concentrators. *Energ. Convers. Manage.* 1998, *39*, 1073-1079.
57. Yousef, H.A. Design and implementation of a fuzzy logic computer-controlled sun tracking system. In *Proceedings of IEEE International Symposium on Industrial Electronics,* Bled, Slovenia, Jul. 12-16, 1999.
58. Kim, T.Y.; Ahn, H.G.; Park, S.K.; Lee, Y.K. A novel maximum power point tracking control for photovoltaic power system under rapidly changing solar radiation. In *Proceedings of ISIE,* Pusan, Korea, Jun. 12-16, 2001.
59. Falbel, G.; Puig-Suari, J.; Peczalski, A. Sun oriented and powered, 3 axis and spin stabilized cubesats. In *Proceedings of IEEE Aerospace Conference,* Big Sky, MT, USA, Mar. 9-16, 2002.
60. Urbano, J.A.; Matsumoto, Y; Asomoxa, R. 5 Wp PV module-based stand-alone solar tracking system. In *Proceedings of 3rd World Conference on Photovoltaic Energy Conversion,* Osaka, Japan, May 11-18, 2003.
61. Jiang, W.; Cao, M. Emulation sunflower. In *Proceedings of the International Symposium on Test and Measurement,* Shenzhen, China, Jun. 1-5, 2003.
62. Luque-Heredia, I.; Martin, C.; Mananes, M.T.; Moreno, J.M., Auger, J.L.; Bodin, V.; Alonso, J.; Diazr, V.; Sala, G. A subdegree precision sun tracker for 1000X

microconcentrator modules, In *Proceedings of the 3rd World Conference on Photovoltaic Energy Conversion,* Osaka, Japan, May 11-18, 2003.

63. Roth, P.; Georgieg, A.; Boudinov, H. Design and construction of a system for sun-tracking, *Renew. Energ.* 2004, *29*, 393-402.
64. Georgiev, A.; Roth, P.; Olivares, A. Sun following system adjustment at the UTFSM. *Energ. Convers. Manage.* 2004, *45*, 1795-1806.
65. Berenguel, M.; Rubio, F.R.; Valverde, A.; Lara, P.J.; Arahal, M.R.; Camacho, E.F.;López, M. An artificial vision-based control system for automatic heliostat positioning offset correction in a central receiver solar power plant. *Sol. Energ.* 2004, *76*, 563-575.
66. Abdallah, S. The effect of using sun tracking systems on the voltage-current characteristics and power generation of flat plate photovoltaics. *Energ. Convers. Manage.* 2004, *45*, 1671-1679.
67. Al-Mohamad, A. Efficiency improvements of photo-voltaic panels using a sun-tracking system. *Appl. Energ.* 2004, *79*, 345-354.
68. Aiuchi, K.; nakamura, M.; Yoshida, K.; Katayama, Y.; nakamura, K. Sun tracking photo-sensor for solar thermal concentrating system. In *Proceedings of International Solar Energy Conference,* Portland, OR, USA, Jul. 11-14, 2004.
69. Alata, M.; Al-Nimr, M.A.; Qaroush, Y. Developing a multipurpose sun tracking system using fuzzy control. *Energ. Convers. Manage.* 2005, *46*, 1229-1245.
70. Kim, I.S. Robust maximum power point tracker using sliding mode controller for the three-phase grid-connected photovoltaic system. *Sol. Energ.* 2007, *81*, 405-414.
71. McCluney, R. Passive optical solar tracking system. *Appl. Optics* 1983, *22*, 3433-3439.
72. Al-Naima, F.M.; Yaghobian, N.A. Design and construction of a solar tracking system. *Solar Wind Technol.* 1990, *7*, 611-617.
73. Lorenz, W. Design guidelines for a glazing with a seasonally dependent solar transmittance. *Sol. Energ.* 1998, *63*, 79-96.
74. Blanco-Muriel, M.; Alarcón-Padilla, D.C.; López-Moratalla, T.; Lara-Coira, M. Computing the solar vector. *Sol. Energ.* 2001, *70*, 431-441.

75. Michalsky, J. J. The asreonomical almanac's algorithm for approximate solar position. *Sol. Energ.* 1988, *40*, 227-235.
