
Masters Theses

Student Theses and Dissertations

Fall 2012

Stochastic model for solar sensor array data

Faris Alfaris

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses



Part of the [Electrical and Computer Engineering Commons](#)

Department:

Recommended Citation

Alfaris, Faris, "Stochastic model for solar sensor array data" (2012). *Masters Theses*. 6947.
https://scholarsmine.mst.edu/masters_theses/6947

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

STOCHASTIC MODEL FOR SOLAR SENSOR ARRAY DATA

by

FARIS ALFARIS

A THESIS

Presented to the Faculty of the Graduate School of the
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2012

Approved by

Dr. J. W. Kimball, Advisor
Dr. M. Ferdowsi
Dr. M. Crow

ABSTRACT

Statistical approaches are often used in time series analysis, for example, to predict the future trend of a time series. Trend forecasting can be applied in many time related parameters such as: solar radiation, generation of electricity and other variables related to time series to improve the efficiency and to set the design requirements. Since the design of any solar energy system requires knowledge of the availability of solar radiation data at the location of interest. Therefore, this research seeks the application of a statistical model to fit the solar radiation time series and predict the future values. There are various methods used to estimate the hourly global solar radiation on the earth surface. However; in this research Meinel and Meinel model was used based on its fit accuracy relying on mean bias error (MBE) and root mean square error (RMSE) tests. The study concerns to two main goals: First, predicting the future produced power of a given solar panel in a series-parallel configuration based on the present data and weather condition in order to improve the performance of the solar panel. Second, there was an attempt to relate all 24 sensors that located on a solar panel so that we can estimate the sun radiation at each part of a hypothetical solar array using one sensor's reading only. In addition, as the availability of the solar radiation related with the climate conditions, an attempt has been made to correct the predicted data under different climate conditions. Linear regression method was used for the purpose of fitting the next point in predicting of the solar radiation, while the covariance, correlation factors and slopes among all sensors are used to relate the whole parts of the panel. The proposed technique yields an acceptable result within an average mean squared error (MSE) of 2.6%.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my research advisor, Dr. Jonathan Kimball, for the valuable guidance and advice. He inspired me greatly to work in this project. His willingness to motivate me contributed tremendously to my project. Also, I would like to thank for providing all needed facilities and related papers. This work would not have been done without his efforts.

Deepest gratitude is also due to the members of the supervisory committee, Dr. M. Crow and Dr. M. Ferdowsi, for serving as members on my thesis committee. Special thanks also to Dr. C. Elmore for his help in this work.

Financial assistance from Saudi Arabian Cultural Mission (SACM) in the form of research assistantship is gratefully acknowledged. Also, I would like to take this opportunity to express my heartfelt gratitude and sincere appreciation to King Saud University who afforded me a full scholarship for master's degree, so I would like to gratefully acknowledge their supporting and confidence in me.

I extend a very special acknowledgement to my parents for their unending words of support and motivation. I will never forget there prays for me, and I am promising them to keep my work at the highest level until there dream of me comes truth. Last but not the least; I dedicate this thesis to my wife, Ghadeer Alrabiah, who always helped me and encouraged me to achieve my goals. I love you Ghadeer.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGMENTS.....	iv
LIST OF ILLUSTRATIONS	vii
LIST OF TABLES	ix
SECTION	
1. INTRODUCTION.....	1
1.1 LITERATURE REVIEW.....	2
1.1.1. Theoretical Studies.....	3
2. EXPERIMENTAL SETUP.....	16
2.1. ARRAY IMPLEMENTATION	16
2.2. SOLAR CELLS AND DAQ SYSTEMS	17
2.3. CIRCUIT BOARDS OF THE SENSOR	20
2.3.1. Sensor Improvement.....	21
2.3.2. Sensors Performance Test.....	22
3. SUN RADIATION MODEL AND LINEAR REGRESSION ANALYSIS.....	24
3.1. SOLAR SPECTRUM MODEL	24
3.2. LINEAR REGRESSION METHOD	31
3.3. CORRELATION METHOD AND ARRAY DATA CONSTRUCTION.....	33
4. DATA PROCESSING AND RESULTS	38
4.1. DATA CALIBRATION AND AVERAGING	38
4.2. SOLAR RADIATION MODEL AND ACTUAL DATA FITTING.....	40
4.3. EXPECTING THE FUTURE TREND.....	43
4.4. DATA FILTERING AND NOISE REJECTION	48
4.5. ARRAY DATA CONSTRUCTION.....	49
5. CONCLUSIONS.....	53
APPENDIX.....	54

REFERENCES..... 73
VITA 75

LIST OF ILLUSTRATIONS

Figure	Page
1.1 (a) Actual and Smoothed, (b) 3-Point Moving Average Series	9
2.1 Schematic Diagram of the Sensor Array.....	17
2.2 Solar Cell Sample. (a) New Solar Cell. (b) Old Solar Cell.....	18
2.3 Plastic Box to Cover the Sensor and the Solar Cell.....	18
2.4 Single Data Acquisition Systems with 8 Inputs.....	19
2.5 Schematic Diagram of the Sensor Circuit Board.....	21
2.6 Actual Sensor Circuit Board	22
3.1 The Orbit of the Earth and the Declination at Different Times of the Year	25
3.2 Sun Angles, Showing Altitude, Azimuth and Hour Angle	26
3.3 Relationship among Zenith Angle, Latitude and Declination at Solar Noon in Winter.....	27
3.4 (a) Average Monthly, (b) Average Cumulative (kWh/m ²) for January in Rolla, Missouri Using Meinel and Meinel Model	29
3.5 The Relationship among the Sun and the Module Angles for Tilted Solar Panel with $\beta=22.6^\circ$	30
3.6 (a) Ish_5 versus Ish_1, (b) Ish_6 versus Ish_2 to Find the Slopes Between each others	34
3.7 Order of Computation all Sensors Starting from B4.....	37
4.1 (a) Uncalibrated Sinsors Readings, (b) Calibrated Sinsors Readings (using two Points calibration way).....	39
4.2 DAQ readings After Averaging Over One Minute.....	40
4.3 Power-Time Curve of Sun Radiation on April 1 st Using Meinel and Meinel Model .	41
4.4 Power-Time Curve of Sun Radiation on April 1 st Using Meinel and Meinel model (after multiplying by power-to-short circuit current ratio).....	42
4.5 Actual Data and Mathematical Model for April 1 st	42
4.6 Comparison Between Subtracting and Dividing the Mathematical Model with Actual Data.....	43
4.7 Applying the Linear Regression Method on the Difference Between the Actual Data and the Mathematical Model	44

4.8 Applying the Linear Regression Method on the Difference Between the Actual Data and the Mathematical Model (enlarged).....	44
4.9 Adding the Predicted Signal to the Model and the Actual Data for April 1 st	45
4.10 Adding the Predicted Signal to the Model and the Actual Data for April 1 st (enlarged)	45
4.11 (c) MSE Error for the Difference Between the Actual and Expected Data (<i>widow size=3</i>)	46
4.12 Comparison Between Subtracting and Dividing the Mathematical Model with Predicted Data (after applying the linear regression for different window sizes).....	47
4.13 Data After Filtering and Noise Rejection (to be compared with Figure 4.9).....	49
4.14 (a) Actual Data, (b) Predicted Data for April 3 rd	50
4.15 (a) Actual Data, (b) Predicted Data for April 3 rd (enlarged)	51
4.16 Difference Between the Predicted Data and Actual Data and the MSE for Sensor A1	52

LIST OF TABLES

Table	Page
3.1 Order of Correlation Process.....	37
4.1 MSE for different Window Size and Different Days for Subtracting and Division ...	48

1. INTRODUCTION

The solar energy has been used as one of renewable energy sources all over the world especially in United States; However, the reliance on the solar energy was not much as the relying on fossil fuels and hydro generation due to its high cost and low efficiency. Recently, the performance of alternative energy technologies is improving, and photovoltaic (PV) systems are no exception. Therefore, on June 26, 1997, the United states Million Solar Roofs Initiative (MSRI) was announced at the United Nations Special Session on Environment and Development in New York [1]. It was proclaimed that “now we will work with business and communities to use the sun’s energy to reduce our reliance on fossil fuels by installing solar panels on 1 million more roofs around our nation by 2010. Capturing the sun’s warmth can help us to turn down the earth’s temperature” [1]. This emphasis on improving the solar panels took a place in many researches and studies.

One of the most popular configurations of the solar array is series-parallel configuration. The arrangement of panels in series boosts the voltage of an array, and a parallel component increases the current generated by the array [2]. Hence, the needed amount of energy for a certain load can be delivered by the solar array. However, the weather conditions such as Rainy, cloudy or partly cloudy weather, contribute to lowering or increasing the received solar energy. Consequently, the design of any solar energy system requires knowledge of the availability of solar radiation data at the location of interest to set the inverters and all used controlling and storage elements in acceptable rating values. Furthermore, to reserve the surplus produced electrical energy,

where the solar panel could produce higher peak KW in partly cloudy day than in a pure sunny day. In addition, keeping the electronic devices used behind the solar array from the over voltage and current should be taken into account to prevent damages. This can be done by predicting the future trend of the sun. Predicting the next points of a time series trend needs different models and techniques. In this project, Meinel and Meinel model was used to plot the sun radiation envelop during the daytime, while the linear regression has been used to calculate the next value in next minute using last 4 readings (window size=4).

As the used panel is in series-parallel configuration, knowing the performance of each part of the panel by using one sensor located on the panel contributes in lowering the used sensors and data collected and this leads to fasting the processing and lowering the network costs.

In this project we designed a sensor to read the short circuit current, open circuit voltage and load current for a solar cell every two seconds. The panel consists of 24 sensors distributed in 3×8 array. Furthermore, for the objective of gathering and analyzing data, we used Excel program to collect the data that measured by sensors, and MATLAB for analyzing and simulating.

1.1. LITERATURE REVIEW

Numerous studies were conducted during the past few years in an attempt to estimate the hourly global solar radiation using several models and made comparisons among the different models. In term of choosing the best model, many researches placed

much emphasis on some tests such as mean bias error (MBE) and root mean square error (RMSE), where both of these indicators evaluate the accuracy of the tested models using the measured and estimated data. However, some other studies relied on plotted graphs and tables to compare among the actual readings and predicted values and then choosing the best model that fits the solar radiation time series accurately. Other studies put more emphasis on using the linear regression and least square methods in smoothing and estimating the data.

1.1.1. Theoretical Studies. The comparison study in [3] concluded that the Collares-Pereira and Rabl model, as modified by Gueymard (CPRG), yields the best performance for estimating mean hourly global radiation incident on a horizontal surface. The study analyzed seven models (Whillier/Liu and Jordan (WLJ), Collares-Pereira and Rabl (CPR), Collares-Pereira and Rabl, Newell, Jain, Baig et al. and Garg Models). Moreover, the study evaluated the ratio of the mean hourly distribution of global radiation over the average day of each average month, and then two tests (MBE and RMSE) were done to Figure out the best model amongst all models. These tests provide information for the models' performance by allowing a term by term comparison of the actual deviation between the calculated value and measured value on short and long term, respectively.

According to the obtained errors, the CPRG model predicts the best result in comparison to other models for different climate conditions and it is given as:

$$r_{CPR} = (a + b \cos \omega)r_o \quad (1)$$

where a and b are linear function of

$$\sin(\omega_o - \pi/3) \quad (2)$$

and they are given by

$$a = 0.4090 + 0.5016 \sin(\omega_o - 1.047) \quad (3)$$

$$b = 0.6609 - 0.4767 \sin(\omega_o - 1.0467) \quad (4)$$

In addition to the CPRG model, the CPR and WLJ models yield better performance than Newell, Baig and Garg models.

A study in [4] used a new approach that contains two phases to predict the hourly solar radiation series. A new model called novel hybrid was used to combine the Autoregressive and Moving Average (ARMA) model to predict the stationary residual series and the controversial Time Delay Neural Network (TDNN) to do the prediction. In this study, four detrend models were used to detrend the solar radiation (Jain's, Baig's, Kaplanis' and Al-Sadah's model). The study concluded that Al-Sadah's model, Equation(5), yields the best result in fitting accuracy based on two indicators: root mean square error (RMSE) and the normalized root mean square error (NRMSE).

$$r_b = a_1 + a_2 h + a_3 h^2 \quad (5)$$

Consequently, this model was used to detrend the solar radiation series for further prediction. However; one drawback has to be considered in applying Al-Sadah's model which is the difficulty of finding the unknown coefficients that are shown in equation (5).

For this purpose, actual data and least square method were used in this study in order to evaluate the unknown values (a_1 and a_2). In addition to that, the Augmented Dickey–Fuller method was applied to test the stationarity of the residual in order to judge the goodness of different detrending models. The study results showed that the best order for the ARMA model is (1,1) according to the auto correlation and practical correlation. Also, the TDNN model found to be much more sensitive than the ARMA model, but not as stable. Furthermore, novel hybrid can produce better prediction where it captures both advantages of ARMA and TDNN models and gives excellent results.

A study in [5] was conducted by A. Katiyar and C. Pandey to develop a new model to evaluate hourly solar radiation on horizontal surface for different locations having different climatic conditions due to the fact that the availability of solar radiation for any place depends upon the climatic conditions of the locality, as well as the correlation for a place may not be suitable for other places of different climatic conditions. The study carried out in five cities in India for five years period (from January 2001 to December 2005). The study analyzed three detrending models (Liu and Jordan, Singh et al. models). In addition to that, a new model called Present model was developed to be appropriate for Indian climate in different climate conditions. Present model is based on ASHRAE model and is given as:

$$I = I_b + I_d \quad (6)$$

$$I_b = AI_N \cos \theta_z + B \quad (7)$$

$$I_N = C.\exp\left[-\frac{D}{\cos\theta_z}\right] \quad (8)$$

$$I_d = E.I_N + F \quad (9)$$

where I_b , I_d and I_N are hourly beam, hourly diffuse and hourly beam radiation in direction of rays, respectively. The values of A , B , C , D , E and F were calculated using least square regression analysis.

θ_z is the zenith angle, which depends upon the latitude of the location

Moreover, the statistical test methods of MBE, RMSE and t-test were used in order to indicate the performance of the used models and they are given as:

$$MBE = \left[\sum \{ (I)_{i,c} - (I)_{i,m} \} / p \right] \quad (10)$$

$$RMSE = \left[\sum \{ (I)_{i,c} - (I)_{i,m} \}^2 / p \right]^{1/2} \quad (11)$$

$$t = \left[(p-1)MBE^2 / (RMSE^2 - MBE^2) \right]^{1/2} \quad (12)$$

where $(I)_{i,c}$ and $(I)_{i,m}$ are the i^{th} calculated and measured values, respectively. p is the total number of observations. The study results showed that the constants A , B , C , D , E and F were varied due to the variation in the Indian climate characteristics and we can find them based on current climate situation by comparing between the estimated and measured values. In addition to this result, the study investigated that the present correlation was more appropriate than the other correlations for all selected locations.

And for the same objective, an attempt has been made in [6] to estimate the hourly global solar radiation for composite climate. The study was carried out for different climate conditions (Clear day, Hazy day, Hazy and cloudy day and cloudy day). Furthermore, a coefficient correlation was calculated to correct the estimated values from Liu and Jordan, Collares-Pereira and Rabl and Al-Sadah's models. Equation shows the used coefficient correlation.

$$r = \frac{N(\sum X_{\text{exp}} \cdot X_{\text{pre}}) - (\sum X_{\text{exp}}) \cdot (\sum X_{\text{pre}})}{\sqrt{N \cdot (\sum X_{\text{exp}}^2) - (\sum X_{\text{exp}})^2} \sqrt{N \cdot (\sum X_{\text{pre}}^2) - (\sum X_{\text{pre}})^2}} \quad (13)$$

The results of the study showed that the estimated and measured values are in close agreement except for low sun angles. In addition, the study concluded that all the three models can satisfactorily be employed to estimate the hourly global solar for the composite climate within the accuracy limit of 10%.

Since the temperature is an important factor that should have been taken into account in the field of solar cells studies, a study in [7] has been done to show the effect of non-uniform temperature and illumination upon PV cells. The study modeled several profiles (Gaussian illumination, Gaussian temperature and shade profiles) to Figure out the impact of the diversity of the temperature and radiation. The study concluded that increasing in temperature leads to get less open circuit voltage ($\Delta V_{oc} = 20 \text{ mV}$ for $\Delta T = 40\text{K}$). In addition, open circuit voltage and short circuit current profiles improve left-right

as the temperate increases from right to left and the illumination rises from left to right, and it achieves maximum output at uniform radiation and temperature.

In another study [8] an attempt has been made to use pervious data for prediction. Three approaches have been used for this objective (Linear Regression, auto Regression and Auto Regression Integrate Moving Average). In addition, the study used other approaches to smooth the actual and predicted series such as: Moving averages, Kernel, nearest neighbor and the locally weighted regression smoothing method. The results showed that 3-point moving average is the best way to smooth the series, based on its low average difference, followed by the nearest neighbor with slight increase in average difference. Accordingly, 3-point moving average was chosen as the smoothing model. Figure 1.1 compares the original series and series smoothed by 3-point moving average method.

