**Dereck Musyimi Mutungi**

# Degradation of photovoltaics in Central Finland: A comparative study of polycrystalline and heterojunction with intrinsic thin layer technologies

Master's Thesis

Masters Degree Programme in Renewable Energy

September 13, 2013

University of Jyväskylä

Department of Physics

**Author:** Dereck Musyimi Mutungi

**Contact information:** `dereck.m.mutungi@student.jyu.fi` , `musyimi@gmail.com`

**Supervisor:** Dr. Jussi Maunuksela

**Title:** Degradation of photovoltaics in Central Finland: A comparative study of polycrystalline and heterojunction with intrinsic thin layer technologies

**Project:** Master's Thesis

**Study line:** Masters Degree Programme in Renewable Energy (Applied Physics)

**Page count:** 81+80

**Abstract:** Photovoltaics have gained popularity since the 2000's. More photovoltaic (PV) systems are being connected to electricity grids thanks to introduction of feed-in tariffs and reduced PV costs. The quality of electricity and lifespan of the photovoltaic module are important in determining the viability of running a photovoltaic system.

Two photovoltaic technologies were investigated, namely: Polycrystalline located at Agora in Jyväskylä and Heterojunction with Intrinsic Thin layer (HIT) located at Saarijärvi. Both sites are in Central Finland. Performance analysis was done using data recorded over three and seven years at Agora and Saarijärvi, respectively using performance ratios and yields.

On average, the performance ratio of the HIT system was higher than polycrystalline system especially below $200\,\mathrm{W/m^2}$ of solar irradiance. The HIT technology had the higher efficiency of the two technologies with between 17.3 % and 19.7 % array efficiency. It has been concluded that the topology of the system is strongly correlated to its performance. Therefore, the topology with the better performance ratio was a hybrid of parallel and series arrangement as seen in Saarijärvi subsystem one (ss1).

The degradation rate of the PV systems using the AutoRegressive Integrated Moving Average (ARIMA) model were found to be $(-0.10 \pm 0.65)$ %/year; $(-0.29 \pm 0.16)$ %/year and $(-0.36 \pm 0.16)$ %/year for Agora, Saarijärvi ss1 and ss2 respectively. The value obtained from Agora was not conclusive and a recommendation of a four year cycle be used to check the degradation rate. Investigation into the causes of degradation of the PV at the Saarijärvi site should be done. In addition the data recording format for Saarijärvi should be revised in order to improve the precision of parameters recorded. The aim is to eliminate the above 100 % inverter efficiency at low irradiance.

Both systems were calculated to have working lives equal to and greater than 25 years.

**Keywords:** photovoltaic, HIT, polycrystalline, degradation, Central Finland, field monitoring, long term performance, heterojunction with intrinsic thin layer, array topology.

# Preface

This work is in partial fulfilment of the Masters Degree Programme in Renewable energy.

I would like to express my gratitude towards my supervisor, Dr Jussi Maunuksela for his guidance and support during my thesis work and the entire study period.

I am grateful to Sharon Gitau for her encouragement to strive for greater heights and Peter 'fly-B2k' Mayavi for reminding me to keep my eye on the goal.

I would like to thank Anna 1 & 2, Charlie, Cyril, Jason, Paulus, Markus, Petri, Ruth, Miisa, Tiinu, Filip, Tero, the entire RE department and many others who I may have failed to mention here by name for their patience, sound advice and friendship. It means a lot.

Last but not least, I would like to thank my parents and siblings for their constant support.

> "Don't know what a slide rule is for - but I do know one and one is two, and if this thesis could impress you, oh what a wonderful master's degree.. la la la la."
> - Sarah Faber.

Jyväskylä September 13, 2013

Dereck Mutungi

# Glossary

| | |
|---|---|
| AC | Alternating current |
| c-Si | Monocrystalline |
| DC | Direct current |
| $E_A$ | Energy from array |
| $E_{in}$ | Total energy input from all power sources |
| $E_{II}$ | Energy into inverter |
| $E_{io}$ | Energy out of inverter |
| $E_{tu}$ | Energy to utility/grid |
| $E_{S,A}$ | In-plane solar energy |
| $F_A$ | Photovoltaic fraction of the total energy input |
| $G_I$ | In-plane irradiance |
| $G_{I,ref}$ | Reference irradiance (1000 W/m$^2$) |
| HIT | Heterojunction with Intrinsic Thin layer |
| $L_c$ | Capture loss |
| $L_s$ | System loss |
| M | Monitoring fraction |
| O | Outage fraction |
| $P_a$ | Power from array |
| $P_o$ | Rated nominal power of array |
| $P_{tu}$ | Power to utility/grid |
| PC | Personal computer |
| PV | Photovoltaic |
| PR | Performance Ratio |
| p-Si | Polycrystalline |
| Si | Silicon |
| Sjarvi | Saarijärvi |
| STC | Standard Test Conditions |
| ss1 | Saarijärvi subsystem 1 |
| ss2 | Saarijärvi subsystem 2 |

| | |
|---|---|
| T | Temperature |
| VA | VoltAmpere |
| $Y_a$ | Array yield |
| $Y_f$ | Final yield |
| $Y_r$ | Reference yield |
| $\eta_I$ | Inverter efficiency |
| $\eta_A$ | Array efficiency |
| $\eta_{tot}$ | Overall efficiency |

# Contents

# 1 Introduction

Energy independence has been the aim of many countries especially in the wake of periodic and unforeseen oil and gas supply disruptions. This coupled with the awareness of effects of increased carbon and greenhouse gas emissions on global warming, has thrust renewable energy into the limelight. The primary source of all the energy on earth is the sun, both directly and indirectly. Improved harnessing of this resource directly would be a step closer to cleaner, affordable and to an extent, a reliable energy source.

Photovoltaic cells (also called solar cells or PV cells) are devices that convert certain wavelengths (less than 1100 nm) of the sun's radiation directly into electricity. There are various solar cell technologies available currently but the silicon wafer-based PV are the most installed and with the largest market share of approximately 85 % as of 2011 [1]. Silicon based PVs are common in many parts of the world due to their relative low cost of manufacture and the abundance of silicon element on earth.

Researchers have strived and continue to work in developing PV cells that are affordable and efficient, that is, as much of the incoming light is utilised as possible in the production of electricity. There is also work on increasing the longevity of the cells and modules. In this thesis work, two silicon based PV technologies will be studied under climatic conditions found in Central Finland. Central Finland is located above the $60^{th}$ north latitude. The photovoltaic systems being investigated are located in Jyväskylä and Saarijärvi. Polycrystalline silicon technology was used in the photovoltaic array in Jyväskylä while heterojunction with intrinsic thin layer technology was used in Saarijärvi photovoltaic system. The Saarijärvi PV system has two subsystems with the same PV technology but differing array topologies. The details of both sites are given in the next section while a full system description is given in the methodology chapter. Both systems have been running for more than three years with data collection done concurrently. The Saarijärvi PV system has been running for four years longer than the Jyväskylä PV system.


Figure 1: The label 'A' is the position of Saarijärvi while the bold dot is Jyväskylä [2].

## 1.1 PV systems locations

A PV system is made up of multiple components namely: PV array, inverters (Balance of systems), wires and cables and in some cases a monitoring system and/or battery storage. The main focus for this research is the PV array and to some extent the inverter will be discussed but not in detail. Figure 1 shows the locations of the two PV systems under study. The polycrystalline PV system is located in Jyväskylä on the rooftop of the Agora building of the University of Jyväskylä at a latitude of 62.2 °N and at an altitude of 95 m. The HIT[1] PV system is located in Saarijärvi on the rooftop of a local school building with a latitude of 62.7 °N and at an elevation of 129 m. At these latitudes, the solar hours per day vary from as little as 4 hours to 20 hours depending on the time of year. In addition, both locations

[1] Heterojunction with intrinsic thin layer

experience snow fall and temperature drops of as low as -30 °C in the winter.

In reference to Köppen-Geiger climate classification, Finland is classified as *Dfb* and *Dfc*. *Dfb* means that for 4 months in a year the temperature is above 10 °C but below 22 °C while *Dfc* means the coldest month has temperatures above -38 °C but cannot fall into *Dfb* category which is cold without dry season and with warm or cold summer depending on location [3]. Central Finland is classified *Dfb*. The classification helps in gaining a general idea of the ambient temperatures to be found in Central Finland. The ambient temperature is a factor that influences the performance of the photovoltaic systems as we shall see later on in chapter four.

## 1.2 Objective

A long term performance analysis of the PV systems under climatic conditions found in Central Finland will be done using data collected over a minimum of three years. This will show how much electricity the PV systems have produced and at what efficiency. The main objective of this study is to find the degradation rate of the photovoltaic systems in Jyväskylä and Saarijärvi using the analysed data. The purpose is to find out if the PV systems are deteriorating and if so, at what rate.

In addition I shall make a comparative study between the two photovoltaic technologies. The purpose is to find out which of the two technologies works better under the field conditions in Central Finland. I shall compare the two different configurations or topologies at the Saarijärvi site to establish if array topology has an influence on electric power production.

## 1.3 Thesis outline

In chapter two, a general overview of the photovoltaic technologies to be studied, that is the polycrystalline and heterojunction with intrinsic thin layer. I will discuss the current status of PV in the world, followed by a literature review on degradation of photovoltaics. I will also touch on the warranties and guarantees and relate it to the topic of degradation.

In chapter three, the methodology used in analysing data will be introduced. Additional information on the PV systems will also be provided. Afterwards, how the data was processed and the results and discussion will be given in chapter four.

The final chapter will present the conclusions that were found and recommendations will also be given.

# 2 Literature Review

## 2.1 Photovoltaic technologies

### 2.1.1 Polycrystalline silicon (p-Si)

Polycrystalline or multicrystalline silicon is the name given to silicon crystals of a certain grain size. It was defined by Basore [4] as having a grain size of less than 1 mm but greater than 1 µm. The processing technique used in making polycrystalline differs from single crystalline silicon (c-Si). In conventional c-Si, the wafers are cut from a single crystalline ingot that has been grown whereas p-Si are made from melted and moulded silicon. The difference in processing results in less material wastage in p-Si production but less orderly crystals being produced. Hence p-Si cells are cheaper to manufacture than c-Si wafers because the production technique is faster when using less orderly placed crystals. The result is lower cell efficiency in p-Si than c-Si. However, this does not diminish the appeal for p-Si. Figure 2 is a microscopic illustration of the p-Si.

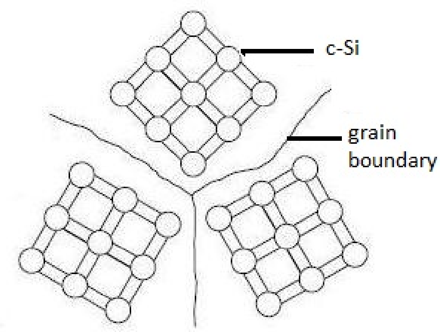



Figure 2: Regions of crystalline Si separated by 'grain boundaries' where bonding is irregular. Adapted from Applied photovoltaics [5].

Polycrystalline silicon cells have been viewed as the low cost alternative to c-Si cells since 1980's. However because of their low efficiency they did not attract much attention from investors and manufacturers until in 1990's when a cell efficiency of 35 % under laboratory conditions was announced [6]. The production and installed capacity has been on the rise ever since.

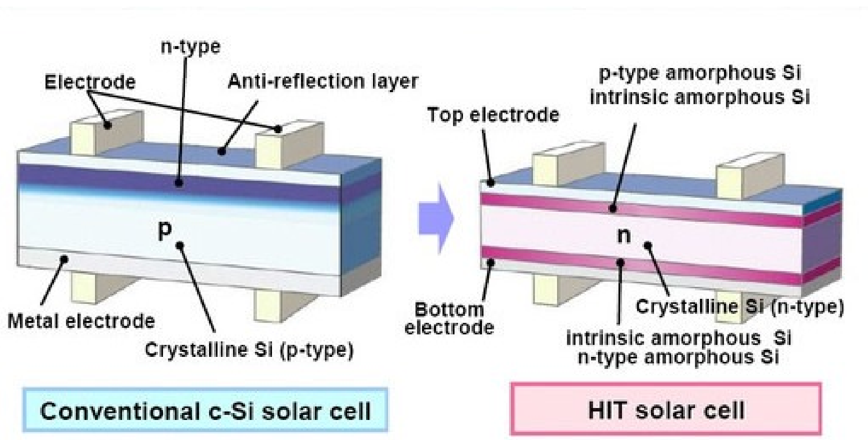



Figure 3: The structural difference between a conventional c-Si and HIT solar cell [7]

The p-Si cell takes on a shimmery appearance in the light because of the irregular crystal bonding. The p-Si PV cells cover a greater surface area on a PV module because it is moulded rather than sawn from a grown crystal. Hence the p-Si are usually uniformly shaped squares or rectangles as opposed to c-Si PV cells which usually have chamfered corners. The typical efficiency of commercially available p-Si cells ranges from 12 % to 17 %.

### 2.1.2 Heterojunction with Intrinsic Thin layer (HIT) cells

Heterojunction with intrinsic thin layer solar cells may also be referred to as silicon heterojunction solar cells. This type of cell is classified as a hybrid solar cell. Developed by Sanyo company in Japan in 1992, it is made up of thick and thin layers of material. Figure 3 shows the structural difference between a conventional c-Si and HIT which is in the number of layers and the make-up of the layers. HIT has a c-Si n-type bounded by a very thin amorphous silicon layers.

The aim of the multi-junction structure is to increase the cell efficiency and to lower manufacturing temperatures to around 200 °C which reduces the silicon degradation during conventional processing (>800 °C) [8]. In addition there are savings on energy costs during manufacturing. Sanyo claims that less materials were used in making the PV cells since HIT are thinner by 150 µm than conventional silicon solar cells [9]. The intrinsic amorphous silicon layer offers contact to both sides of the c-Si, while providing passivation and extra stability to the system [10]. Passivation is defined as growth of an oxide layer on the surface of a semiconductor to provide electrical stability by isolating the transistor surface from electrical and chemical conditions in the environment. This reduces reverse-current leakage, increases breakdown voltage, and raises power dissipation rating [11].

HIT are better known for their relatively higher cell and module efficiency as compared to conventional monocrystalline. In 2009, Sanyo announced that it had achieved a conversion efficiency of 23 % and were researching into bifacial modules [12]. In the field, HIT has been noted to perform better than most other technologies. In France, HIT was noted to perform exceptionally well [13]. Using performance index as a performance measure, Leloux et al. [13] reported HIT having a performance index of 88.7 % while other c-Si ranged from 79.3 % to 87.9 %. Sasitharanuwat et al. [14] reported the HIT had the highest efficiency and the least module temperature for the same irradiance level of three technologies that were being investigated. The other two technologies are p-Si and amorphous silicon. The same result was also found in India with the same three technologies tested [15]. What would be of interest is the performance of the HIT in a location like Central Finland where unlike France and Thailand, the irradiation and ambient temperature are comparatively lower.

HIT cells have a darker and more uniform look when compared to p-Si cells. HIT cells in a module often do not occupy as much space as the p-Si cells. In general, commercial HIT cells are more efficient than p-Si with cell efficiency ranging from 16 % to 21 %.

## 2.2 Current status of photovoltaics

Manufacturing and installed capacity of PV systems have increased significantly over the past 12 years. In 2000 the PV installed capacity was less than one $GW_p$[1] and it grew to 70 $GW_p$ by the end of 2011 [1]. Currently, the country with one of the highest growth rates in manufacturing is China. Typical factors contributing to increased dissemination of this form of renewable energy are:

- The reduced cost of manufacture due to improved manufacturing techniques.

[1]Gigawatt peak power

- Research and innovation has increased the application of the solar cell and module. An example is solar chargers for laptops and mobile phones; street lamps powered by solar panels;
- Public awareness of environmental pollution and protection.
- Government initiatives towards energy independence. Concurrently, reducing their carbon emission especially countries who have signed the Kyoto protocol.
- Introduction of feed-in tariffs in some countries have caused an increase in the installed capacity. For example in Israel between 2008 and 2011, capacity went up by 130 MW [1].

In light of the nuclear disaster in 2011 at Fukushima Daiichi, governments such as Germany have planned to phase out their nuclear power plants by 2022 [16]. The energy difference is going to be filled by other energy sources such as solar. In addition, previous experience with oil supply disruptions and price fluctuations makes the push for renewable energy even stronger.

Although solar power use is on the rise, it still is not yet cost competitive to other forms of energy. Traditional energy sources such as fossil and nuclear energy have been cheap to produce per kilowatt hour. However their prices are predicted to be on the rise as the amount of oil and uranium supply decreases due to finite resources. The price of solar electricity has been on the downward trend for the past decade because of government support in form of subsidies and tax exemptions, reduced manufacturing costs and improved module efficiencies. However, more informative data needs to be presented on field-tested systems. Studies of how long a PV system can be viable, how much electricity it can produce during its life time and performance at various climatic conditions will give confidence to financiers on return on investment, which will lead to greater capital access for home owners and companies who would want to switch to a green source of energy.

## 2.3 Performance characterisation

At a consumer level, certain questions require answers. These questions are such as: how much electricity can I get out of a PV system; how much does the system cost; what is the efficiency and how long will the system payback time be. Many manufacturers have provided the answers to most of these questions. However, answers to some of the questions are not satisfactory especially when comparing two or more systems of different technology, in different locations and of different sizes.

To make comparison easier, a guideline for assessing, analysing and presenting data was published in 1995 by the institute for systems engineering and informatics which is part of the European Commission's Joint research centre (JRC) in Ispra, Italy. The aim of the guide was to standardize reporting of data from PV plants. A key factor in the guide is the use of normalised energy which is the ratio of the energy output to the peak power capacity of the PV plant or reference solar irradiation (in the case of solar energy) over a given period such as a day, month or year. Normalising energy is convenient in analysing systems [17]. The performance indicators that were established will be presented in detail in section 3.3.

It should be noted that the performance indicators developed by JRC are not universal. In the US, the long term performance analysis is done with Photovoltaics for Utility Scale Application (PVUSA) rating analysis. As a result many publications from Europe often use performance ratio (PR) while the ones from United States of America use PVUSA rating. Other parts of the world may use PR or normalised energy without necessarily using terminology defined by the JRC. In some instances, both PR and PVUSA are used to compare the same system [18, 19]. Marion et al. [19] presented that the degradation rates calculated using the PR and PVUSA were not different in value and also pointed out that the PR is useful for identifying existing issues. For this reason, change in PR value would be a useful indicator of degradation in a PV system.

## 2.4 Degradation

Degradation affects the performance of the PV system in terms of its efficiency, the life span and the quality of the energy output. A material is termed as *degraded* when its properties have been eroded or worn out physically and/or the chemical complexity has been reduced [20]. Degradation may occur by physical and/or chemical modes. Chemical degradation occurs through chemical reactions of the materials such as rusting of the support structure or yellowing of the laminate on the solar module. The effects are often irreversible. In physical degradation, the interaction of two physical entities causes the degradation. An example of a physical degradation is bulking of an array support structure due to the weight of the PV modules.

According to Edson and Dyk [21], the main modes of degradation are: front surface soiling; optical degradation; cell degradation; mismatched cells; light induced degradation and temperature induced degradation. The majority of the degradation modes are inevitable due to the exposure of the system to the weather elements. Considering the location of the PV systems, during winter, snow cover on the PV array causes front surface soiling and if left to accumulate may exert weight on the array panels and structure. In the summer, light induced degradation is more prominent than in the winter.

### 2.4.1 Degradation measurement methods

Several analytical methods have been used in various papers [18, 22–25] to quantify degradation. The methods used to describe the degradation rate of PV systems include I-V (current-voltage) measurements, performance ratio analysis (least square fit, classical decomposition, moving average) and Photovoltaic for utility scale applications (PVUSA).

Each method has merits and demerits. Performance ratio and PVUSA use continuous data for analysis while I-V curve use data that is periodically measured. I-V is best done under predetermined set conditions or STC[2]. Emery [26] discussed some of the disadvantages of I-V measurement. He points out that variation in reference cell calibration, correction for spectral variation can affect the measurements. He further questions the accuracy and repeatability of the I-V measurements. Although, much improvement has been done since the publication of this study, still some questions arise such as the PV cell area definition and whether from simulator to simulator the reference cell is calibrated according to a common standard. Another disadvantage is the cost. It is expensive to rent a solar simulator that can measure modules or arrays. I-V curve tracers used in the field are useful in fault detection but not accurate in long term degradation rate determination. Furthermore, many of the re-

[2]Standard Test Conditions

ports published using the I-V curve method use one or two readings and compare it with the manufacturer's specifications at STC and not with first time readings from the system after installation. This produces significant errors [27]. In the PVUSA analysis, the calculations are relatively more complex than PR calculation because the former takes into account irradiation, windspeed and temperature while the later does not take account the temperature or windspeed. For the purpose of this study, I shall use the performance ratio for analysing the PV systems.

The determination of a suitable duration required for a PV system to be monitored in order to get an accurate degradation rate and the type of method to analyse data is continuous. As technology is being developed, manufacturers and researchers want to use less time to test before launching a product. Apart from accelerated tests, field tests are used to deteremine degradation rates and are considered to be the closest to real application situations. Osterwald et al. [28] suggested that the adequate period to determine an acceptably accurate degradation rate should be no less than three years. Jordan and Kurtz [24] further reiterated this by stating that complete observation cycles (3 to 5 years) were required to get accurate degradation rates. In addition, they concluded that ARIMA (AutoRegressive Integrated Moving Average) method of analysing data was the most robust in comparison to classical time series decomposition and linear fit to adjusted data. ARIMA takes into account seasonal variations, data shifts due to changes within the hardware and it is also not over-sensitive to outlier data. Dunea et al. [29] succesfully used ARIMA in the forecasting energy output and found it useful for analysis. It is the work by Dunea et al. that helped Jordan and Kurtz [24] to consider ARIMA as a statistical method to analyse time series data.

### 2.4.2 Significance of degradation studies

Degradation of all machines and structures is inevitable but the rate at which the degradation takes place is of importance. Knowing the causes and devising preventive measures help the PV systems to last a longer time and increase their reliability. The research into degradation has far reaching implications beyond just the realm of the energy production. An example is the warranty period, which depends on the degradation rate of the PV panel which is linked to accessibility to financial capital. Singh et al. [30] asked for concrete results from researchers so as to increase confidence for loan applications to purchase PV panels.

With an increased number of PV systems being connected to grids, the quality of the energy is important. Coupled with the performance and life expectancy of the system, this influences the system's *levelized cost of electricity*[3] (LCOE); that is used for cost comparison between different energy sources. A thorough analysis would involve capital cost, operations and maintenance costs, fuel costs, performance, discount cost, degradation cost and replacement cost [31]. Caution should be exercised when doing a comparison since it depends on assumptions, investment conditions and technology. For PV, the initial cost of the system and interest will be the bulk of the cost since no maintenance is done to the system. If it is proven that the lifetime of the system is longer than the warranty, the LCOE reduces. But for financial purposes, the manufacturers have to adjust the warranties accordingly to the life expectancy of the PV. This will result in improved investor confidence in PVs.

### 2.4.3 Relationship of degradation rate with warranties and guarantees

First, we need to differentiate between the warranty and guarantee. A guarantee is usually free and is a promise about an item by the manufacturer or company while a warranty acts like an insurance policy for which you must pay a premium [32]. In addition, a warranty is a legal contract but a guarantee is a promise to solve certain problems and issues. Some manufacturers use the terms freely and interchangeably. An example is both terms are found in the Sanyo technical datasheet and are used interchangeably while the Naps website [33], only uses the word warranty. It should be noted that the end of the warranty period is not synonymous to the end of the lifetime of the module.

Many PV module manufacturers offer warranties such as product, workmanship and power warranties [9, 33]. In this study, power warranty will be considered. It ranges from 10 to 25 years for the PV systems in question. Values provided by PV manufacturers are obtained from field studies, qualification and accelerated testing, and statistical projections. Qualification and accelerated tests may not wholly be reliable. An instance was in 2006 when PV failure was caused by electrical arcing and fires in module junction boxes due to inade-

[3] This is the total cost of the system divided by the lifetime of the system.

quate soldering during manufacturing [34]. The PV modules had undergone thorough tests, however at that time there was no testing procedure for detecting that type of fault.

The reason for this subsection is to highlight the differences between the companies when it comes to guaranteed power after a certain period. Skoczek et al. [35] and Vázquez et al. [36] mention many manufacturers offering warranties of 80 % of the nominal maximum power or an equivalent of 20 to 25 years lifetime. However, some manufacturers use 80 % of minimum output power on delivery as basis for their power warranty. This difference leaves the consumer susceptible to being taken advantage of. In a hypothetical case if a customer returns a Sanyo module rated at 190 W that is performing at 140 W after 15 years, the manufacturer can claim that the company is still within its guaranteed power supply and refuse the returned module. From the Sanyo technical datasheet [9], the maximum power is 190 W, the warranted minimum power on delivery is 180.5 W while there is an output of tolerance of +10/$-5$ %. The power warranty is for 20 years (80 % of the minimum power). Therefore the minimum power that can be guaranteed by the manufacturer at the end of the warranty period is:

$$180.5\text{W} \times 80\% \times 95\% = 137.18\text{W}$$

where 80 % is percentage of the minimum power at the end of the warranty period, 95 % is the lower output percentage (This was found by using the lower power output tolerance of $-5$ %.). This (137.18 W) is the floor value, below which the module is considered to be a liability. It is about 73.7 % of the maximum power. The module can have a much longer lifetime than the guarantee. It all comes down to the requirement of the end-user. For power warranties of 25 or more years to be achieved, the degradation rate has to be less than 0.5 % per year [36]. Estimating the lifetime of a PV is not easy. According to Skoczek et al. [35], after 20 years of field aging, out of 204 modules only 35 had failed the warranty criteria (90 % of maximum power after 10 years and 80 % of maximum power after 20 years). In addition, they noted lifetime of the PV is not marked by a catastrophic failure at a fixed point in time but rather by degradation.