76. Beshears, D. L.; Capps, G. J.; Earl. D. D.; Jordan, J. K.; Maxey, L. C.; Muhs, J. D.; Leonard, T. M. Tracking systems evaluation for the Hybrid Lighting System. In *Proceedings of International Solar Energy Conference*, Kohala Coast, HI, USA, Mar. 15-18, 2003.
77. Abdallah, S.; Nijmeh, S. Two axes sun tracking system with PLC control. *Energ. Convers. Manage.* 2004, *45*, 1931-1939.
78. Reda, I.; Andreas, A. Solar position algorithm for solar radiation applications. *Sol. Energ.* 2004, *76*, 577-589.
79. Chen, F.; Feng, J.; Hong, Z. 2006 Digital sun sensor based on the optical vernier measuring principle. *Meas. Sci. Technol.* 2006, *17*, 2494-2498.
80. Chen, F.; Feng, J. Analogue sun sensor based on the optical nonlinear compensation measuring principle. *Meas. Sci. Technol.* 2007, *18*, 2111-2115.
81. Grena, R. An algorithm for the computation of the solar position. *Sol. Energy* 2008, *82*, 462-470.
82. Chen, Y.T.; Lim, B.H.; Lim, C.S. General sun tracking formula for heliostats with arbitrarily oriented axes. *J. Sol. Energ. Eng.* 2006, *128*, 245-250.
83. Chen, Y.T.; Chong, K.K.; Lim, C.S. Report of the first prototype of non-imaging focusing heliostat and it's application in high temperature solar furnace. *Sol. Energ.* 2002, *72*, 531-544.
84. Chong, K.K.; Siaw, F.L.; Wong, C.W.; Wong, G.S. Design and construction of non-imaging planar concentrator for concentrator photovoltaic system. *Renewab. Energ.* 2009, *34*, 1364-1370.
85. Chong, K.K.; Wong, C.W. General formula for on-axis sun tracking system and its application in improving tracking accuracy of solar collector. *Sol. Energ.* 2009, *83*, 298-305.
86. F. Rubio, M. Ortega, and F. Gordillo, "Application of New Control Strategy for Sun Tracking", in *Energy Conversion and Management*, vol. 48 (7), 2007, pp. 2174-2184.

APPENDIX A

PROGRAM TO LINK WITH ADVANTECH DAQ DRIVER

```
//------------------------DIGITAL OUTPUT-------------------------------------//

extern LONG lDriverHandle;
extern PT_DioWritePortByte      ptDioWritePortByte;
extern PT_DioGetCurrentDOByte   ptDioGetCurrentDOByte;
extern char szErrMsg[80];
extern LRESULT ErrCde;
USHORT uDoValue = 0x00;

//-----------------------DIGITAL INPUT----------------------------------------//

static      PT_DioReadPortByte  ptDioReadPortByte;  // structure for DioReadPortByte
table
static      DEVFEATURES     DevFeatures;    // structure for device features
static      LONG   DriverHandle = (LONG)NULL;        // driver handle
static      PT_DeviceGetFeatures  ptDevFeatures;
static      PT_DioSetPortMode  ptDioPortMode;
USHORT      gwDevice = 0, gwSubDevice = 0;      // Device index
SHORT       gnNumOfDevices, gnNumOfSubdevices;  // number of installed devices

//--------------------------PULSE---------------------------------------------//

PT_CounterPulseStart ptCounterPulseStart;
extern PT_CounterPulseStart ptCounterPulseStart;
```

```
//-----------------------------------------------------------------------//
//                      VARIABLES                              //
//-----------------------------------------------------------------------//

const SHORT     MAX_DEVICES = 100;

LONG lDriverHandle;                                     //Driver handle
PT_DioWritePortByte      ptDioWritePortByte;                  // DioWriteByte table
PT_DioGetCurrentDOByte   ptDioGetCurrentDOByte;
char        szErrMsg[80];                               // used for MESSAGEBOX function
LRESULT     ErrCde;                                     // return error code

DEVLIST     DeviceList[MAX_DEVICES];                         //Device list array
DEVLIST     SubDeviceList[MAX_DEVICES];                       //Sub Device list array
SHORT       OutEntries;
USHORT uDoReadValue = 0x00;
//-----------------------------------------------------------------------//
//-----------------------------------------------------------------------//

__fastcall TfrmDigout::TfrmDigout(TComponent* Owner)
       : TForm(Owner)
{
}
//---------------------------------------------------------------------
void __fastcall TfrmDigout::FormCreate(TObject *Sender)
{
//----------------------------------------------------------------------//