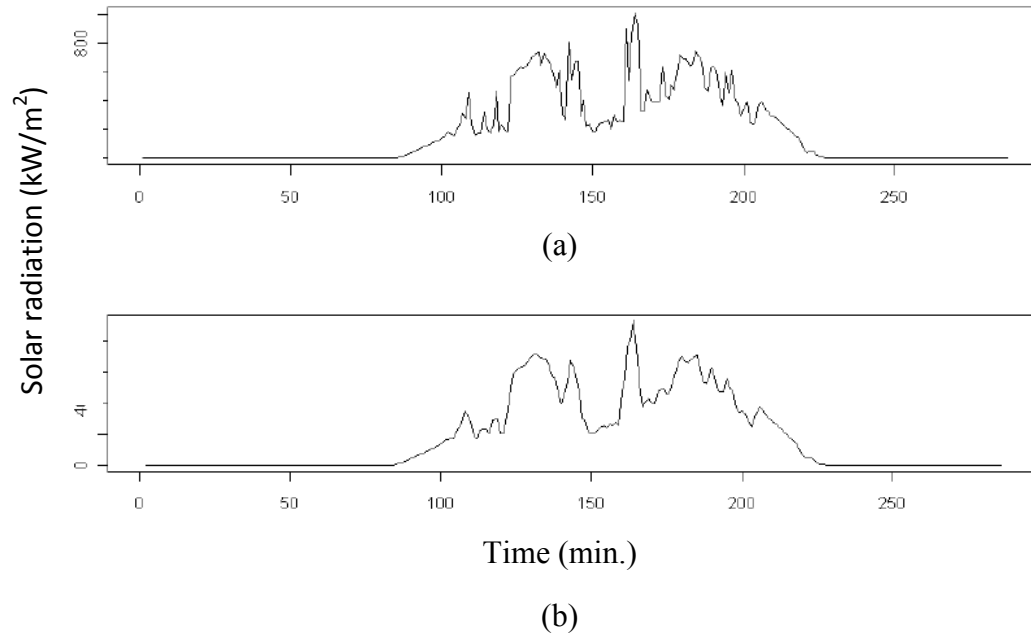


Figure 1.1: (a) Actual and Smoothed, (b) 3-Point Moving Average Series

Furthermore, the study concluded that Linear Regression has better performance than other models such as Auto Regression and Auto Regression Integrate Moving Average. However, the Linear Regression method requires a number of previous data for prediction. Data simulation has been made to show that a list of 10 to 15 past data values yields optimal result.

Several empirical models have been used to calculate solar radiation utilizing available meteorological, climatological, and geographical parameters such as sunshine hours, air temperature, latitude, precipitation, relative humidity, and cloudiness. However; the comparison study in [9] used sunshine duration as the main parameter for estimating the global solar radiation. This study carried out in Turkey for 18 provinces to represent the different weather conditions all over the country. Moreover, seven different

empirical equations (Linear, quadratic, linear logarithmic, logarithmic, exponential, linear exponential and exponent) were used to estimate the monthly average daily global solar radiation on a horizontal surface. The statistical test methods of the mean absolute bias error (MABE), mean relative error (MRE), MBE, RMSE and correlation coefficient (r) were used in order to indicate the performance of the models.

The results showed that the models of the linear logarithmic and the linear exponential give generally the best results while the logarithmic and exponential models exhibit worse performance than the other models.

In the same context, a study in [10] proposed a practical and reliable approach for the prediction of the photovoltaic power generation using solar irradiance as the input. Solar irradiance is modeled as the sum of a deterministic component and a Gaussian noise signal. The deterministic part of the solar irradiance was estimated by a sinusoidal equation as follows:

$$I(t) = I_{\max} \sin\left(\frac{\pi.t}{L}\right) \quad (14)$$

where t is the time based on the sampling period, I is solar irradiance for each sample, I_{\max} is the maximum amount of solar irradiance of the previous day and L is the length of the day. The shaping filter for the Gaussian noise was calculated using spectral analysis and ARMA. The results of the two approaches were compared with the measured irradiance. The study concluded that better estimated were obtained using spectral analysis than those obtained with the ARMA, particularly for lower sampling rates.

As the condition of the weather is playing a significance role in prediction the total received energy to a certain solar panel, a study in [11] has been conducted to quantify the significance of partly cloudy conditions in rising the received energy where the change in weather conditions can have a large effect on the characteristics of the solar radiation. The study identified three distinctly different classes of weather: overcast, partly cloudy and sunny. The classification was based upon a spectral analysis of the wavelet transform which involves two methods to access the fluctuations within the hour. These two methods are known as: The clearness index and the distribution of the intra-hourly values.

$$\psi(j) = \frac{1}{T_o} \int_0^{T_o} \frac{1}{2^j} |W_j(t)|^2 dt \quad (15)$$

Equation (15) used to calculate the spectral density for different frequency ranges of the wavelet transform in order to use it in the classification procedure. T_o is the interval over the wavelet transform and W_j are the detailed coefficients at scale j . For the classification the mean spectral density (Ψ_{class}) is calculated by adding up the spectral density Ψ over the scales j that correspond to a short time frame and a longer time frame. The probability density function of the exponential distribution was given by:

$$f(x) = \lambda e^{-\lambda \cdot x} \quad (16)$$

Where λ is the rate parameter and it can be estimated from the sample values as following:

$$\lambda = \frac{n}{\sum_{i=1}^n x_i} \quad (17)$$

Therefore, the clearness index was calculated by using the following equation:

$$K(t) = \frac{G(t)}{I_o E_o(t) \cos(\theta_i(t))} \quad (18)$$

Where $K(t)$ is the clearness index signal, $G(t)$ the local solar irradiation [W/m^2], I_o is the solar constant ($1367 W/m^2$), E_o the eccentricity correction factor and θ_i is the optimal angle of incidence.

The study showed that the sun radiation at cloudy conditions becomes higher at the edges of the reflection. This rising is because the clouds are reflecting the light as a mirror. This was illustrated in a Figure shows the power time series of a photovoltaic (PV) module for a sunny and partly cloudy days. It was showed in the Figure of the partly cloudy day, there is a cloudy start of the day, as indicated by the low level of radiation, but just after noon the sky becomes partly cloudy and there are fluctuations of up to 0.75 (which is higher than the peak of clear sky).

The paper revealed how much large reserve requirements due to solar power are in partly cloudy conditions. In addition, the variations in reserve requirements due to

solar power can vary in a wide range; therefore, a significant decrease in required reserves is achieved by the aggregation of multiple sites across the location of interest. Also, the study proved that for a single site, the reserves required 99% of the time are on average 0.45 p.u. for sunny and overcast days, while for partly cloudy days the Figure rises to 0.64 p.u. which we can use it for reserve planning.

The authors in [1] defined a new model for the sun spectrum which is called Meinel and Meinel. This model uses less number of coefficients. They started by calculating the distance between the sun and the earth by using the following equation:

$$d = 1.5 \times 10^{11} \left\{ 1 + 0.017 \sin \left[\frac{360(n-93)}{365} \right] \right\} m, \quad (19)$$

where n represents the day of the year.

Another defined parameter was the declination angle δ , and it was given for any day of the year as:

$$\delta = 23.45^\circ \sin \left[\frac{360(n-80)}{365} \right] \quad (20)$$

where 23.45° represents the angle of the polar axis of the earth with the sun. The authors also proposed a method to calculate the clock time at which solar noon occurs. This clock time can be found by:

$$t = 12 + \frac{\varphi_L - \phi_n}{15} \times 60 \quad (21)$$

where φ_L is the latitude where the point of interest is at, ϕ_n is the longitude at which the solar noon occur relative to the local time zone and $(60 \text{ min}/15^\circ)$ is used because the earth needs 60 minutes to rotate 15 degrees.

The book introduced some more angles in order to specify the location of the sun at any given time, the altitude angle α which is the angle between the horizon and the incident solar beam in a plane determined by the zenith and the sun. Also, the book defined the azimuth angle Ψ which measures the sun's angular position east or west of south. Azimuth angle is needed to describe the angular deviation of the sun from directly south. In addition, the hour angle ω has been introduced. The hour angle is the angular displacement of the sun from solar noon in the plane of apparent travel of the sun, this parameter is useful to describe the position of the sun referring to the noon time.

The hour angle ω can be defined as the difference between noon and desired time of day in term of 360° rotation in 24 hours.

$$\omega = \frac{12 - T}{24} \times 360^\circ = 15(12 - T)^\circ \quad (22)$$

where T is the time of the day.

Two very useful relationships have been discussed; sunrise time ω_s and daylight duration DH and they were defined as:

$$\omega_s = \cos^{-1}(-\tan \phi \tan \delta) \quad (23)$$

$$DH = \frac{48}{360} \omega_s = \frac{\cos^{-1}(\tan \phi \tan \delta)}{7.5} h \quad (24)$$

One important relationship among ϕ , δ and ω to determine the position of the sun in term of α at a given location, date and time was defined as:

$$\sin \alpha = \sin \delta \sin \phi + \cos \delta \cos \phi \cos \omega \quad (25)$$

To find the final form of the sun radiation model, the authors used equation (25) to find the air mass coefficient AM to include the length of the path through the atmosphere. This path length is generally compared with a vertical path directly to sea level. Therefore, the air mass was calculated by using:

$$AM = 1/\sin \alpha \quad (26)$$

Hence, the received energy of the sun radiation at a specific location on a specific day and time can be estimated by:

$$I = 1367 \times (0.7)^{AM} \quad (27)$$

According to the fit accuracy of Meinel model, we have chosen it to model the sun radiation in this study. Next section discusses the process of collecting the radiation data.

2. EXPERIMENTAL SETUP

2.1. ARRAY IMPLEMENTATION

In order to collect the data from real solar radiation, a physical sensor array was used for this purpose. This chapter describes the implementation of the array, as well as, the sensors that used to capture the received power. The project setup and hardware preparations have been developed from previous work from another paper done by Beth Yount and Nisha Nagarajan [12]-[2]. The sensor array used for this paper was designed to be the same physical size as 24 Kyocera KD135GX 125W panels set-up in a traditional series-parallel configuration and adjustable legs so that it could collect data at 0° and 26° , the latter being the pitch of a typical household roof [2]. As illustrated in Figure 2.1, The sensor array is in rectangular shape with dimensions of 17.5 ft x 10 ft steel structure. The sensor boxes that placed on the steel base are distributed in matrix form starts with A1 and ends at C8. This way of arrangement was used to ease the defining and access of each sensor. The boxes are separated between each other by a column and row in distance of 2.5 ft x 5 ft respectively. All solar cells are identical and glued directly to the box lid.

Note to mention that we have used solar cells inside a small black container, but we replaced them by new sensors come alone without edges to prevent their impact on the projection of the sun's light on the cells. The solar cells then were soldered to be connected to the sensors via wiring cable.

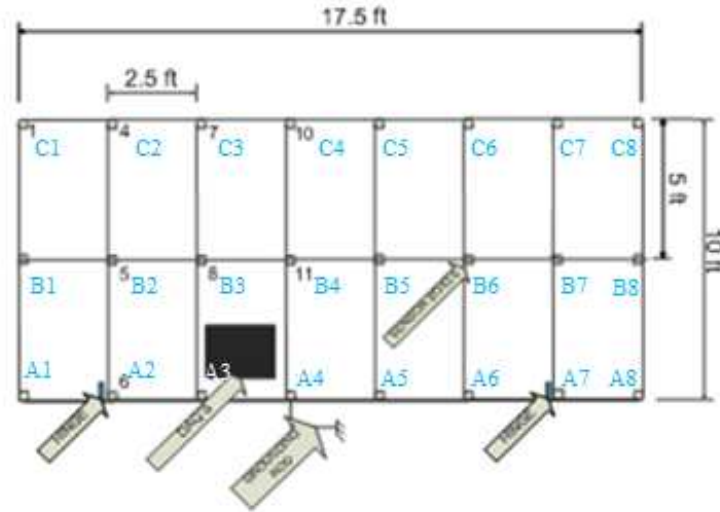


Figure 2.1: Schematic Diagram of the Sensor Array

2.2. SOLAR CELLS AND DAQ SYSTEMS

The used solar cells are supplied by Solar Made company [13] and the rating of each sensor was 0.5V 125mA. The area of each cell was approximately 5cm^2 (2.5cm x 2cm) as shown in Figure 2.2 (a). Furthermore, to keep the sensors and the solar cells from damage and dust, we put them in plastic boxes with removable lids. The cells in Figure 2.2 (a) were used for previous work. However, all solar cells have been replaced by new ones without containers as shown in Figure 2.2 (b). The reason of replacing was because the old cells came with black containers with boundary edges, these edges block the sun radiation during sunset and sunrise which impact on the accuracy of the sensors, as well as, the new solar cells come approximately in identical size more than the old ones. Figure 2.3 shows the plastic box contains a solar cell, circuit board of the sensor, 9V dry battery and a CAT5e cable connection.

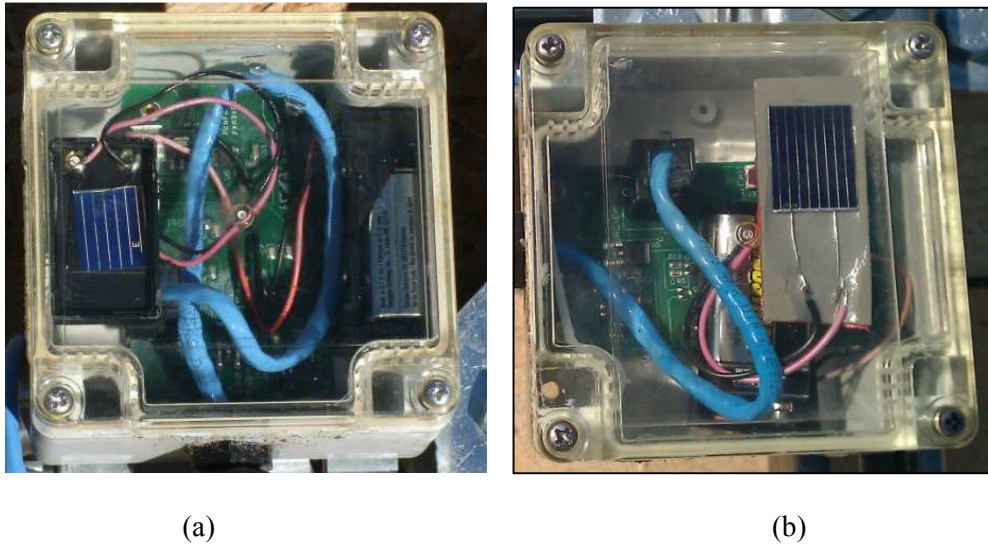


Figure 2.2: Solar Cell Sample. (a) New Solar Cell. (b) Old Solar Cell



Figure 2.3: Plastic Box to Cover the Sensor and the Solar Cell

The CAT5e cable was used to transfer the read data, by the sensor, to the main data acquisition systems (DAQs) where the data was stored into a 1GB Compact Memory card inserted into each DAQ. Figure 2.4 shows single data acquisition systems with 8 Inputs. The DAQs were R-Engine-A programmable controllers manufactured by Tern, Inc. [14]. Since each DAQ consists of 8 data Ethernet imports, 3 DAQs have been used to cover all 24 data sources.

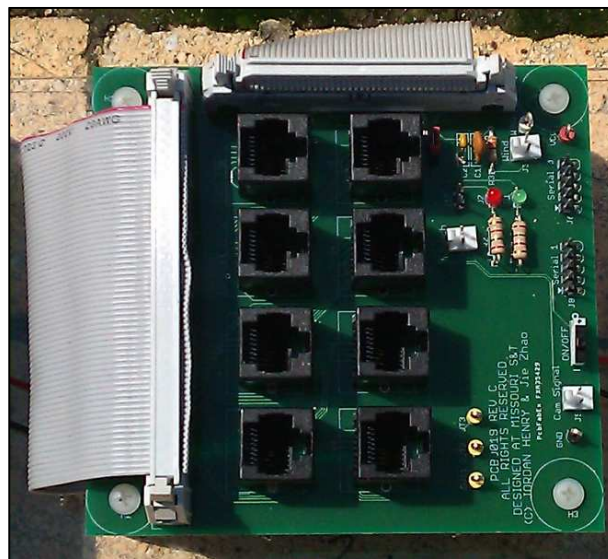


Figure 2.4: Single Data Acquisition Systems with 8 Inputs

The three DAQs then were put inside an enclosed suitcase and keep them running to get data (short circuit current, open circuit voltage, load current, time and date) from each sensor every other second. In addition, we used a very low watt light bulb to heat the DAQs up a little bit during the winter, where they stopped running in temperature

below 40F as indicated in their specifications. The stored data in the Compact Memory cards was in Microsoft Excel table format and the memories have been replaced by empty ones every week to assure that the DAQs are still running and to make some analysis on the recorded data. A simple analysis and graphs were then done with Microsoft Excel due to its simplicity, however, for more accuracy and higher degree of freedom in data manipulating, MATLAB¹ software was used to perform the detailed analysis and to write the final code to achieve the main objective of this project.

2.3. CIRCUIT BOARDS OF THE SENSOR

As mentioned previously, the aim of each sensor is to measure three main quantities (short circuit current, open circuit voltage and load current). For this purpose, a circuit board has been design to read these three values as voltage signals and then transmit the readings to DAQs via CAT5e cable. Figure 2.5 shows the schematic diagram of the sensor. The used elements for this circuit are listed in a table at the end of the paper.

¹ MATLAB is a registered trademark of The MathWorks, Inc.