Figure 4 shows increasing numbers of publication over the years and changing module warranty periods. As the systems got older and the number of publications increased, more information was found. The red line indicates the manufacturers' warranty period also in-

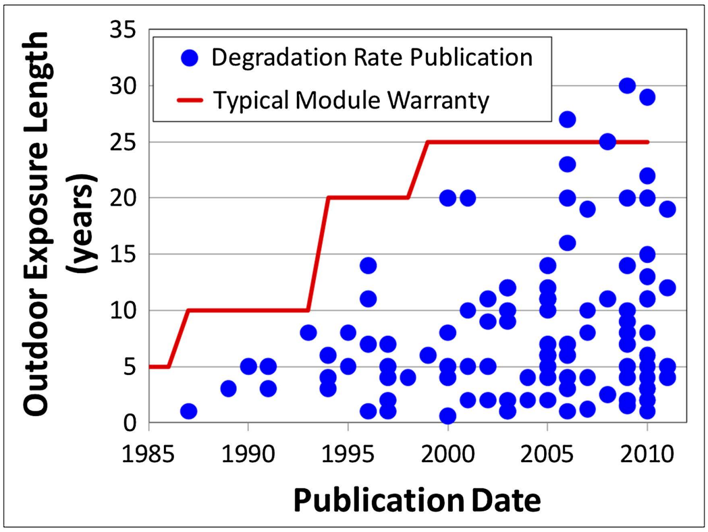



Figure 4: The outdoor exposure length against the year of publication of degradation rate [27]

creasing periodically in line with the outdoor exposure period. The behaviour of the PV systems have been documented but waiting for 30 years is not an option to determine the degradation rate in the field. Faster and accurate ways have to be used and use of the performance indicators and ARIMA would provide one option.

## 2.5 Degradation rates reported for northern latitudes

The term *northern latitude*, has been used for areas above the 40°N. Although Finland is at a high latitude, the difference in global yearly irradiation (calculated using EC PVGIS[4] [37]) solar resources between Finland (1054 kW h/$m^2$) and Germany (1157 kW h/$m^2$) is not large. The main difference in the dissemination of PVs in Germany and Finland is feed-in tariffs [37]. The colder temperatures experienced in Central Finland would be an advantage in terms of power production because generally cell efficiency increases as ambient temperatures decrease. Figure 5 shows geographical distribution of reported degradation rates with the circle sizes indicating the number of published reports at various locations. The majority of publications on degradation are from Japan, Central Europe, Middle East and North America. There is only one published report from a site that has a higher latitude than

[4]European Commission Photovoltaic Geographical Information System

Jyväskylä which is Nunavut, Canada (63.4 °N).

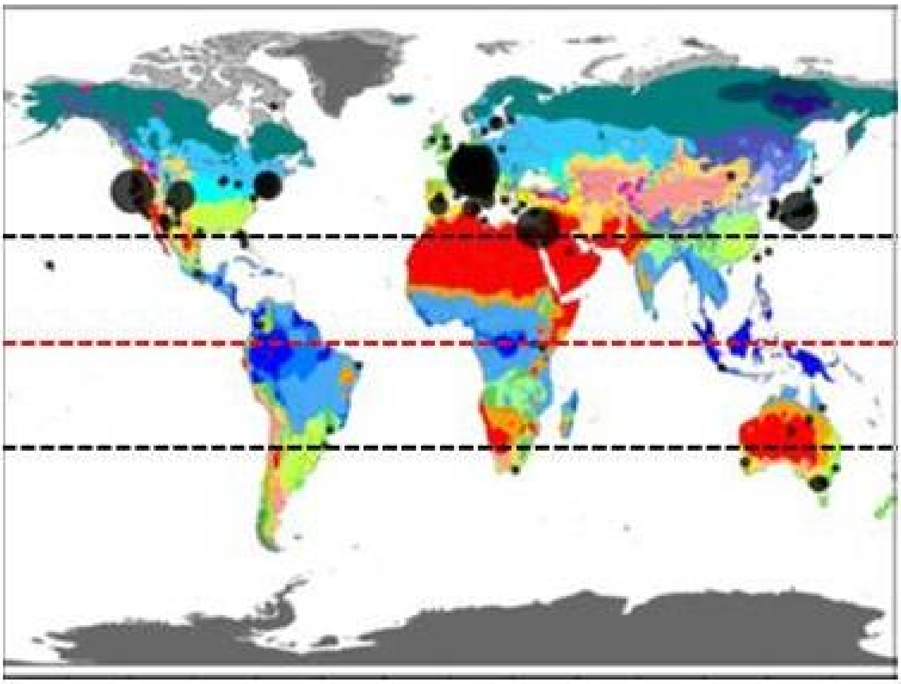


Figure 5: Geographic distribution of reported degradation rates overlaid on a Köppen-Geiger climate map with the equator and the tropic of Cancer and Capricorn [38]

The oldest of the silicon based technologies is the monocrystalline and it has the longest monitoring periods of all the PV types. Polycrystalline have not been left far behind in terms of monitoring. Table 1 summarises some of the findings presented in literature for polycrystalline arrays. In summary, none of the papers had noted any signs of visual degradation of the arrays. Palmblad et al. [40] went as far as conservatively stating the module looked as a good as new after 22 years of service. So far there has been no paper encountered that reported the performance of a polycrystalline array or system at a latitude above 62°N.

The relatively young age of the HIT means that long term monitoring such as those seen for mono and polycrystalline panels have not been encountered. Jiang and Lim [43] reported PR value of 0.75-0.8 and array efficiency between 14.6-15.1 % but no degradation rate since it was only for a one year period and the location was Singapore (1.35°N). Makrides et al. [18] presented degradation rates for different PV technologies. They reported that HIT

[5] Verma et al. state the polycrystalline was performing within 90 % of the rated power in the 10th year.

Table 1: Locations of p-Si systems and reported degradation rates

| Location | latitude | Altitude (m) | Monitoring period (years) | Method of evaluation | Degradation rate |
|---|---|---|---|---|---|
| Ispra, Italy [39] | 45.8°N | 220 | 20 | I-V | 4.42 % in 20 years |
| Huvudsta, Sweden [40] | 59.4°N | 17 | 22 | I-V | 2.0 % in over 22 years |
| Casaccia, Italy [41] | 44.4°N | 925 | 10 | Array efficiency trend line | 0.4 % per year |
| Grimstad, Norway [42] | 58.1°N | 33 | 10 | I-V | Not stated explicitly[5] |

had a first year degradation of -4.73 % and subsequent degradation of -0.10%/year. For the multicrystalline PV, the degradation rates were higher for the first year period as compared to three-year period with values of -2.40 %/year to -1.47 %/year and -0.06 %/year to 0.01 %/year, respectively. The method used was PR and PVUSA with outage and irradiance filtering (>800 W/m$^2$). The site was in Cyprus (35°N). So far no paper was found that presented long term field test performance of HIT at latitude above 60 °N.

Its worth noting that many of the research papers reporting performance use the I-V characteristic as a measure of the performance of the module.

## 2.6 Effect of location and climate on the PV system performance

Central Finland has low solar insolation (in comparison to the tropics or Central Europe), low average temperatures and snow fall between 60 cm and 90 cm at its deepest. With these factors, a PV array that works well in low irradiance and inverter that has good conversion efficiency at low partial load is recommended.

In cold conditions, PV cells work better because the heat generated in the cells is dissipated away faster. The electrons vibrate less and the bandgap energy increases. This has the effect of reducing the short circuit current but increasing the open circuit voltage. The magnitude of the voltage change is much greater than that of the current change. Ultimately, the fill factor[6] is improved and the maximum power greater. Ross and Royer [44] state that peak power and open-circuit voltage for most types of modules tend to improve by approximately 0.3 % to 0.5 % for every degree Celsius drop in cell temperature.

Rarely does the field conditions match STC which are ambient temperature of 25 °C, wind speed of $1\,\mathrm{m\,s^{-1}}$ and $1000\,\mathrm{W/m^2}$ irradiation. Field conditions in Central Finland are characterised by low irradiance level therefore the amount of power produced would not frequently reach the peak energy value. At low irradiance, the requirement of having an inverter that has minimum start up losses is of importance. Fig. 6 shows the efficiency curves for three different types of inverters as plotted by Notton et al. [45]. The curves show the differences in the three inverters at varied partial loads of the inverters. Type 1 has the largest start up loss while type 3 has the largest load loss. Type 2 is the most ideal with the least start up and load losses. Although, type 2 is the best, selecting an inverter involves more than just efficiency but also cost. For instance, if a large percentage energy into the inverter is at low partial load then using type 3 would be adequate at the expense of good performance at higher partial loads; if type 3 is cheaper than type 2. The inverter efficiency would influence the final output power so the inverter selection is important.

There is a temptation of having undersized inverters to save on costs when the amount of irradiance level is low. But Burger and Rüther [46] point out that undersizing may result in energy losses especially with PV technologies with high temperature coefficients of power operating at sites in cold climates.

In summary, the literature review has presented the background information required and now the technique or method used to process and analyse data will be presented in the following chapter.

[6] Fill factor is the ratio between the maximum achiveable power of the solar cell/module to the product of its open circuit voltage and short circuit current.

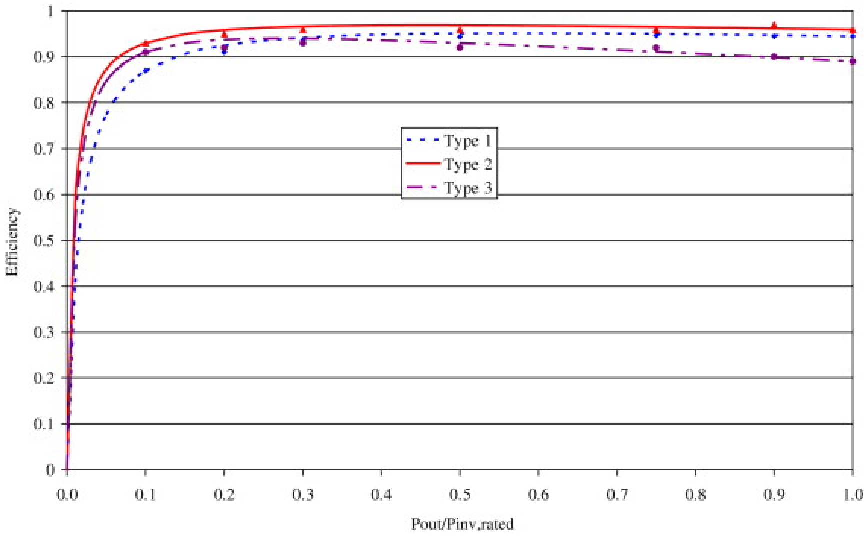



Figure 6: Inverter efficiencies of three different types [45]

# 3 Methodology

This chapter contains the description of the PV systems, data recording formats and methods used to analyse the data.

## 3.1 Description of PV systems

### 3.1.1 Agora photovoltaic system

The Agora building is situated in the Mattilanniemi campus of the University of Jyväskylä. Mounted on its roof on one rack are 20 Naps NP130GK photovoltaic modules each containing 36 polycrystalline silicon cells. The modules are connected in series. The array is free-standing, inclined at 40° and south facing. The system is grid-connected through a Sunny Boy SB-2500 inverter. The array did not have any visible degradation upon inspection. Table 2 below shows a summary of the system.

Table 2: Summary of the Agora PV system [47].

| | |
|---|---|
| **System name** | Agora PV system |
| **Technology** | Polycrystalline silicon |
| **Manufacturer and model** | Naps systems Oy, NP130GK |
| **Date of installation** | Fall 2008 |
| **Starting date of data collection** | $17^{th}$ June 2009 |
| **System size (W)** | 2600 |
| **Area** ($m^2$) | 19.83 |
| **Number of modules** | 20 |
| **Module efficiency ( %)** | 13.1 |
| **Nominal module maximum power (W)** | 130 |

The Sunny Boy SB-2500 inverter has a maximum efficiency of 94.1 % or 93.2 % Euro-eta[1] efficiency [48]. It has a maximum AC apparent power output of 2500 V A and nominal power

[1] This is a performance criterion for inverters.

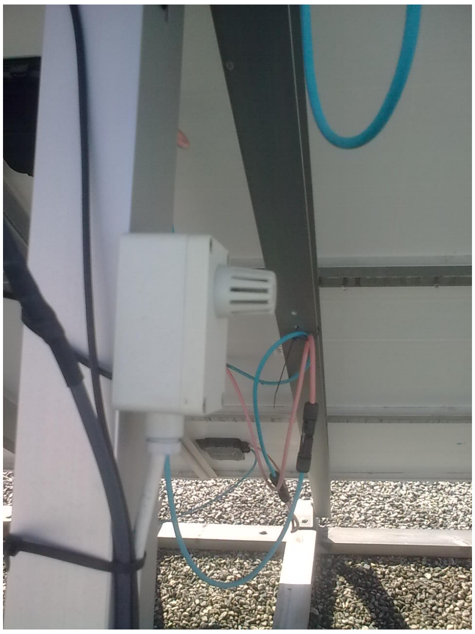
Figure 7: A photograph of the Agora polycrystalline PV system.

output of 2300 W [48]. The inverter is connected to a Sunny Boy Control datalogger. There was no maintenance done on the solar panels during its operation including snow clearing in the winter.

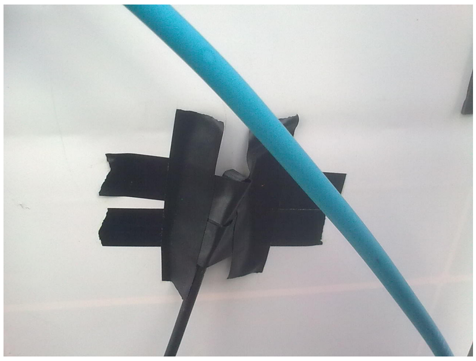
The ambient temperature is measured using a thermometer located on the back support structure of the array (Figure 8(a)) while the cell temperature is measured using a thermocouple attached to the back side of the PV module as shown in Figure 8(b). The solar irradiance is measured using a reference cell that is located on the same plane as the array. The thermometer, thermocouple and reference cell are connected to the Sunny Boy Control.

### 3.1.2 Saarijärvi photovoltaic system

The PV system in Saarijärvi is mounted on a local school's roof and it is also free-standing. The Saarijärvi PV system has HIT Sanyo solar modules connected to two subsystems: (ss1) 27 modules with three parallel strings each containing nine modules in series with a total power rating of 5130 W and (ss2) six modules in series and a total power rating of 1140



(a) Thermometer.



(b) Thermocouple.

Figure 8: (a) The thermometer was used to measure the ambient temperature and (b) the thermocouple was used to measure the cell temperature of the Agora PV system, respectively.

W.[2] Each of the subsystems were connected to different inverters. Figure 10 shows the configuration of the PV system. The inverters used were Fronius IG: the smaller of the two subsystems was connected to Fronius IG 15 with maximum efficiency of 94.2 % and a European efficiency of 91.4 % [49] while the larger subsystem was connected to a Fronius IG 60 with a maximum efficiency of 94.5 % and European efficiency of 93.5 %. Like the Agora PV system, this system did not undergo regular maintenance. Table 3 shows a summary of the system.

Ambient & cell temperatures, wind speed and irradiance were measured using Fronius PT 1000 temperature sensors, cup anemometer and monocrystalline Si-sensor (irradiance), respectively. These devices were connected to a sensor card located in the Fronius IG1 (inverter: Fronius IG 60).

The system has encountered a number of data interruptions. Approximately 270 days worth

[2] Often the installed power is calculated using the minimum power upon delivery which is about 95 % of the maximum rated power. For the sake of uniformity in analysis, I have used maximum rated power only.

Table 3: Summary of the Saarjärvi PV system.

| | |
|---|---|
| **System name** | Saarjärvi school PV system |
| **Technology** | HIT |
| **Manufacturer and model** | Sanyo, HIP-190NE1 |
| **Date of installation** | Fall 2005 |
| **Starting date of data collection** | $7^{th}$ October 2005 |
| **System size (W)** | 6270 |
| **Area** ($m^2$) | 37.95 |
| **Number of modules** | 33 |
| **Module efficiency ( %)** | 16.5 |
| **Nominal module maximum power (W)** | 190 |

Figure 9: The PV system on the roof of the local school building in Saarijärvi [50].

of data is missing. The years for which data was lost were 2007, 2008, 2010 and 2011. In 2007 and 2008, the computer monitoring the system crashed and data recorded for approximately 190 days was lost. In 2007, the data was lost in late spring and summer when the PV system is usually most active. All the years except 2009 and 2012, have seen also inverter errors which have compromised data recording and a further 80 days were lost.

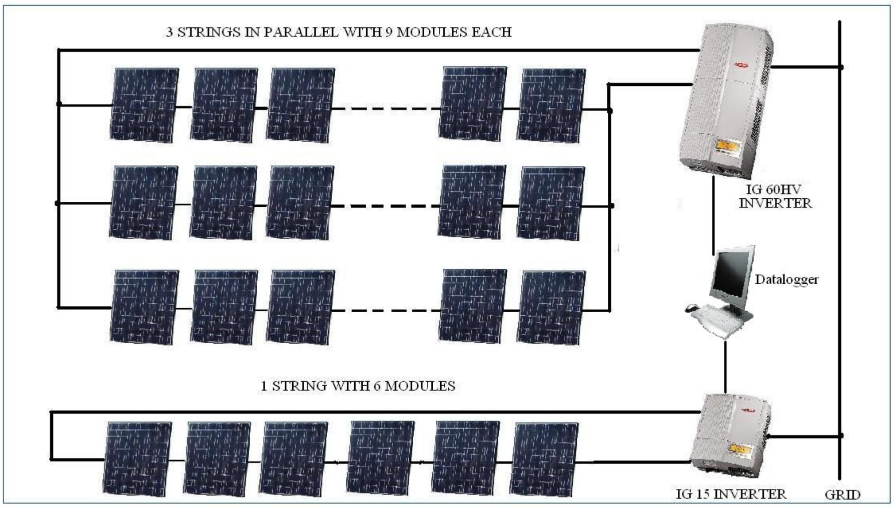



Figure 10: The topology of the Saarijärvi PV system.

## 3.2 System monitoring

The PV monitoring was done in accordance with the IEC 61724:1998 standard [51][3]. The recorded parameters were used in calculation of performance indices as shown in Section 3.3. A brief description of what parameters were recorded and how they were recorded is given for each PV system is given in the next two subsections.

### Agora PV system

Data recording began on $17^{th}$ June 2009 and the parameters recorded were: date/time; solar irradiance; cell temperature, ambient temperature; voltage & current from array; voltage & current to grid; power to the grid; energy to the grid; AC frequency and status. Initially the data was recorded at 15 minute intervals then on $27^{th}$ June 2009, the interval was shortened to one minute. The reason for reducing the interval was to increase the number of readings. Data samples were taken every ten seconds and the average over one minute was written to a file in the computer. The data was only recorded when the system was producing electricity. Data was stored in csv (comma separated variables) format and each day was stored in its own file. The computer used to monitor the system was located on site in a room adjacent to

[3]The BS EN 61724:1998 is based on the IEC 61724:1998 for which was adopted by the European Union under the standard name EN 61724:1998. It has not been altered, just translated to English.

the PV array and it was a stand-alone PC.

Periodically the data was collected by university staff and stored in a database. The Agora PV system has worked well with only two interruptions which occurred on 22.06.2009 from 11:15 to 14:15 and 18.12.2011 to 21.12.2011. Both occasions were due to inverter error. The second interruption had little bearing on the performance of the system since it occurred during the period of the shortest daylight period of the year.

**Saarjärvi PV system**

The Saarijärvi PV system is older than the Agora PV system. The data recording began $7^{th}$ October 2005. The data recorded was saved on a mySQL database for remote access. For the purpose of this thesis, the parameters that were extracted from the database were date/time, irradiance, cell & ambient temperatures, array voltage & current for both sub-systems and power to grid. Unlike the PV system in Agora, this system recorded data 24-hours a day all year round. Like the Agora system, the data was initially recorded every 15 minutes, then it was reduced to 10 minutes on $28^{th}$ February 2008. The reason for the 10 minute intervals is the minimum allowed data recording interval allowed by the Fronius datalogger.

## 3.3 Performance indices

There are many ways of evaluating the performance of a system. One of the methods is to use yields, performance ratios and losses. There are three types of yields namely; reference, array and final yields. All three are normalised energy values and have the basic unit of hours. As previously mentioned, these variables enable comparison of PV systems regardless of technology, location or size. The performance ratio and losses are derived from the yields.

### 3.3.1 Reference yield

The reference yield can be described as the number of sun hours at reference irradiation equivalent to the in-plane solar energy for an entire day. According to the IEC 61724:1998 standard [51], the equation required to calculate the daily reference yield is:

$$Y_r = \frac{\tau_r \times (\sum_{day} G_I)}{G_{I,ref}} \tag{3.1}$$

where $Y_r$ is the reference yield,$\tau_r$ is the recording interval (h), $G_I$ is the in-plane irradiation on the module (W/m$^2$) and $G_{I,ref}$ is the reference irradiance of 1000 W/m$^2$. The sum is over a period of a day. The unit of the reference yield is $\mathrm{W\,h/m^2} \big/ \mathrm{W/m^2/d}$. I reduced it to $\mathrm{h\,d^{-1}}$ where $h$ is hours and $d$ is day.

### 3.3.2 Array yield

A group of connected solar modules make up an array. The energy produced by this array is normalised by the nominal power of the array. In this study, the rated nominal power for the modules were used to get the peak power of the arrays. For this thesis, *nominal power* refers to the maximum rated power for a module as described by the manufacturer. Array yield is the amount of hours the array would operate at its rated nominal power to produce the same amount of energy in a day. The daily array yield is given by:

$$Y_a = \frac{\tau_r \times (\sum_{day} P_a)}{P_o} \tag{3.2}$$

where $Y_a$ is the array yield, $P_a$ is power from the array (W) and $P_o$ is rated nominal power (W). The sum is over a period of a day and the unit of measure is $\mathrm{h\,d^{-1}}$.

### 3.3.3 Final yield

The power from the array can be stored in batteries, supplied to a building or fed to a wider electricity grid. The last two options are done via an inverter. The final yield is a product of the array yield and the load efficiency. In this case, it is the same as the ratio of the daily energy to the grid to the rated nominal power:

$$Y_f = \frac{\tau_r \times (\sum_{day} P_{tu})}{P_o} \tag{3.3}$$

where $Y_f$ is the final yield and $P_{tu}$ is power to the grid/utility (W). The sum is over a period of a day and the unit of measure is $\mathrm{h\,d^{-1}}$.

### 3.3.4 Performance ratio and losses

The performance ratio is the ratio of the final yield to the reference yield. This is different from efficiency since it takes into account the available solar resource and not power output at STC:

$$PR = \frac{Y_f}{Y_r} \tag{3.4}$$

where $PR$ is the performance ratio. This is a dimensionless variable. The performance ratio will be used to determine the rate of degradation in conjunction with ARIMA, which is described in the next section.

The losses are of two types: the capture loss and the system loss. The capture loss is the difference between the reference yield and the array yield. It describes the amount of energy lost in solar conversion in terms of time. Losses can occur through heat, array inefficiencies (cell mismatch, degradation of laminate) or module surface soiling. The capture loss can be described mathematically as follows:

$$L_c = Y_r - Y_a \tag{3.5}$$

where $L_c$ is the capture loss. Units is $\mathrm{h\,d^{-1}}$.

The system loss covers all losses after the array to the utility. Thus it is the difference between the array yield and the final yield. Losses may be due to DC/AC conversion, wire resistance and inverter threshold. The equation for the system loss is:

$$L_s = Y_a - Y_f \tag{3.6}$$

where $L_s$ is the system loss. Unit is $\mathrm{h\,d^{-1}}$.

## 3.4 ARIMA

ARIMA is the acronym for AutoRegressive Integrated Moving Average. It was originally proposed by Box and Jenkins in 1970 [52]. It is also referred to as the Box-Jenkins model. It is a statistical tool used to analyse time series data and is commonly used in econometrics. ARIMA process checks present data in relation to past data and this can be used to determine the possible outcome in the future. It is robust in the sense that it can handle outliers (data outside the normal range), missing data and noise. For the data obtained from the PV systems, when the parameters are plotted against time, it is expected that a seasonal element to the data will be observed also outlier values.

ARIMA has been proposed for analytical work by Jordan et al. [24] based on its success to model solar radiation data and grid connected PV power production. The ARIMA model is described by the following equation:

$$\phi(B)\nabla^d z_t = \theta(B)a_t$$

where $\phi(B)$ is a stationary autoregressive operator, $\nabla$ is the differencing operator, $d$ is the difference, $z_t$ is an autoregressive process of a given order, $\theta(B)$ is a moving average process of a given order, $B$ is a backward shift operator and $a_t$ is a purely random process with mean zero and variance $\sigma_z^2$ [52, 53].

ARIMA model can also be described in terms of (p,d,q) where the p,d and q are the autoregressive, integrated and moving average orders. A more detailed description of how to calculate the orders can be found in the books by Box [52] and Chatfield [53]. For the purpose of this study, a basic understanding was required and with the help of statistical software, the orders were calculated.

The solar resource follows a seasonal pattern and thus it may be inferred that the recorded data would follow a similar pattern. Therefore it would be appropriate to use a seasonal ARIMA (SARIMA) for modelling. The SARIMA model is described in two parts $(p,d,q) \times (P,D,Q)_{period}$ where the first part is the non-seasonal terms while the second is the seasonal terms. The *period* to be specified is the number of intervals; for example, '7' for daily data in a week, '52' for weekly data in a year or '12' for monthly data in a year. For this work, I used R statistics software to calculate the SARIMA data fitted values and also used a built in function that generated SARIMA non-seasonal and seasonal order numbers, that is, p,d,q,P,D and Q. The SARIMA model was used to get 'ARIMA' equivalent data from the performance indices.

## 3.5 Data analysis

The Agora and Saarijärvi systems began data monitoring from months in the middle of the year. Since both of them were older than the recommended minimum age of three years for evaluation to be done, I decided to start analysing the data for both systems at the beginning of the next respective calender years. This was to enable analysis of the system with full cycles. As earlier stated, the Saarijärvi system has two subsystems with differing topologies but the same PV technology. A separate analysis of the two subsystem was done to check if there is a marked difference in the behaviour of the subsystems. The csv files for both systems were read and data extracted using Matlab. Due to the different formats within the files, different Matlab codes were written. The programs can be seen in the appendix section A.