//     Get the list of current Advantech I/O Devices in your system        //
//----------------------------------------------------------------------//
   bOpen = False;
   ErrCde = DRV_DeviceGetNumOfList(&gnNumOfDevices);
   if(ErrCde != SUCCESS)
   {
      DRV_GetErrorMessage(ErrCde,(LPSTR)szErrMsg);
      Application->MessageBoxA((LPCSTR)szErrMsg, "Driver Message");
      return;
   }

   if(gnNumOfDevices > MAX_DEVICES)
   {
      gnNumOfDevices = MAX_DEVICES;
   }
   ErrCde = DRV_DeviceGetList((DEVLIST far *)DeviceList, gnNumOfDevices,
&OutEntries);
   if(ErrCde != SUCCESS){
      DRV_GetErrorMessage(ErrCde,(LPSTR)szErrMsg);
      Application->MessageBoxA((LPCSTR)szErrMsg, "Driver Message");
      return;
   }

   for(int i=0; i<gnNumOfDevices; i++){
      lstDevice->Items->Add(AnsiString(DeviceList[i].szDeviceName));
   }
   lstDevice->ItemIndex = 0;

   lstDeviceClick(Sender);
}
//---------------------------------------------------------------------------
void __fastcall TfrmDigout::lstDeviceClick(TObject *Sender)
{
//--------------------------add by mining---------------------------------//
   PT_DioWriteBit ptDioWriteBit;
```

```
    ptDioWriteBit.port=0;
    ptDioWriteBit.bit=0;

//---------------------------end by mining---------------------------------//
//----------------------Get the port of each device------------------------//

    AnsiString strPort("Port: ");
    lstChannel->Clear();
    gnNumOfSubdevices = DeviceList[lstDevice->ItemIndex].nNumOfSubdevices;
    if(gnNumOfSubdevices > MAX_DEVICES){
        gnNumOfSubdevices = MAX_DEVICES;
    }
    if(gnNumOfSubdevices >0){
        ErrCde = DRV_DeviceGetSubList(DeviceList[lstDevice->ItemIndex].dwDeviceNum, (DEVLIST far*)SubDeviceList,gnNumOfSubdevices, &OutEntries);
        if (ErrCde != SUCCESS){
            DRV_GetErrorMessage(ErrCde, (LPSTR)szErrMsg);
            Application->MessageBoxA((LPCSTR)szErrMsg, "Driver Message");
            return;
        }
        for(int j=0; j<gnNumOfSubdevices; j++){

        }
    }else{

    }
    if(0 == gnNumOfSubdevices){
        lstChannel->Enabled = True;

        if(True == bOpen){
            DRV_DeviceClose((LONG far* )&lDriverHandle);                //close the old
opened Device
            bOpen = False;
        }

        ErrCde = DRV_DeviceOpen(DeviceList[lstDevice->ItemIndex].dwDeviceNum,
(LONG far*)&lDriverHandle);
        if(ErrCde != SUCCESS){
            DRV_GetErrorMessage(ErrCde,(LPSTR)szErrMsg);
            Application->MessageBox((char *)szErrMsg, "Driver Message", MB_OK);
            return;
        }
        bOpen = True;
        ptDevFeatures.buffer = (LPDEVFEATURES)&DevFeatures;
        ptDevFeatures.size = sizeof(DEVFEATURES);
        if ((ErrCde = DRV_DeviceGetFeatures(lDriverHandle,
            (LPT_DeviceGetFeatures)&ptDevFeatures)) != SUCCESS){
            DRV_GetErrorMessage(ErrCde,(LPSTR)szErrMsg);
            DRV_DeviceClose((LONG far *)&lDriverHandle);
            bOpen = False;
            Application->MessageBox((char *)szErrMsg, "Driver Message", MB_OK);
            return;
        }

//-------------------------add by mining-------------------------------------//
//------------judge whether this device supports DO function------------------//

        if(ptDevFeatures.buffer->usMaxDOChl==0)
        {
            ErrCde = DRV_DioWriteBit(lDriverHandle,&ptDioWriteBit);
            DRV_GetErrorMessage(ErrCde,(LPSTR)szErrMsg);
            DRV_DeviceClose((LONG far *)&lDriverHandle);
            bOpen = False;

            Application->MessageBox((char *)szErrMsg, "Driver Message", MB_OK);
            return;

        }
        else
        {

        }

//------------------------end by mining--------------------------------------//

        for(int i=0; i<(DevFeatures.usMaxDOChl+7)/8; i++){
            strPort = "Port: ";
            strPort = strPort + AnsiString(i);
            lstChannel->Items->Add(strPort);
        }
        lstChannel->ItemIndex = 0;
    }

}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::cmdExitClick(TObject *Sender)
{
Close();
}
```

APPENDIX B