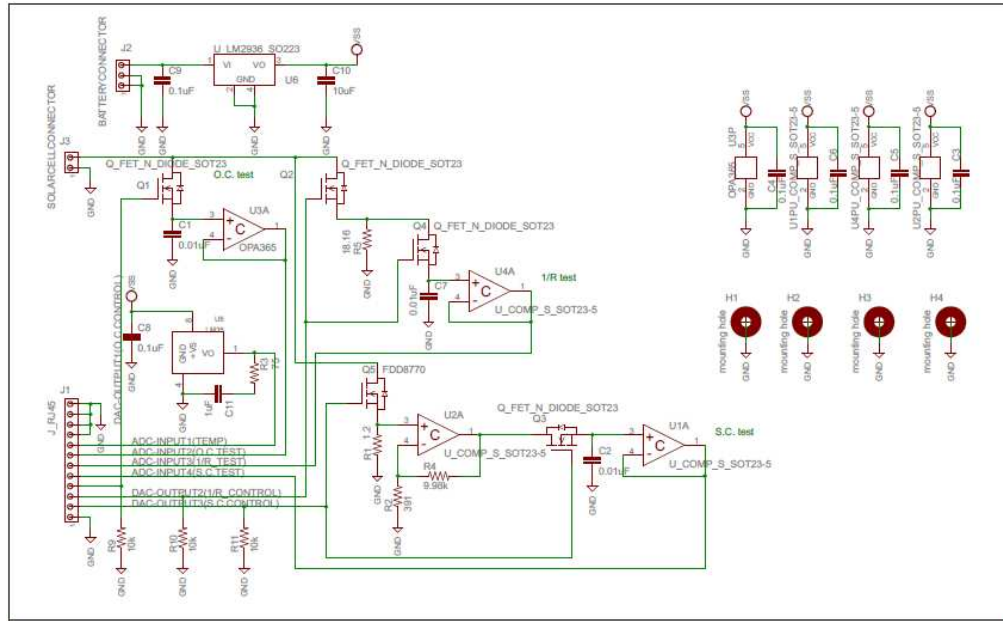


Figure 2.5: Schematic Diagram of the Sensor Circuit Board (Drawn by J. Dickherber)

The most important indicator from the sensor circuit is the short circuit current measurement, and due to the fact that the short circuit current is directly proportional to the instantaneous received power, the short circuit current reading was used to present the Energy-Time pattern after multiplying by a proper factor. More details will be showed later in this paper.

2.3.1. Sensor Improvement. It was observed that another problem has appeared on the sun radiation reading despite of changing the solar cells. After testing the circuit in the laboratory we found out that the circuit gets saturated after reaching a certain current value and we could solve this problem by changing the resistor R_4 from $10k\Omega$ to $6.8k\Omega$, where the resistor R_4 affect the gain of the short circuit current as shown in the formula:

$$Gain = R1 \times \left(1 + \frac{R4}{R2}\right) \quad (28)$$

So, the total gain has been dropped from 31.81 to 22.01. An actual circuit board is shown in Figure 2.6, the circuits making and testing have been done in the laboratory in prior of placing them in the panel.

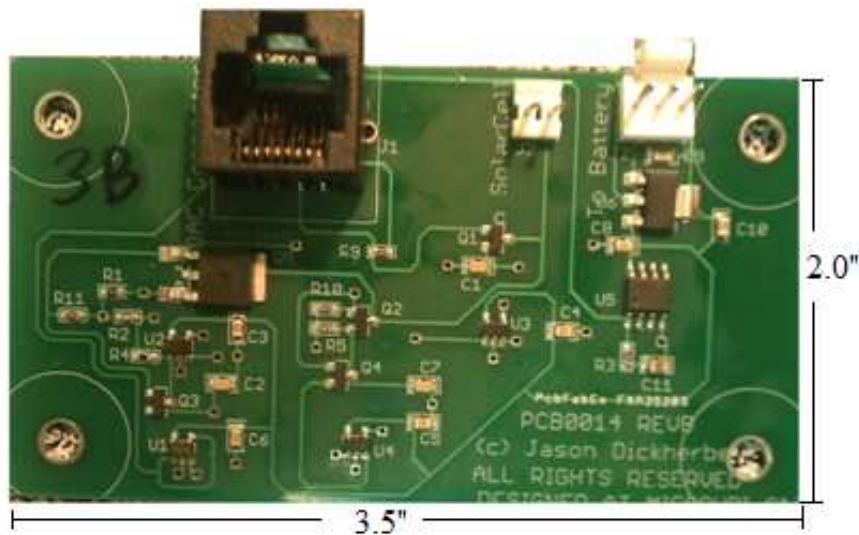


Figure 2.6: Actual Sensor Circuit Board

2.3.2. Sensors Performance Test. Some circuit boards might give wrong data due to manufacturing problems, lack in soldering paste or damage in integrated circuits or wires. Consequently, a calibration test has been conducted on the whole 24 sensor boards

to assure that they read all desired quantities in right way. The procedures of the test were taken from Jason Dickherber's suggestion test.

Since the calibration of the solar sensor boards is not reliable using PV cells due to the variance in power output by the PV cells in different locations and lighting. Power supply and test circuits are necessary to emulate the PV cells. The operating points were tested: Cell Open Circuit (OC), Short Circuit (SC) and a Load Test (I/R). The sensor board is accepted if the difference between the reading from the test circuit and from the DAQ is less than 2%. All sensors passed 2% in their reading have been fixed and re-check again.

3. SUN RADIATION MODEL AND LINEAR REGRESSION ANALYSIS

This section describes the model that used to fit the sun radiation versus time curve. Also, it touches upon the linear regression method that used in future data prediction. In addition, further analysis and work applied on the read data such as: Data averaging, data calibration and sensors relating will be introduced.

3.1. SOLAR SPECTRUM MODEL

As mentioned in the literature review, Meinel and Meinel model is the best fit to the sun radiation based on its mean bias error (MBE) and root mean square error (RMSE) tests [1]. This model is distinguished by its accuracy and less number of used parameters. In addition, this model sets the sunrise and sunset times without using extra equations as in other models. The final form of the equation that gives the solar spectrum can be given as:

$$I = 1367 \times (0.7)^{AM} \quad (29)$$

where I is the sun radiation energy (kW/m^2) and AM is the air mass coefficient. In order to drive equation (29), reference [1] started by defining the declination δ and it was given for any day of the year as:

$$\delta = 23.45^\circ \sin \left[\frac{360(n-80)}{365} \right] \quad (30)$$

Where n represents the day of the year and 23.45° represents the angle of the polar axis of the earth with the sun. The reference considered that the angles north of the equator are positive, while they are negative when they locate south of the equator. Figure 3.1 shows the declination at different times of the year.

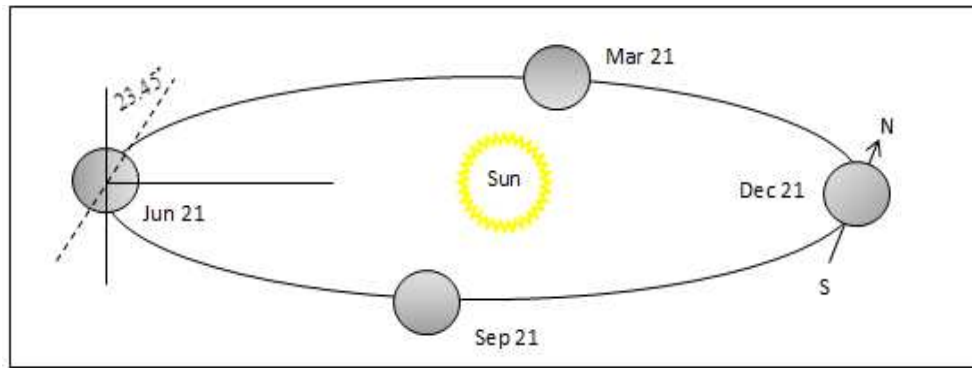


Figure 3.1: The Orbit of the Earth and the Declination at Different Times of the Year

The authors also proposed a method to calculate the clock time at which solar noon occurs. This clock time can be found by:

$$t = 12 + \frac{\varphi_L - \phi_n}{15} \times 60 \quad (31)$$

where φ_L is the latitude where the point of interest is at, ϕ_n is the longitude at which the solar noon occur relative to the local time zone and $(60 \text{ min}/15^\circ)$ is used because the earth needs 60 minutes to rotate 15 degrees. Furthermore, in order to specify the position of the

sun, the authors specified two more coordinates besides the distance between the sun and the earth which was assumed to be constant. These two coordinates are the solar altitude α and the azimuth Ψ .

The figure below (Figure 3.2) illustrates the angles and the relation among them. It demonstrates the altitude angle α as the angle between the horizon and the incident solar beam in a plane determined by the zenith and the sun. The azimuth angle Ψ measures the sun's angular position east or west of south. Azimuth angle is needed to describe the angular deviation of the sun from directly south. In addition, the Figure shows the hour angle ω , which is the angular displacement of the sun from solar noon in the plane of apparent travel of the sun. This parameter is useful to describe the position of the sun referring to the noon time.

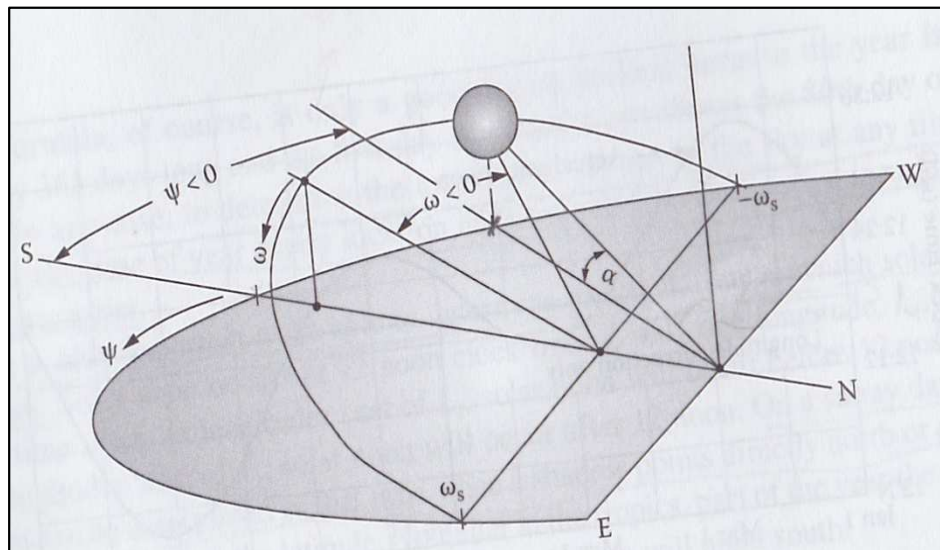


Figure 3.2: Sun Angles, Showing Altitude, Azimuth and Hour Angle [1]

The hour angle ω can be defined as the difference between noon and desired time of day in term of 360° rotation in 24 hours.

$$\omega = \frac{12-T}{24} \times 360^\circ = 15(12-T)^\circ \quad (32)$$

Where T is the time of the day.

Figure 3.3 shows the relationship among different sun angles that described above.

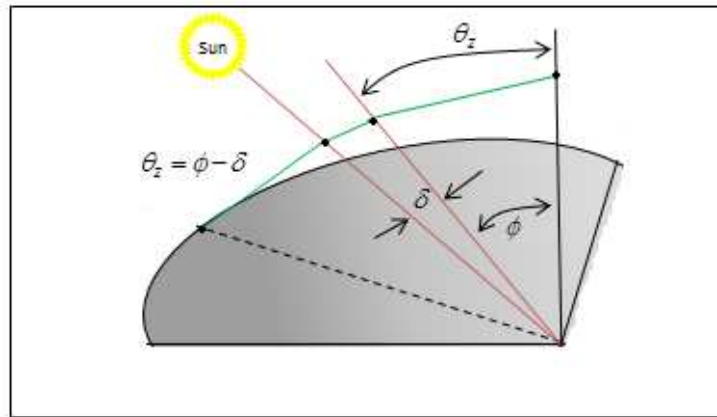


Figure 3.3: Relationship among Zenith Angle, Latitude and Declination at Solar Noon in Winter

One important relationship among ϕ , δ and ω to determine the position of the sun in term of α at a given location, date and time was defined as:

$$\sin \alpha = \sin \delta \sin \phi + \cos \delta \cos \phi \cos \omega \quad (33)$$

To find the final form of the sun radiation model, the authors used equation (33) to find the air mass coefficient AM to include the length of the path through the atmosphere. This path length is generally compared with a vertical path directly to sea level. Therefore, the air mass was calculated by using:

$$AM = \frac{1}{\sin \alpha} \quad (34)$$

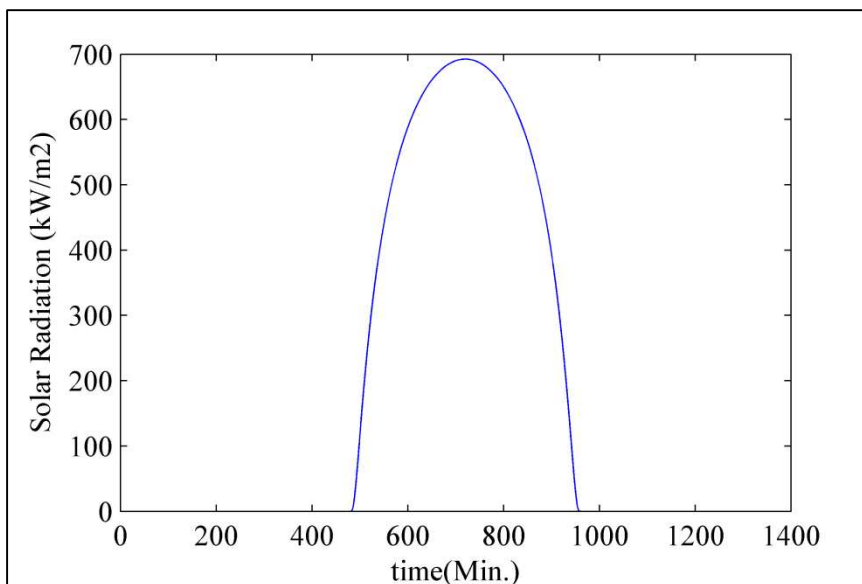
Hence, the received energy of the sun radiation at a specific location on a specific day and time can be estimated by equation (29)

$$I = 1367 \times (0.7)^{AM} \quad (35)$$

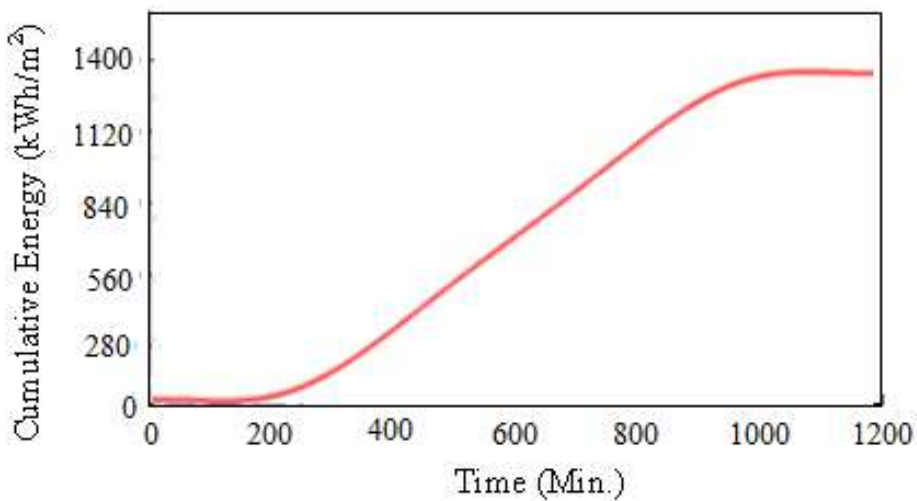
This equation is, of course, for $AM = I$. However, according to Meinel and Meinel [1], a better fit to the observed data is given by:

$$I = 1367 \times (0.7)^{AM^{0.678}} \quad (36)$$

The average monthly and cumulative energy for January has been obtained using equation (36) as shown in Figure 3.4



(a)



(b)

Figure 3.4: (a) Average Monthly, (b) Average Cumulative (kWh/m²) for January in Rolla, Missouri Using Meinel and Meinel Model

The power incident on a PV module depends not only on the power contained in the sunlight, but also on the angle between the module and the sun. When the absorbing

surface and the sunlight are perpendicular to each other, the power density on the surface is equal to that of the sunlight (in other words, the power density will always be at its maximum when the PV module is perpendicular to the sun). However, as the angle between the sun and a fixed surface is continually changing, the power density on a fixed PV module is less than that of the incident sunlight.

The following figure (Figure 3.5) shows how to calculate the radiation incident on a titled surface with a slope (5:12). The sun radiation on the tilted module can be found from equation (36) which is used for finding the radiation on a horizontal surface. As the array tilting to the south, a new α must be calculated using the following equation:

$$\alpha_2 = \alpha + \beta \quad (37)$$

α and β are shown in figure 3.5.

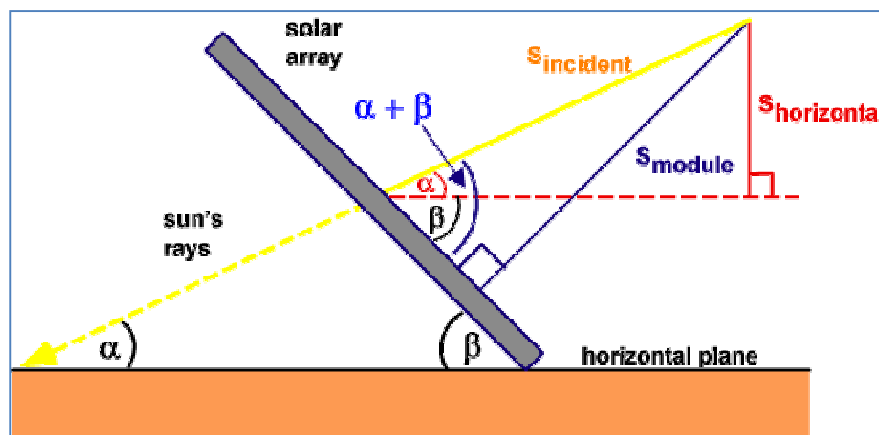


Figure 3.5: The Relationship among the Sun and the Module Angles for Tilted Solar Panel with $\beta=22.6^\circ$

Equation (37) shows that we need to add the array angle ($\beta=22.6^\circ$) to the original α to indicate the tilting of the array toward the south whereas the array itself locates at the north hemisphere of the earth and then we substitute the new alpha α_2 in equation (33) to find the air mass AM .