The number of hours that the PV system was monitored was calculated and the value was noted. This was done by setting the commencement date at the beginning of a given year and end date at the last day of the year in the reporting period. Next, data was filtered using some exclusions such as cut-in irradiance for each system. For the Saarijärvi system, data with no array power was also excluded.

To calculate the energy produced by the array, the following formula was used:

$$E_A = \sum_{\tau_m} (U_{pv} \times I_{pv} \times \tau_r) \tag{3.7}$$

where $E_A$ is energy from the array (Wh), $\tau_m$ is the monitoring period (h), $U_{pv}$ and $I_{pv}$ are the voltage (V) and current (A) from the array, respectively. $\tau_r$ is the recording interval (h). The energy of the array is the summation over the monitoring period.

The grid energy and in-plane solar energy were determined in a similar manner:

$$E_{tu} = \sum_{\tau_m} (P_{tu} \times \tau_r) \tag{3.8}$$

$$E_{S,A} = \sum_{\tau_m} (G_I \times A_A \times \tau_r) \tag{3.9}$$

where $E_{tu}$ is energy to the grid (Wh), $P_{tu}$ is the power to the grid (W), $E_{S,A}$ is the in plane solar energy (Wh), $G_I$ is the irradiance per square unit (W/m$^2$) and $A_A$ is the area of the array (m$^2$). Both equations are summations over the monitoring period.

The temperatures and irradiance were averaged over a day when performing time series analysis. It should be noted that average values for some days from the Agora PV system could not be obtained since it only recorded when the system was on:

$$T_{mean} = \frac{\sum_{day} T}{\sum_{day} \tau_r} \tag{3.10}$$

$$G_{I,mean} = \frac{\sum_{day} G_I}{\sum_{day} \tau_r} \tag{3.11}$$

where $T_{mean}$ is the average temperature (°C per day) and $T$ is the instantaneous temperature, $G_{I,mean}$ is the average irradiance per day and $G_I$ is the instantaneous irradiance. The unit used for irradiance per day is W/m$^2$ d.

The average values of the irradiance and the temperature is only for periods when the systems are producing power. The temperatures were neglected when there was no sunlight or power production.

For the energy balances of the PV systems, the equations to calculate them were obtained from Blaesser and Munro [54]. The monitoring fraction was found as follows:

$$M = \frac{t_M}{\tau} \tag{3.12}$$

where $M$ is the monitoring fraction, $t_M$ is the total time of monitoring activity (h) and $\tau$ is the reporting period (h). In this case, $\tau$ is three years for the Agora PV and seven years for the Saarijärvi system. The monitoring fraction shows how long the system has been under observation within a given reporting period. For instance, if a system has 24 hour recording over the whole reporting period, the monitoring fraction would be one.

The outage fraction sums up the number of hours when the system was not available due to a fault in the system. The Agora system had a status column in the csv file to show if there was a fault detected. For the Saarijärvi system, the number of hours in which there was missing data was presumed to be equivalent to the outage hours. The following equation was used:

$$O = \sum_{\tau} \frac{t_{NAV}}{\tau} \tag{3.13}$$

where $O$ is the outage fraction and $t_{NAV}$ is the duration which the system is unavailable (h).

The mean daily irradiation on an array plane is taken as the product of the sum of the total irradiation over the reporting period and 24 hours. The result is divided by the total monitoring time. The unit is W h/m$^2$ d. The following equation was used:

$$[Wh/m^2d] = 24.t_r \sum_{\tau} G_I/t_M \begin{cases} \tau & \text{if } M = 1, \\ t_M & \text{if } M < 1. \end{cases} \tag{3.14}$$

where 24 is the number of hours in a day, $t_r$ is the recording interval (h) and $t_M$ is the total monitoring time (h).

There is no power that is drawn from the grid to run the system with the exception of the computers that are recording the data. During the analysis, the computer has been assumed not to be part of the PV electricity generation system. However, this should be considered in future works when analysing the PV system in its totality.

The array, system and inverter efficiencies are all ratios of inputs to outputs. The array efficiency is the ratio of the array power output to the in-plane solar energy while the system efficiency is the useful power output (whether to the grid or battery) to the in-plane solar energy. The inverter efficiency is the ratio of the energy out of the inverter to the energy into the inverter. The three equations can be described as follows:

$$\eta_{A,mean} = \frac{E_{A,\tau}}{E_{S,A,\tau}} \tag{3.15}$$

$$\eta_{tot} = \frac{E_{use,PV,\tau}}{E_{S,A,\tau}} \tag{3.16}$$

$$\eta_I = \frac{E_{io,\tau}}{E_{II,\tau}} \tag{3.17}$$

where $\eta_{A,mean}$, $\eta_{tot}$ and $\eta_I$ are the efficiencies of the array, system and inverter, respectively. $E_{A,\tau}$ is the energy from the array during the reporting period (Wh), $E_{S,A,\tau}$ is the solar energy in-plane to the array (Wh), $E_{use,PV,\tau}$ is the useful energy contributed by the PV during the reporting period (Wh), $E_{II,\tau}$ is the energy into the inverter (Wh) and $E_{io,\tau}$ is the energy out of the inverter (Wh). For this system, it was assumed that losses between the array and the inverter and the inverter to the grid were negligible. Therefore $E_{II,\tau}$ is equal to $E_{A,\tau}$ and $E_{use,PV,\tau}$ is equal to $E_{io,\tau}$.

All the equations (3.7-3.17) were used in Matlab to obtain values for different performance parameters.

In the analysis of effect of irradiance and cell temperature on performance indices, the year 2012 was selected as the base year because both systems in Agora and Saarijärvi did not have any missing data for that year. The values of the irradiance and cell temperature values were averaged over 10-minute periods and used in the analysis. Normalised values of array efficiency and inverter power were obtained by dividing by the module efficiencies at STC and rated power output, respectively.

Using curve fitting toolbox in Matlab, data for smoothed curves were obtained. The scatter plots and the smoothed curves were plotted using Originlab. Equations of polynomial curves were obtained and general trends could be established. The rate of change of array efficiency with respect to cell temperature could be calculated using the equations of the curves.

The ARIMA model values were obtained using R. First, weekly PR values were calculated. This was done so as to obtain as many values as possible in order to improve accuracy. The in-built forecast package, *auto.arima*, was used to find the order values (p,d,q,P,D,Q) of the ARIMA. The order values were used to obtain new values that can be called PR-ARIMA values. The PR-ARIMA values and raw PR values are stored in a mat file which were exported to Originlab for plotting. In Originlab, linear lines of best fit are drawn for both the PR values and the PR-ARIMA values. The equations from the lines and standard deviations are used to calculate the degradation rate.

# 4 Results and discussion

In this chapter, the results obtained from the two sites are presented. The performance indices over time and the effect of two factors, namely: temperature and irradiance, are presented and discussed. This will lead to degradation rate analysis.

## 4.1 Energy balances

### 4.1.1 Agora PV system

The guidelines used for analysis and presentation of data were published by the Joint Research Centre (JRC) [54]. Tables 4 and 5 show summaries of energy balances, climatic data and other performance parameters.

The monitoring fraction shown in Table 4 is less than 50 % because data recording only occurred when there was sufficient light or when the system was producing electricity. Therefore during the night hours and winter months no data was written. This also explains why the daily irradiation levels are much higher than those in Saarijärvi which will be presented in the next section 4.1.2.

So far the system's mean performance ratio is 0.83. The overall inverter efficiency is the same as the maximum efficiency and higher than the Euro-efficiency (93.2 %).

In the stacked graph (Fig. 11), during the months corresponding to January, November and December (Taking 1,13 and 25 to represent January), there are low yields due to low light conditions. In the period of February and March, there is high capture loss compared to the final yield due to partial snow cover on the array.

March and April show a reduction in the capture loss and a small increase in the system loss (the green colour). This happens when the snow melts from the array surface, the ambient temperatures are still low, the irradiance levels are increasing and the day length is increasing. The residual snow on the ground reflects light thus increasing the irradiation on the array. The system loss occurs between the array and the grid. The cause may be inverter loss or

Table 4: Summary of energy balance of Agora PV system

| | | | |
|---|---|---|---|
| | **Agora** | | |
| ***Energy balances*** | | | |
| **Technology**: | p-Si | **Period**: | 1/1/10-31/12/12 |
| **Nominal power**, $P_o$[$kW_p$]: | 2.6 | **Total Array Area**, A[$m^2$]: | 19.83 |
| **Total time of monitoring**, $t_m$[h]: | 11,984 | **Monitoring fraction**, M: | 0.46 |
| **Outage fraction**, O: | 0.00035 | | |
| ***Climatic data*** | | | |
| **Recorded total solar energy in array plane**, $E_{S,A}$ [kWh]: | | | 57,894 |
| **3 Year mean daily irradiation in array plane**,[kWh/$m^2$/d]: | | | 5.85 |
| ***System balances*** | | | |
| **Total array output energy**, $E_A$[kWh]: | | 6698 | |
| **Energy supplied to utility grid**, $E_{tu}$ [kWh]: | | 6307 | |
| **Energy drawn from utility grid**, $E_{fu}$ [kWh]: | | 0 | |
| **Energy from back-up generator**, $E_{bu}$ [kWh]: | | 0 | |
| **Total energy to DC or AC loads**, [kWh]: | | 0 | |
| **Total input energy**, $E_{in}$ [kWh]: | 6698 | **PV_Fraction**, $F_A$: | 1 |
| **Useful energy** $E_{use}$ [kWh]: | 6307 | $E_{use,PV}$ [kWh]: | 6307 |
| ***BOS component balances and efficiencies*** | | | |
| **Energy** [kWh] **to inverters**, $E_{II}$: | 6698 | **from inverters**, $E_{io}$: | 6307 |
| **Energy Efficiency of inverters** , $\eta_I$: | | 0.94 | |

Table 5: Annual indices of performance for Agora PV system.

| | 2010 | 2011 | 2012 |
|---|---|---|---|
| **Final yield** ($\mathrm{h\,y^{-1}}$) | 790.8 | 870.5 | 764.5 |
| **Array yield** ($\mathrm{h\,y^{-1}}$) | 839.3 | 924.2 | 812.6 |
| **Reference yield** ($\mathrm{h\,y^{-1}}$) | 977.4 | 1028.9 | 913.2 |
| **Performance ratio** | 0.81 | 0.85 | 0.84 |
| **Mean array efficiency**, $\eta_A$: | | 0.116 | |
| **Overall plant efficiency**, $\eta_{tot}$: | | 0.109 | |

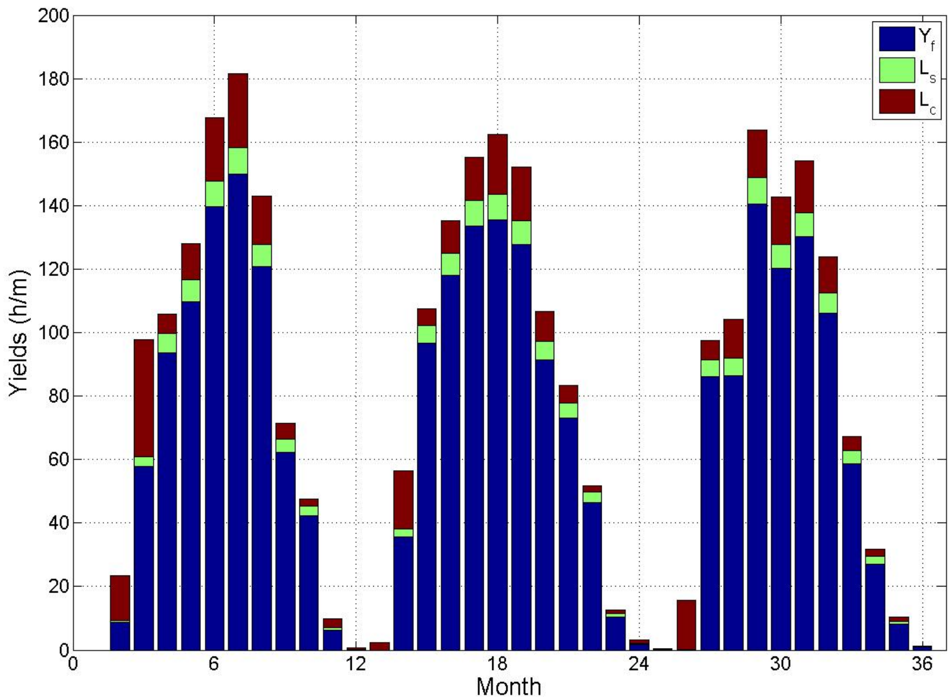



Figure 11: Stacked graph of the performance indices for Agora for the monitoring period (The units h/m are hours per month)

resistance in the wires. This will be explored later in this chapter.

In the period between May and July, the temperature increases and this causes the capture loss to increase once again. The change in system loss is not as adverse as the capture loss throughout this period. The reason can be attributed to the inverter efficiency and other energy losses between the grid connection and the array being more or less constant. The reference yield levels are periodic and for the same month in different years, the levels may be different such as month 30 (June 2012) in Figure 11 was lower than month 6 and 18 (June 2010 and 2011) due to greater cloud cover in the summer of 2012.

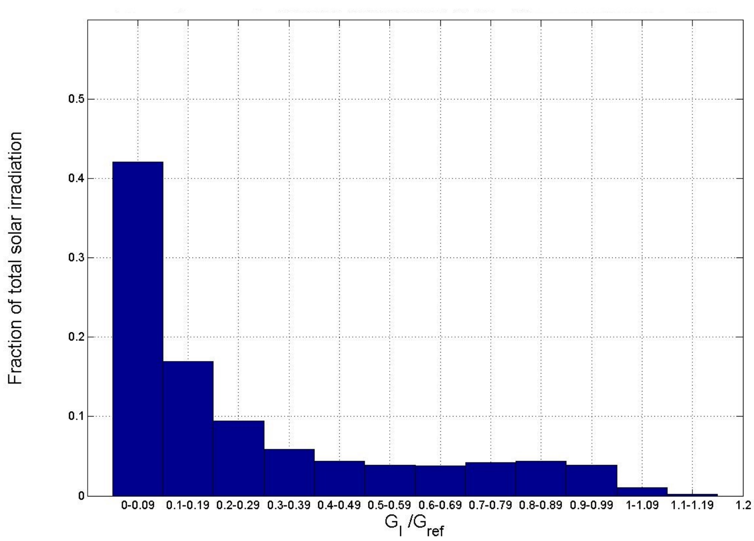



Figure 12: Histogram showing the normalised distribution of in-plane radiance in 2012 for the Agora PV system

Figure 12 shows the normalised distribution of the in-plane radiance, it can be seen that about 70 % of the in-plane irradiation is contributed by solar irradiance below 300 W/m$^2$. About 2 % is contributed by irradiance level above 1000 W/m$^2$. The histogram was based on the irradiance data in 2012. The low light conditions are prevalent in this location. The histogram is helpful in predicting the quality and quantity of power output. Furthermore, the suitability of p-Si for this location and the sizing of the inverter in this PV system will be checked in a later section.

### 4.1.2 Saarijärvi PV system

Tables 6, 7 and 8 show summaries of energy balances, climatic data and other performance parameters.The monitoring fraction of the Saarijärvi system is nearly 90% due to continuous monitoring. The outage hours are equivalent to 270 days and the reasons for this have been stated in section 3.1.2. As previously mentioned, the daily in-plane irradiation in Saarijärvi is twice as low as the one in Agora. The reason is attributed to the equation used to derive the value. The equation derived from the JRC guidelines [54], equation (3.14), was used for both PV system even though the resulting output differs. If the mean daily irradiation takes into account night/no light period monitoring, it would explain why there is large difference between the two values. To verify the results, using PVGIS [55], the coordinates of Agora and inclination of 40° for the PV array were used and the value found was 2.91 kWh/m$^2$d. For a similar PV inclination, the value found for Saarijärvi was 2.93 kWh/m$^2$d. When compared to the calculated values of 5.85 kWh/m$^2$d and 2.44 kWh/m$^2$d for Agora and Saarijärvi respectively, there is a large variation in the Agora values. The difference can be attributed to the monitoring fraction. Agora has a monitoring fraction of 0.46 while Saarijärvi has 0.89. I propose that the equation 3.14 be multiplied by the monitoring fraction $M$. With the new formula, the resulting mean daily irradiation values would be 2.69 kWh/m$^2$d and 2.17 kWh/m$^2$d for Agora and Saarijärvi systems respectively.

Tables 7 and 8 display the annual performance indices for the Saarijärvi PV systems. We can see that typically PR ranges from 0.78 to 0.89 with the exception of 2007 when the ss1 has a higher performance ratio than the other subsystem yet the array and subsystem efficiency of ss2 is greater than ss1. The reason behind the lower efficiencies in ss1 is the period in which the inverter malfunctioned. Another difference between the systems is the inverter efficiencies, where subsystem 1 has a more efficient inverter. However, the efficiency calculated is above the maximum efficiency stated by the manufacturer for a Fronius IG 60 HV (94.5 %). The inverter efficiency of the ss2 was within the range of the maximum efficiency and the European efficiency. The inverter performances will be analysed on their own in section 4.3.

Figure 13 shows stacked bar graph of the performance indices for the Saarijärvi PV systems. There are gaps in months 16, 17 and 30 due to data loss caused by the computer crashes. In ss1, for months 19 and 20, there was no final yield since there was an inverter malfunction in the subsystem. The 2nd, 3rd and 5th period coincide with 2007, 2008 and 2011 where data interruptions were most noted. The effects of the various interruptions can be observed

Table 6: Summary of energy balance of Saarijärvi PV system

| **Saarijärvi** | | | |
|---|---|---|---|
| ***Energy balances*** | | | |
| **Technology**: | HIT | **Period**: | 1/1/06-30/12/12 |
| **Nominal power**, $P_o$[$kW_p$]: | 6.27 | **Total Array Area**, A[$m^2$]: | 37.95 |
| **Total time of monitoring**, $t_m$[h]: | 54,857 | **Monitoring fraction** M: | 0.89 |
| **Outage fraction** ,O: | 0.11 | | |
| ***Climatic data*** | | | |
| **Recorded total solar energy in array plane**, $E_{S,A}$ [kWh]: | | | 212,010 |
| **7 Year mean daily irradiation in array plane**, [kW h/$m^2$/d]: | | | 2.44 |
| ***System balances*** | | | |
| | | **Subsystem 1** | **Subsystem 2** |
| **Total array output energy**, $E_A$[kWh]: | | 23,797 | 5,589 |
| **Energy supplied to utility grid**, $E_{tu}$ [kWh]: | | 22,809 | 5,172 |
| **Energy drawn from utility grid**, $E_{fu}$ [kWh]: | | 0 | 0 |
| **Energy from back-up generator**, $E_{bu}$ [kWh]: | | 0 | 0 |
| **Total energy to DC or AC loads**, [kWh]: | | 0 | 0 |
| **Total input energy**, $E_{in}$ [kWh]: | 29,386 | **PV_Fraction**, $F_A$: | 1 |
| **Useful energy** $E_{use}$ [kWh]: | 27,981 | $E_{use,PV}$ [kWh]: | 27,981 |
| ***BOS component balances and efficiencies*** | | | |
| | | **Subsystem 1** | **Subsystem 2** |
| **Energy [kWh] to inverters**, $E_{II}$: | | 23,797 | 5,589 |
| **from inverters**, $E_{io}$: | | 22,809 | 5,172 |
| **Energy Efficiency of inverters** , $\eta_I$: | | 0.958 | 0.925 |

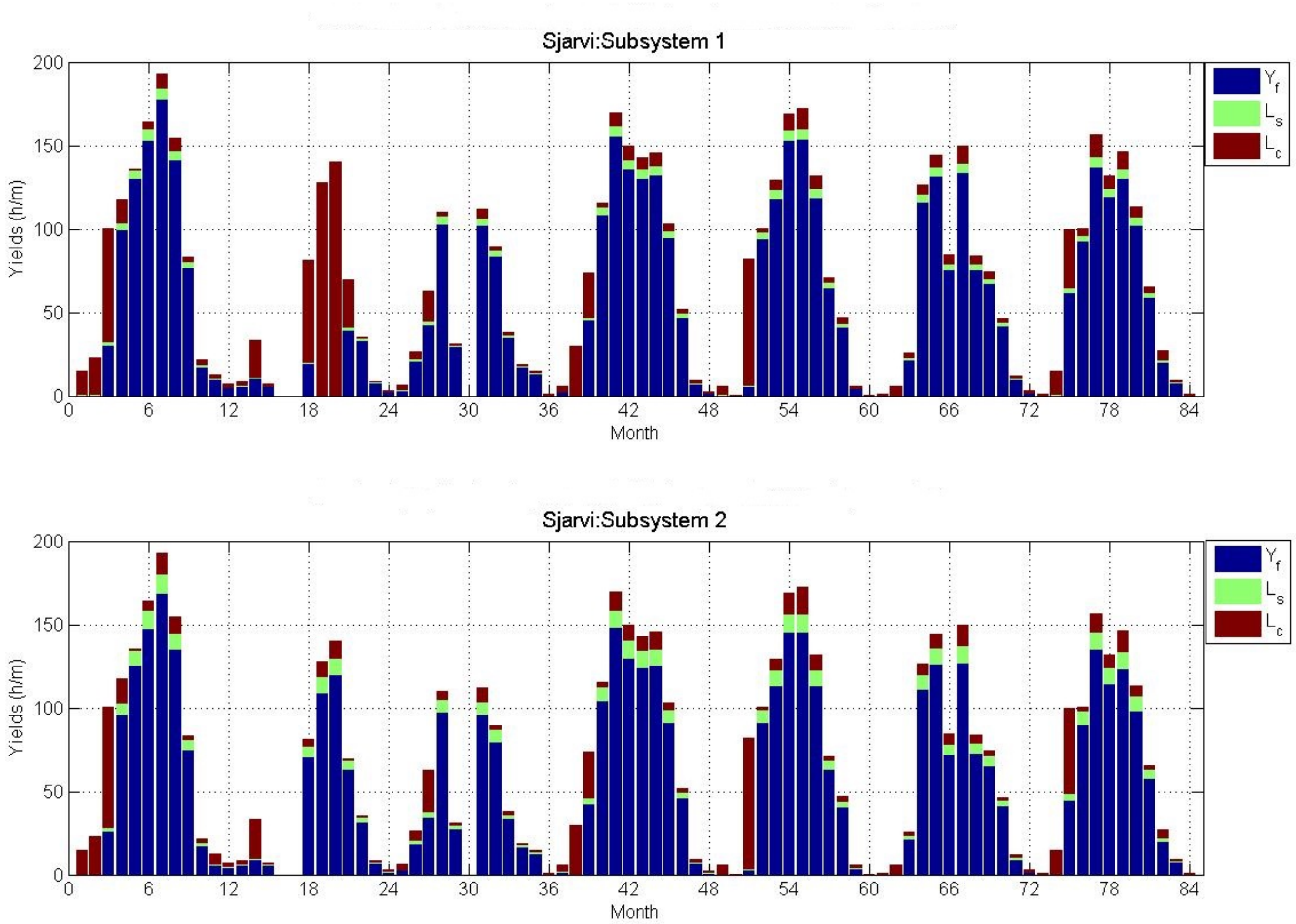



Figure 13: Stacked bar graph of the performance indices for Saarijärvi PV system. Note the change in pattern between months 12 and 24. (The units h/m are hours per month).

Table 7: Annual indices of performance for Saarijärvi ss1

| | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
|---|---|---|---|---|---|---|---|
| **Final yield** ($h y^{-1}$) | 843 | 124 | 453 | 862 | 756 | 676 | 732 |
| **Array yield** ($h y^{-1}$) | 878 | 131 | 473 | 899 | 788 | 705 | 765 |
| **Reference yield** ($h y^{-1}$) | 1027 | 514 | 510 | 999 | 916 | 756 | 865 |
| **Performance ratio** | 0.82 | 0.24 | 0.89 | 0.86 | 0.83 | 0.89 | 0.85 |
| **Mean array efficiency**, $\eta_A$: | | | 0.137 | | | | |
| **Overall subsystem efficiency**, $\eta_{tot}$: | | | 0.131 | | | | |

Table 8: Annual indices of performance for Saarijärvi ss2

| | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
|---|---|---|---|---|---|---|---|
| **Final yield** ($h y^{-1}$) | 803 | 425 | 422 | 823 | 721 | 650 | 693 |
| **Array yield** ($h y^{-1}$) | 863 | 461 | 457 | 888 | 779 | 703 | 752 |
| **Reference yield** ($h y^{-1}$) | 1027 | 514 | 510 | 999 | 916 | 756 | 865 |
| **Performance ratio** | 0.78 | 0.83 | 0.83 | 0.82 | 0.79 | 0.86 | 0.80 |
| **Mean array efficiency**, $\eta_A$: | | | 0.145 | | | | |
| **Overall subsystem efficiency**, $\eta_{tot}$: | | | 0.134 | | | | |

in the graphs (Fig. 13). Again the months of February and March show the highest capture loss due to snow cover on the arrays.

Figure 14 is a histogram of normalised distribution of in-plane radiance in Saarijärvi which is similar in shape to Figure 12. Below 300 $W/m^2$ the in- plane irradiance contribution is about 70 % of the total irradiation. The histogram confirms that both sites are exposed to identical solar irradiation levels.

### Summary

Comparing the two sites over three years (2010, 2011 and 2012), the final yields of the Agora PV system are greater than ss1 and ss2 in Saarijärvi. In addition, the reference yield for the

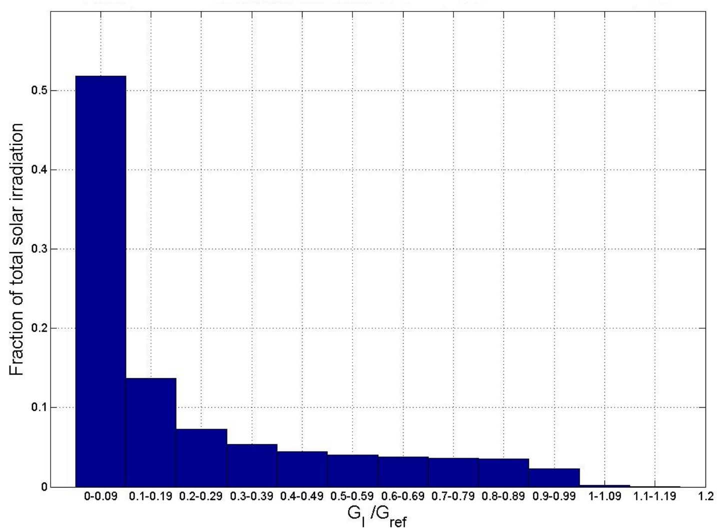



Figure 14: Histogram showing the normalised distribution of in-plane radiance in Saarijärvi

same year but different sites also have large differences. The two main reasons were: (1) the data recording interruptions at the Saarijärvi site in 2010 and 2011 and (2) the precision of the values of the collected data. The Agora irradiance data was recorded to three decimal places while at Saarijärvi the irradiance values were integers only. The sum of several decimal values can have an effect on the total sum.