PROGRAM FOR DUAL AXIS SOLAR TRACKING OPERATION

```
int StartCount()
{
        for (x=0;x<100000000000;x++)
        {
                Sleep (1000);
        }
        return 0;
}

void __fastcall TfrmDigout::Button1Click(TObject *Sender)
{
double N;
        SYSTEMTIME Local_time;
        GetLocalTime(&Local_time);
        day = Local_time.wDay;
        month = Local_time.wMonth;
        start_hour=Edit1->Text.ToInt();
        start_min=Edit2->Text.ToInt();
        Timer1->Enabled=true;
        Panel10->Caption=Local_time.wYear;
        Panel12->Caption=Local_time.wDay;
            if (month==1)
                {
                Panel11->Caption="January";
                N=day;
```

```
    }
else if (month==2)
    {
    Panel11->Caption="February";
    N=31+day;
    }
else if (month==3)
    {
    Panel11->Caption="March";
    N=31+28+day;
    }
else if (month==4)
    {
    Panel11->Caption="April";
    N=31+28+31+day;
    }
else if (month==5)
    {
    Panel11->Caption="May";
    N=31+28+31+30+day;
    }
else if (month==6)
    {
    Panel11->Caption="June";
    N=31+28+31+30+31+day;
    }
 else if (month==7)
    {
    Panel11->Caption="July";
    N=31+28+31+30+31+30+day;
    }
```

```
     else if (month==8)
        {
        Panel11->Caption="August";
        N=31+28+31+30+31+30+31+day;
        }
     else if (month==9)
        {
        Panel11->Caption="September";
        N=31+28+31+30+31+30+31+31+day;
        }
     else if (month==10)
        {
        Panel11->Caption="October";
        N=31+28+31+30+31+30+31+31+30+day;
        }
    else if (month==11)
        {
        Panel11->Caption="November";
        N=31+28+31+30+31+30+31+31+30+31+day;
        }
      else if (month==12)
        {
        Panel11->Caption="December";
        N=31+28+31+30+31+30+31+31+30+31+30+day;
        }
Panel6->Caption=N;
hour = Local_time.wHour;
min = Local_time.wMinute;
sec = Local_time.wSecond;
Panel7->Caption=hour;
Panel8->Caption=min;
```

```
     Panel9->Caption=sec;
     Panel13->Caption=lati;
     Panel14->Caption=longi;
     a = (284 + N)*360/365;
     X = 360*(N - 81)/365.2422;
     CreateThread(0,0,(LPTHREAD_START_ROUTINE)&StartCount,0,0,0);
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer4Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x01;

ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Panel15->Caption="Motor 1 paused";
Timer4->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer3Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x01;
ptDioWritePortByte.state = 0x01;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
```

```
Timer3->Enabled=false;
Timer4->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer7Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x04;
ptDioWritePortByte.state = 0x01;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer7->Enabled=false;
Timer8->Enabled=true;
}
//---------------------------------------------------------------------------
void __fastcall TfrmDigout::Timer9Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x08;
ptDioWritePortByte.state = 0x01;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer9->Enabled=false;
Timer10->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer8Timer(TObject *Sender)
```

```
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x04;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Panel16->Caption="Motor 2 paused";
Timer8->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer10Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x08;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Panel16->Caption="Motor 2 paused";
Timer10->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer5Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x02;
ptDioWritePortByte.state = 0x01;
```

```
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer5->Enabled=false;
Timer6->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer6Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x02;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Panel15->Caption="Motor 1 paused";
Timer6->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer1Timer(TObject *Sender)
{
SYSTEMTIME Local_time;
GetLocalTime(&Local_time);