3.2. LINEAR REGRESSION METHOD

In statistics, regression analysis consists of techniques for modeling the relationship between a dependent variable (also called response variable) and one or more independent variables (also known as explanatory variables or predictors) [15]. Furthermore, in linear regression method, the dependent variable is modeled as a linear function of independent variable so that the model gives the best fit to the actual data. In the same context, for a time series variable, linear regression method often used to calculate the next value if the relationship between two sets of data (x and y) is linear. Hence, we can apply this method on our sun radiation data with using the sun radiation model (Meinel and Meinel model) as a reference. The parameters of the regression model can be found by using several techniques. Each technique results in different parameters values and thus different estimation. However, the most commonly used criterion is the least squares method [15]. In this paper, the simple linear regression was used where the system which needs to be modeled has one dependent variable (sun radiation) and one independent variable (time). The simple linear regression model can be written as:

$$y_{(k+1)} = \beta_{0,k} + \beta_{1,k} \times x_{k+1} + \varepsilon \quad (38)$$

Where y is the dependent variable, x is the independent variable, β_0 and β_1 are y - intercept and slope, respectively, and ε is a random error. The parameters are estimated based on predefined criterion. The most commonly used criterion is the least squares method. The principle of the least square method is to find the values of β_0 and β_1 such that the sum of the squared distance between actual data and fitted data reaches the minimum among all possible choices of regression coefficients β_0 and β_1 [15]. Therefore, based on least squares method, the y - intercept and slope parameters can be found using:

$$\beta_1 = \frac{n \sum_i^k (x_i y_i) - \sum_i^k x_i \sum_i^k y_i}{n \sum_i^k (x_i^2) - \left(\sum_i^k x_i \right)^2} \quad (39)$$

$$\beta_0 = \frac{\sum_i^k y_i - \beta_1 \sum_i^k x_i}{n} \quad (40)$$

where $i=k-n$ and k is the number of the current value (i.e. if we reached to the 25th reading, the $k=25$). The coefficient n is the number of past data taken into account to calculate the next value. For example, if we used three last readings to predict the next reading, the $n=3$). The parameter n is also called “Window Size”, and for our last example, *window size*=3. Different window sizes affect the accuracy of the estimator model. However, in our linear regression model, the window size was chosen to be equal 4 due to its fit accuracy relying on the Mean Squared Error (MSE). More details in different window size sets will be shown in the next section. In addition to the previous

parameters and variables, correlation coefficient is needed for further work and it can be calculated by:

$$r = \frac{n \sum (xy) - \sum x \sum y}{\sqrt{[n \sum (x^2) - (\sum x)^2][n \sum (y^2) - (\sum y)^2]}} \quad (41)$$

One important vector which plays an important role in regression method is regression residual e_i . Regression residual defined as the difference between the actual response y_i and the fitted response \hat{y}_i .

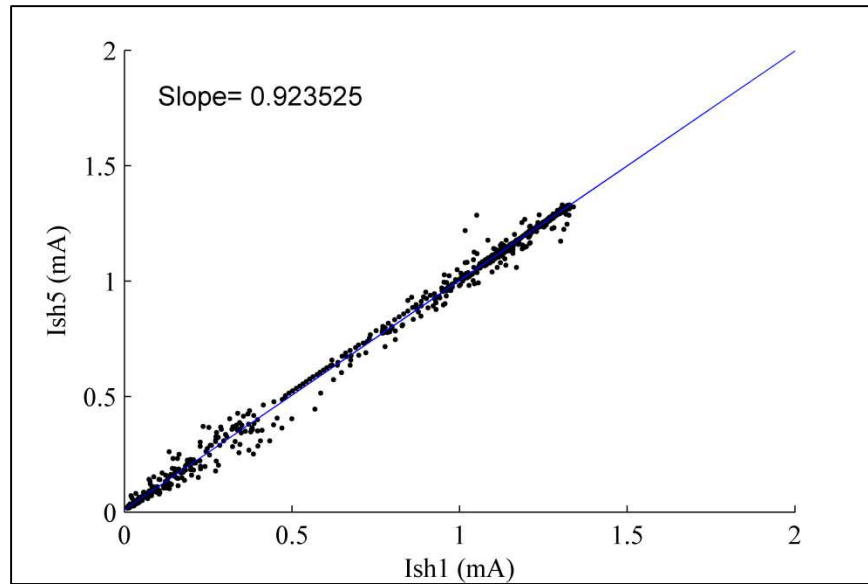
$$e_i = y_i - \hat{y}_i \quad (42)$$

It should be noted that regression residual is observable, but the error term in the regression model ε is unobservable, therefore, in this paper we applied the linear regression method on the difference between the actual sun radiation data and the sun spectrum model (Meinel and Meinel model), where the random error ε has been generated using a special function in MATLAB.

3.3. CORRELATION METHOD AND ARRAY DATA CONSTRUCTION

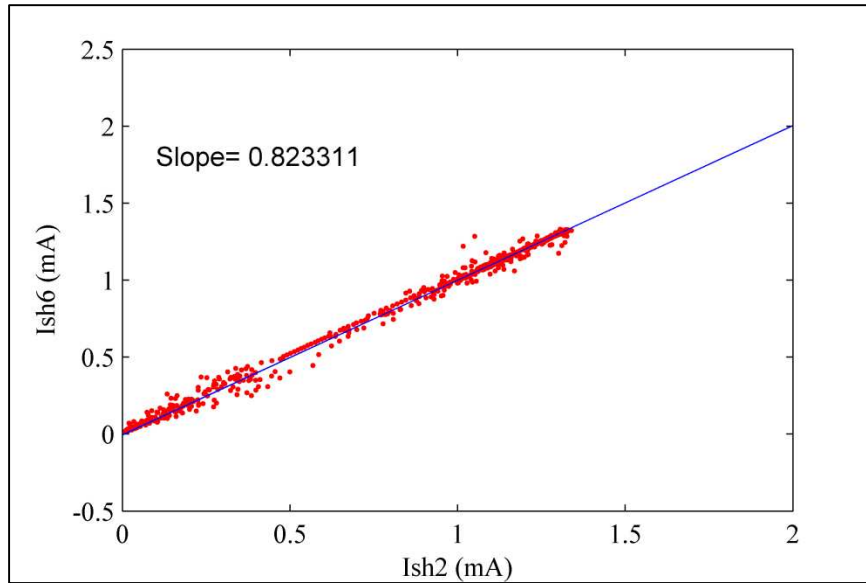
One main aim of the project is to build up the data of the whole array using one sensor's reading. The method that we used to make a relationships among all sensors was finding the slope between each neighbor sensors, as well as, the correlation coefficients (variance matrix has been tried but it was found to be useless where the values were very

close to each other). In order to find the slope between two sensor readings, MATLAB has been used to plot each two neighbor sensors and then calculate the slope after applying the polynomial fitting to the pattern as shown in Figure 3.6.



(a)

Figure 3.6: (a) Ish_5 versus Ish_1, (b) Ish_6 versus Ish_2 to Find the Slopes Between each others



(b)

Figure 3.6: (a) Ish_5 versus Ish_1, (b) Ish_6 versus Ish_2 to Find the Slopes Between each others (cont.)

The correlation coefficients, which obtains how two variables related to each other, were found from the covariance matrix and slandered deviation values so that the correlation coefficient between sensor1 (x) and sensor2 (y) can be found by:

$$R_{(x,y)} = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} \quad (43)$$

Where $R_{(x,y)}$ is the correlation coefficient between x and y , $\text{cov}(x,y)$ is the covariance matrix, σ_x and σ_y are the standard deviation of x and y , respectively. These two values,

slope and correlation coefficient, were used in process of building up the whole array radiation data from one sensor only by using the following equation:

$$i_y = \left[\frac{\sum_{x=j}^n i_x \times (R_{(y-x)} + slope_{(y-x)})}{(R_{(y-x)} + slope_{(y-x)})} \right] + \varepsilon \quad (44)$$

where i_y is the calculated short circuit current, i_x is the current related to i_y by $slope_{(y-x)}$ and correlation coefficient $R_{(x,y)}$ where x represents the name of each sensor located side-by-side with i_y . ε is the random error from equation (38) and it has been generated using ($randn(m,n)$) function in MATLAB. This function returns an m-by-n matrix containing pseudorandom values drawn from the standard normal distribution, and then we multiplied these random values by the slandered deviation σ of each sensor's reading separately to get its own reading. Note to mention that using the slope and correlation coefficient combined together in equation (44) yields better correlation than using one of them alone in process of obtaining the other 23 sensors. The flow chart (Figure3.7) explains the processing of constructing the array's data. First of all, we started with B4 sensor's reading to generate the data of sensors (B5, B3, A4 and C4). These five sets of data have been used to calculate other four sensor's data (A3, C3, A5 and C5). Then we selected row 5 (A5, B5 and C5) to calculate B6, and row 3 (A3, B3 and C3) to calculate B2. After that we repeated the calculation process over the whole array, so we calculated A6 from (B5, A5 and B6), C6 from (B5, C5 and B6), A2 from (B3, A3 and B2), C2 from (B3, C3 and B2), and again we used rows 6 and 2 to calculate B7 and B1, respectively.

Then we applied the same sequence as before, so we calculated A7 from (B6, A6 and B7), C7 from (B, C6 and B7), A1 from (B2, A2 and B1), C1 from (B2, C2 and B1). Finally, we calculated A8 from (B7, A7 and B8) and C8 from (B7, C7 and B8). All calculations of Figure 3.7 have been done using equation (44). Table 3.1 shows the order of calculation process of Figure 3.7.

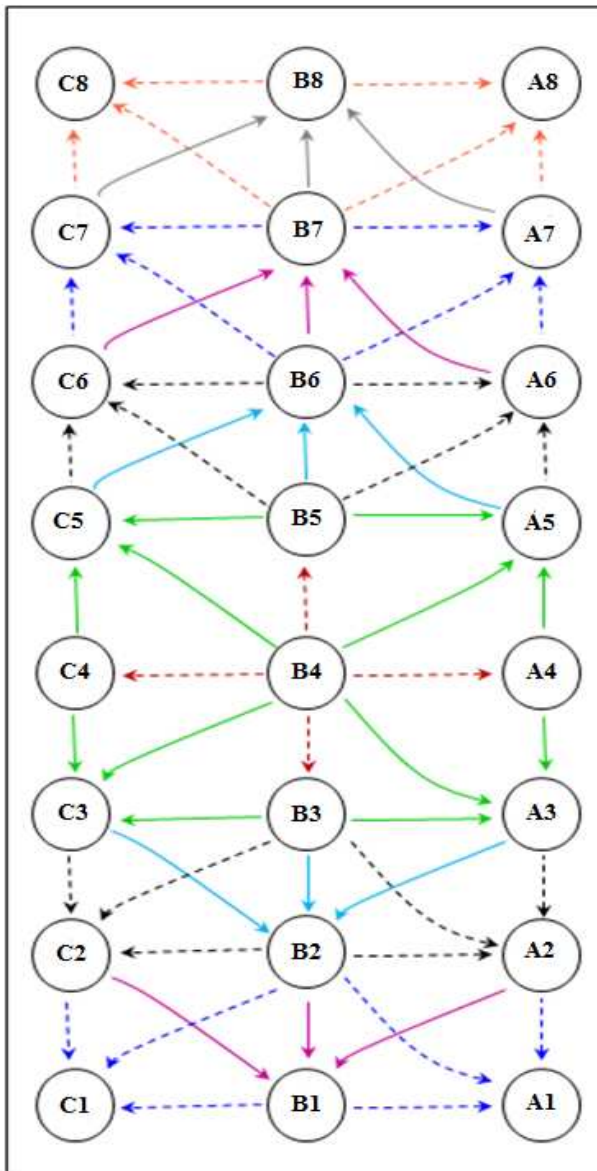


TABLE 3.1: Order of Correlation Process

Order	Arrow
1	
2	
3	
4	
5	
6	
7	
8	

Figure 3.7: Order of Computation all Sensors Starting from B4

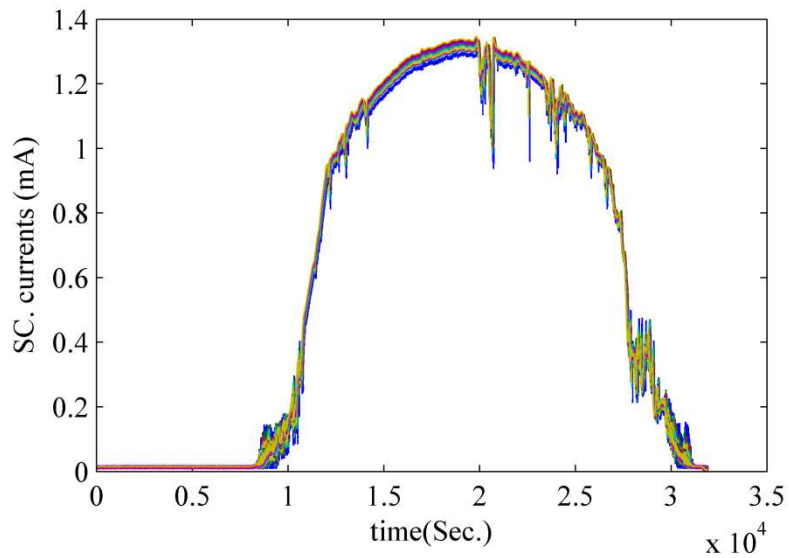
4. DATA PROCESSING AND RESULTS

4.1. DATA CALIBRATION AND AVERAGING

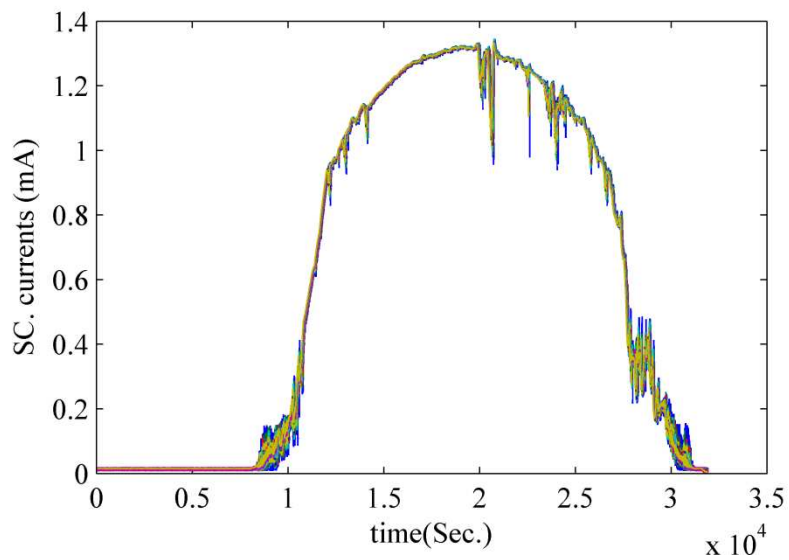
All readings from the 24 sensors have been collected and imported from the DAQ system to MATLAB software through USB memory adaptor. The array was located in a place open to the sunlight most of the day it was laid flat. Before working on the data, two main statistical processing have been applied: Data calibration and averaging. Data calibration has been done to assure that all solar cells set to be identical. The calibration process was conducted by multiplying all data by such a factor to meet them up at a reference sensor reading so that:

$$i_{short(new)} = i_{short(old)} \times \frac{(i_{ref.(m)}/i_{old(m)}) + (i_{ref.(n)}/i_{old(n)})}{2} \quad (45)$$

where $i_{short(new)}$ is the new short circuit current to be calculated, $i_{short(old)}$ is the old short circuit current to be calibrated, i_{ref} is the reference current, m and n are two points through the day and they were chosen so that the calibration factor will be approximately the average amongst all factors throughout the day (one calibration factor has been tested and it gives less efficient results than two points calibration). Furthermore, the data have been averaged over one minute using MATLAB. Figure 4.1 and Figure 4.2 show the initial process (data calibration and averaging) which precedes the main analytical processes.



(a)



(b)

Figure 4.1: (a) Uncalibrated Sensors Readings, (b) Calibrated Sensors Readings (using two points calibration way)

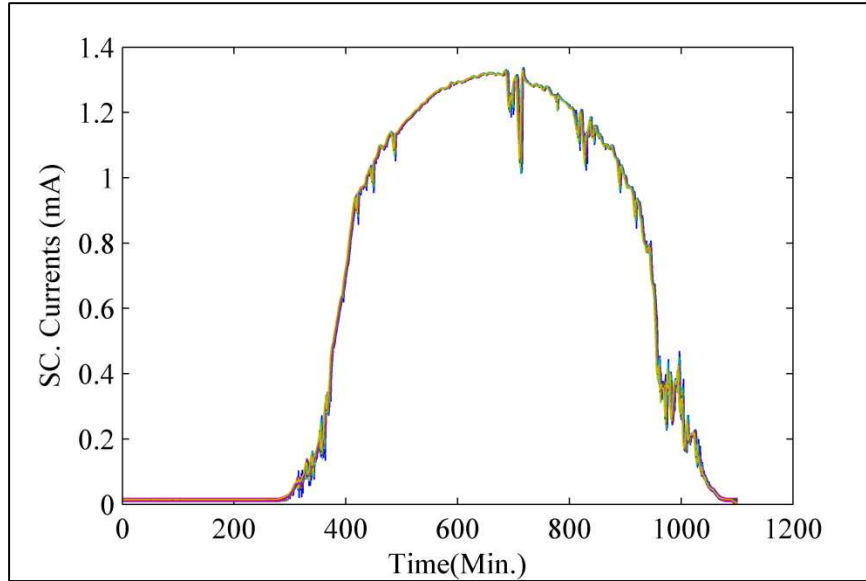


Figure 4.2: DAQ readings After Averaging Over One Minute

4.2. SOLAR RADIATION MODEL AND ACTUAL DATA FITTING

Figure 4.3 demonstrates the power versus time curve of Meinel and Meinel model calculated for April 1st. The time axis scale has been converted to minute scale instead of hours to be appropriate to compare with actual data.