## 4.2 Behaviour of Performance indices over time

The seasonal changes have a strong influence on ambient temperature and the amount of available solar energy to the arrays. During late autumn and early winter, there is snow and low light levels while during late winter and early spring there is still snow but more insolation. Looking at the left hand side of Figure 15 shows a scatter plot of performance ratio (PR) in a time series. There are periodic trough shaped crests formed between the months of April and September. The lowest part of the trough is during the middle of summer. The same pattern can be seen in Figure 16. With reference to Fig. 16, although the same PV technology is in both graphs, subsystem 2 has a greater difference between the lowest PR value in the summer and the highest PR value for the year as compared to subsystem 1. This means the subsystem 2 is more sensitive to temperature and irradiance level change than subsystem 1.

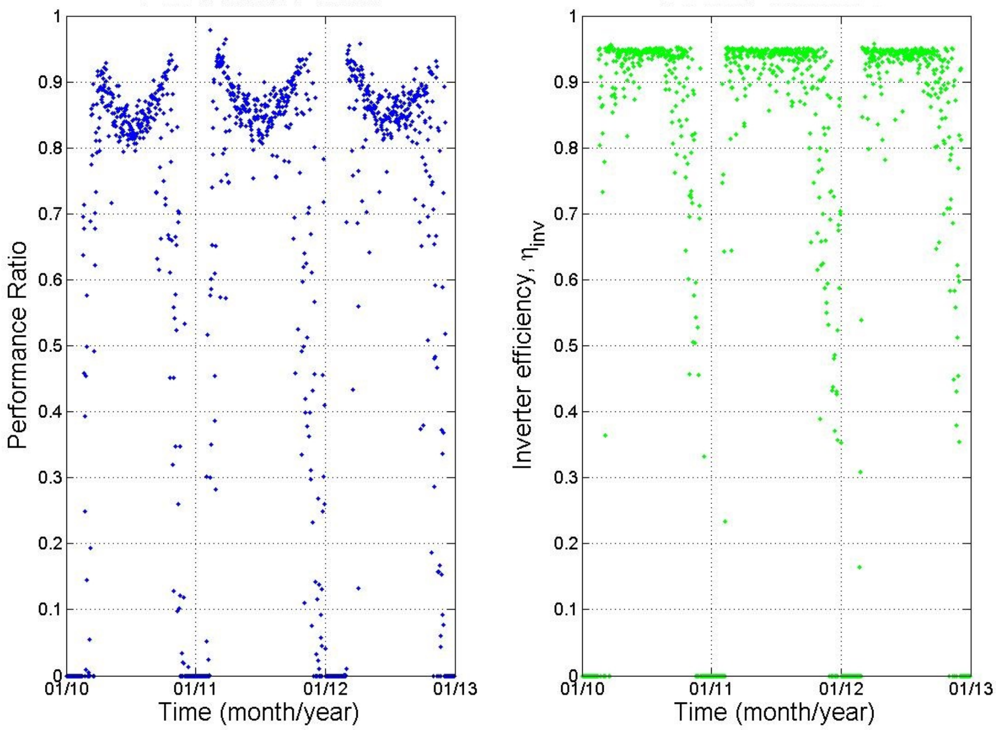



Figure 15: Performance ratio (left) and inverter efficiency (right) as a function of time for Agora system

The inverter efficiency also shows seasonality but this is more closely related to availability of array power. In Figure 15 on the right, there are several data points between zero and the maximum efficiency of the inverter during the months of low irradiation. In comparison, Figure 17 shows significantly fewer points within the same range (zero to maximum efficiency). This can be attributed to inverter behaviour in low light conditions. Instances where the inverter efficiency show values above 100 % may be caused by rounding off error during calculation. The behaviour of the inverters at partial load and different performance ratios will be investigated later in section 4.4.

System and array efficiencies plots have similar patterns to the PR time plot as shown in Figures 18 and 19. The system efficiency is lower than the array efficiency due to losses between the array and the grid. In Fig. 19, the system disruption marked by missing data

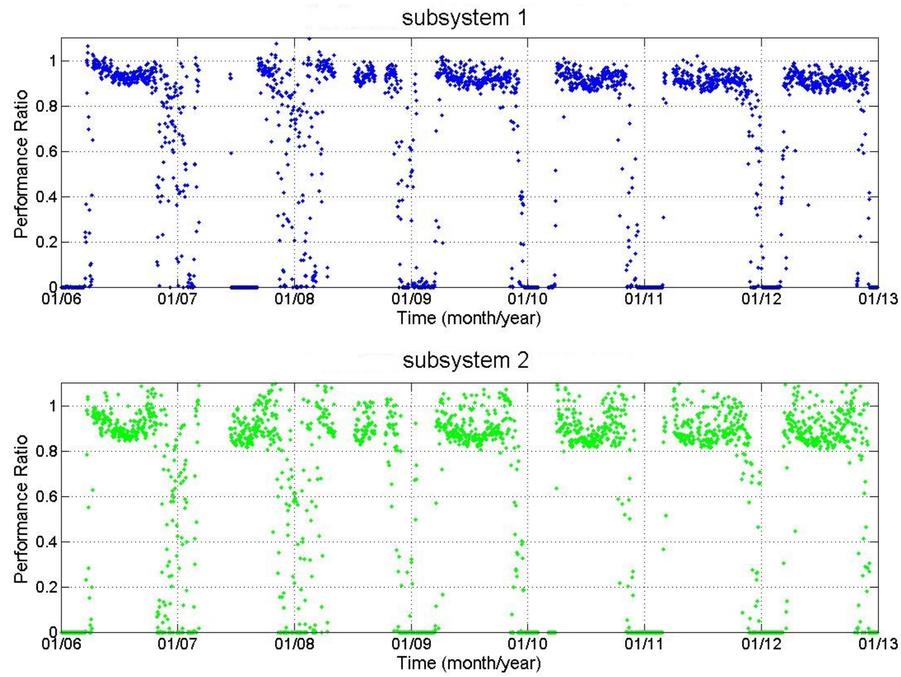



Figure 16: Performance ratio as a function of time for Saarijärvi ss1 and ss2

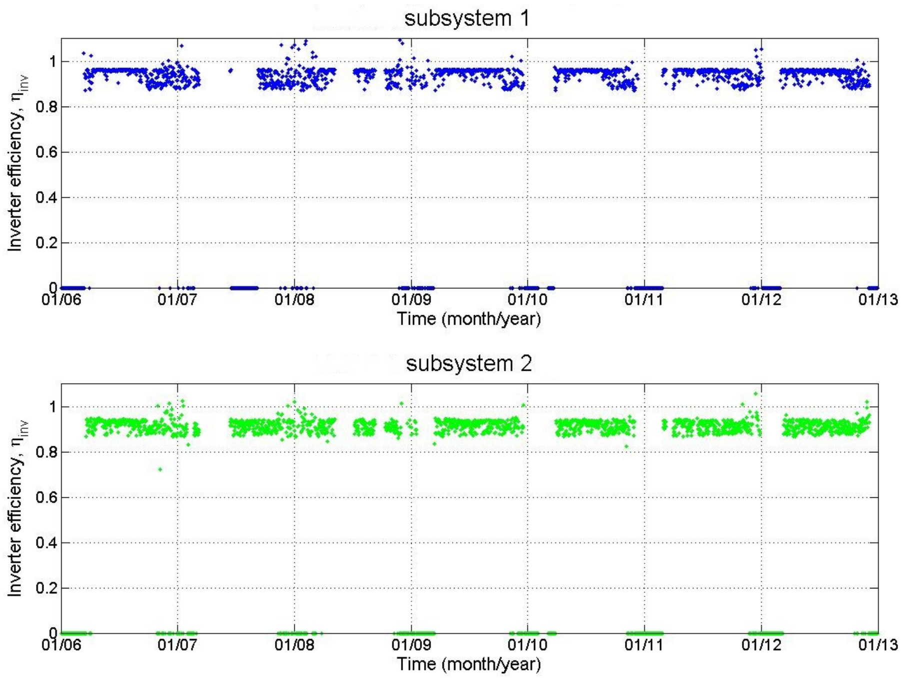



Figure 17: Inverter efficiencies as a function of time for Saarijärvi ss1 and ss2

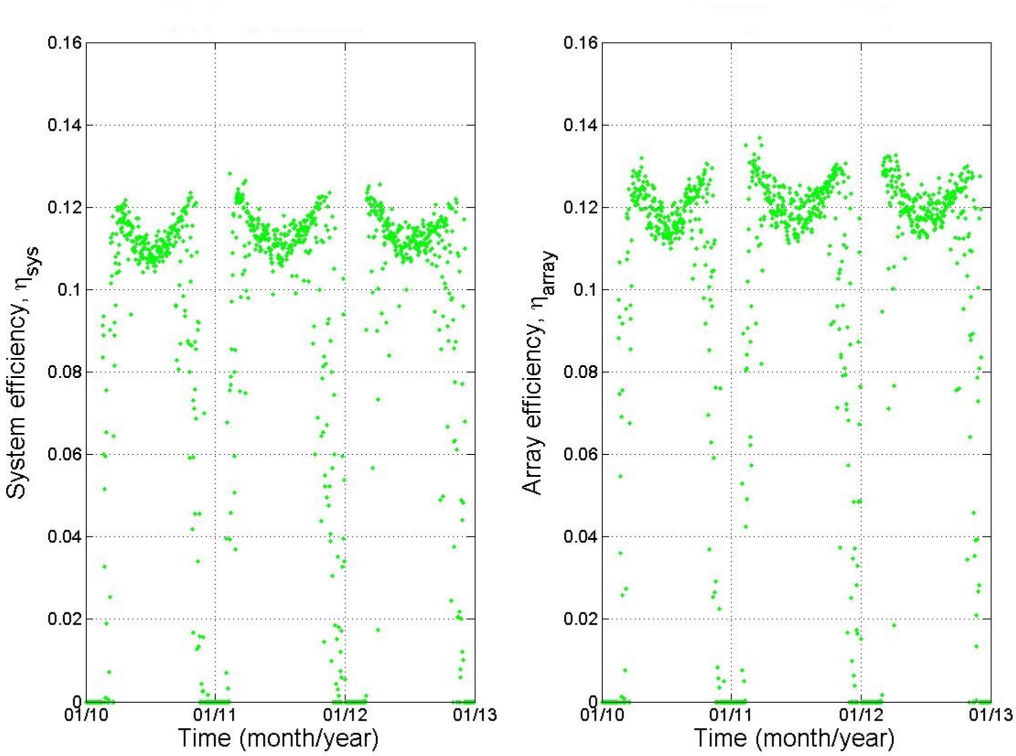



Figure 18: System (left) and array (right) efficiencies as functions of time for Agora PV system

points can be seen between 2007 and 2009. It should be noted that ss2 has wider range of array efficiency values during the peak insolation period as compared to ss1. That is, the trough thickness is greater in ss2 than in ss1. It can be deduced that the difference in the subsystem configurations is a factor. The two configurations will be compared later on.

Figure 20 shows the capture loss over the reporting period. It confirms that losses occurring during the summer months are due to cell temperature increase. The irregular spikes in the capture loss before the small humps in the scatter plot are due to snow cover on the array but not on the reference cell. In early spring, the temperatures may still be well below zero but the sun hours are long but with little to no array power. This loss can be avoided by manual snow removal if the site is visited regularly or by increasing the angle of the array to 45°. The additional cost of adjusting the array angle or possibility of damaging the array during cleaning may not make the exercise attractive. Also the additional power output may

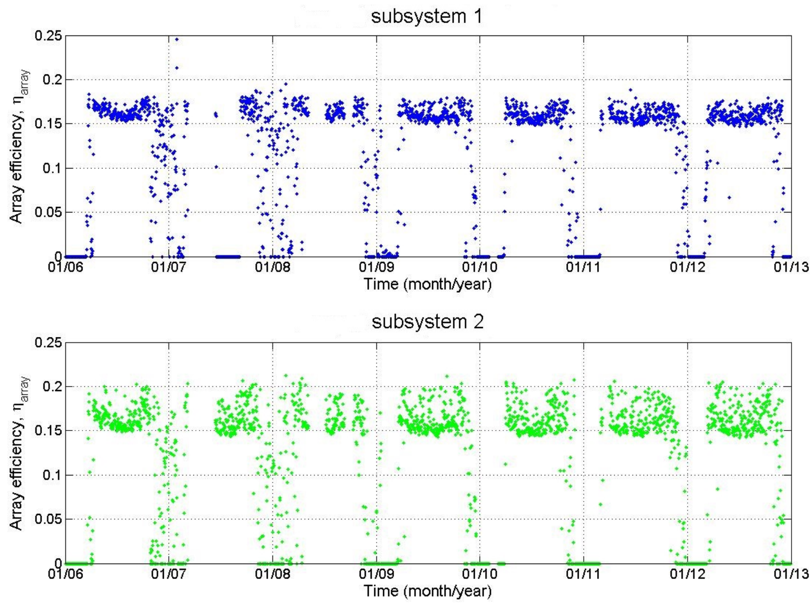



Figure 19: Array efficiencies as functions of time for Saarijärvi PV ss1 and ss2

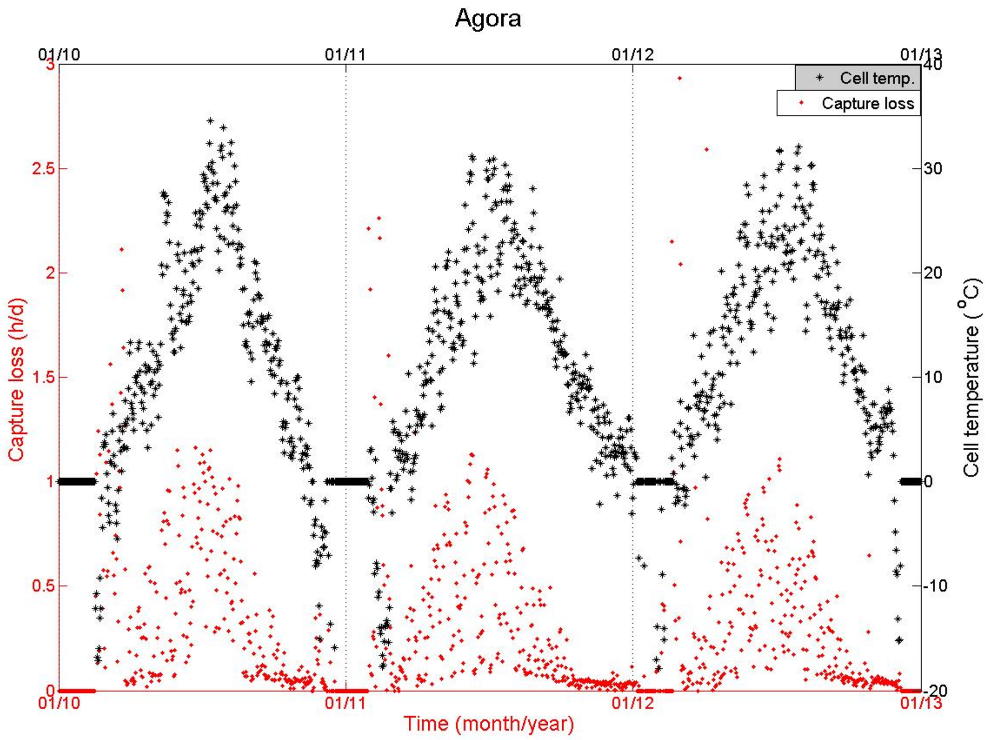



Figure 20: Capture loss and cell temperature as a function of time for Agora system

be insignificant. A similarly shaped scatter plot of capture loss was found for the Saarijärvi system. The plot can be found in the Appendices (Fig. 44). The PR and array efficiencies are affected by the irradiance and/or cell temperature. The following section will deal with irradiation and temperature effects.

## 4.3 Effects of temperature and irradiance on the PV systems

It is generally known that increase in the cell temperature causes a reduction in the PV cell efficiency due to increased recombination rates of the internal carriers (holes and electrons).To find out to what extent this affects the PV systems under investigation, I began by verifying the module temperature distribution over the Agora PV array also to ascertain the accuracy of the thermocouple recorded cell temperature in comparison to manually recorded cell temperature. In addition a comparison between the cell and ambient temperatures was done. Afterwards, the cut-in irradiance for the respective PV arrays was found. Finally, the irradiance and temperature were used to determine their effect on the PV systems performance ratio and array efficiencies.

### 4.3.1 Cell and ambient temperature

One of the assumptions made was that the cell temperature is the same for the entire module and array. The thermocouple such as the one shown in Figure 8(b), measures the temperature of one module which is taken to represent the entire array. It is located at the top right corner module in the array. To test this assumption, the back side temperature of modules in the Agora PV array were measured and compared to the temperature recorded by the datalogger. To illustrate this, Figure 21 shows the temperature distribution on the Agora PV array with wind conditions below $2\,\mathrm{m\,s^{-1}}$. Data recorded by the datalogger at the same time shows a temperature range of 21.3 °C to 22.5 °C which is different from the temperatures recorded using a hand held infrared thermometer (24.8 to 30.4) °C. An explanation for the difference is the infrared thermometer measures instantaneous temperature while the recorded cell temperature is an average over one minute.

From the Figure 21, the coolest part of the array is on the top right hand corner which is

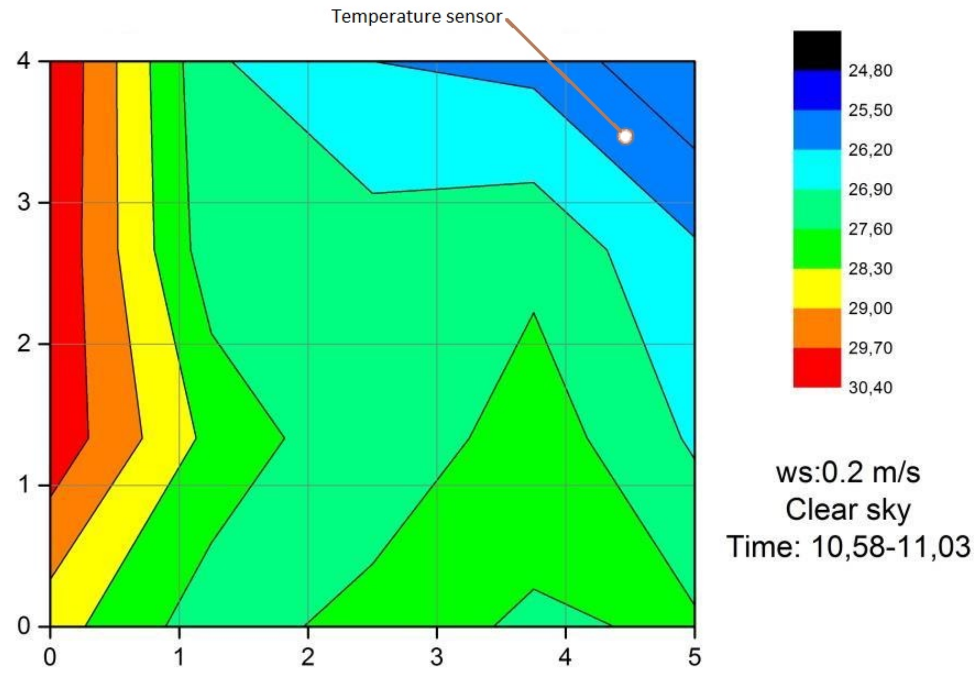



Figure 21: Temperature distribution on the back surface of the Agora PV array with a colour bar in degree Celsius and ws stands for windspeed

the position of the thermocouple while the hottest section is on the upper left hand corner. The reason for the difference in cell temperatures is the airflow, with the right side of the array being exposed to direct wind while the opposite side is partly protected. I suggest the thermocouple be moved to the third column and closer to the centre for a better mean cell temperature reading.

It was also assumed in the long run that the wind had a negligible effect on the cell temperature with relation to ambient temperature. To check if there was a large difference between the daily average ambient and cell temperatures, Figures 22 and 23 were plotted and it can be seen that the difference was small. The cell temperature will be used in further analysis since it directly affects the working of PVs.

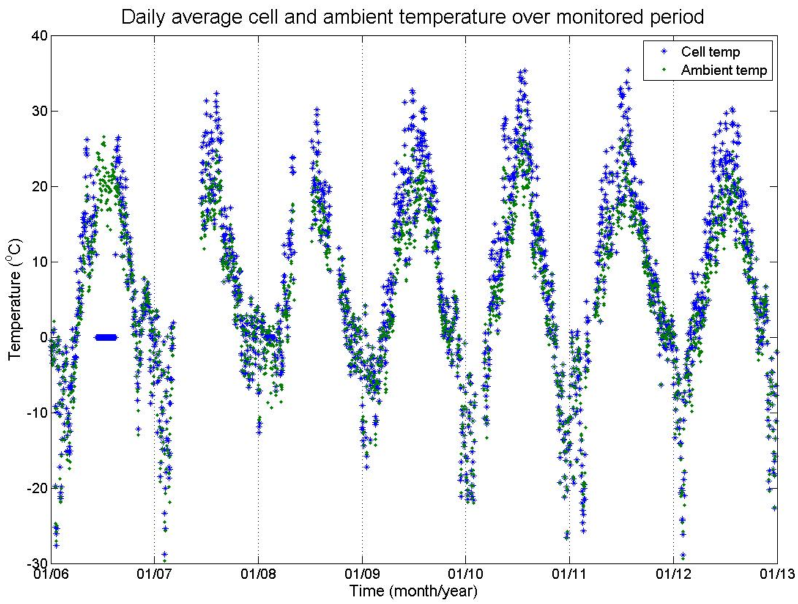



Figure 22: Cell and ambient temperatures measured in Saarijärvi

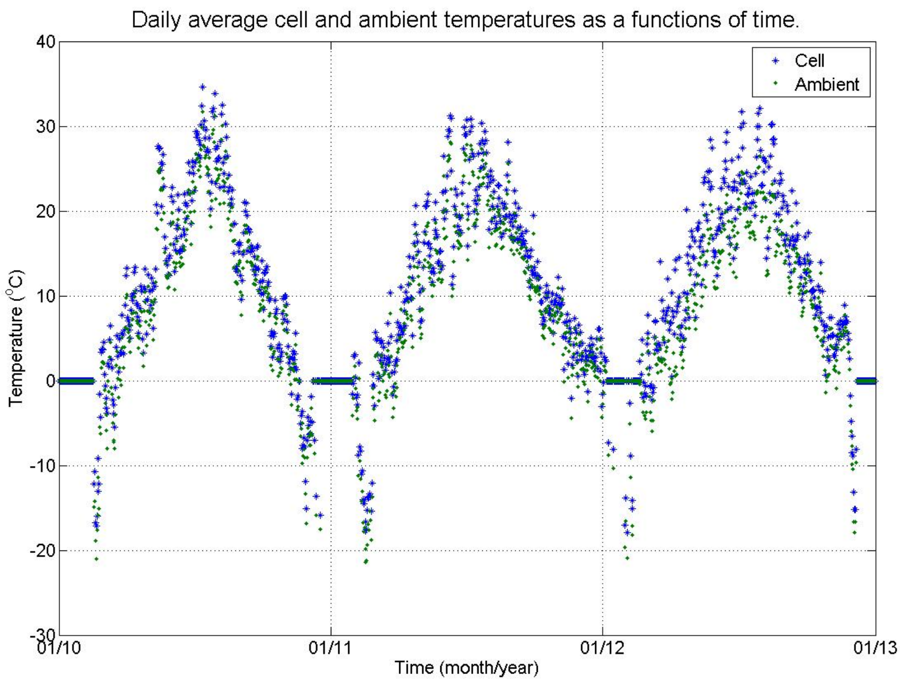



Figure 23: Cell and ambient temperatures measured in Agora

### 4.3.2 Cut-in irradiance

PV systems often start working above a certain irradiance level. The initial irradiance is used to overcome any internal resistance in the array and to start up the power conditioning equipment. This minimum irradiance is referred to as cut-in irradiance. For an ideal system, the array power is produced from irradiance level just above zero. However in practice irradiance value starting from 5 W/m$^2$ is acceptable. This is true for two out of the three cases. For the Agora system and Saarijärvi ss2, the cut-in irradiance is approximately 10 W/m$^2$ for both. This value was obtained from the scatter plot of the normalised array power as a function of instantaneous irradiance (see Figures 24 and 25). However, the scatter plot for ss1 shows array power even at irradiance close to zero yet ss1 and ss2 are both HIT with the same inverter manufacturer.

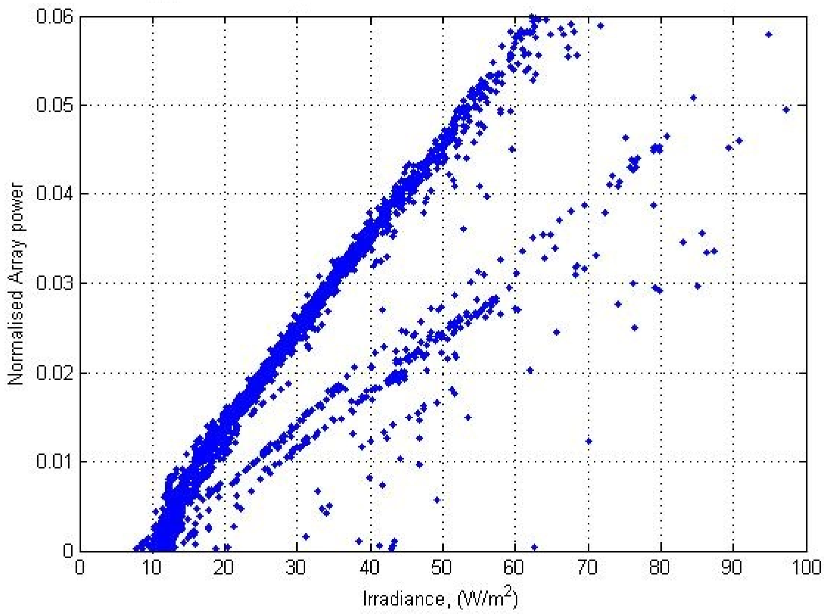



Figure 24: An example of the normalised array power as a function of the irradiance for the month of December 2011. The graph was used to determine the cut in irradiance for Agora PV system.