decl = 23.45*sin(a*pi/180);
decl2 = decl*pi/180;
lati2 = lati*pi/180;
X2 = X*pi/180;
T = 12 + (0.1236*sin(X2)-0.0043*cos(X2))+(0.1583*sin(2*X2)+0.0608*cos(2*X2));
lon = (120 - longi)/15;
```

```
Tc2 = hour+0.016666666666667*min+0.00027777777777778*sec;
Tc = Tc2 + (0.000277777777777778)*x;
Ts = Tc + (T/60) - lon;
H = (Ts - 12)*15;
H2 = H*pi/180;
elev = asin(cos(lati2)*cos(decl2)*cos(H2)+sin(lati2)*sin(decl2));
elev2 = elev*180/pi;
azim = -sin(H2)*cos(decl2)/cos(elev);
if (azim>1)
          {
          azim = 2 - azim;
          }
else if (azim<-1)
        {
         azim = -(2+azim);
        }

if (azim<-0.9999)
        }
        azim2 = asin(azim);
        }
else if ( azim>0.0)
        {
        azim2 = (pi/2) + ((pi/2) - asin(azim));
        }
else if ( azim<0.0)
        {
        azim2 = -(pi/2) - ((pi/2) + asin(azim));
        }
else
        {
          azim2 = asin(azim);
          }
azim3 = azim2*180/pi;
Panel1->Caption="Standby position!";
Panel2->Caption="Standby position!";
Panel3->Caption=Local_time.wHour;
Panel4->Caption=Local_time.wMinute;
Panel5->Caption=Local_time.wSecond;
if (Local_time.wHour>=start_hour && Local_time.wMinute>=start_min)
          {
          Panel1->Caption=elev2;
          Panel2->Caption=azim3;
          start_hour=0;
          start_min=0;
if (Panel3->Caption==Panel7->Caption && Panel4->Caption==Panel8->Caption &&
Panel5->Caption==sec+1)
                    {
                    old_elev=elev2;
                    Panel17->Caption=old_elev;
                    old_azim=azim3;
                    Panel18->Caption=old_azim;
                    Timer2->Enabled=true;
                    }
          }
else if (Local_time.wHour>start_hour)
          {
          Panel1->Caption=elev2;
          Panel2->Caption=azim3;
          start_hour=0;
          start_min=0;
```

```
if (Panel3->Caption==Panel7->Caption && Panel4->Caption==Panel8->Caption &&
Panel5->Caption==sec+1)
                    {
                    old_elev=elev2;
                    Panel17->Caption=old_elev;
                    old_azim=azim3;
                    Panel18->Caption=old_azim;
                    Timer2->Enabled=true;
                    }
          }
if (Timer2->Enabled==false)
          {
          Panel15->Caption="Motor stops!";
          }
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer2Timer(TObject *Sender)
{
if(elev2 >= old_elev + 0.02)
          {
          Timer3->Enabled=true;
          old_elev=elev2;
          Panel17->Caption=old_elev;
          Panel15->Caption="Motor 1 rotating clockwise!";
          }
else if(elev2 <= old_elev - 0.02)
          {
          Timer5->Enabled=true;
          old_elev=elev2;
          Panel17->Caption=old_elev;
```

```
            Panel15->Caption="Motor 1 rotating counter-clockwise!";
            }
else if(azim3 >= old_azim + 0.01)
            {
            Timer7->Enabled=true;
            old_azim=azim3;
            Panel18->Caption=old_azim;
            Panel16->Caption="Motor 2 rotating clockwise!";
            }
if(azim3 <= old_azim - 0.01)
            {
            Timer9->Enabled=true;
             old_azim=azim3;
            Panel18->Caption=old_azim;
            Panel16->Caption="Motor 2 rotating counter-clockwise!";
            }
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Button2Click(TObject *Sender)
{
Timer1->Enabled=false;
Timer2->Enabled=false;
}
//---------------------------------------------------------------------------
```

APPENDIX C

PROGRAM FOR INITIALIZATION OF THE SOLAR TRACKER

```
double a,X,N,decl,decl2,lati2,X2,T,lon,Tc,Ts,H,H2,elev,elev2,azim,azim2,azim3, pi =
3.1416,lati =3.533333,longi=103.466667,azim4,elev3,azim_ex,elev_ex, time1,time2;
int month,day,hour,min;

void __fastcall TfrmDigout::Button1Click(TObject *Sender)
{
hour = Edit3->Text.ToInt();
min = Edit4->Text.ToInt();
day = Edit2->Text.ToInt();
month = Edit1->Text.ToInt();
        if (month==1)
        {
        Panel3->Caption="January";
        N=day;
        }
        else if (month==2)
        {
        Panel3->Caption="February";
        N=31+day
        }
        else if (month==3)
        {