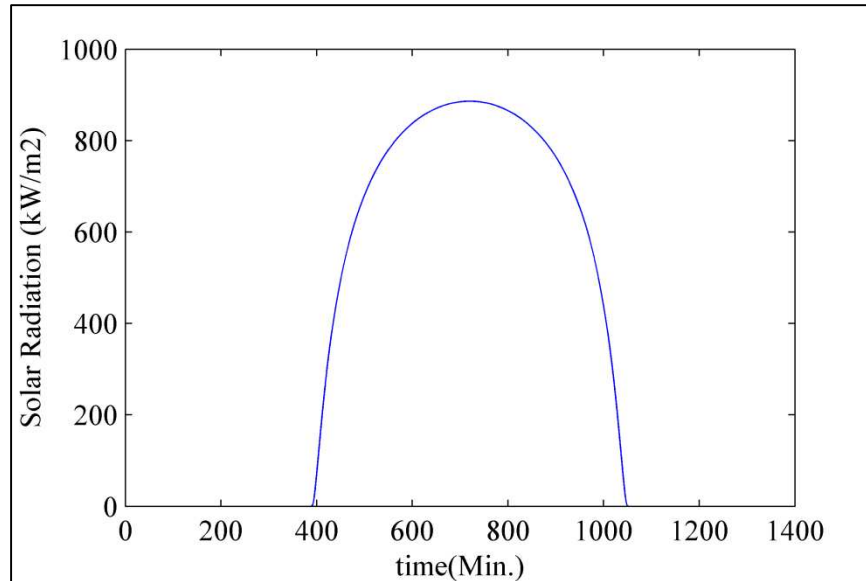


Figure 4.3: Power-Time Curve of Sun Radiation on April 1st Using Meinel and Meinel Model [1]

Note to mention that we have manipulated with the main equation of the Meinel and Meinel model to convert the y-axis of the sun radiation spectrum from (kW) to (mA) by multiplying by such factor. This factor was the maximum reading of the actual data divided by maximum value of the model where the relationship between the output power and the short circuit current of a solar cell is almost linear [1]. In addition, azimuth angle ($\Psi=7.0^\circ$) has been included to the model calculations. This process has been done to match the mathematical model to the actual data, and then to plot it over the actual data to be ready for analyzing as shown in Figures 4.4 and 4.5.

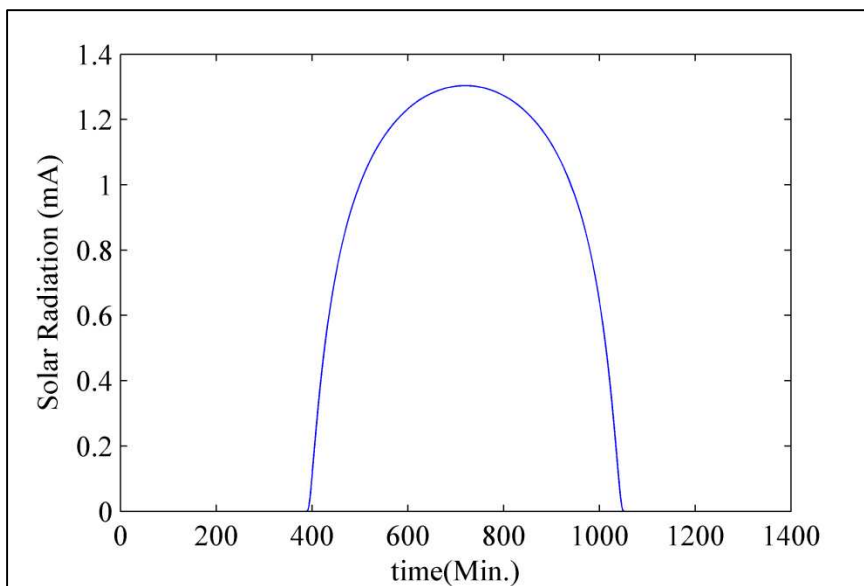


Figure 4.4: Power-Time Curve of Sun Radiation on April 1st Using Meinel and Meinel model (after multiplying by power-to-short circuit current ratio)

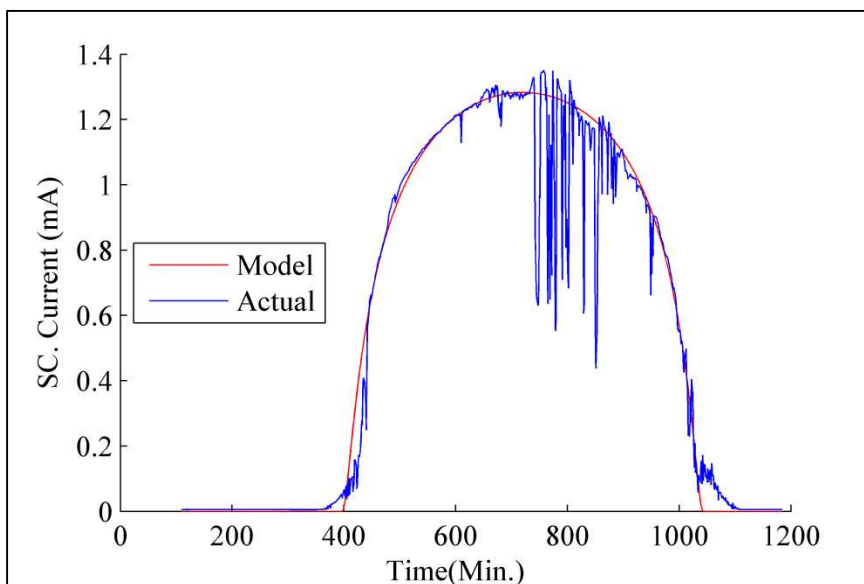


Figure 4.5: Actual Data and Mathematical Model for April 1st

4.3. EXPECTING THE FUTURE TREND

As discussed in section 3, the difference between the actual data and the mathematical model for sun radiation has been predicted using linear regression method. In addition to subtraction way, division has been tried. Figure 4.6 shows the comparison between subtracting and division. Relying on Figure 4.6, subtracting the mathematical model from the actual data gives less mean squared error than the division way.

After deciding the difference between the actual data and the mathematical model to be predicted, linear regression has been applied on the subtracted data using equation (38), and then we added the predicted signal to Meinel and Meinel model for the same day to get the predicted sun radiation as shown in Figures 4.7, 4.8, 7.9, 7.10 and 7.11.

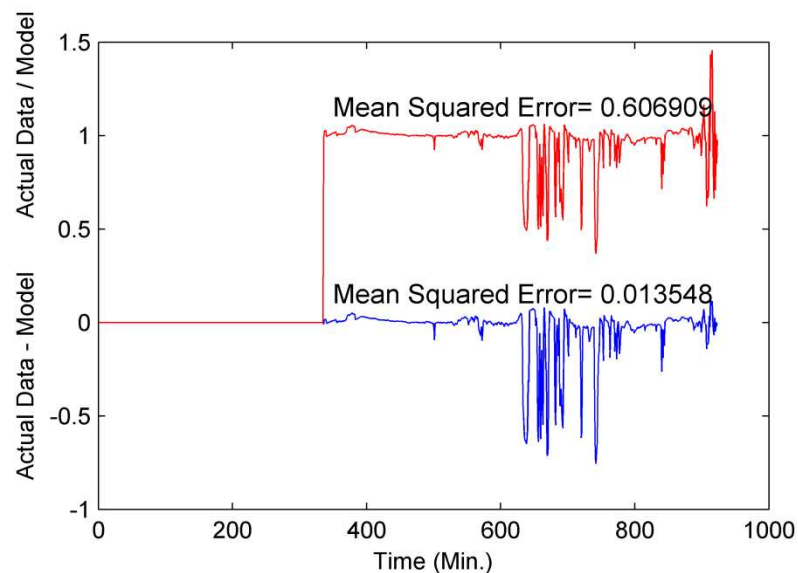


Figure 4.6: Comparison Between Subtracting and Dividing the Mathematical Model with Actual Data

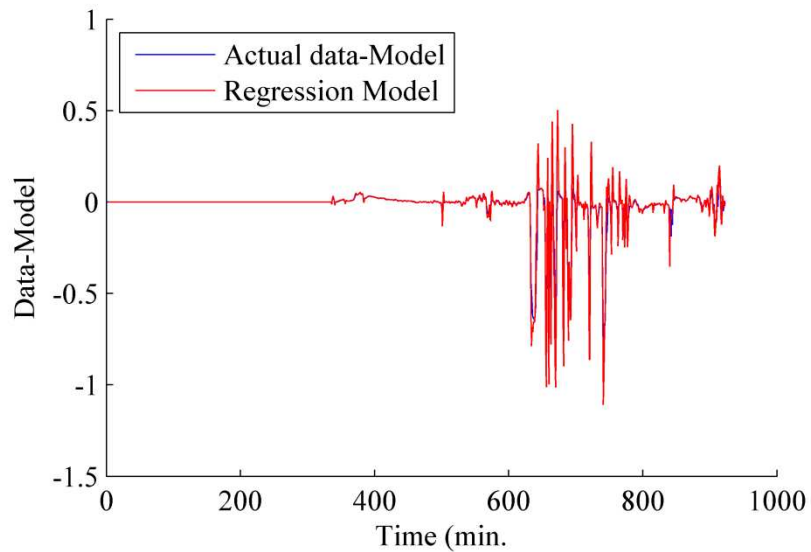


Figure 4.7: Applying the Linear Regression Method on the Difference Between the Actual Data and the Mathematical Model

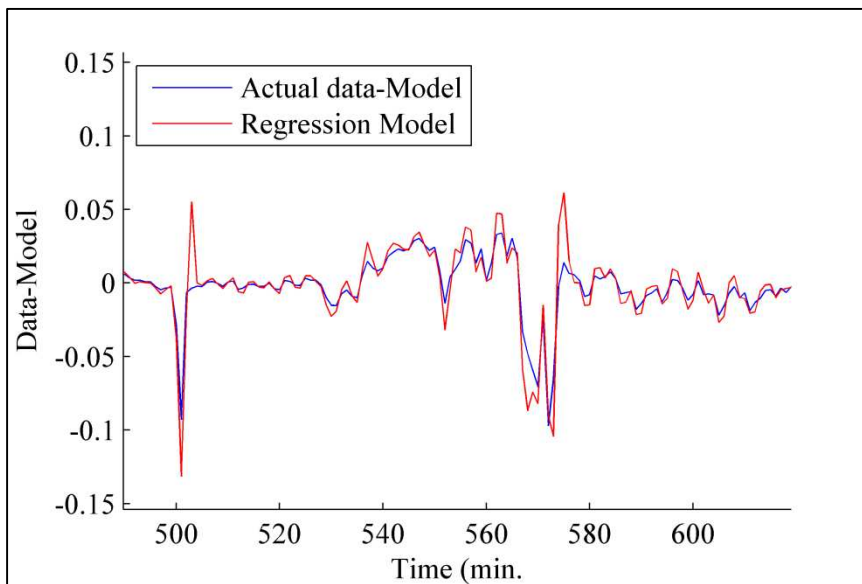


Figure 4.8: Applying the Linear Regression Method on the Difference Between the Actual Data and the Mathematical Model (enlarged)

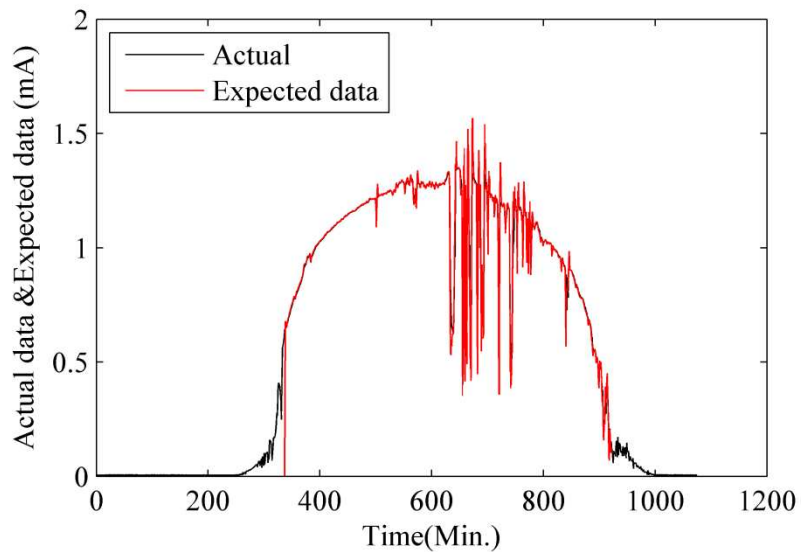


Figure 4.9: Adding the Predicted Signal to the Model and Meinel Model and the Actual Data for April 1st

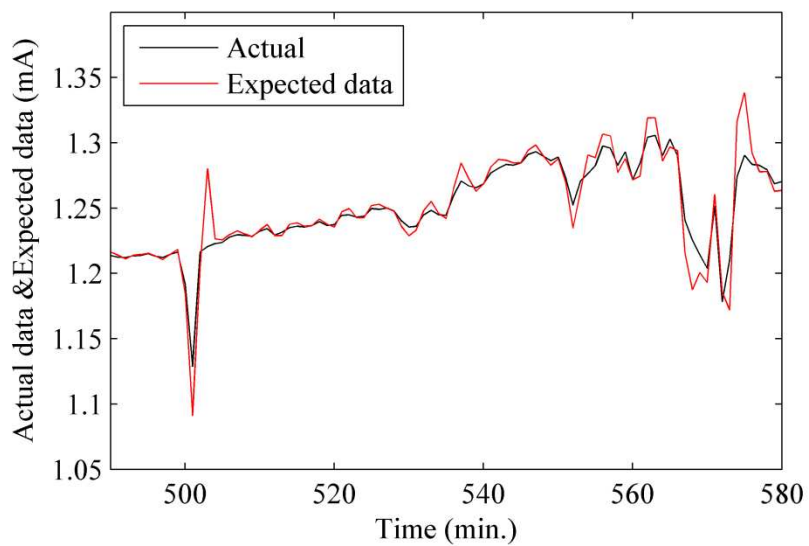


Figure 4.10: Adding the Predicted Signal to the Model and Meinel Model and the Actual Data for April 1st (enlarged)

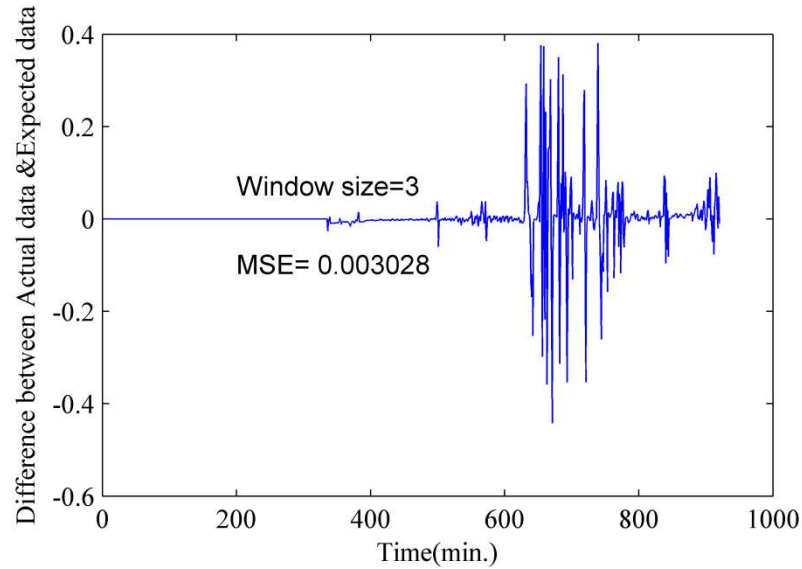


Figure 4.11: MSE Error for the Difference Between the Actual and Expected Data
(*widow size=3*)

Further comparison has been made to assure that subtracting is better than division, as well as, to select the best window size of the linear regression equations (38), (39) and (40). In Figure 4.12 four different values of window size (3, 4, 5 and 6) were tested for subtracting and division. Table 4.1 summarizes a comparison between subtracting and dividing the mathematical Model with the predicted data after applying the linear regression for different window sizes for four days.