One reason for this difference is the topology. The ss1 is a mixture of parallel and series while ss2 is in series. Electrons absorb energy from incoming photons to move from the valence band to the conduction band and to conduct electricity. The ss1 has three branches of parrallel connections making it produce three times as much current as ss2 which is only in series. Also noting that each string of the branches of ss1 has nine modules and the string in ss2 has six modules, it means that the voltage difference at the terminals is in favour of ss1. Therefore the power generation occurs at a much lower irradiance in ss1 than ss2. Another possibility is that the initial low irradiance is used to power the inverter (maximum power point tracking) and datalogger and in Saarijärvi PV system, only one subsystem supplies the power to both inverters.

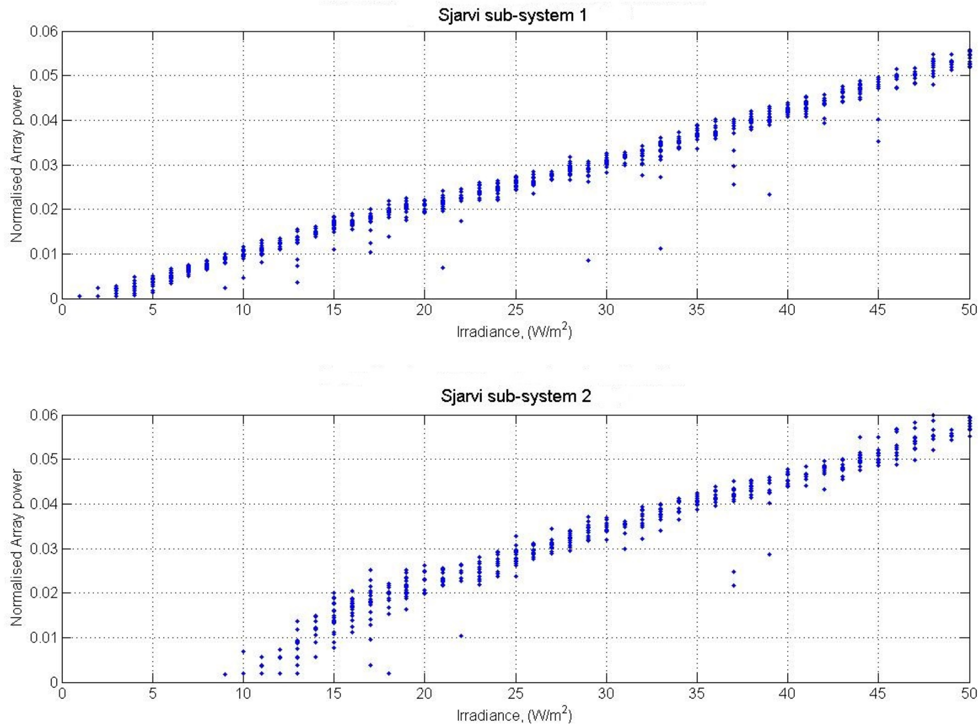



Figure 25: An example of the normalised array power as a function of the irradiance for the month of October 2012. The graph was used to determine the cut in irradiance for Saarijärvi PV system.

The appendices C.2 contains the graphs for all the months in a calender year for both systems. The plots are useful for fault detection. An example is Fig. 24 which has one main linear trend and several smaller ones below it. The smaller ones indicate a below par performance brought on by a mode of degradation. In this case, it is the array being soiled by snow in the month of December. Fig. 25 shows data for October which in comparison has few data points below the main linear data points.

The value of the cut-in irradiance was used to filter raw data to be analysed.

### 4.3.3 Performance ratio

From the time series section (Sec. 4.2) it was found that PR values have a seasonal pattern. The effect of cell temperature and irradiance on the PR of the system was verified by plotting PR as a function of irradiance and the data points colour coded to represent the cell temperature as shown in Figs. (26 - 28).

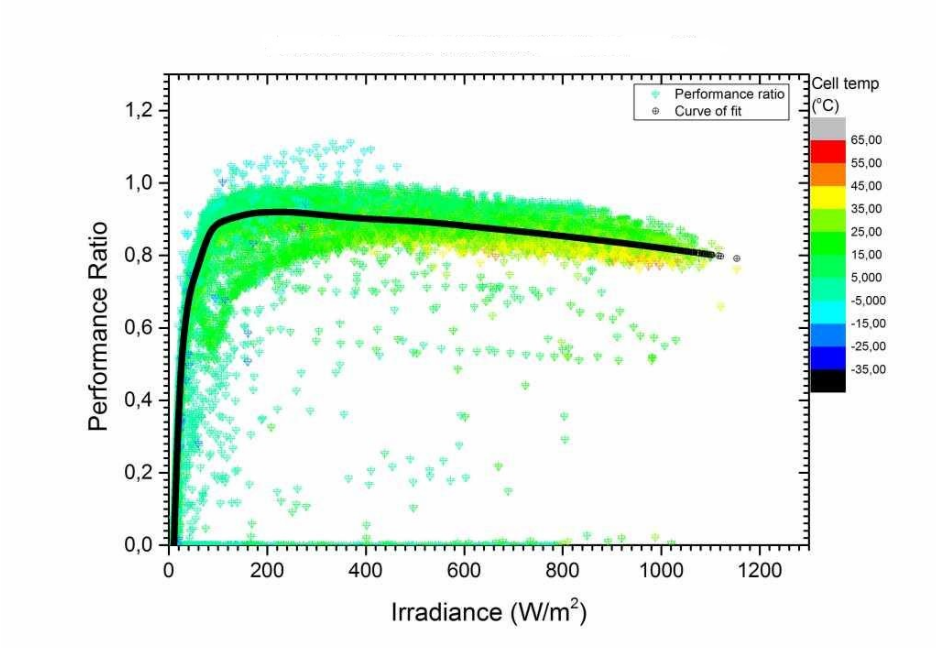



Figure 26: Performance ratio as a function of irradiance for Agora PV system. The colour bar indicates the cell temperature.

Figure 26 shows a rapidly increasing PR in Agora PV until it peaks at about 100 W/m$^2$. The plot also shows the decrease in PR as the temperature increases for a given irradiance level. For a temperature of between 5 °C and 10 °C, PR is approximately 0.9. In comparison, the PR value for Saarijärvi (Figs. 27 and 27) climbs more rapidly and reaches its peak value before 20 W/m$^2$. The shape of both curves differ due to topology difference. Comparing the PRs of the three systems in Fig. 29, we can see that ss2 has the highest PR and that ss1

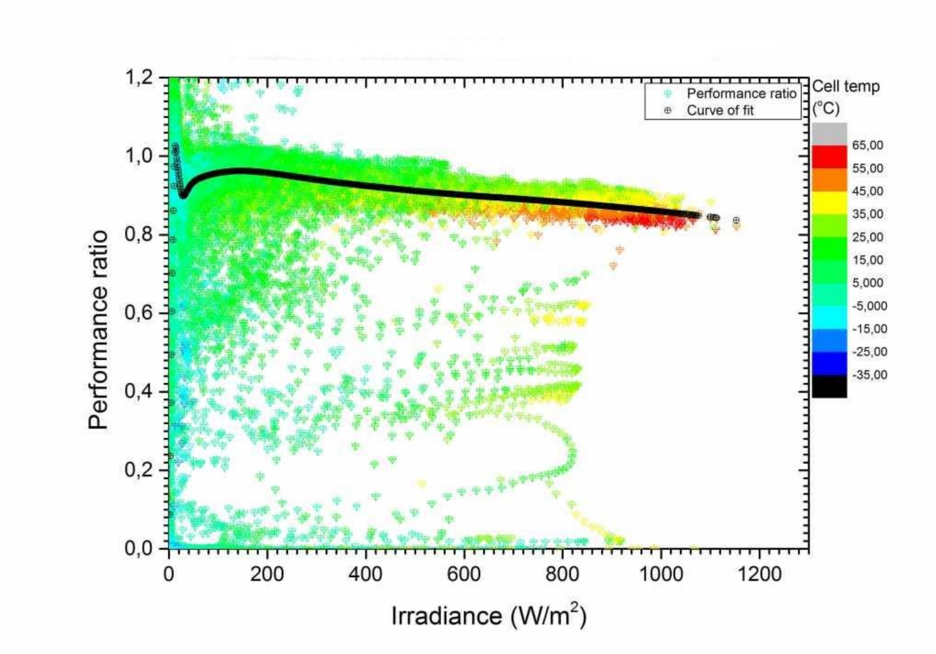



Figure 27: Performance ratio as a function of irradiance for Saarijärvi ss1

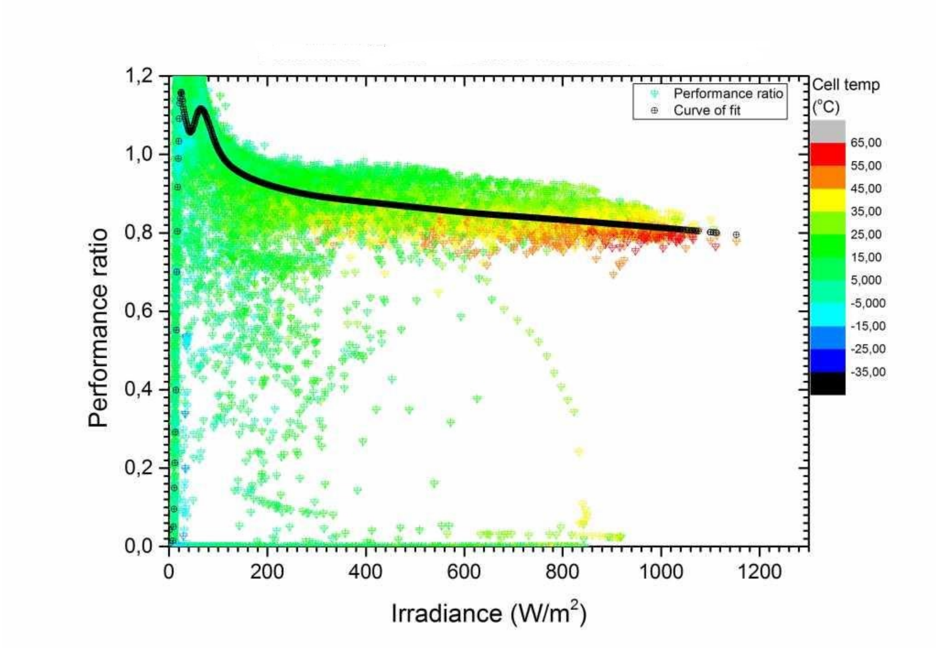



Figure 28: Performance ratio as a function of irradiance for Saarijärvi ss2

achieves its highest PR at the lowest irradiance level compared to the rest. Above 120 W/$m^2$, ss1 has the best PR while the PR of Agora is better than ss2 above 220 W/$m^2$.

All three performance ratio plots show initially an increasing trend followed by a steady decrease as the irradiance increases. It should be pointed out that the Saarijärvi curves are marked by two maxima. The cause of which can be a possible recording error in the winter months if the reference cell was partially obstructed by snow. The recording technique at low irradiance needs to be looked into. The first maxima occurred when the temperature was between -5 °C and -15 °C (See Figs.27 and 28 ).

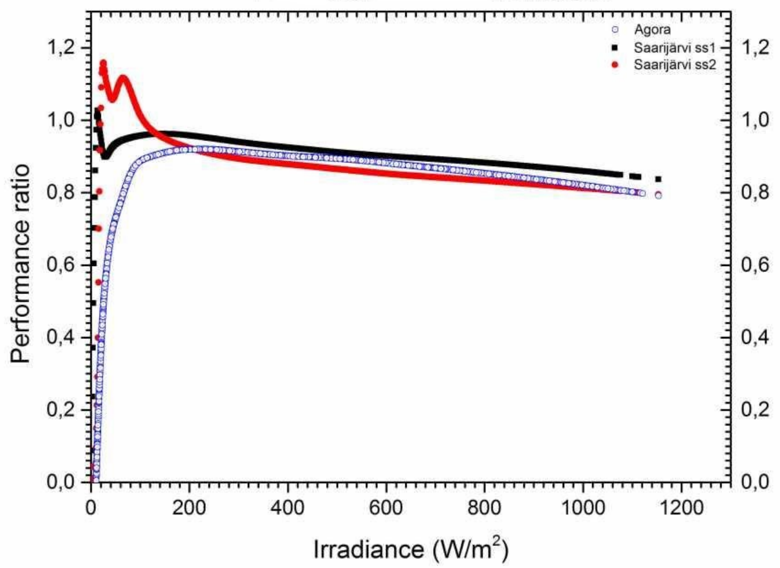



Figure 29: Performance ratio as a function of irradiance for all three systems

#### 4.3.4 Array efficiency

The performance ratio looks at a system in its entirety meaning from the array to the grid and everything in between. In order to see how the arrays behave at different irradiances and temperatures, I normalised the array efficiency using the module efficiency provided by the manufacturer at STC and plotted them as functions of irradiance.

Figure 30 shows the normalised array efficiencies as function of irradiance. In all four plots, the normalised array efficiency had rapid rise and a steady decline as the irradiance increased. In Figs. 30(a)-30(c), the influence of cell temperature on the array efficiency is evident. For a given irradiance level, the efficiency of the array is greater at a lower cell temperatures.

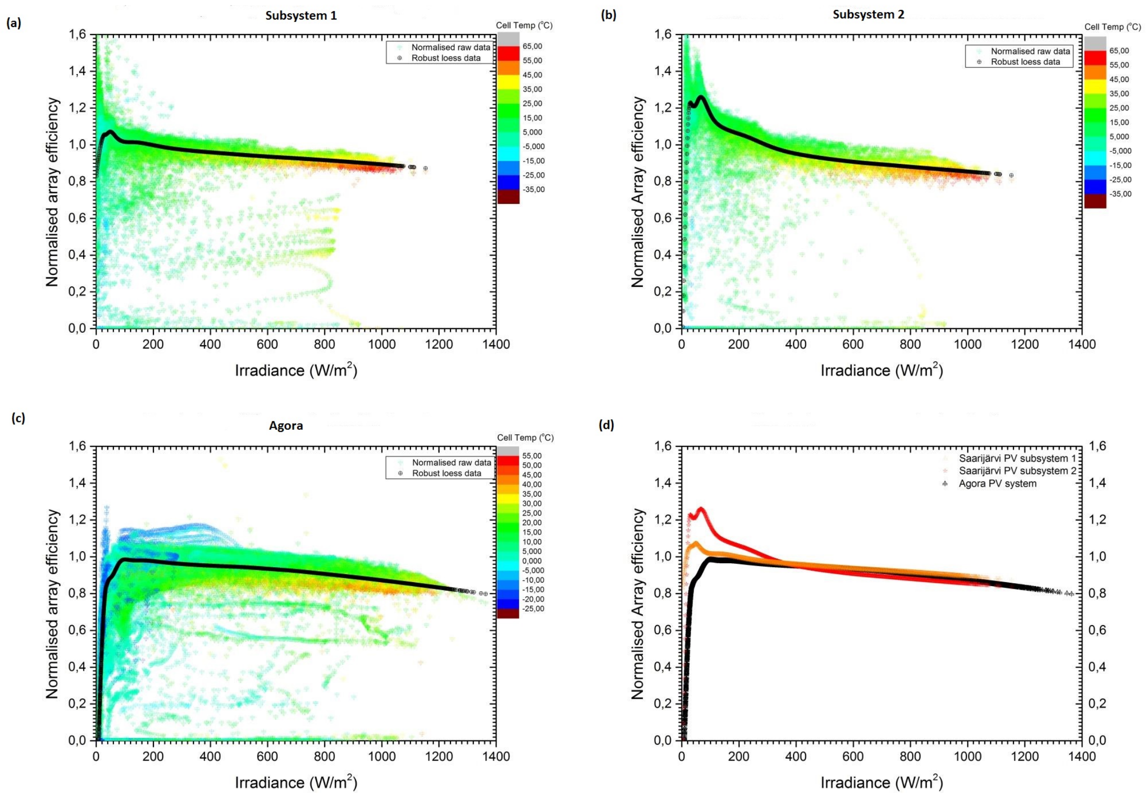



Figure 30: From top left: Scatter plot of normalised array efficiency as a function of irradiance: (a) Saarijärvi PV ss1; (b) Saarijärvi PV ss2; (c) Agora PV system; (d) smoothed curves for the three PV arrays (Agora-(Black), ss1-(Orange), ss2-(Red)).

Furthermore, the three plots show that the efficiencies peak below 100 W/m$^2$ which makes the PVs suitable for Central Finland climate since a large fraction of in-plane irradiance is below 100 W/m$^2$. The difference in the curve shapes can be attributed to the PV technology and topology differences. Both Saarijärvi curves peak above the stated module efficiency at STC meaning that they perform well in cold and low insolation conditions. However, when comparing the two Saarijärvi PV array curves, it can be seen that the ss2 reaches a higher efficiency but declines faster than ss1.

When all three curves are plotted together (see Fig. 30(d)), below 400 W/m$^2$, there are differences in the normalised array efficiencies. The Saarijärvi curves (the red and orange) have higher normalised efficiencies than the Agora one. This would mean that at low light the

HIT technology found in Saarijärvi would be better suited than the polycrystalline found in the Agora system. At around 400 W/m$^2$, all the arrays operate at the same percentage of the arrays efficiencies at STC ($\approx$ 95%). Beyond the 400 W/m$^2$, ss1 has the least rate of decline while ss2 has the highest. It should be noted that ss2 would still have a greater array efficiency than the Agora PV. According to Ross and Royer [44], modules with poor low light performance peak at higher irradiance levels (such as Agora PV array). They also stated that it is difficult for consumers to differentiate between good and poor low light performers because manufacturers only specify performance parameters at STC. For PVs to be better accepted in high latitudes and cold climates, manufacturers should show proof of performance of their products in the aforementioned conditions.

In summary, considering the insolation level below 400 W/m$^2$ is nearly 80 % according the histogram (Fig. 14), ss2 is the most efficient of the three arrays. When considering the full possible irradiance range, ss1 is better suited than ss2.

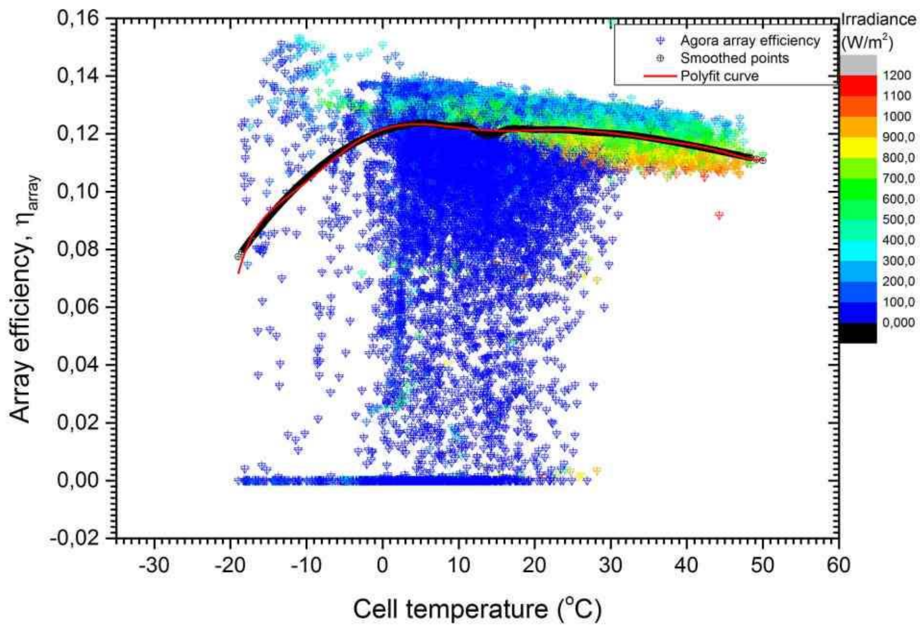



Figure 31: Agora array efficiency as a function of cell temperature. A smoothed data set was used to find a line of fit. The irradiance values are daily averages.

The effect of cell temperature on the array efficiency is shown in Fig. 31, 32 and 33 for Agora and Saarijärvi ss1 and ss2 respectively. In Fig. 31, the cell temperature range is from

-20 °C to 50 °C. The irradiance level is predominantly below 200 $W/m^2$ at temperatures below zero degrees celsius. From the graph, it can be seen that the array efficiency is highest at low temperatures and irradiance between 300 and 400 $W/m^2$. In order to find a general expression for the behaviour of the array at Agora, a polynomial curve of order 10 was plotted using the smoothed data. From the plotted polynomial fit, the maximum array efficiency was found to be (12.34±0.01) % at a temperature of 4.5 °C. When compared to the value in the Naps datasheet and the energy balance (13.1 % and 11.6 %, respectively), it is approximately the mean of the two values. The array efficiency temperature coefficient can be found using the derivative of the polyfit curve equation which is given as:

$$\begin{aligned}\frac{\mathrm{d}y}{\mathrm{d}x} &= a_1 + 2a_2(x) + 3a_3(x)^2 + 4a_4(x)^3 + 5a_5(x)^4 \\ &\quad + 6a_6(x)^5 + 7a_7(x)^6 + 8a_8(x)^7 + 9a_9(x)^8 \qquad \text{for all } x \in \mathbf{R}\end{aligned}$$

where $a_1 = 8.909 \times 10^{-04}$, $a_2 = -1.192 \times 10^{-04}$, $a_3 = 1.231 \times 10^{-06}$, $a_4 = 3.689 \times 10^{-07}$, $a_5 = -1.076 \times 10^{-08}$, $a_6 = -4.635 \times 10^{-10}$, $a_7 = 2.814 \times 10^{-11}$, $a_8 = -5.026 \times 10^{-13}$ and $a_9 = 3.0904 \times 10^{-15}$. The result is multiplied by 100 to get the unit as $\%\,°C^{-1}$. Unfortunately, there was no value in the Naps datasheet with which to compare.

Similar plotted graphs were done for Saarijärvi ss1 and ss2 (Figs.32 and 33). There was a decrease in the array efficiency in both ss1 and ss2 above 11 °C. Both ss1 and ss2 have high array efficiency at low irradiance levels and it is evident by the amount of blue colour in the upper part of the plots.

Comparing the Agora and Saarjärvi systems at cell temperature below zero degrees, there is a relatively rapid decline in array efficiency in the Saarijärvi systems as compared to the Agora one. One hypothesis is that the snow on the Saarijärvi arrays does not melt until much later in the spring when it is warmer which may not be the case at Agora. The Agora array was exposed to irradiance above 500 $W/m^2$ at subzero degrees celsius (This is evident in the plot Fig. 31) but the same cannot be said of the arrays at Saarijärvi. Site visits during winter would be useful in determining the cause. The Saarijärvi systems also have a sudden increase below -20 °C. It is worth observing that the smoothed data points go below zero value because of a limitation in the smoothing function used. For this reason,

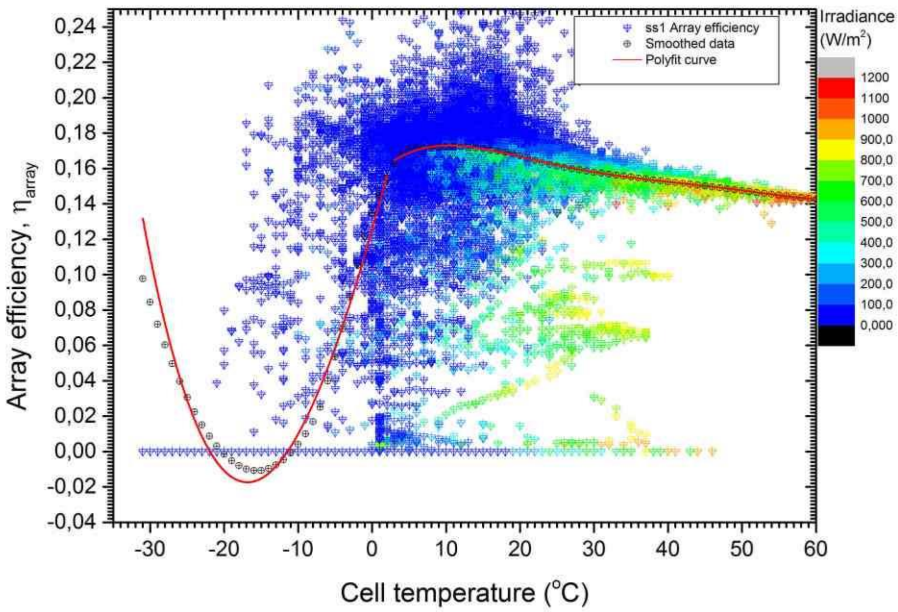



Figure 32: Saarijärvi subsystem 1 array efficiency as a function of cell temperature. The irradiance values are daily averages.

low temperature values below -5 °C will be disregarded for now.

Using the polynomial fitted curves on the smoothed data points, the maximum array efficiencies for ss1 and ss2 are: (17.29±0.01) % at a temperature of 10.3 °C and (19.71±0.01) % at a temperature of 8.9 °C respectively. These values compared to the one provided in the Sanyo datasheet (16.5 % at STC) are relatively higher. This would mean that the arrays are generally more efficient in cold temperature than at STC.

The ss2 array is more sensitive to temperature increase as compared to the ss1 array. This can be observed by the steeper gradient in the curve in ss2 (Fig.33) as compared to ss1 above zero degrees celsius. The array efficiency temperature coefficient for ss1 is

$$\frac{\mathrm{d}y}{\mathrm{d}x} = a_1 + 2a_2(x) + 3a_3(x)^2 + 4a_4(x)^3 + 5a_5(x)^4 \qquad \text{for } x \geq 0$$

where $a_1 = 0.004$, $a_2 = -3.611 \times 10^{-04}$, $a_3 = 1.096 \times 10^{-05}$, $a_4 = -5.520 \times 10^{-07}$ and $a_5 = 7.935 \times 10^{-10}$. The result is multiplied by 100 to get the units as $\%\,°\mathrm{C}^{-1}$.