        Panel3->Caption="March";
        N=31+28+day;
        }
```

```
else if (month==4)
{
Panel3->Caption="April";
N=31+28+31+day;
}
else if (month==5)
{
Panel3->Caption="May";
N=31+28+31+30+day;
}
else if (month==6)
{
Panel3->Caption="June";
N=31+28+31+30+31+day;
}
else if (month==7)
{
Panel3->Caption="July";
N=31+28+31+30+31+30+day;
}
else if (month==8)
{
Panel3->Caption="August";
N=31+28+31+30+31+30+31+day;
}
else if (month==9)
{
Panel3->Caption="September";
N=31+28+31+30+31+30+31+31+day;
}
else if (month==10)
```

```
        {
        Panel3->Caption="October";
        N=31+28+31+30+31+30+31+31+30+day;
        }
        else if (month==11)
        {
        Panel3->Caption="November";
        N=31+28+31+30+31+30+31+31+30+31+day;
        }
        else if (month==12)
        {
        Panel3->Caption="December";
        N=31+28+31+30+31+30+31+31+30+31+30+day;
        }
azim_ex=Edit5->Text.ToDouble();
elev_ex=Edit6->Text.ToDouble();
a = (284 + N)*360/365;
X = 360*(N - 81)/365.2422;
decl = 23.45*sin(a*pi/180);
decl2 = decl*pi/180;
lati2 = lati*pi/180;
X2 = X*pi/180;
T = 12 + (0.1236*sin(X2)-0.0043*cos(X2))+(0.1583*sin(2*X2)+0.0608*cos(2*X2));
lon = (120 - longi)/15;
Tc = hour+0.016666666666667*min;
Ts = Tc + (T/60) - lon;
H = (Ts - 12)*15;
H2 = H*pi/180;
elev = asin(cos(lati2)*cos(decl2)*cos(H2)+sin(lati2)*sin(decl2));
elev2 = elev*180/pi;
azim = -sin(H2)*cos(decl2)/cos(elev);
```

```
if (azim>1)
{
        azim = 2 - azim;
}
else if (azim<-1)
{
        azim = -(2+azim);
}
azim2 = asin(azim);
azim3 = azim2*180/pi;
azim4 = azim3 - azim_ex;
elev3 = elev2 - elev_ex;
Panel1->Caption = azim4;
Panel2->Caption = elev3;
Panel4->Caption = day;
Panel5->Caption = hour;
Panel6->Caption = min;
time1 = elev2/0.001;
time2 = azim4/0.002;
Panel7->Caption = time1;
Panel8->Caption = time2;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer1Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x01;
ptDioWritePortByte.state = 0x01;
```

```
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer1->Enabled=false;
Timer2->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer2Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x01;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer2->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer4Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x02;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer4->Enabled=false;
}
//---------------------------------------------------------------------------
```

```
void __fastcall TfrmDigout::Timer3Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x02;
ptDioWritePortByte.state = 0x01;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer3->Enabled=false;
Timer4->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer5Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x04;
ptDioWritePortByte.state = 0x01;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer5->Enabled=false;
Timer6->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer7Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x08;
ptDioWritePortByte.state = 0x01;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer7->Enabled=false;
Timer8->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer6Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x04;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer6->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Timer8Timer(TObject *Sender)
{
ptDioWritePortByte.port = 0;
ptDioWritePortByte.mask = 0x08;
ptDioWritePortByte.state = 0x00;
ErrCde =
DRV_DioWritePortByte(lDriverHandle,(LPT_DioWritePortByte)&ptDioWritePortByte
);
Timer8->Enabled=false;
}
```

```
//---------------------------------------------------------------------------

void __fastcall TfrmDigout::Button2Click(TObject *Sender)
{
if(elev3 > 0)
            {
            Timer1->Enabled=true;
            }
else if(elev3 < 0)
            {
            Timer5->Enabled=true;
            }
else if(azim4 > 0)
            {
            Timer3->Enabled=true;
            }
if(azim3 < 0)
            {
            Timer5->Enabled=true;
            }
}
//---------------------------------------------------------------------------
```