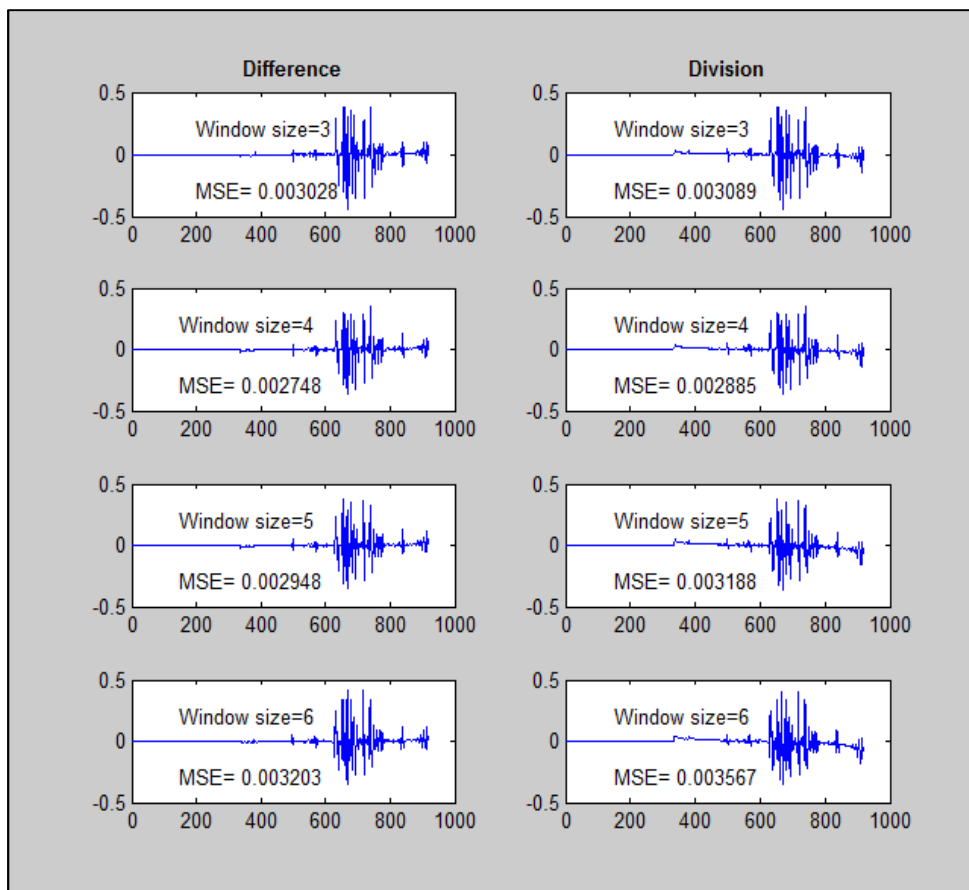


Figure 4.12: Comparison Between Subtracting and Dividing the Mathematical Model with Predicted Data (after applying the linear regression for different window sizes)

TABEL 4.1: MSE for Different Window Size and Different Days for Subtracting and Division

Day	Subtracting (Window size)				Dividing (Window size)			
	3	4	5	6	3	4	5	6
April 1st	0.00302	0.0027	0.00294	0.0032	0.00308	0.00288	0.00318	0.00356
April 2nd	0.00309	0.00211	0.00243	0.0037	0.00289	0.00297	0.00299	0.00307
April 3rd	0.00292	0.00301	0.00337	0.0039	0.00366	0.00302	0.0032	0.00309
April 4th	0.00391	0.00238	0.00289	0.0041	0.00376	0.00406	0.0040	0.0038
Mean	0.00323	0.0025	0.0029	0.0037	0.0033	0.00323	0.0033	0.0033

The results from Figure 4.12 and table 4.1 showed that applying linear regression for subtracting the mathematical model from the actual data with window size equals 4 yields better mean squared error.

4.4. DATA FILTERING AND NOISE REJECTION

Based on the fact that the solar radiation during a partly cloudy weather condition goes higher than the normal peak at sunny day as explained in [11], the predicting signal may exceed the sun light limitation due to the sharp change in the solar radiation.

Accordingly, high radiation filter (HRF) has been used to reject the unacceptable values.

In process of designing the HR filter, we assumed that the readings that go higher than 20% of the values calculated from Meinel and Meinel model to be rejected and reduced to 70% of its value. In addition, our sensors give strange readings with lots of noise at

sunrise and sunset times so that we clipped out approximately 100 minutes after sunrise and before sunset. Figure 4.13 shows the data that plotted in Figure 4.8 after filtering and rejecting the sunrise and sunset noise.

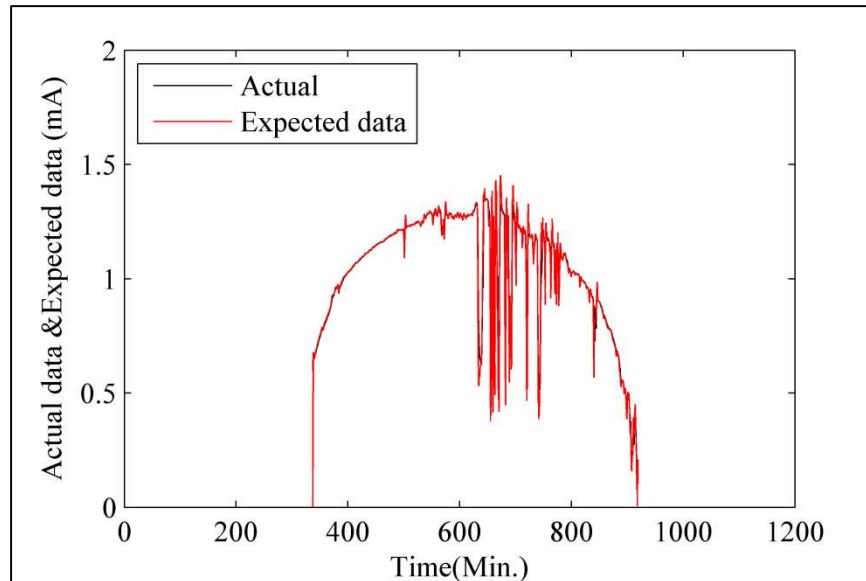
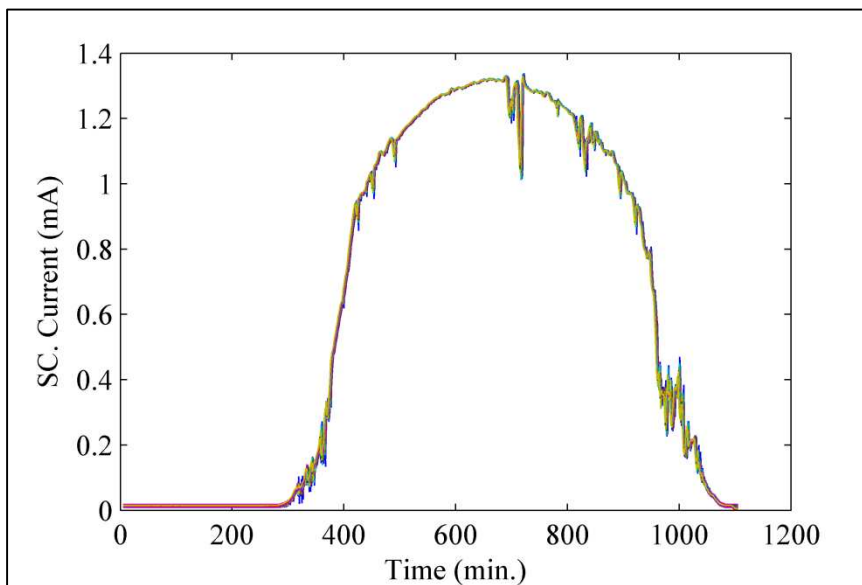


Figure 4.13: Data After Filtering and Noise Rejection (to be compared with Figure 4.9)

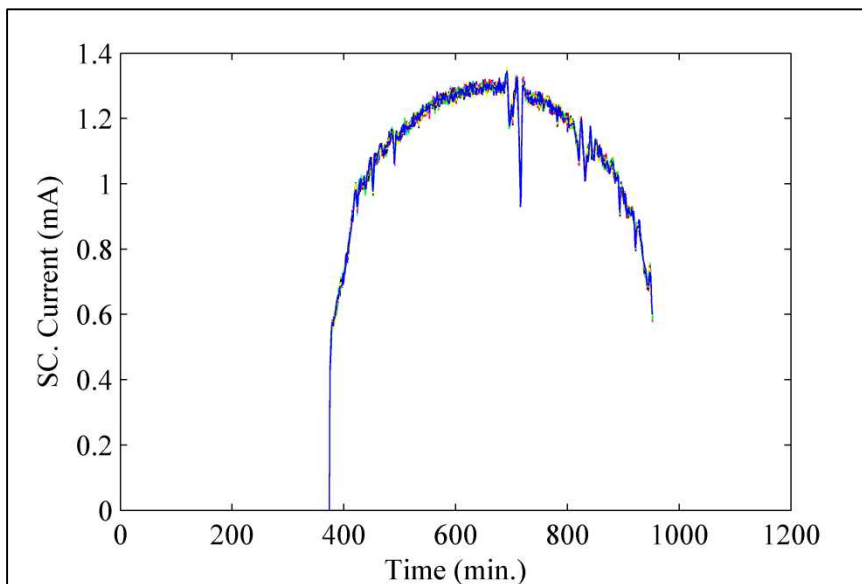
4.5. ARRAY DATA CONSTRUCTION

Generating all sensors' readings from only one sensor reduces the costs of control circuits and DAQ systems, as well as, it speeds up the data analyzing with a smaller room of data storage. As shown in Figure 3.7, all sensors readings can be created from sensor B4 (center of the array) using equation (44). Further work has been done to combine the previous processes (data prediction and generation) into one step. Figure 4.14 compares all array actual data and sensors data after predicting B4 sensor and generate the other 23

sensors' data using linear regression method, slopes among the sensors and correlation factors. Figure 4.15 shows the data of Figure 4.14 between 420 and 900 minutes.

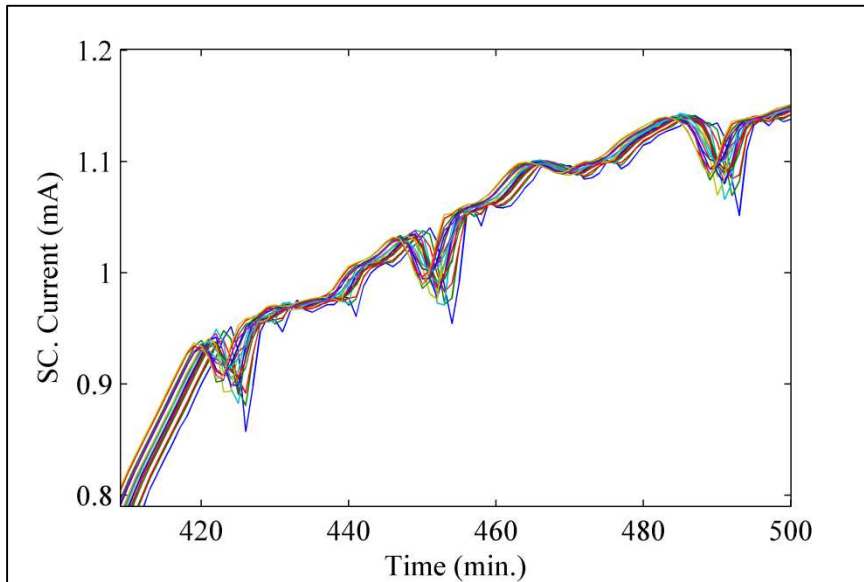


(a)

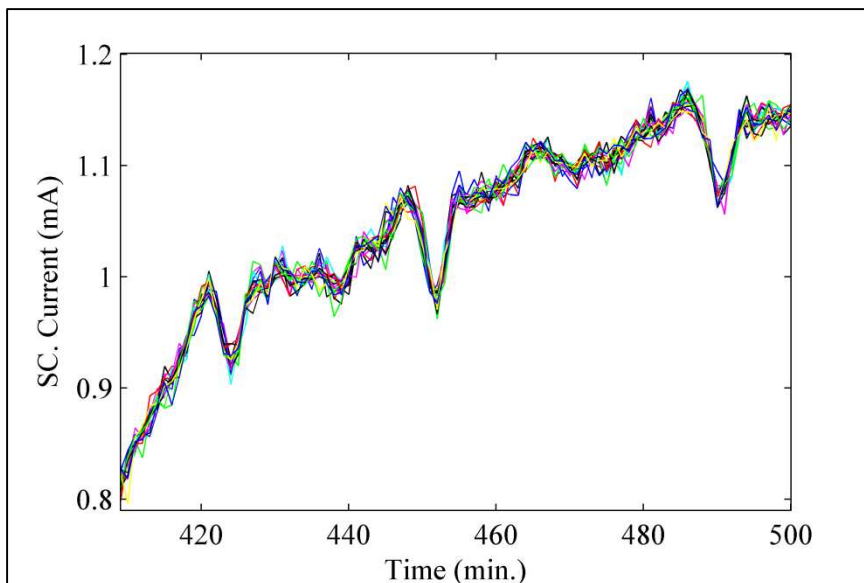


(b)

Figure 4.14: (a) Actual Data, (b) Predicted Data for April 3rd



(a)



(b)

Figure 4.15: (a) Actual Data, (b) Predicted Data for April 3rd
(enlarged)

The final results have been tested by finding MSE individually for each sensor and then we calculated the average error among them. The test ended up with 2.6% as the average of the MSEs. Figure 4.16 shows the difference between the predicted data and actual data and the MSE for sensor A1.

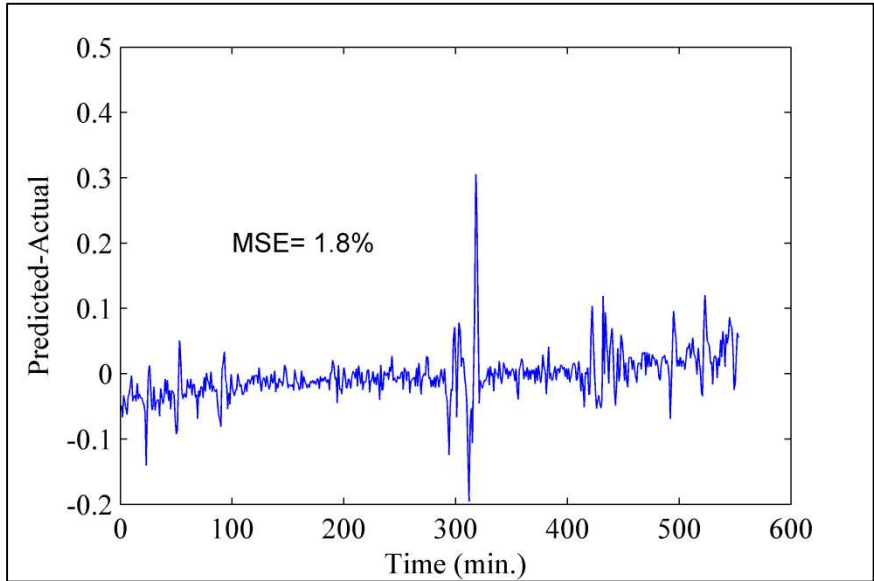


Figure 4.16: Difference Between the Predicted Data and Actual Data and the MSE for Sensor A1.

5. CONCLUSIONS

Using of the M and M model along with linear regression method to predict the future trend of the solar radiation yields better results with using a window size equals four for the regression equations. The results also showed that relating different parts of a solar panel is possible. In addition, predicting the next minute reading of each part of a solar panel using model fitting and correlation method yields an acceptable result with an average (MSE) of 2.6%. This technique can contribute in speeding up the data analyzing with a smaller room of data storage, set the ratings of PV system elements and protects the electrical converters (or inverters) from low and over voltage (current), as well as, it reduces the costs of control circuits and DAQ system. Furthermore, the study concluded that the sun radiation during a cloudy weather may exceed the maximum limit, where it reached to 1.07 p.u. (Figure 4.5), and that is due to the reflected radiation by the clouds. This habit may lead to a negative effect on the data prediction. Therefore, high radiation filter (HRF) has been used to correct the calculated radiation at the peak time.

APPENDIX

```

                                % MAIN CODE

clear all
close all
clc

data=csvread('all120sen.csv', 96291, 3, [96291 3 128210 40]);
sc_current=data(:,5:28);
t_m=data(:,1)*60+data(:,2);
T_m=[min(t_m):max(t_m)];
t_s=(data(:,1)*60*60)+(data(:,2)*60)+(data(:,3));
day=95; % APRIL 3rd

% CALIPRATION PROCESS
m=length(sc_current);

for i=1:24

aa(i)=sc_current(17704,16)/sc_current(17704,i);

b(i)=sc_current(22899,16)/sc_current(22899,i);

c(i)=(aa(i)+b(i))/2;
end

for n=1:24
for i=1:m

    sc_current_cal(i,n)=sc_current(i,n)*c(n);

end
end

figure(1)
plot(sc_current_cal)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
xlabel('time(Sec.)')
ylabel('SC. currents (mA)')
savemyfig3

figure(2)
plot(sc_current)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
xlabel('time(Sec.)')
ylabel('SC. currents (mA)')

```

```

savemyfig3

% AVERAGING OVER 1 MIN.
for l=min(t_m):max(t_m)

    meancurrents((l-min(t_m)+1),:
) = mean(sc_current_cal(t_m==l,1:20),1);

end

figure(3)
plot(meancurrents)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
xlabel('Time (Min.)')
ylabel('SC. currents (mA)')
savemyfig3

% APPLYING Meinel and Meinel MODEL *****
[I,Ireal]=model(day);

% CONVERTING FROM POWER TO SHORT CIRCUIT CURRENT
a=max(meancurrents(:,11));
b=max(I(day,:));
c=a/b;

I2=I(day,:)*c;
figure(4)
plot(T_m,I2)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
xlabel('Time (Min.)')
ylabel('Solar Radiation (mA)')
savemyfig3

figure(5)
plot(T_m,Ireal(day,:))
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
xlabel('Time (Min.)')
ylabel('Solar Radiation (kW/m2)')
savemyfig3

% COVARIANCE, CORRELATION COEFFICIENTS AND VARIANCE
COVARIANCE=cov(meancurrents);