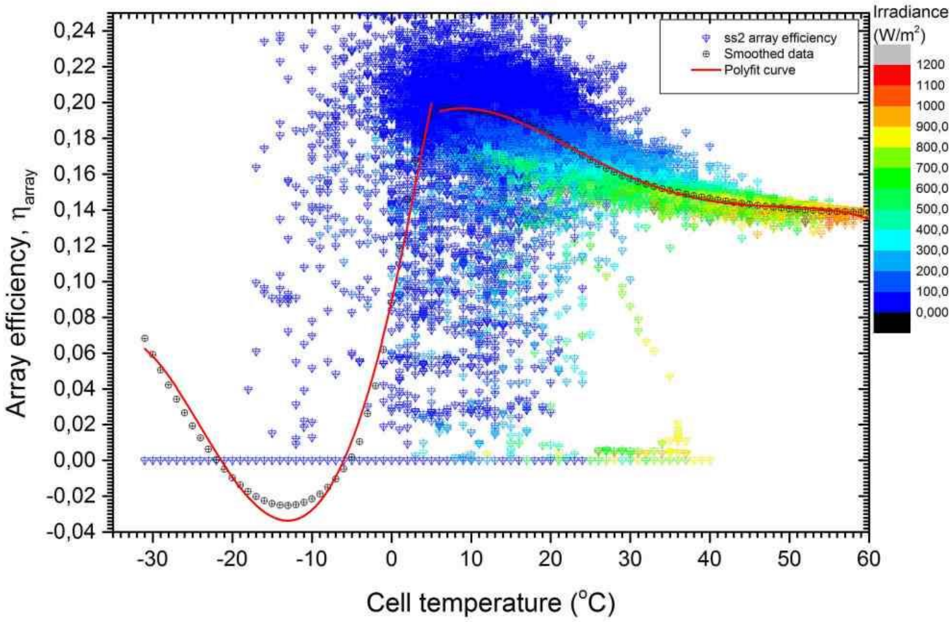



Figure 33: Saarijärvi subsystem 2 array efficiency as a function of cell temperature. The irradiance values are daily averages.

While the array efficiency temperature coefficient for Saarijärvi ss2 is described as follows:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = a_1 + 2a_2(x) + 3a_3(x)^2 + 4a_4(x)^3 + 5a_5(x)^4 \qquad \text{for all } x \geq 0$$

where $a_1 = 0.005$, $a_2 = -3.911 \times 10^{-04}$, $a_3 = 8.443 \times 10^{-06}$ and $a_4 = -5.905 \times 10^{-08}$. The result is multiplied by 100 to get the units as $\%\,°\mathrm{C}^{-1}$.

Normalised array efficiencies using the module efficiencies at STC values were plotted in Fig. 34. At cell temperatures (0 °C$\leq$ cell temp. $\leq$ 30 °C), the Saarijärvi arrays have better efficiencies than the Agora array. Below zero degrees, the Agora array performs best. However, this is a cautious conclusion since the amount of snow cover on the different arrays is unknown during the winter time. Future work can be done to gain a better understanding.

Comparing the histogram and probability plots (Fig. 35) for the cell temperatures at the two sites, the probability of the temperature being in the range of ($-2.5$ °C$\leq$ cell temp. $\leq$ 32.5 °C) is 91.18 % and 76.36 % for Agora and Saarijärvi respectively. The mean and standard deviation for the cell temperatures in Agora and Saarijärvi were found to be (16.40 $\pm$ 10.58) °C and (15.10 $\pm$ 15.00) °C respectively.

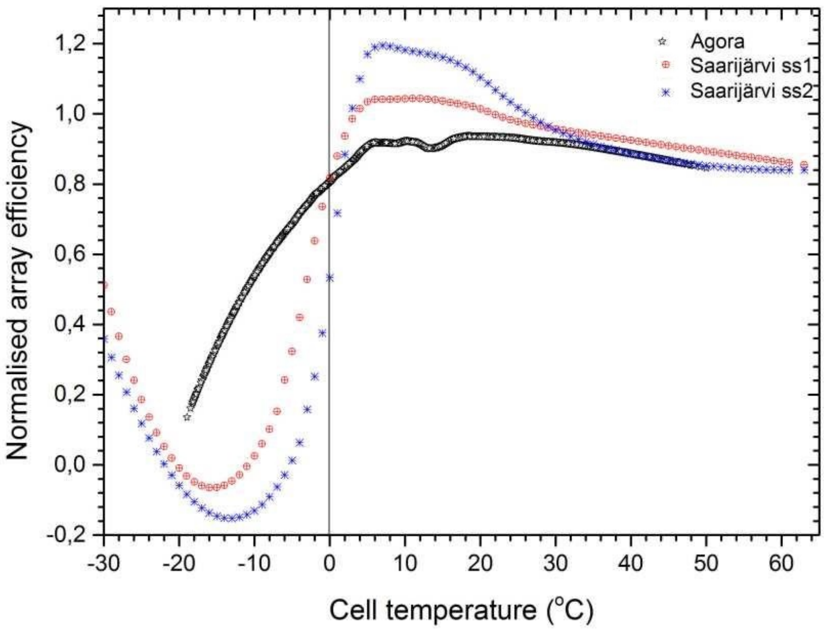


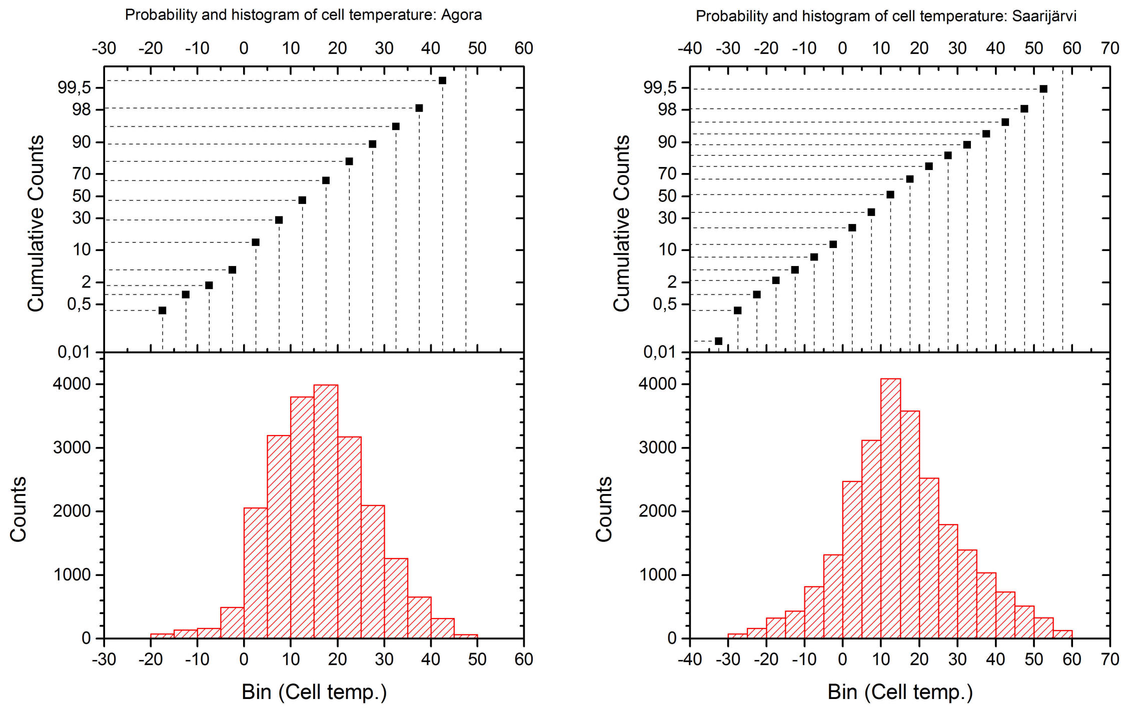

Figure 34: Normalised array efficiencies as a function of cell temperature.

In summary, the HIT PV found in Saarijärvi has generally higher array efficiency for temperatures above zero degrees than polycrystalline PV found in Agora. The p-Si has generally a better array efficiency than HIT at sub-zero temperature using the data collected.

## 4.4 Inverter efficiency

The inverter at both sites are not affected by ambient temperature since they are located indoors so they are only affected by the input power (array power). Figure 36 shows the behaviour of the inverters at different PR's and array yields. The inverter in Agora has a linear increase from zero to 0.8 for array yields between zero and $1\,\mathrm{h\,d^{-1}}$. The low array yield values coincide with low light conditions or short daylight hours. When compared to the inverters in the Saarijärvi system, they have above 80 % efficiency under all conditions. This suggests that the Saarijärvi inverters have a lower conversion loss than the Agora inverters hence they handle partial load better.

Figure 37 shows the inverter efficiency as a function of partial load for all three inverters. The




(a) (b)

Figure 35: Probability and histograms of the cell temperatures of the arrays.

partial load in this case is the ratio of the input power to the rated capacity of the inverter. In the legend of the figure, Agora, Sjarvi 1 and Sjarvi 2 correspond to the Sunny Boy inverter 2500, Fronius IG 60HV and Fronius IG 15, respectively. Sjarvi 1 and Sjarvi 2 correspond to ss1 and ss2, respectively. Some observations from the plot were noted. First, the fronius and sunny boy inverters have different curve shapes. Second, the inverter in Saarijärvi ss2 is oversized since the partial load maximum is 0.7 of the power input capacity of the inverter. The inverter input rated capacity is 1610 W which is larger than the array maximum output of 1140 W. Finally, the fronius inverters have efficiencies that exceed the practical limit of 100 %.

The sunny boy inverter follows the same shape as in Fig. 6 but the Fronius inverters do not. The reason for this is the rounding off error in the measurement of the current from the PV. While Agora PV current is measured in milliamperes, the Saarijärvi current is measured in amperes to 2-decimal places. In addition, the Agora system has a shorter recording and

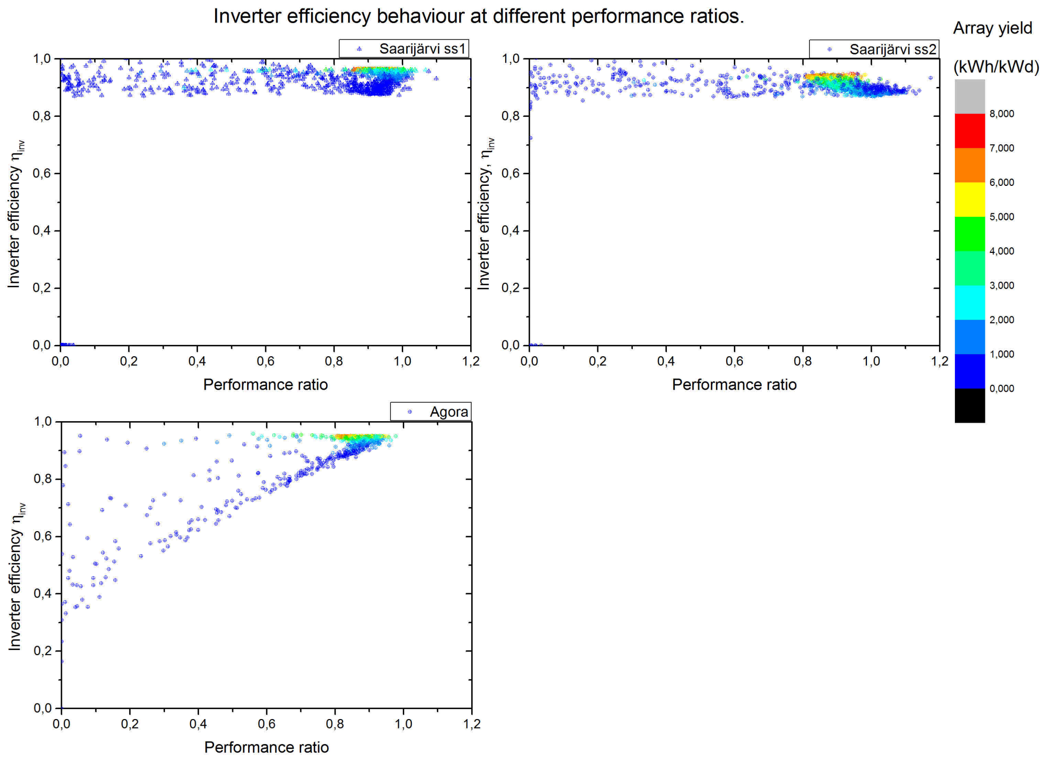



Figure 36: The inverter efficiencies as a function of performance ratio. From bottom left clockwise: Agora; Saarijärvi ss1; ss2.

sampling interval compared to Saarijärvi thus has greater accuracy. The result of which is some values above 100 % at low partial load.

The inverter with the least efficiency at low partial loads is the Agora inverter because at partial loads below 0.1, Sjarvi 1 and Sjarvi 2 have minimum efficiencies of 63 % and 70 % respectively while Agora has a minimum of 10 % efficiency. Assuming that the first decreasing portion of the inverter efficiency curves for Sjarvi 1 and Sjarvi 2 were ignored, Sjarvi 1 would have the least start up loss while Sjarvi 2 would have the most start up loss. Sjarvi 1 inverter has less load loss than the Agora inverter. For cold climates, the response of the Sjarvi 1 inverter (Fronius IG 60 HV) would be ideal. Oversizing of an inverter in this case has resulted in decreased efficiency of the inverter. But how proportion of the partial load is received by the inverter? To find out, a histogram of DC and AC power was plotted. The histogram aided in determining the suitability of the inverter for the given PV system.

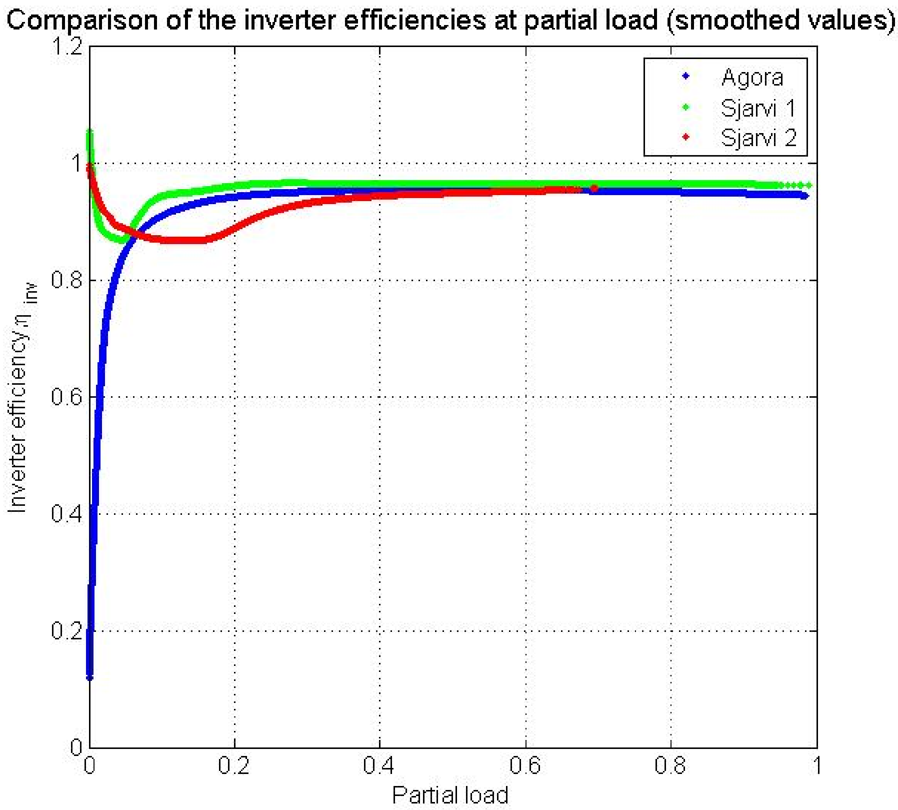



Figure 37: The three inverter efficiencies as functions of partial load to inverter.

DC and AC power histograms for each subsystem were plotted as shown in Figures 38, 39 and 40. For the three figures, initially the AC bar was taller than the DC bar. Again, the sensitivity of the datalogger at low light and partial power below 10 % causes this discrepancy. Between 10 % and 20 % normalised power, all the systems show losses between DC to AC conversion. From 20 % to 80 % the inverter efficiency is high. Afterwards, the conversion rate drops again as the arrays reach their maximum power output.

In Figure 38, 45 % to 50 % of the total energy produced in Agora system is less than 10 % of the nominal array power. Less than 10 % of the total energy is produced above 70 % of the nominal power for the Agora PV array. In Saarijärvi, less than 10 % of the nominal array power contributes between 50 % and 60 % of the total energy (See Figure 39 and 40). Less than 2 % of the total energy produced by all three systems was above 90 % of the nominal array power.

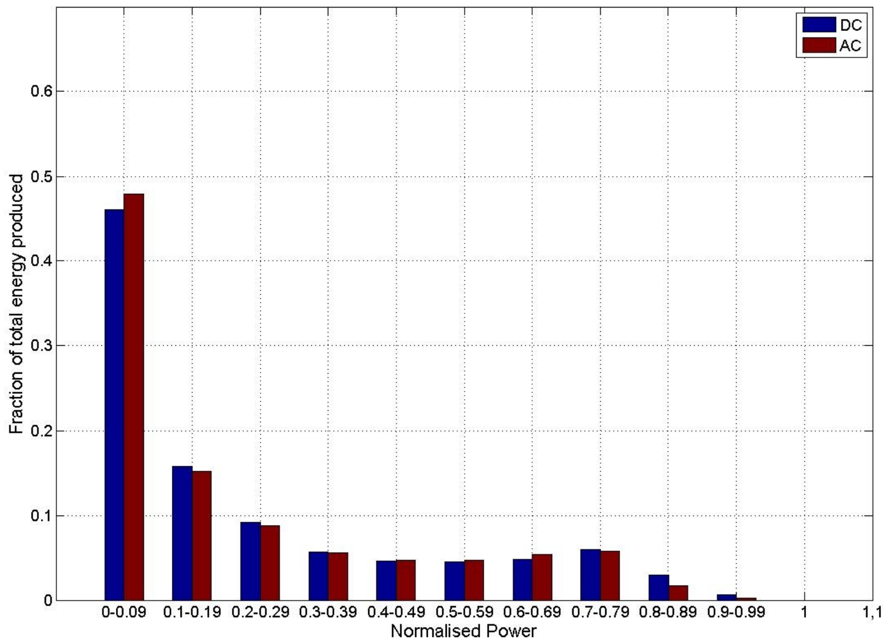



Figure 38: Histogram of normalised power for Agora PV system.

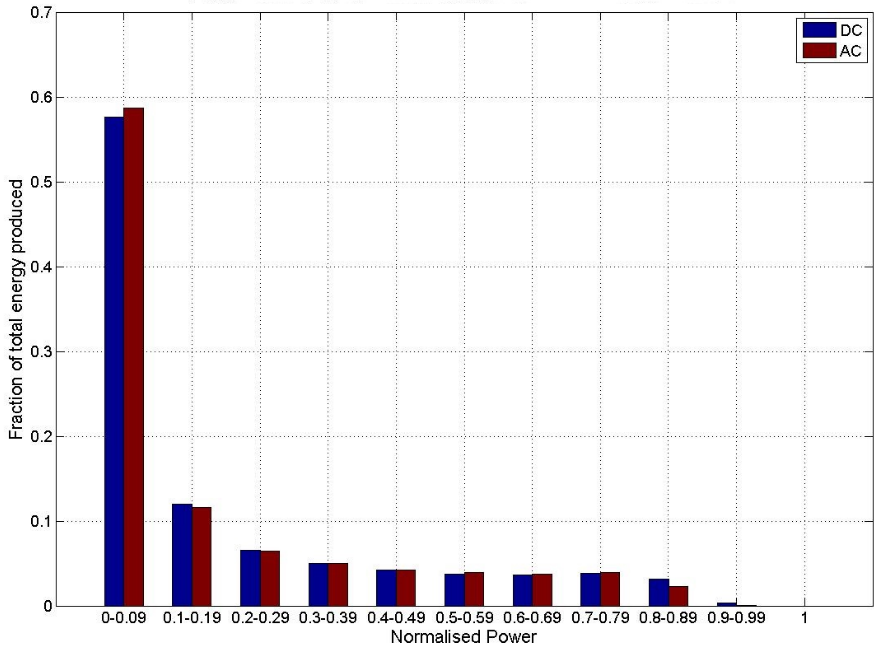



Figure 39: Histogram of normalised power for Saarijärvi PV subsystem 1.

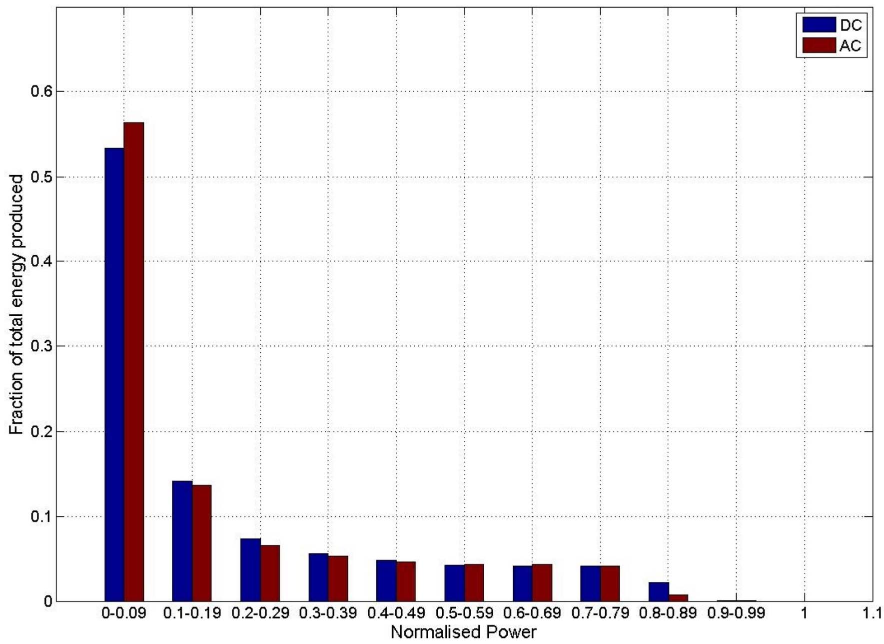



Figure 40: Histogram of normalised power for Saarijärvi PV subsystem 2.

Undersizing of the inverter by 90 % of the nominal array power would not cause a significant difference in the total energy out from the PV systems. The effect of oversizing the inverter is seen in the ss2 histogram (Fig. 40) which shows a larger conversion loss between 0.1 and 0.5 of the normalised array power as compared to ss1 (Fig. 39). This accounts for approximately 30 % of the total energy. These losses can be avoided by appropriate inverter sizing.

## 4.5 Degradation rates

The ARIMA calculations were done in R and the corresponding code can be found in the appendices (Appendix B). The main reason for using ARIMA models was to find new values that took into account previous data and the current data. This new dataset took into account outlier values, missing data and any data drift from the equipment. An advantage of using the ARIMA model is the possibility to forecast what the performance would be within a predetermined period.

The ARIMA values and corresponding 'raw' values were saved in mat files and these files were used to plot the values on a graph using OriginPro 9. I took a value of 74 % or 0.74 as a lower limit. This was in order to use values of PR when there was sufficient irradiation and the array was mostly free of snow. Another reason was that the system would be deemed a commercial 'liability' below this performance ratio. The values coincided with the months of late March to late October. Since STC conditions are difficult to attain in the field, a range of values would suffice. In order to increase the number of data points to get a trend, average weekly PR values were used to perform the calculations.

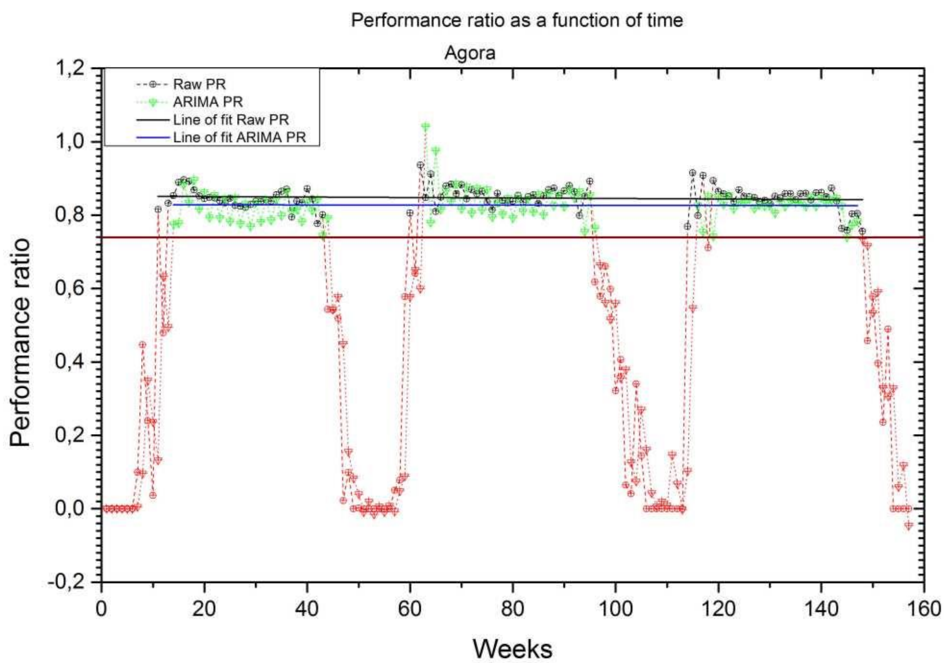



Figure 41: Graph showing the degradation of the Agora PV system. The ARIMA model used was ARIMA$(2,0,2)(0,0,1)_{52}$.