```

```

CORRELATION=corrcoef(meancurrents);
FF=CORRELATION;
varia=var(meancurrents);

% Slopes bet. sensors
[p11_14,p11_8,p11_12,p11_10,p15_14,p15_11,p15_12,p13_10,p13_11,p13_14,p
9_8,p9_11,p9_12,p7_8,p7_10,p7_11,p17_13,p17_14,p17_15,p18_14,p18_15,...

p18_17,p16_13,p16_14,p16_17,p5_7,p5_8,p5_9,p6_5,p6_8,p6_9,p4_5,p4_7,p4_
8,p20_16,p20_17,p20_18,p19_16,p19_17,p19_20,p21_6,p21_17,p21_20,p2_4,p2
_5,p2_6,p3_2,p3_5,p3_6,p1_2,p1_4,p1_5,p23_19,p23_20,p23_21,p22_19,p22_2
0,p22_23,p24_23,p24_20,p24_21]=slopes(meancurrents);

% MEAN AND STANDARD DEVIATION FOR ALL SENSORS
ave=mean(meancurrents); %mean values
m=length(meancurrents); %length of the data file
stan=sqrt(COVARIANCE); %standard deviation

start=375;
endd=952; % REJECTING THE NOISE
L=375:952;

% Normally distributed pseudorandom numbers
E=randn(length(L),24);

% DATA EXPECTING AND CORRELATION
maximump=max(I2);

for i=start:endd
    A=meancurrents(i-4,11)-I2(1,i-4-78);
    B=meancurrents(i-3,11)-I2(1,i-3-78);
    C=meancurrents(i-2,11)-I2(1,i-2-78);
    D=meancurrents(i-1,11)-I2(1,i-1-78);

    expected=LinearRegression(i,A,B,C,D);

    i_expected=expected+I2(i+4-78);

    i11(i)=i_expected;
    i14(i)=i11(i)*((p11_14(1)+FF(14,11))/2)+(stan(11,14)*E(i-
start+1,14));
    i8(i)=i11(i)*((p11_8(1)+FF(8,11))/2)+(stan(11,8)*E(i-start+1,8));
    i12(i)=i11(i)*((p11_12(1)+FF(12,11))/2)+(stan(11,12)*E(i-
start+1,12));
    i10(i)=i11(i)*((p11_10(1)+FF(10,11))/2)+(stan(11,10)*E(i-
start+1,10));

i15(i)=((i14(i)*(p15_14(1)+FF(14,15))+i11(i)*(p15_11(1)+FF(11,15))+i12(

```

$$i) * (p15_12(1) + FF(12, 15)) / (p15_14(1) + FF(14, 15) + p15_11(1) + FF(15, 11) + p15_12(1) + FF(15, 12)) + (stan(15, 14) * E(i - start + 1, 15) + stan(15, 11) * E(i - start + 1, 15) + stan(15, 12) * E(i - start + 1, 15));$$

$$i13(i) = ((i14(i) * (p13_14(1) + FF(14, 13)) + i11(i) * (p13_11(1) + FF(11, 13)) + i10(i) * (p13_10(1) + FF(10, 13))) / (p13_14(1) + FF(13, 14) + p13_11(1) + FF(13, 11) + p13_10(1) + FF(13, 10))) + (stan(13, 14) * E(i - start + 1, 13) + stan(13, 11) * E(i - start + 1, 13) + stan(13, 10) * E(i - start + 1, 13));$$

$$i9(i) = ((i8(i) * (p9_8(1) + FF(8, 9)) + i11(i) * (p9_11(1) + FF(11, 9)) + i12(i) * (p9_12(1) + FF(12, 9))) / (p9_8(1) + FF(9, 8) + p9_11(1) + FF(9, 11) + p9_12(1) + FF(9, 12))) + (stan(9, 8) * E(i - start + 1, 9) + stan(9, 11) * E(i - start + 1, 9) + stan(9, 12) * E(i - start + 1, 9));$$

$$i7(i) = ((i8(i) * (p7_8(1) + FF(8, 7)) + i11(i) * (p7_11(1) + FF(11, 7)) + i10(i) * (p7_10(1) + FF(10, 7))) / (p7_8(1) + FF(7, 8) + p7_11(1) + FF(7, 11) + p7_10(1) + FF(7, 10))) + (stan(7, 8) * E(i - start + 1, 7) + stan(7, 11) * E(i - start + 1, 7) + stan(7, 10) * E(i - start + 1, 7));$$

$$i17(i) = ((i14(i) * (p17_14(1) + FF(14, 17)) + i13(i) * (p17_13(1) + FF(13, 17)) + i15(i) * (p17_15(1) + FF(15, 17))) / (p17_14(1) + FF(17, 14) + p17_13(1) + FF(17, 13) + p17_15(1) + FF(17, 15))) + (stan(17, 13) * E(i - start + 1, 17) + stan(17, 14) * E(i - start + 1, 17) + stan(17, 15) * E(i - start + 1, 17));$$

$$i18(i) = ((i17(i) * (p18_17(1) + FF(17, 18)) + i15(i) * (p18_15(1) + FF(15, 18)) + i14(i) * (p18_14(1) + FF(14, 18))) / (p18_17(1) + FF(18, 17) + p18_15(1) + FF(18, 15) + p18_14(1) + FF(18, 14))) + (stan(18, 17) * E(i - start + 1, 18) + stan(18, 15) * E(i - start + 1, 18) + stan(18, 14) * E(i - start + 1, 18));$$

$$i16(i) = ((i14(i) * (p16_14(1) + FF(14, 16)) + i17(i) * (p16_17(1) + FF(17, 16)) + i13(i) * (p16_13(1) + FF(13, 16))) / (p16_14(1) + FF(16, 14) + p16_13(1) + FF(16, 13) + p16_17(1) + FF(16, 17))) + (stan(16, 14) * E(i - start + 1, 16) + stan(16, 13) * E(i - start + 1, 16) + stan(16, 17) * E(i - start + 1, 16));$$

$$i5(i) = ((i9(i) * (p5_9(1) + FF(9, 5)) + i8(i) * (p5_8(1) + FF(8, 5)) + i7(i) * (p5_7(1) + FF(7, 5))) / (p5_9(1) + FF(5, 9) + p5_8(1) + FF(5, 8) + p5_7(1) + FF(5, 7))) + (stan(5, 9) * E(i - start + 1, 5) + stan(5, 8) * E(i - start + 1, 5) + stan(5, 7) * E(i - start + 1, 5));$$

$$i6(i) = ((i5(i) * (p6_5(1) + FF(5, 6)) + i8(i) * (p6_8(1) + FF(8, 6)) + i9(i) * (p6_9(1) + FF(9, 6))) / (p6_5(1) + FF(6, 5) + p6_9(1) + FF(6, 9) + p6_8(1) + FF(6, 8))) + (stan(6, 5) * E(i - start + 1, 6) + stan(6, 9) * E(i - start + 1, 6) + stan(6, 8) * E(i - start + 1, 6));$$

$$i4(i) = ((i5(i) * (p4_5(1) + FF(5, 4)) + i7(i) * (p4_7(1) + FF(7, 4)) + i8(i) * (p4_8(1) + FF(8, 4))) / (p4_5(1) + FF(4, 5) + p4_7(1) + FF(4, 7) + p4_8(1) + FF(4, 8))) + (stan(4, 5) * E(i - start + 1, 4) + stan(4, 7) * E(i - start + 1, 4) + stan(4, 8) * E(i - start + 1, 4));$$

$$i20(i) = ((i18(i) * (p20_18(1) + FF(18, 20)) + i17(i) * (p20_17(1) + FF(17, 20)) + i16(i) * (p20_16(1) + FF(16, 20))) / (p20_18(1) + FF(18, 20) + p20_17(1) + FF(17, 20) + p20_16(1) + FF(16, 20))) + (stan(20, 18) * E(i - start + 1, 20) + stan(20, 17) * E(i - start + 1, 20) + stan(20, 16) * E(i - start + 1, 20));$$

```
i) * (p20_16(1)+FF(16,20)) / (p20_16(1)+FF(20,16)+p20_17(1)+FF(20,17)+p20_18(1)+FF(20,18)) + (stan(20,16)*E(i-start+1,20)+stan(20,17)*E(i-start+1,20)+stan(20,18)*E(i-start+1,20));
```

```
i19(i) = ((i16(i) * (p19_16(1)+FF(16,19)) + i17(i) * (p19_17(1)+FF(17,19)) + i20(i) * (p19_20(1)+FF(20,19))) / (p19_16(1)+FF(19,16)+p19_17(1)+FF(19,17)+p19_20(1)+FF(19,20)) + (stan(19,16)*E(i-start+1,19)+stan(19,17)*E(i-start+1,19)+stan(19,20)*E(i-start+1,19));
```

```
i21(i) = ((i18(i) * (p21_18(1)+FF(18,21)) + i17(i) * (p21_17(1)+FF(17,21)) + i20(i) * (p21_20(1)+FF(20,21))) / (p21_18(1)+FF(21,18)+p21_17(1)+FF(21,17)+p21_20(1)+FF(21,20)) + (stan(21,18)*E(i-start+1,21)+stan(21,17)*E(i-start+1,21)+stan(21,20)*E(i-start+1,21));
```

```
i2(i) = ((i4(i) * (p2_4(1)+FF(4,2)) + i6(i) * (p2_6(1)+FF(6,2)) + i5(i) * (p2_5(1)+FF(5,2))) / (p2_4(1)+FF(2,4)+p2_5(1)+FF(2,5)+p2_6(1)+FF(2,6)) + (stan(2,4)*E(i-start+1,2)+stan(2,5)*E(i-start+1,2)+stan(2,6)*E(i-start+1,2));
```

```
i3(i) = ((i2(i) * (p3_2(1)+FF(2,3)) + i5(i) * (p3_5(1)+FF(5,3)) + i6(i) * (p3_6(1)+FF(6,3))) / (p3_2(1)+FF(3,2)+p3_5(1)+FF(3,5)+p3_6(1)+FF(3,6)) + (stan(3,2)*E(i-start+1,3)+stan(3,5)*E(i-start+1,3)+stan(3,6)*E(i-start+1,3));
```

```
i1(i) = ((i2(i) * (p1_2(1)+FF(2,1)) + i4(i) * (p1_4(1)+FF(4,1)) + i5(i) * (p1_5(1)+FF(5,1))) / (p1_2(1)+FF(1,2)+p1_4(1)+FF(1,4)+p1_5(1)+FF(1,5)) + (stan(1,2)*E(i-start+1,1)+stan(1,4)*E(i-start+1,1)+stan(1,5)*E(i-start+1,1));
```

```
i23(i) = ((i21(i) * (p23_21(1)+FF(21,23)) + i19(i) * (p23_19(1)+FF(19,23)) + i20(i) * (p23_20(1)+FF(20,23))) / (p23_21(1)+FF(23,21)+p23_19(1)+FF(23,19)+p23_20(1)+FF(23,20)) + (stan(23,21)*E(i-start+1,23)+stan(23,19)*E(i-start+1,23)+stan(23,20)*E(i-start+1,23));
```

```
i22(i) = ((i23(i) * (p22_23(1)+FF(23,22)) + i19(i) * (p22_19(1)+FF(19,22)) + i20(i) * (p22_20(1)+FF(20,22))) / (p22_23(1)+FF(22,23)+p22_19(1)+FF(22,19)+p22_20(1)+FF(22,20)) + (stan(22,23)*E(i-start+1,22)+stan(22,19)*E(i-start+1,22)+stan(22,20)*E(i-start+1,22));
```

```
i24(i) = ((i23(i) * (p24_23(1)+FF(23,24)) + i21(i) * (p24_21(1)+FF(21,24)) + i20(i) * (p24_20(1)+FF(20,24))) / (p24_23(1)+FF(24,23)+p24_21(1)+FF(24,21)+p24_20(1)+FF(24,20)) + (stan(24,23)*E(i-start+1,24)+stan(24,21)*E(i-start+1,24)+stan(24,20)*E(i-start+1,24));
```

```
i1n=i1(i);
i1_f(i)=Data_Filtering(maximump,i1n);
```

end

```

predicted_data=[i1',i2',i3',i4',i5',i6',i7',i8',i9',i10',i11',i12',i13'
,i14',i15',i16',i17',i18',i19',i20',i21',i22',i23',i24'];

```

```

% DTAT PLOTTING

```

```

figure(6)
plot([1+4:1101+4],meancurrents)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
ylabel('SC. Current (mA)')
xlabel('Time (min.)')
savemyfig3

```

```

figure(7)
plot(predicted_data)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
ylabel('SC. Current (mA)')
xlabel('Time (min.)')
savemyfig3

```

```

%-----

```

```

figure(8)
xlim([409 500.01])
ylim([0.79 1.201])
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
ylabel('SC. Current (mA)')
xlabel('Time (min.)')
savemyfig3

```

```

%-----

```

```

figure(9)
f=plot(i1,'r');
hold on
ff=plot(i1_f,'b');
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
ylabel('SC. Current (mA)')
xlabel('Time (min.)')
set(f,'Displayname','Without Filtering')
set(ff,'Displayname','With Filtering')
legend('Location','west')
savemyfig3

```

```

%-----
%-----

%FUNCTIONS

% (1) MODEL

function [I,Ireal]=model(day)

data=csvread('all20sen.csv', 96291, 3, [96291 3 128210 40]);
sc_current=data(:,5:28);
t_m=data(:,1)*60+data(:,2);
t_s=(data(:,1)*60*60)+(data(:,2)*60)+(data(:,3));
T_m=[min(t_m):max(t_m)];

% N = %number of the day
% T = time in hours
lat= 37.9; %latitude
longitude=91.76; %longitude
% declination=23.45*sin((360/365)*(N-80));

    m=length(T_m);

for N=day;
declination=23.45*sin((360/365)*(N-80)*(pi/180));
i=0;
for mm=1:m
for T=T_m(mm)/60 % we can use min(T_m)/60:1/60:max(T_m)%
    w=15*(12-T);
sin_alpha(N,i+1)=sin(declination*(pi/180))*sin(lat*(pi/180))...
    +cos(declination*(pi/180))*cos(lat*(pi/180))*cos(w*(pi/180));
alpha1(N,i+1)=asin(sin_alpha(N,i+1))*(180/pi)-8.3;

sin_alpha1(N,i+1)=sin(alpha1(N,i+1)*(pi/180));
i=i+1;
end
end
end

for N=day
for i=1:m
    if sin_alpha1(N,i)<0
        sin_alpha2(N,i)=0;
    else
        sin_alpha2(N,i)=sin_alpha1(N,i);
    end
end
end
end

```



```

for N=day
    for i=1:m
        am(N,i)=1/sin_alpha2(N,i);

        I(N,i)=(1367*(0.7)^( (am(N,i) )^(0.678) ))*(1.807/949.4); %SHORT
CIRCUIT CURRENT SCALE
    end
end
Ireal=I/(1.807/949.4); % POWER SCALE

% (2) LINEAR REGRESSION

function [expected]=LinearRegression(i,A,B,C,D)

diff1(1)=A;
diff1(2)=B;
diff1(3)=C;
diff1(4)=D;

z=i+4;

slop=(4*((z-1)*diff1(1)+(z-2)*diff1(2)+(z-3)*diff1(3)+(z-
4)*diff1(4))-((z-1)+(z-2)+(z-3)+(z-
4))* (diff1(1)+diff1(2)+diff1(3)+diff1(4)))/(4*((z-1)^2+(z-2)^2+(z-
3)^2+(z-4)^2)-(((z-1)+(z-2)+(z-3)+(z-4))^2));

b=((diff1(1)+diff1(2)+diff1(3)+diff1(4))-slop*((z-1)+(z-2)+(z-
3)+(z-4)))/4;

expected=slop*(z)+b;

% (3) SLOPES

function
[p11_14,p11_8,p11_12,p11_10,p15_14,p15_11,p15_12,p13_10,p13_11,p13_14,p
9_8,p9_11,p9_12,p7_8,p7_10,p7_11,p17_13,p17_14,p17_15,p18_14,p18_15,...

p18_17,p16_13,p16_14,p16_17,p5_7,p5_8,p5_9,p6_5,p6_8,p6_9,p4_5,p4_7,p4_
8,p20_16,p20_17,p20_18,p19_16,p19_17,p19_20,p21_6,p21_17,p21_20,p2_4,p2
_5,p2_6,p3_2,p3_5,p3_6,p1_2,p1_4,p1_5,p23_19,p23_20,p23_21,p22_19,p22_2
0,p22_23,p24_23,p24_20,p24_21]=slopes(meancurrents)

% Finding slopes among sensors
p11_8=polyfit(meancurrents(:,11),meancurrents(:,8),1);