Figure 41 shows the rate of degradation for the Agora PV system. The ARIMA model value obtained was $(2,0,2)(0,0,1)_{52}$. The calculated maximum PR values at the beginning of the monitoring period were $0.852 \pm 0.007$ and $0.829 \pm 0.009$ for the raw and ARIMA datasets respectively. Using the line gradients, the degradation rates were found to be $(-0.40 \pm 0.44)$ %/year and $(-0.10 \pm 0.65)$ %/year for raw and ARIMA respectively. The error was determined from the standard deviation of the slope. From the latter values, it is inconclusive to state if there is degradation or not since the error values are larger than the degradation percentages. Looking at figure 41, generally the value of PR goes up in the midsection as compared to the preceding and succeeding sections. Therefore there was no clear trend.

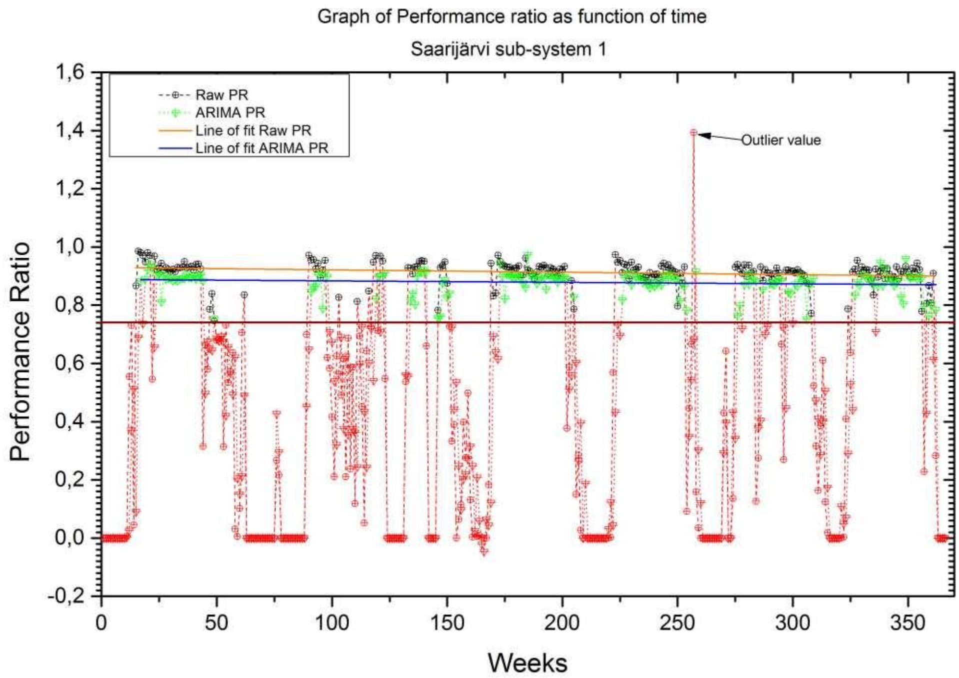



Figure 42: Graph showing the degradation of the Saarijärvi PV ss1. The ARIMA model used was ARIMA$(5,0,0)(1,0,0)_{52}$.

Figure 42 shows the calculated PR values for ss1 from the graph, which are $0.930 \pm 0.006$ and $0.889 \pm 0.006$ for the raw and ARIMA datasets respectively. The corresponding degradation rates are $(-0.45 \pm 0.16)$ %/year and $(-0.29 \pm 0.16)$ %/year. When attempting to find a seasonal ARIMA using the PR data from Saarijärvi ss2, the result from the auto.arima function gave a non-seasonal value of ARIMA$(3,0,0)$ model. The resulting plot is Fig. 43. Using the values obtained from the linear line of best fit, maximum PRs were $0.911 \pm 0.007$ and $0.879 \pm 0.006$ for raw and ARIMA model datasets respectively. The corresponding degradation rate values found were $(-0.32 \pm 0.18)$ %/year and $(-0.36 \pm 0.16)$ %/year for the raw and ARIMA datasets, respectively. The degradation rates that have been calculated using the ARIMA model values are between (-0.10 and -0.36) % per year. Using the initial calculated PR value for the corresponding systems and taking a PR of 0.8 as a base value, the calculated working life for the PV systems are: $(35.0 \pm 53.8)$ years; $(34.6 \pm 19.5)$ years and $(25.0 \pm 10.6)$ years for Agora and Saarijärvi ss1 and ss2, respectively. Although all PV

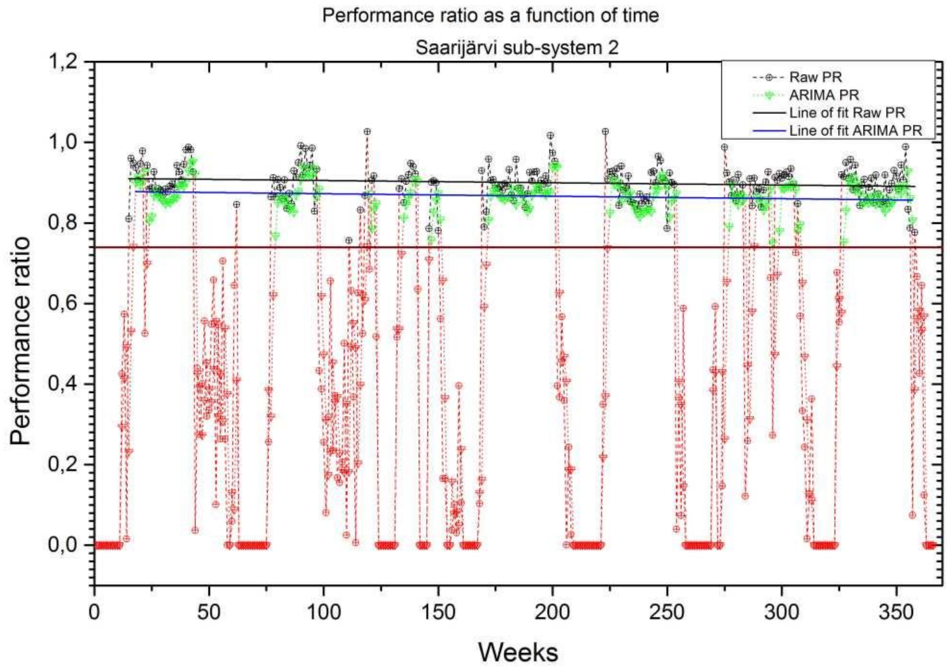



Figure 43: Graph showing the degradation of the Saarijärvi PV ss2. The ARIMA model used was ARIMA$(3,0,0)$.

systems show above 25 year working life, the Agora PV system does not show a clear degradation since the standard deviation value is much greater than the degradation rate value. However, if the p-Si array is similar to the one found in Huvudsta, Sweden [40], the array will show little degradation after a longer period than the HIT arrays at Saarijärvi. To confirm this assumption, degradation rate analysis should be done after another additional full year of operation to confirm this trend.

The degradation rates for ss1 and ss2 are different by 0.07 %/year yet they are the same PV technology. The variance is attributed to the inverters and topology. The efficiency of the inverters affects the final yield which in turn affects the performance ratio. This can be seen in Tables 7 and 8. With reference to the calculated working life, this difference can be equivalent to about 9 years of system life gained or lost.

# 5 Conclusion

A study was undertaken on two PV systems located in Central Finland. The Agora PV system which is made up of a polycrystalline PV array located in Jyväskylä and the Saarijärvi PV system made up of two heterojunction with intrinsic thin layer PV arrays located in Saarijärvi. Both systems are grid-tied. The aim of the study has been to find the degradation rate of the PV systems and to report on the long term performance of the same. This led to a comparison of the performance of the two PV technologies and the effect of different array topologies on energy production. Finally, an estimation on the working life expectancy of the PV system was calculated.

Despite the data recording interruptions, the PV arrays at both sites worked well with no maintenance. The inverter at the Agora site has also worked without disruptions for the reported period but one of the inverters at Saarijärvi had a few breakdowns which were resolved. The difference in array topology has strong influence on power production with reference to the data obtained from the Saarijärvi site. With regard to the PV technologies, the HIT proved to perform better than p-Si under similar field conditions in Central Finland. In addition, the importance of proper inverter selection was brought out. Finally, the life expectancy of the PV system exceeded those given in the warranty periods of the respective PV module manufacturers.

This thesis study was used to quantify the degradation but not the mode of the degradation. Investigation into the main causes of the deterioration of the PV system especially in Saarijärvi should be undertaken. An assumption has been made that the calculated degradation rate of the PV system is inclusive of all components (PV array, inverters and wiring). Further study into whether any one component has a more dominating influence on the degradation rate should be done. Further study into the degradation rate of Agora PV system should be undertaken so as to get a more accurate deterioration rate.

The study has thus far proven that solar electricity can be viable as far north as Central Finland with predicted PV system working life of 25 years and more.

# Bibliography

[1] A. Jäger-Waldau, “JRC scientific and policy reports PV status report 2012”, Joint Research Centre of the European Commission, Ispra, Italy, technical, 2012.

[2] Google Maps, *Map of Finland: Positions of Jyväskylä and Saarijärvi*, 2013. [Online]. Available: `https://maps.google.com/` (visited on 26/04/2013).

[3] M. C. Peel, B. L. Finlayson and T. A. McMahon, “Updated world map of the Köppen-Geiger climate classification”, *Hydrology and Earth System Sciences Discussions*, vol. 4, no. 2, pp. 1633–1644, 2007.

[4] P. A. Basore, “Defining terms for crystalline silicon solar cells”, *Progress in Photovoltaics: Research and Applications*, vol. 2, no. 2, pp. 177–179, 1994.

[5] S. R. Wenham, M. A. Green, M. E. Watt and R. Corkish, *Applied photovoltaics*, 2nd ed. Earthscan, 2007, ch. Semiconductors and p-n junctions, pp. 31–41.

[6] A. Braga, S. Moreira, J. Zampieri P. Bacchin and P. Mei, “New processes for the production of solar-grade polycrystalline silicon: A review”, *Solar energy materials and solar cells*, vol. 92, no. 4, pp. 418–424, 2008.

[7] Cleave Energy Inc, *HIT solar cell*, N/A. [Online]. Available: `http://www.cleave-energy.com/images/sanyo\_HIT-solar-cell-spec.jpg` (visited on 05/01/2013).

[8] N. Hernandez-Como and A. Morales-Acevedo, “Hetero-junction (HIT) silicon solar cell model for AMPS-1D simulation”, in *2008 5th International Conference on Electrical Engineering, Computing Science and Automatic Control*, IEEE, Nov. 2008, pp. 449–454.

[9] Sanyo, *Technical Description HIP-190NE1*, 2004.

[10] T. Razykov, C. Ferekides, D. Morel, E. Stefanakos, H. Ullal and H. Upadhyaya, “Solar photovoltaic electricity: Current status and future prospects”, *Solar Energy*, vol. 85, no. 8, pp. 1580–1608, 2011.

[11] Farlex. (2013). The free dictionary, [Online]. Available: `http://encyclopedia2.thefreedictionary.com/Passivation+layer` (visited on 12/03/2013).

[12] T. Mishima, M. Taguchi, H. Sakata and E. Maruyama, “Development status of high-efficiency HIT solar cells”, *Solar Energy Materials and Solar Cells*, vol. 95, no. 1, pp. 18–21, Jan. 2011.

[13] J. Leloux, L. Narvarte and D. Trebosc, “Review of the performance of residential PV systems in France”, *Renewable and Sustainable Energy Reviews*, vol. 16, no. 2, pp. 1369–1376, Feb. 2012.

[14] A. Sasitharanuwat, W. Rakwichian, N. Ketjoy and S. Yammen, “Performance evaluation of a 10kWp PV power system prototype for isolated building in Thailand”, *Renewable Energy*, vol. 32, no. 8, pp. 1288–1300, Jul. 2007.

[15] V. Sharma, A. Kumar, O. Sastry and S. Chandel, “Performance assessment of different solar photovoltaic technologies under similar outdoor conditions”, *Energy*, vol. 58, pp. 511–518, Sep. 2013.

[16] S. Evans. (28th Dec. 2012). BBC News - Germany: Nuclear power plants to close by 2022, [Online]. Available: `http://www.bbc.co.uk/news` (visited on 30/05/2011).

[17] G. Blaesser, “PV system measurements and monitoring the European experience”, *Solar Energy Materials and Solar Cells*, vol. 47, no. 1-4, pp. 167–176, Oct. 1997.

[18] G. Makrides, B. Zinsser, G. E. Georghiou, M. Schubert and J. H. Werner, “Degradation of different photovoltaic technologies under field conditions”, *2010 35th IEEE Photovoltaic Specialists Conference*, pp. 2332–2337, Jun. 2010.

[19] B. Marion, J. Adelstein, K. Boyle, H. Hayden, B. Hammond, T. Fletcher, B. Canada, D. Narang, A. Kimber, L. Mitchell, G. Rich and T. Townsend, “Performance parameters for grid-connected PV systems”, in *Conference Record of the Thirty-first IEEE Photovoltaic Specialists Conference, 2005.*, IEEE, 2005, pp. 1601–1606.

[20] *Merriam-Webster.com*, ‘Degrading’, 2013. [Online]. Available: `http://www.merriam-webster.com` (visited on 08/04/2013).

[21] E. Meyer and E. van Dyk, “Assessing the Reliability and Degradation of Photovoltaic Module Performance Parameters”, *IEEE Transactions on Reliability*, vol. 53, no. 1, pp. 83–92, 2004.

[22] J. E. Granata, W. E. Boyson, J. A. Kratochvil and M. A. Quintana, “Long-term performance and reliability assessment of 8 PV arrays at Sandia National Laboratories”, in *2009 34th IEEE Photovoltaic Specialists Conference (PVSC)*, IEEE, Jun. 2009, pp. 001 486–001 491.

[23] D. Jordan and S. Kurtz, “Thin-film reliability trends toward improved stability”, in *37th IEEE Photovoltaic Specialists Conference 2011*, IEEE, Jun. 2011, pp. 827–832.

[24] D. Jordan and S. Kurtz, “Analytical improvements in PV degradation rate determination”, in *2010 35th IEEE Photovoltaic Specialists Conference*, IEEE, Jun. 2010, pp. 2688–2693.

[25] S. Pulver, D. Cormode, A. Cronin, D. Jordan, S. Kurtz and R. Smith, “Measuring degradation rates without irradiance data”, in *2010 35th IEEE Photovoltaic Specialists Conference*, IEEE, Jun. 2010, pp. 1271–1276.

[26] K. A. Emery, “Solar simulators and I-V measurement methods”, *Solar Cells*, vol. 18, no. 1, pp. 251–260, 1986. [Online]. Available: `http://www.sciencedirect.com`.

[27] D. C. Jordan and S. R. Kurtz, “Photovoltaic Degradation Rates-an Analytical Review”, *Progress in Photovoltaics: Research and Applications*, vol. 21, no. 1, pp. 12–29, 2013.

[28] C. Osterwald, J. Adelstein, J. Cueto, B. Kroposki, D. Trudell and T. Moriarty, “Comparison of Degradation Rates of Individual Modules Held at Maximum Power”, in *2006 IEEE 4th World Conference on Photovoltaic Energy Conference*, IEEE, 2006, pp. 2085–2088.

[29] A. Dunea, D. Dunea, V. Moise and M. Olariu, “Forecasting methods used for performance’s simulation and optimization of photovoltaic grids”, in *2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502)*, vol. 4, IEEE, 2001, p. 5.

[30] P. P. Singh and S. Singh, “Realistic generation cost of solar photovoltaic electricity”, *Renewable Energy*, vol. 35, no. 3, pp. 563–569, 2010.

[31] NREL. (2012). Energy Analysis, [Online]. Available: `http://www.nrel.gov/analysis/tech\_lcoe.html` (visited on 09/04/2013).

[32] Shoosmiths LLP. (2011). What's the difference between a guarantee and a warranty?, [Online]. Available: http://www.access-legal.co.uk/free-legal-guides/whats-the-difference-between-a-guarantee-and-a-warranty-1314.htm (visited on 11/04/2013).

[33] Naps Systems Oy. (2009). Naps grid connected modules, [Online]. Available: http://www.napssystems.com (visited on 09/04/2013).

[34] C. R. Osterwald and T. J. McMahon, "History of accelerated and qualification testing of terrestrial photovoltaic modules: A literature review", *Progress in Photovoltaics: Research and Applications*, vol. 17, no. 1, pp. 11–33, 2009.

[35] A. Skoczek, T. Sample and E. D. Dunlop, "The results of performance measurements of field-aged crystalline silicon photovoltaic modules", *Progress in Photovoltaics: Research and Applications*, vol. 17, no. 4, pp. 227–240, 2009.

[36] M. Vázquez and I. Rey-Stolle, "Photovoltaic module reliability model based on field degradation studies", *Progress in Photovoltaics: Research and Applications*, vol. 16, no. 5, pp. 419–433, 2008.

[37] K. Sipilä, R. Pasonen, K. Mäki and R. Alanen, "Arctic solar energy solutions", VTT, 2012, ch. Potential of solar energy in nordic countries, pp. 23–29. [Online]. Available: http://www.vtt.fi.

[38] D. C. Jordan, J. H. Wohlgemuth and S. R. Kurtz, "Technology and Climate Trends in PV Module Degradation Preprint", in *2012 27th European Photovoltaic Solar Energy Conference and Exhibition*, Frankfurt, Germany, Sep. 2012. [Online]. Available: http://www.nrel.gov/docs/fy13osti/56485.pdf.

[39] D. Polverini, M. Field, E. Dunlop and W. Zaaiman, "Polycrystalline silicon PV modules performance and degradation over 20 years", *Progress in Photovoltaics: Research and Applications*, 2012.

[40] L. Palmblad, C. Martinsson, J. Hedström and M. Andersson, "Long term Performance of PV modules-Results from Swedish case studies", Stockholm, 2007, pp. 1–8. [Online]. Available: http://www.aforsk.se.

[41] A. De Lillo, S. Li Causi and S. Castello, “Long-term performance of Casaccia plant”, in *Photovoltaic Energy Conversion, 2003. Proceedings of 3rd World Conference on*, (Osaka Japan), vol. 3, 2003.

[42] D. Verma, M. Tayyib, T. O. Saetre and O.-M. Midtgard, “Outdoor Performance of 10 year Old a-Si and Poly-Si Modules in southern Norway conditions”, in *2012 38th IEEE Photovoltaic Specialists Conference*, IEEE, Jun. 2012, pp. 2368–2371.

[43] F. Jiang and A. Lim, “Performance of heterojunction with intrinsic thin layer (HIT) PV module in Singapore”, in *2010 Conference Proceedings IPEC*, IEEE, Oct. 2010, pp. 95–98.

[44] M. Ross and J. Royer, *Photovoltaics in cold climates*, 1st ed. James & James, 1999, ch. The array, structure and electronics, pp. 39–46.

[45] G. Notton, V. Lazarov and L. Stoyanov, “Optimal sizing of a grid-connected PV system for various PV module technologies and inclinations, inverter efficiency characteristics and locations”, *Renewable Energy*, vol. 35, no. 2, pp. 541–554, Feb. 2010.

[46] B. Burger and R. Rüther, “Inverter sizing of grid-connected photovoltaic systems in the light of local solar resource distribution characteristics and temperature”, *Solar Energy*, vol. 80, no. 1, pp. 32–45, Jan. 2006.

[47] *Technical Description NAPS NP130GK*, Naps Systems Oy, 2010. [Online]. Available: `http://www.napssystems.com` (visited on 14/02/2013).

[48] SMA Solar Technology AG, *Sunny Boy 2500 / 3000 Installation guide*, 2012. [Online]. Available: `http://files.sma.de`.

[49] Fronius, *FRONIUS IG Technical data*, 2011. [Online]. Available: `http://www.fronius.com` (visited on 12/05/2013).

[50] University of Jyväskylä. (2005). Panels on the roof: Saarijärvi PV system, [Online]. Available: `https://www.jyu.fi` (visited on 01/04/2013).

[51] Electrotechnical Sector Committee, *Photovoltaic system performance monitoring — Guidelines for measurement , data exchange and analysis*, British Standards committee, 1998, p. 20.

[52] G. Box, G. Jenkins and G. Reinsel, *Time series analysis: Forecasting and control*, 3rd ed. Chapman and Hall/CRC, 1994, ch. Linear nonstationary models, pp. 89–127.

[53] C. Chatfield, *The analysis of time series: An introduction*, 5th ed. Chapman and Hall/CRC, 1999, ch. Probability models for time series, pp. 27–45.

[54] G Blaesser and D Munro, “Guidelines for the assessment of photovoltaic plants: Document B”, Institute for systems Engineering and Informatics, Tech. Rep. 4, 1995.

[55] European Commission Joint Research Centre. (2012). Photovoltaic geographical information system, [Online]. Available: `http://re.jrc.ec.europa.eu/pvgis/` (visited on 23/03/2013).

# Appendices

## A Matlab code

Matlab was used to analyse data and plot different graphs. The comments at the beginning of each section of code describes the aim or process being executed. The recording formats for both PV systems differ.

```
%% This program has been written by Dereck Mutungi

%It extracts values from the csv files for Agora system into matrices.
    Due
%to the large size of the data, another file Agoratemp.m was written to
%extract the temperature.
%% Start with a clean slate
clc; clear;close all;
%% Open file
% Please make sure that the csv and this script is of the same folder. An
% automatic search script is not included in this m-file
%files= dir();
strDir = 'H:\university notes\Thesis\Agora data\';
fnames = dir(strcat(strDir, '*.csv'));
int numfids;
numfids = 0;
numfids = length(fnames);
% Line 8 of the file contains the format of the csv which will be helpful
% in extracting the data in the file.
%% Format and settings
fmt        = '%s%f%f%f%f%f%f%f%f%f%f%s%s%f%f%f%f%f%f%f%f%f%f%s%s';
cnfg       = {'Delimiter', ';', 'HeaderLines', 8};
% Try also: When 'CollectOutput' set to 1, textscan concatenates
% consecutive output cells with the same data type into a single array.

%% File extraction
% Note that the variable data will be of class cell, we can easily
    extract
% the cell of interest once textscan is done getting the data from the
```

```
    csv
% file
for i = 1: numfids
fyl_csv      = fopen(strcat(strDir, fnames(i).name));
data         = textscan(fyl_csv,fmt,cnfg{:});
fyl_csv      = fclose(fyl_csv);
kusanya{i}   = data;
end
%% Looking for errors in the data
% The status of the system is in column M and N
count=0;
w = 'Fehler';
f = 'Stoerung';
fileID = fopen('error.txt','w');
fprintf(fileID,'%s','');
fclose(fileID);
for filenumber = 1:i
    for counter = 1: length(kusanya{1,filenumber}{1,13})
        if strcmp(kusanya{1,filenumber}{1,13}{counter,1},w) ...
                || strcmp(kusanya{1,filenumber}{1,13}{counter,1},f)
            count = count+1;
            kusanya{1,filenumber}{1,2}(counter,1)= NaN;
            fileID = fopen('error.txt','a');
            fprintf(fileID,'%s\n',kusanya{1,filenumber}{1,1}{counter,1});
            fclose(fileID);
        end
    end
end

clear counter filenumber w strDir fileID f
%%  Required data
% Extract data
% Remove very last row for consistency in elements (see orig csv data)
%                                   B O P Q R T W
for k = 1:i
val          = cell2mat(kusanya{1,k}( [2 15 16 17 18 20 23] ));
% Since this is a time series data, it's always nice to keep them neatly
```

```
    in
% a structure
if(size(val,1)<size(val,2))
    val          = val(1,:);
    Agora.val {k}  = val;
else
  val          = val(1:length(val)-1,:);
   Agora.val {k}  = val;
end
time         = datenum(kusanya{1,k}{1},'dd.mm.yyyy HH:MM:SS');

if(size(time,1)== size(time,2))
    time          = time(1,:);
else
    time          = time(1:length(time)-1,:);

end
Agora.time {k} = time;

end

% Use datestr(time) to see if time has been successfully read
times    = datestr(time(1,:))

%For cleaner workspace remove all unnecessary used variables
clear cnfg fmt fyl_csv  times numfids fnames data time val

Agora.time1=Agora.time{1};
Agora.val1   = Agora.val{1};
for j  = 1:k-1
    Agora.time1 = vertcat(Agora.time1,Agora.time{j+1});
    Agora.val1 = vertcat(Agora.val1,Agora.val{j+1});
end

% For cleaner workspace remove all unnecessary used variables
clear i j