```

```
p11_12=polyfit(meancurrents(:,11),meancurrents(:,12),1);
p11_10=polyfit(meancurrents(:,11),meancurrents(:,10),1);
p11_14=polyfit(meancurrents(:,11),meancurrents(:,14),1);

p15_14=polyfit(meancurrents(:,15),meancurrents(:,14),1);
p15_11=polyfit(meancurrents(:,15),meancurrents(:,11),1);
p15_12=polyfit(meancurrents(:,15),meancurrents(:,12),1);

p13_10=polyfit(meancurrents(:,13),meancurrents(:,10),1);
p13_11=polyfit(meancurrents(:,13),meancurrents(:,11),1);
p13_14=polyfit(meancurrents(:,13),meancurrents(:,14),1);

p9_8=polyfit(meancurrents(:,9),meancurrents(:,8),1);
p9_11=polyfit(meancurrents(:,9),meancurrents(:,11),1);
p9_12=polyfit(meancurrents(:,9),meancurrents(:,12),1);

p7_8=polyfit(meancurrents(:,7),meancurrents(:,8),1);
p7_10=polyfit(meancurrents(:,7),meancurrents(:,10),1);
p7_11=polyfit(meancurrents(:,7),meancurrents(:,11),1);

p17_13=polyfit(meancurrents(:,17),meancurrents(:,13),1);
p17_14=polyfit(meancurrents(:,17),meancurrents(:,14),1);
p17_15=polyfit(meancurrents(:,17),meancurrents(:,15),1);

p18_14=polyfit(meancurrents(:,18),meancurrents(:,14),1);
p18_15=polyfit(meancurrents(:,18),meancurrents(:,15),1);
p18_17=polyfit(meancurrents(:,18),meancurrents(:,17),1);

p16_13=polyfit(meancurrents(:,16),meancurrents(:,13),1);
p16_14=polyfit(meancurrents(:,16),meancurrents(:,14),1);
p16_17=polyfit(meancurrents(:,16),meancurrents(:,17),1);

p5_7=polyfit(meancurrents(:,5),meancurrents(:,7),1);
p5_8=polyfit(meancurrents(:,5),meancurrents(:,8),1);
p5_9=polyfit(meancurrents(:,5),meancurrents(:,9),1);

p6_5=polyfit(meancurrents(:,6),meancurrents(:,5),1);
p6_8=polyfit(meancurrents(:,6),meancurrents(:,8),1);
p6_9=polyfit(meancurrents(:,6),meancurrents(:,9),1);

p4_5=polyfit(meancurrents(:,4),meancurrents(:,5),1);
p4_7=polyfit(meancurrents(:,4),meancurrents(:,7),1);
p4_8=polyfit(meancurrents(:,4),meancurrents(:,8),1);

p20_16=polyfit(meancurrents(:,20),meancurrents(:,16),1);
p20_17=polyfit(meancurrents(:,20),meancurrents(:,17),1);
p20_18=polyfit(meancurrents(:,20),meancurrents(:,18),1);

p19_16=polyfit(meancurrents(:,19),meancurrents(:,16),1);
```

```

p19_17=polyfit(meancurrents(:,19),meancurrents(:,17),1);
p19_20=polyfit(meancurrents(:,19),meancurrents(:,20),1);

p21_18=polyfit(meancurrents(:,21),meancurrents(:,18),1);
p21_17=polyfit(meancurrents(:,21),meancurrents(:,17),1);
p21_20=polyfit(meancurrents(:,21),meancurrents(:,20),1);

p2_4=polyfit(meancurrents(:,2),meancurrents(:,4),1);
p2_5=polyfit(meancurrents(:,2),meancurrents(:,5),1);
p2_6=polyfit(meancurrents(:,2),meancurrents(:,6),1);

p3_2=polyfit(meancurrents(:,3),meancurrents(:,2),1);
p3_5=polyfit(meancurrents(:,3),meancurrents(:,5),1);
p3_6=polyfit(meancurrents(:,3),meancurrents(:,6),1);

p1_2=polyfit(meancurrents(:,1),meancurrents(:,2),1);
p1_4=polyfit(meancurrents(:,1),meancurrents(:,4),1);
p1_5=polyfit(meancurrents(:,1),meancurrents(:,5),1);

p23_19=polyfit(meancurrents(:,23),meancurrents(:,19),1);
p23_21=polyfit(meancurrents(:,23),meancurrents(:,21),1);
p23_20=polyfit(meancurrents(:,23),meancurrents(:,20),1);

p22_19=polyfit(meancurrents(:,22),meancurrents(:,19),1);
p22_23=polyfit(meancurrents(:,22),meancurrents(:,23),1);
p22_20=polyfit(meancurrents(:,22),meancurrents(:,20),1);

p24_21=polyfit(meancurrents(:,24),meancurrents(:,21),1);
p24_23=polyfit(meancurrents(:,24),meancurrents(:,23),1);
p24_20=polyfit(meancurrents(:,24),meancurrents(:,20),1);

figure(10)
hold on
t=0:2;
y=polyval(p1_5,t);
plot(meancurrents(:,1),meancurrents(:,5),'.k',t,y)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
xlabel('Ish1 (mA)')
ylabel('Ish5 (mA)')
text(.1,1.8,sprintf('Slope= %f',p1__5(1)))
savemyfig3

figure(11)
t=0:2;
y=polyval(p4_8,t);
plot(meancurrents(:,1),meancurrents(:,5),'.b',t,y)

```

```

set(gcf, 'Position', [200 200 4.5*96 3*96])
set(gca, 'FontName', 'Times')
set(gca, 'FontSize', [10])
xlabel('Ish4 (mA)')
ylabel('Ish8 (mA)')
text(.1,1.8,sprintf('Slope= %f',p4__8(1)))
savemyfig3

figure(12)
t=0:2;
y=polyval(p2_6,t);

plot(meancurrents(:,1),meancurrents(:,5),'.r',t,y)
set(gcf, 'Position', [200 200 4.5*96 3*96])
set(gca, 'FontName', 'Times')
set(gca, 'FontSize', [10])
xlabel('Ish2 (mA)')
ylabel('Ish6 (mA)')
text(.1,1.8,sprintf('Slope= %f',p2__6(1)))
savemyfig3

% (4) FILTER

function [il_f]=Data_Filtering(maximump,iln)

    if iln>maximum*1.2
        il_f=0.7*iln;
    else
        il_f=iln;
    end
end

% -----
% Comparison Between Difference and Division for Different Window Size
% on April 1st, 2012

% SUBTRACTING AND DIVISION
start=336;
endd=923;
for y=start:endd
    diff1(y)=meancurrents(y,1)-I2(y);
end

for y=start:endd
    div(y)=meancurrents(y,1)/I2(y);
end

% MSE ERROR
error1=mse(diff1);

```

```

error2=mse(div);

figure(13)

hold on
f=plot(T_m,I2,'r');
ff=plot(T_m,meancurrents(:,1));
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
set(f,'Displayname','Model')
set(ff,'Displayname','Actual')
legend('Location','west')
xlabel('Time (Min.)')
ylabel('SC. Current (mA)')
line([445;445],[0;2])
line([1034;1034],[0;2])
text(460,0.1,'X1=445 Min.')
text(800,0.1,'X2=1034.')
hold off
savemyfig3

figure(14)

hold on
f=plot(T_m,I2,'r');
ff=plot(T_m,meancurrents(:,1));
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
set(f,'Displayname','Model')
set(ff,'Displayname','Actual')
legend('Location','west')
xlabel('Time (Min.)')
ylabel('SC. Current (mA)')
hold off
savemyfig3

figure(15)
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[9])
plot(diff1)
text(350,0.15,sprintf('Mean Squared Error= %f',error1))
hold on
plot(div,'r')
text(350,1.15,sprintf('Mean Squared Error= %f',error2))
ylabel('          Actual Data - Model          Actual Data / Model')
xlabel('Time (Min.)')
savemyfig3

% LINEAR REGRESSION
%A) Difference

j=length(diff1);

```

```

>window=3
for z=4:j

    slop=(3*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3))-((z-1)+(z-2)+(z-3))*(diff1(z-1)+diff1(z-2)+diff1(z-3)))/(3*((z-1)^2+(z-2)^2+(z-3)^2)-((z-1)+(z-2)+(z-3))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3))-slop*((z-1)+(z-2)+(z-3)))/3;

    expected1(z-3,:)=slop*(z)+b;

end

>window=4
for z=5:j

    slop=(4*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3)+(z-4)*diff1(z-4))-((z-1)+(z-2)+(z-3)+(z-4))*(diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)))/(4*((z-1)^2+(z-2)^2+(z-3)^2+(z-4)^2)-((z-1)+(z-2)+(z-3)+(z-4))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4))-slop*((z-1)+(z-2)+(z-3)+(z-4)))/4;

    expected2(z-4,:)=slop*(z)+b;

end

>window=5
for z=6:j

    slop=(5*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3)+(z-4)*diff1(z-4)+(z-5)*diff1(z-5))-((z-1)+(z-2)+(z-3)+(z-4)+(z-5))*(diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5)))/(5*((z-1)^2+(z-2)^2+(z-3)^2+(z-4)^2+(z-5)^2)-((z-1)+(z-2)+(z-3)+(z-4)+(z-5))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5))-slop*((z-1)+(z-2)+(z-3)+(z-4)+(z-5)))/5;

    expected3(z-5,:)=slop*(z)+b;

end

>window=6
for z=7:j

    slop=(6*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3)+(z-4)*diff1(z-4)+(z-5)*diff1(z-5)+(z-6)*diff1(z-6))-((z-1)+(z-2)+(z-3)+(z-4)+(z-5)+(z-6))*(diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5)+diff1(z-6)))/(6*((z-1)^2+(z-2)^2+(z-3)^2+(z-4)^2+(z-5)^2+(z-6)^2)-((z-1)+(z-2)+(z-3)+(z-4)+(z-5)+(z-6))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5)+diff1(z-6))-slop*((z-1)+(z-2)+(z-3)+(z-4)+(z-5)+(z-6)))/6;

```

```

    expected4(z-6,:) = slop*(z)+b;

end

%B) Division

diff_stor=diff1;
diff1=div;
j=length(div);

>window=3
for z=4:j

    slop=(3*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3))-((z-1)+(z-2)+(z-3))*(diff1(z-1)+diff1(z-2)+diff1(z-3)))/(3*((z-1)^2+(z-2)^2+(z-3)^2)-((z-1)+(z-2)+(z-3))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3))-slop*((z-1)+(z-2)+(z-3)))/3;

    expecteda(z-3,:) = slop*(z)+b;

end

>window=4
for z=5:j

    slop=(4*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3)+(z-4)*diff1(z-4))-((z-1)+(z-2)+(z-3)+(z-4))*(diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)))/(4*((z-1)^2+(z-2)^2+(z-3)^2+(z-4)^2)-((z-1)+(z-2)+(z-3)+(z-4))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4))-slop*((z-1)+(z-2)+(z-3)+(z-4)))/4;

    expectedb(z-4,:) = slop*(z)+b;

end

>window=5
for z=6:j

    slop=(5*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3)+(z-4)*diff1(z-4)+(z-5)*diff1(z-5))-((z-1)+(z-2)+(z-3)+(z-4)+(z-5))*(diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5)))/(5*((z-1)^2+(z-2)^2+(z-3)^2+(z-4)^2+(z-5)^2)-((z-1)+(z-2)+(z-3)+(z-4)+(z-5))^2);

    b=((diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5))-slop*((z-1)+(z-2)+(z-3)+(z-4)+(z-5)))/5;

    expectedc(z-5,:) = slop*(z)+b;

end

```

```

>window=6
for z=7:j

    slop=(6*((z-1)*diff1(z-1)+(z-2)*diff1(z-2)+(z-3)*diff1(z-3)+(z-
4)*diff1(z-4)+(z-5)*diff1(z-5)+(z-6)*diff1(z-6))-((z-1)+(z-2)+(z-3)+(z-
4)+(z-5)+(z-6))*(diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-
5)+diff1(z-6)))/(6*((z-1)^2+(z-2)^2+(z-3)^2+(z-4)^2+(z-5)^2+(z-6)^2)-
(((z-1)+(z-2)+(z-3)+(z-4)+(z-5)+(z-6))^2));
b=((diff1(z-1)+diff1(z-2)+diff1(z-3)+diff1(z-4)+diff1(z-5)+diff1(z-6))-
slop*((z-1)+(z-2)+(z-3)+(z-4)+(z-5)+(z-6)))/6;

    expectedd(z-6,:)=slop*(z)+b;

end

%Adding and multiplying the expected noise to the solar radiation
model:

>window=3
m=length(expected1);
for i=start:endd-3

windo3_diff(i)=expected1(i)+I2(i+3);
windo3_div(i)=expecteda(i)*I2(i);

end

>window=4
m=length(expected2);
for i=start:endd-4

windo4_diff(i)=expected2(i)+I2(i+4);
windo4_div(i)=expectedb(i)*I2(i);

end

>window=5
m=length(expected3);
for i=start:endd-5

windo5_diff(i)=expected3(i)+I2(i+5);
windo5_div(i)=expectedc(i)*I2(i);

end

>window=6
m=length(expected4);

for i=start:endd-6

windo6_diff(i)=expected4(i)+I2(i+6);
windo6_div(i)=expectedd(i)*I2(i);

```



```

end

%Subtracting the expected trends from the actual readings
>window=3
for i=start:endd-3
    diff_after_expect_diff3(i)=meancurrents(i+2,1)-windo3_diff(i);

    diff_after_expect_div3(i)=meancurrents(i+2,1)-windo3_div(i);
end
>window=4
for i=start:endd-4
    diff_after_expect_diff4(i)=meancurrents(i+3,1)-windo4_diff(i);

    diff_after_expect_div4(i)=meancurrents(i+3,1)-windo4_div(i);
end
>window=5
for i=start:endd-5
    diff_after_expect_diff5(i)=meancurrents(i+4,1)-windo5_diff(i);

    diff_after_expect_div5(i)=meancurrents(i+4,1)-windo5_div(i);
end
>window=6

for i=start:endd-6
    diff_after_expect_diff6(i)=meancurrents(i+5,1)-windo6_diff(i);

    diff_after_expect_div6(i)=meancurrents(i+5,1)-windo6_div(i);
end

%           MSE FOR ALL
>window=3
msed3=mse(diff_after_expect_diff3);
msedv3=mse(diff_after_expect_div3);
>window=4
msed4=mse(diff_after_expect_diff4);
msedv4=mse(diff_after_expect_div4);
>window=5
msed5=mse(diff_after_expect_diff5);
msedv5=mse(diff_after_expect_div5);
>window=6
msed6=mse(diff_after_expect_diff6);
msedv6=mse(diff_after_expect_div6);

%-----
----
figure(16)

subplot(4,2,1)
plot(diff_after_expect_diff3);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(200,0.2,'Window size=3')
text(200,-0.3,sprintf('MSE= %f',msed3))
%*****

```

```

subplot(4,2,3)
plot(diff_after_expect_diff4);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=4')
text(150,-0.3,sprintf('MSE= %f',msed4))
%*****
subplot(4,2,5)
plot(diff_after_expect_diff5);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=5')
text(150,-0.3,sprintf('MSE= %f',msed5))
%*****
subplot(4,2,7)
plot(diff_after_expect_diff6);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=6')
text(150,-0.3,sprintf('MSE= %f',msed6))
%*****
subplot(4,2,2)
plot(diff_after_expect_div3);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=3')
text(150,-0.3,sprintf('MSE= %f',msedv3))
%*****
subplot(4,2,4)
plot(diff_after_expect_div4);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=4')
text(150,-0.3,sprintf('MSE= %f',msedv4))
%*****
subplot(4,2,6)
plot(diff_after_expect_div5);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=5')
text(150,-0.3,sprintf('MSE= %f',msedv5))
%*****
subplot(4,2,8)
plot(diff_after_expect_div6);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(150,0.2,'Window size=6')
text(150,-0.3,sprintf('MSE= %f',msedv6))
savemyfig3
%*****

```

```

figure(17)
f=plot(meancurrents(:,1),'k');
hold on
ff=plot([3:end-1],windo3_diff,'r');
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
set(f,'Displayname','Actual')
set(ff,'Displayname','Expected data')
legend('Location','northwest')
xlabel('Time(min.)')
ylabel('Actual data &Expected data (mA)')
savemyfig3

figure(18)
plot(diff_after_expect_diff3);
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[10])
text(200,0.07,'Window size=3')
text(200,-0.1,sprintf('MSE= %f',msed3))
xlabel('Time(min.)')
ylabel('Difference between Actual data &Expected data')
savemyfig3

figure(19)

hold on
k=1:length(diff_stor);
f=plot(k,diff_stor);
ff=plot([3:max(k)-1], expected1,'r');
set(gcf,'Position',[200 200 4.5*96 3*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[11])
set(f,'Displayname','Actual data-Model')
set(ff,'Displayname','Regression Model')
legend('Location','northwest')

xlabel('Time (min.)')
ylabel('Data-Model')
hold off
savemyfig3

```

REFERENCES

- [1] R. A. M. a. J. Ventre, *Photovoltaic Systems Engineering*, 3 ed: CRC Press, pp. 12-42, 2010.
- [2] N. Nagarajan, "Development of a graphical user interface for the study of parallel-connected solar array," Missouri University of Science and Technology, 2011.
- [3] M. J. A. a. G. N. Tiwari, "Study of Models for Prediction the Mean Hourly Global Radiation from Daily Summation," *Open Environmental Sciences*, vol. 2, pp. 34-38, 2008.
- [4] J. W. a. C. K. Chan, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN " *Solar Energy*, pp. 808-817, January. 2011.
- [5] A. K. K. a. C. K. Pandey, "Correlation for estimation of hourly solar radiation," *International Energy & Environment Foundation*, vol. 2, pp. 191-198, 2011.
- [6] M. J. A. a. G. N. Tiwari, "Estimation of Hourly Global Solar Radiation for Composite Climate," *Open Environmental Sciences*, vol. 2, pp. 6-14, 2008.
- [7] J.-L. Domenech-Garret, "Cell behaviour under different non-uniform temperature and radiation combined profiles using a two dimensional element model," *Solar Energy*, pp. 256-264, 2011.
- [8] C. C. Wu Ji, JW Loh, FH Choo, LH Chen, "Solar Radiation Prediction Using Statistical Approaches," to be presented at International Conference on Information and Communication Systems, 2009.
- [9] K. Bakirci, "Correlation for estimation of daily global solar radiation with hours of bright sunshine in Turkey," *Elsevier Ltd.*, pp. 485-501, 2009.
- [10] M. A. a. M. F. M. Hassanzadeh, "Practical Approach for Sub-Hourly and Hourly prediction of PV Power Output," to be presented at National Science Foundation 2010.
- [11] G. R. a. M. G. M. Nijhuis, "Classification technique to quantify the significance of partly cloudy conditions for reserve requirements due to photovoltaic plants," *PowerTech, IEEE Trondheim*, pp. pp. 1-7, 2011.
- [12] J. W. K. a. B. Yount, "Managing a solar array project: Analysing insulation & motivation," pp. 50.
- [13] "SolarMade Company," accessed on, <http://www.solarmade.com/>.

- [14] "Tern Inc.," accessed on, <http://www.tern.com/portal/content.asp?contentid=689>.
- [15] X. Y. a. X. Su, *Linear Regression Analysis: Theory and Computing*: World Scientific, pp. 2-17, 2009.

VITA

Faris Ebrahim Alfaris was born on November 15, 1986 in Riyadh, Saudi Arabia. He completed his schooling in Prince Faisal bin Turki, Sudeer, Saudi Arabia. Faris received his Bachelor of Engineering (B.E.) degree in Electrical and Control from the King Saud University, Riyadh, Saudi Arabia in February 2009.

He began his Master of Science program in Electrical Engineering at Missouri University of Science and Technology in August 2010. He obtained his Master's degree in December 2012.