```

```
Agora.val1(isnan(Agora.val1))=0;
save FulldataAgora.mat Agora k
```

code/Matcode/AgoraData.m

```
%This code has been written by Dereck Mutungi 26/4/2013
%%
%Due to the large size of data, the data was split into two and will be
%later merged.
%% Start with a clean slate
clc; clear;close all;
%% Open file
%This is a continuation of data extraction. The ambient and cell
%temperature of the PV system
strDir = 'H:\university notes\Thesis\Agora data\';
fnames = dir(strcat(strDir, '*.csv'));
int numfids;
numfids = 0;
numfids = length(fnames);

%% Format and settings
fmt          = '%s%f%f%f%f%f%f%f%f%f%f%f%f%s%s%f%f%f%f%f%f%f%f%f%f%f%f%f%s%s';
cnfg         = {'Delimiter', ';', 'HeaderLines', 8};


%% File extraction
% Note that the variable data will be of class cell, we can easily
    extract
% the cell of interest once textscan is done getting the data from the
    csv
% file
for i = 1: numfids
fyl_csv      = fopen(strcat(strDir, fnames(i).name));
data         = textscan(fyl_csv,fmt,cnfg{:});
fyl_csv      = fclose(fyl_csv);
kusanya{i}   = data;
end
%% Looking for errors in the data
```

```
% The status of the system is in column M and N
count=0;
w = 'Fehler';
f = 'Stoerung';
fileID = fopen('error.txt','w');
fprintf(fileID,'%s','');
fclose(fileID);
for filenumber = 1:i
    for counter = 1: length(kusanya{1,filenumber}{1,13})
        if strcmp(kusanya{1,filenumber}{1,13}{counter,1},w) ...
                ||strcmp(kusanya{1,filenumber}{1,13}{counter,1},f)
            count = count+1;
            kusanya{1,filenumber}{1,2}(counter,1)= NaN;
            fileID = fopen('error.txt','a');
            fprintf(fileID,'%s\n',kusanya{1,filenumber}{1,1}{counter,1});
            fclose(fileID);
        end
    end
end

clear counter filenumber w strDir fileID f
%% Required data
% The columns in the CSV fiel with the required temperatures. Cell(C) and
% Ambient (D)
%                                    C D
for k = 1:i
val          = cell2mat(kusanya{1,k}( [3 4] ));
if(size(val,1)<size(val,2))
    val          = val(1,:);
    Agora1.val {k}  = val;
    Agora1.val{k}(isnan(Agora1.val{k}))=0;
else
  val          = val(1:length(val),:);
   Agora1.val {k}  = val;
    Agora1.val{k}(isnan(Agora1.val{k}))=0;
end
time         = datenum(kusanya{1,k}{1},'dd.mm.yyyy HH:MM:SS');
```

```
if(size(time,1)== size(time,2))
    time        = time(1,:);
else
    time        = time(1:length(time),:);

end
Agora1.time{k} = time;

end



% Use datestr(time) to see if time has been successfully read
times   = datestr(time(1,:))

% For cleaner workspace remove all unnecessary used variables
clear cnfg fmt fyl_csv  times numfids fnames data time val



% For cleaner workspace remove all unnecessary used variables
clear i j


save Agoratemp.mat Agora1 k
```

code/Matcode/Agoratemp.m

```
%% This program has been written by Dereck Mutungi.

%It extracts values from the csv files for Saarijarvi system into
    matrices.

%%  Start with a clean slate
clc; clear;close all;
%%  Open file
```

```
% Please make sure that the csv and this script is of the same folder. An
% automatic search script is not included in this m-file
strDir = 'H:\university notes\Thesis\SjarviData\';
fnames = dir(strcat(strDir, '*.csv'));
int numfids;
numfids = 0;
numfids = length(fnames);
% Line 1 of the file contains the format of the csv which will be helpful
% in extracting the data in the file. Let's just collect the timestamp
% as a string.
%% Format and settings
fmt          = '%s%f%f%f%f%f%f%f%f%f%f';
cnfg         = {'Delimiter', ',', 'HeaderLines', 1};

%% File extraction
% Using textscan to get data from the csv files
for i = 1: numfids
fyl_csv      = fopen(strcat(strDir, fnames(i).name));
data         = textscan(fyl_csv,fmt,cnfg{:});
fyl_csv      = fclose(fyl_csv);
kusanya{i}   = data;
end

%% Required data
% Remove very last row for consistency in elements
%
for k = 1:i
val          = cell2mat(kusanya{1,k}( [2 3 4 5 6 7 8 9 10 11] ));
val          = val(1:length(val)-1,:);
Sjarvi.val {k}  = val;
Sjarvi.val{k}(isnan(Sjarvi.val{k}))=0;
time         = datenum(kusanya{1,k}{1},'dd/mm/yyyy HH:MM');
time         = time(1:length(time)-1,:);
Sjarvi.time {k} = time;

end
% Use datestr(time) to see if time has been successfully read
```

```
times     = datestr(time(1,:))


% For cleaner workspace remove all unnecessary used variables
clear cnfg data fmt fyl_csv time times val fnames i kusanya numfids
    strDir

save SjarviData.mat Sjarvi k
```

code/Matcode/SjarviData.m

The data was filtered, monitoring range set and daily average values were used to plot time series graphs to observe the behaviour of performance indices over time. Not all the plots derived were shown in the main body. Again, the programming has been intentionally separated for each system for easier tracking and the data sizes were still too large to combine.

```
%This code has been written by Dereck Mutungi

% This program is used to get full calender years... 1st January 2010 to
% 31st December 2012. Preliminary average values are calculated.
%%
clear all; close all; clc;
load FulldataAgora.mat
load Agoratemp.mat
%% Selecting data in that period using loops.
p=0;
for j=1:k
    Agora1.val{j}(isnan(Agora1.val{j}))=0;
    Agora.val{j}(isnan(Agora.val{j}))=0; % To ensure no NaN values are left
    % Condition to select data between 1/1/2010 and 31/12/2012
    if ((Agora1.time{j}(1,1)>=734139) && (Agora1.time{j}(1,1)<735235))
    p=p+1;
    Agora1.day{p}(1)= Agora1.time{j}(1,1);
    Agora.day{p}(1)= Agora.time{j}(1,1);
    Agora.data{p}=Agora.val{j};
    Agora.time2{p}=Agora.time{j};
    Agora1.data{p}=Agora1.val{j};
        else
        continue
    end
end
%This condition is to check if the file has got more than two record values
%Recording only occurs when there is activity from the array system.
for j2=1:p
    l=1;
    if (length(Agora1.data{j2})~=2)
        Agora.daily_irradiance{j2}=sum(Agora.data{j2}(:,1));
```

```
        Agora1.daily_celltemp{j2}=sum(Agora1.data{j2}(:,1));
        Agora1.daily_ambtemp{j2}=sum(Agora1.data{j2}(:,2));
        Agora.Arraypower{j2}=sum(Agora.data{j2}(:,2)...
                                  .*(Agora.data{j2}(:,7)./1000)); %W

    else
        Agora1.daily_celltemp{j2}(1) = (Agora1.data{j2}(1,1));
        Agora1.daily_ambtemp{j2}(1) = (Agora1.data{j2}(1,2));
        Agora.daily_irradiance{j2}(1) = (Agora.data{j2}(1,1));
        Agora.Arraypower{j2}(1)=Agora.data{j2}(1,2)...
                                  .*(Agora.data{j2}(1,7)./1000);
    end

end
%converting the cells to matrices
Agora1.daily_Sumtemp = cell2mat(Agora1.daily_celltemp);
Agora1.daily_SumAmbtemp = cell2mat(Agora1.daily_ambtemp);
Agora.daily_Sumirradiance = cell2mat(Agora.daily_irradiance);
Agora.daily_SumArraypower = cell2mat(Agora.Arraypower);
Agora.day=cell2mat(Agora.day);
Agora1.day=cell2mat(Agora1.day);
for t=1:p
    % To get the average temperature that was recorded.
    % This is NOT the average over the period of 24 hours but the average
    % over the recording period.
Agora1.daily_avgcelltemp(t)= Agora1.daily_Sumtemp(t)/...
    ((length(Agora1.data{t})-1));
Agora1.daily_avgAmbtemp(t)= Agora1.daily_SumAmbtemp(t)/...
    ((length(Agora1.data{t})-1));
Agora.daily_avgIrradiance(t)= Agora.daily_Sumirradiance(t)...
    /((length(Agora.data{t})-1));
Agora.daily_avgArraypower(t)= Agora.daily_SumArraypower(t)...
    /((length(Agora.data{t})-1));
end

Agora1.daily_avgcelltemp(isnan(Agora1.daily_avgcelltemp))=0;
Agora1.daily_avgAmbtemp(isnan(Agora1.daily_avgAmbtemp))=0;
```

```
Agora.daily_avgIrradiance(isnan(Agora.daily_avgIrradiance))=0;
Agora.daily_avgArraypower(isnan(Agora.daily_avgArraypower))=0;
Agora.daily_avgCelltemp = Agora1.daily_avgcelltemp;
Agora.daily_avgAmbtemp = Agora1.daily_avgAmbtemp;
clear j2 k Agora1
save Agoracycle.mat Agora p
```

code/Matcode/Range_selection.m

```
%This Matlab code has been written by Dereck Mutungi
% This program is used to get  full calender years... 1st January 2006 to
% 31st December 2012. There are also daily average values calculated
    later.
%%
clear all; close all; clc;
load SjarviData.mat
%% Selecting data in that period
p=0;
for j=1:k
    Sjarvi.val{j}(isnan(Sjarvi.val{j}))=0;
    Sjarvi.val{j}(isnan(Sjarvi.val{j}))=0; % To remove NaN numbers
    p=p+1;
    for x=1:length(Sjarvi.val{j})
    % Condition to select data between 1/1/2006 and 31/12/2012
    if ((Sjarvi.time{j}(x,1)>=732678) && (Sjarvi.time{j}(x,1)<735235))

    Sjarvi.year{p}= Sjarvi.time{j};
    Sjarvi.data{p}=Sjarvi.val{j};
        else
        continue
    end
    end
end
% To get rid of any empty cells with no values
Sjarvi.year=Sjarvi.year(~cellfun('isempty',Sjarvi.year));
Sjarvi.data=Sjarvi.data(~cellfun('isempty',Sjarvi.data));
q=1;
% Organising the data into days
```

```
for  j=1:length(Sjarvi.year)
      l=1;  t=1;
     [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Sjarvi.year{j}(1,1));
     Sjarvi.daily_SumAmbtemp{j}(1) = (Sjarvi.data{j}(1,8)); % Amb temperature
     Sjarvi.daily_SumCelltemp{j}(1) = (Sjarvi.data{j}(1,7)); % Cell temperature
     for x = 2 : length(Sjarvi.data{j})
     %Converting the date into a vector
    [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Sjarvi.year{j}(x,1));
       if (D(x-1)== D(x)) && (M(x-1) == M(x))
        Sjarvi.daily_SumAmbtemp{j}(l) = Sjarvi.daily_SumAmbtemp{j}(l)+...
            (Sjarvi.data{j}(x,8));
        Sjarvi.daily_SumCelltemp{j}(l) = Sjarvi.daily_SumCelltemp{j}(l)+...
            (Sjarvi.data{j}(x,7));
        Sjarvi.day(q)=Sjarvi.year{j}(x,1);
        t=t+1;
       else
           Sjarvi.daily_avgAmbtemp{j}(l)=Sjarvi.daily_SumAmbtemp{j}(l)/t;
           Sjarvi.daily_avgCelltemp{j}(l)=Sjarvi.daily_SumCelltemp{j}(l)/t;
           Sjarvi.dailyCelltemp(q)=Sjarvi.daily_avgCelltemp{j}(l);
           Sjarvi.dailyAmbtemp(q) = Sjarvi.daily_avgAmbtemp{j}(l);
          %initialising the next day energy.
        l=l+1;
        q=q+1;
        Sjarvi.daily_SumAmbtemp{j}(l) = (Sjarvi.data{j}(x,8));
        Sjarvi.daily_SumCelltemp{j}(l) = (Sjarvi.data{j}(x,7));
        t=1;
        continue;
       end
     end
Sjarvi.daily_avgAmbtemp{j}(l)=Sjarvi.daily_SumAmbtemp{j}(l)/t;
Sjarvi.daily_avgCelltemp{j}(l)=Sjarvi.daily_SumCelltemp{j}(l)/t;
Sjarvi.dailyCelltemp(q)=Sjarvi.daily_avgCelltemp{j}(l);
Sjarvi.dailyAmbtemp(q) = Sjarvi.daily_avgAmbtemp{j}(l);
```

```
Sjarvi.day(q)=Sjarvi.year{j}(x,1);

end
%% calculating the daily average irradiance in-plane of the array
SumIrradiance{1}(1)=0;
avgIrradiance(1)=0;
q=1;
for j=1:length(Sjarvi.year)
     l=1; t=1;
    [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Sjarvi.year{j}(1,1));
    SumIrradiance{j}(1) = (Sjarvi.data{j}(1,9)); % Amb temperature
   for x = 2 : length(Sjarvi.data{j})
    %Converting the date into a vector
   [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Sjarvi.year{j}(x,1));
       if (D(x-1)== D(x)) && (M(x-1) == M(x))
           if (Sjarvi.data{j}(x,9)>0)
             SumIrradiance{j}(1) = SumIrradiance{j}(1)+(Sjarvi.data{j}(x
   ,9));
             t=t+1;
           end
       else
           Sjarvi.avgIrradiance(q)=SumIrradiance{j}(1)/t;
          %initialising the next day energy.
         l=l+1;
         q=q+1;
         SumIrradiance{j}(1) = (Sjarvi.data{j}(x,9));
         t=1;
         continue;
       end
    end
Sjarvi.avgIrradiance(q)=SumIrradiance{j}(1);

end
% To free up more memory, get rid of the unnecessary variables
clear D H M MN S Y j k l p q t x
save Sjarvicycle.mat Sjarvi
```

code/Matcode/Range_selectionSjarvi.m

Data filtering was based on the outcome of the cut-in irradiance. It was noted that ss1 recorded power to the grid before the array generated any current. These values were filtered out.

```
% Written by Dereck Mutungi.
% This code is used to generate graphs of normalised array power against
% irradiance.The graphs are monthly for the year 2011: Agora
%%
clear all; close all; clc;
load Agoracycle.mat
%%
% Initialising the variables
Arraypower{1}{1}=0;
irradiance{1}{1}=0;

%Combination of loops and conditions to extra array power, irradiance and
%time
for i = 1:p
    [Y(i) M(i) D(i) H(i) MN(i) S(i)] =  datevec(Agora.time2{i}(1,1));
    if (Y(i)==2011)
        for month = 1:12
            if (M(i)==month)
                for day=1:eomday(2011,M(i))
                    if (D(i)==day)
                        Arraypower{month}{day}=Agora.data{i}(:,2) ...
                            .*(Agora.data{i}(:,7)./1000);
                      irradiance{month}{day}= Agora.data{i}(:,1);

                        else
                            continue
                    end
                end
            else  continue
            end
        end
    else continue
    end
end
```

```
%%
%combination of loops and conditions to filter out zero power so as assign
%a new matrix with non-zero power values
for m=1:12
    for d=1:length(Arraypower{m})
        x=0;
        for l=1:length(Arraypower{m}{d})

            if (Arraypower{m}{d}(l)>0)
                x=x+1;
                Arraypower1{m}{d}(x)=Arraypower{m}{d}(l);
                irradiance1{m}{d}(x)=irradiance{m}{d}(l);
            else
                continue
            end
        end
    end
end
%%
% Use of loops to draw the graphs and visually inspect the cut-in
% irradiance. (later found to be 10W/m^2)
for m=1:12
    for d=1:length(Arraypower1{m})
      figure(m)
  title({['Normalised array power as a function of irradiance: month ' num2str(m) ];...
      'Agora'})
  plot(irradiance1{m}{d}, Arraypower1{m}{d}./2600,'.')
  ylabel('Normalised Array power');
  xlabel('Irradiance , (W/m^2)');
  ylim([0,1]); %The narrow band shows the cut in irradiance (0.06, 100)
  xlim([0,1200]);
  hold on
  grid on

    end
```

```
end

% Graphs will be used to check for cut-in irradiance and any anomaly
    during
% operation. This is done by checking for points that are below a
    clustered
% linear gradient.
```

code/Matcode/ArrayPowerVsIrradiance.m

```
% Written by Dereck Mutungi.
% This code is used to generate graphs of normalised array power against
% irradiance.The graphs are monthly for the year 2009: Saarijarvi
%%
clear all; close all; clc;
load Sjarvicycle.mat
%%
% Initialising the variables
Arraypower1{1}{1}(1)=Sjarvi.data{1}(1,3).*(Sjarvi.data{1}(1,4));
irradiance{1}{1}(1)=Sjarvi.data{1}(1,9);
Arraypower2{1}{1}(1)=Sjarvi.data{1}(1,5).*(Sjarvi.data{1}(1,6));
%Combination of loops and conditions to get array power and irradiance
    into
%structures.
[Y(1) M(1) D(1) H(1) MN(1) S(1)] =  datevec(Sjarvi.year{1}(1,1));
day=1; w=1; month=1;
for i = 1:length(Sjarvi.year)
    for x=2:length(Sjarvi.year{i})
     [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Sjarvi.year{i}(x,1));
     if (Y(x)==2012)
          if (M(x)==M(x-1))
               if (D(x-1)==D(x))
                    w=w+1;
                     Arraypower1{month}{day}(w)=Sjarvi.data{i}(x,3)
    ...
                         .*(Sjarvi.data{i}(x,4));
                     Arraypower2{month}{day}(w)=Sjarvi.data{i}(x,5)
    ...
```

```
                                    .*(Sjarvi.data{i}(x,6));
                                irradiance{month}{day}(w)= Sjarvi.data{i}(x,9);

                        else

                            day=day+1;
                            w=1;
                            Arraypower1{month}{day}(w)=Sjarvi.data{i}(x,3)...
                                    .*(Sjarvi.data{i}(x,4));
                                Arraypower2{month}{day}(w)=Sjarvi.data{i}(x,5)
    ...
                                    .*(Sjarvi.data{i}(x,6));
                                irradiance{month}{day}(w)= Sjarvi.data{i}(x,9);
                                    continue
                        end

                else
                    month=month+1;
                    day=1;
                    w=1;
                    Arraypower1{month}{day}(w)=Sjarvi.data{i}(x,3)...
                                    .*(Sjarvi.data{i}(x,4));
                    Arraypower2{month}{day}(w)=Sjarvi.data{i}(x,5)...
                                    .*(Sjarvi.data{i}(x,6));
                    irradiance{month}{day}(w)= Sjarvi.data{i}(x,9);
                    continue
                end
            else continue
            end
        end
end
%%
%combination of loops and conditions to filter out zero power values.
for m=1:12
    for d=1:length(Arraypower1{m})
        x=0;t=0;
        for l=1:length(Arraypower1{m}{d})
```

```

                if (Arraypower1{m}{d}(1)>0)
                    x=x+1;
                    Arraypower1a{m}{d}(x)=Arraypower1{m}{d}(1);
                    irradiance1{m}{d}(x)=irradiance{m}{d}(1);
                end
                if (Arraypower2{m}{d}(1)>0)
                    t=t+1;
                    Arraypower2a{m}{d}(t)=Arraypower2{m}{d}(1);
                    irradiance2{m}{d}(t)=irradiance{m}{d}(1);
                end

            end
        end
end
%%
% Use of loops to draw the graphs and visually inspect the cut-in
% irradiance.
for m=1:12
      for d=1:length(Arraypower1a{m})
        figure(m)
        subplot(2,1,1);
   title({'Normalised array power as a function of irradiance';...
        ['Sjarvi sub-system 1: month ' num2str(m)]})
   plot(irradiance1{m}{d}, Arraypower1a{m}{d}./5130,'.')
   ylabel('Normalised Array power');
   xlabel('Irradiance, (W/m^2)');
   ylim([0,1]); %The narrow band shows the cut in irradiance (0.06,50)
   xlim([0,1200]);
   hold on
   grid on
      end
        for d=1:length(Arraypower2a{m})
   figure(m)
   subplot(2,1,2);
   plot(irradiance2{m}{d}, Arraypower2a{m}{d}./1140,'.')
   title({'Normalised array power as a function of irradiance';...
```

```
        ['Sjarvi sub-system 2:month ' num2str(m)]})
    ylabel('Normalised Array power');
    xlabel('Irradiance , (W/m^2)');
    ylim([0,1]); %The narrow band shows the cut in irradiance
    xlim([0,1200]);
    hold on
    grid on

        end
end

% (later found to be 10W/m^2)
```

code/Matcode/ArrayPowerVsIrradianceSjarvi.m

```
%Written by Dereck  Mutungi

% This code will be used to filter out data that has irradiance below cut
% in (10W/m^2) and array current is 0
%%
clear all; close all; clc;
% loading the full cycle matrix. See Range_selection.m
load Agoracycle.mat
%Combination of loops and conditions to produce the filtered data by
%irradiance
%%
for i = 1:p
    x=0;
    for l=1:length(Agora.data{i}(:,1))
        if (length(Agora.data{i}(:,1))==1)
            Agora.Timefiltered{i}=Agora.time2{i};
            Agora.datafiltered{i}=Agora.data{i};
        elseif ((Agora.data{i}(l,1)>10)||(Agora.data{i}(l,7)>0))
            x=x+1;
                Agora.datafiltered{i}(x,:)=Agora.data{i}(l,:);
                Agora.Timefiltered{i}(x,:)=Agora.time2{i}(l,:);
        else
            continue
```

```
            end
        end
end
%%
% After filtering some days had insufficient irradiance hence no current
    from
% the array, but the day would be useful for further analysis.
for i=1:p
    if (size(Agora.datafiltered{i})==0)
        Agora.datafiltered{i}=[0,0,0,0,0,0,0];
        Agora.Timefiltered{i}=Agora.time2{i}(1,1);
    end
end
save AgoraFiltered.mat Agora p
```

code/Matcode/Filtered_by_G_I.m

```
%Written by Dereck Mutungi

% This code will be used to filter out data that has irradiance below cut
% in (10W/m^2) for sub system 2 and above zero for sub-system 1
%%
clear all; close all; clc;
% loading the full cycle matrix. See Range_selection.m
load Sjarvicycle.mat
%%
%Combination of loops and conditions to produce the filtered data by
%irradiance
for i = 1:length(Sjarvi.year)
    x=0;t=0; w=0;
    for l=1:length(Sjarvi.data{i})
        % Filtered by irradiance
        if (Sjarvi.data{i}(l,9)~=0)
            x=x+1;
                Sjarvi.datafiltered1{i}(x,:)=Sjarvi.data{i}(l,:);
                Sjarvi.Yearfiltered1{i}(x,:)=Sjarvi.year{i}(l,:);
        end
        %Filtered by 10W/m^2
```

```
            if (Sjarvi.data{i}(1,9)>10)
                t=t+1;
                    Sjarvi.datafiltered2{i}(t,:)=Sjarvi.data{i}(1,:);
                    Sjarvi.Yearfiltered2{i}(t,:)=Sjarvi.year{i}(1,:);
            end
            %Filtered by current value and power
            if ((Sjarvi.data{i}(1,4)>0)&&(Sjarvi.data{i}(1,1)>0))
                w=w+1;
                    Sjarvi.datafiltered3{i}(w,:)=Sjarvi.data{i}(1,:);
                    Sjarvi.Yearfiltered3{i}(w,:)=Sjarvi.year{i}(1,:);
            end
        end
end

save SjarviFiltered.mat Sjarvi
```

code/Matcode/Filtered_by_G_I_Sjarvi.m

Calculations for performance indices and a time series plot of the same was done using the following code.

```
%This code has been written by Dereck Mutungi
% The program calculates grid & array energy, yields, PR and efficiencies.
%%
clc; clear;close all;
%% Load the Agora matrix. See Filtered_by_G_I.m
% This matrix also contains data filtered by irradiance and current. Plus
% original data.
load AgoraFiltered.mat
% Selecting the date and adding the power and getting the energy per day
% in Wh
%%
for j=1:length(Agora.Timefiltered)
    l=1;
    [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Agora.Timefiltered{j}(1,1));
    Agora.datafiltered{j}(isnan(Agora.datafiltered{j}))=0;
    Agora.daily_Arrayenergy{j}(1) = (Agora.datafiltered{j}(1,2)...
        .*Agora.datafiltered{j}(1,7).*(0.25/1000)); % energy in Wh... the current in mA
    Agora.daily_Gridenergy{j}(1) = (Agora.datafiltered{j}(1,6)* 0.25); % energy in Whr


for x = 2 : length(Agora.datafiltered{j})
  % checks for cell size since some days have no data thus size mismatch
    if ((length(Agora.Timefiltered{j}))==(length(Agora.datafiltered{j})))
     %Converting the date into a vector
        [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Agora.Timefiltered{j}(x,1));
            if (D(x-1)== D(x)) && (M(x-1) == M(x))
                if ((MN(x)-MN(x-1))== 15)||((MN(x)-MN(x-1))== -45)
                    Agora.daily_Arrayenergy{j}(l) = ...
                        Agora.daily_Arrayenergy{j}(l)+...
```

```
                    (Agora.datafiltered{j}(x,2).*Agora.datafiltered{j}(x,7).*(0.25/1000));
                 Agora.daily_Gridenergy{j}(1) =...
                     Agora.daily_Gridenergy{j}(1)+(Agora.datafiltered{j}(x,6).*0.25);
            elseif ((MN(x)-MN(x-1))== 1)||((MN(x)-MN(x-1))== -59)
                Agora.daily_Arrayenergy{j}(1) =...
                    Agora.daily_Arrayenergy{j}(1)+...
           (Agora.datafiltered{j}(x,2).*Agora.datafiltered{j}(x,7).*(1/60000));
                Agora.daily_Gridenergy{j}(1) = Agora.daily_Gridenergy{j}(1)+...
                (Agora.datafiltered{j}(x,6).*(1/60));
            else
                Agora.daily_Arrayenergy{j}(1) = Agora.daily_Arrayenergy {j}(1)+ 0;
                Agora.daily_Gridenergy{j}(1) = Agora.daily_Gridenergy {j}(1)+ 0;
            end
        end
    else
       Agora.daily_Arrayenergy {j}(1)= 0;
       Agora.daily_Gridenergy {j}(1)= 0;
       continue;
     end
end

end
clear D H M MN S Y  j l x

for j=1:length(Agora.time2)
     l=1;
     [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Agora.time2{j}(1,1));
     Agora.daily_irradiation{j}(1) = (Agora.data{j}(1,1)* 0.25);% energy in Whr/sq m

for x = 2 : length(Agora.data{j})
```

```
  % checks for cell size since some days have no data thus size mismatch
    if ((length(Agora.data{j}))==(length(Agora.time2{j})))
     %Converting the date into a vector
       [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Agora.time2{j}(x,1));
           if (D(x-1)== D(x)) && (M(x-1) == M(x))
               if ((MN(x)-MN(x-1))== 15)||((MN(x)-MN(x-1))== -45)
                    Agora.daily_irradiation{j}(1) = Agora.daily_irradiation{j
    }(1)+...
              (Agora.data{j}(x,1).*0.25);
                elseif ((MN(x)-MN(x-1))== 1)||((MN(x)-MN(x-1))== -59)
                   Agora.daily_irradiation{j}(1) = Agora.daily_irradiation{j
    }(1)+...
              (Agora.data{j}(x,1).*(1/60));
               else
                    Agora.daily_irradiation{j}(1) = Agora.daily_irradiation {
    j}(1)+ 0;
               end
           end
    else
       Agora.daily_irradiation {j}(1)= 0;
       continue;
     end
end

end
clear D H M MN S Y  j l x
%%
%Daily energy in kWhr
dailyArrayEnergy = cell2mat(Agora.daily_Arrayenergy);
Agora.daily_Arrayenergy= dailyArrayEnergy/1000;
dailyGridEnergy = cell2mat(Agora.daily_Gridenergy);
Agora.daily_Gridenergy = dailyGridEnergy/1000;
dailyirradiation = cell2mat(Agora.daily_irradiation);
%The area of the array is 19.83 sq m
Agora.daily_Solarenergy = (dailyirradiation/1000).*19.83; %Kwh
clear dailyArrayEnergy dailyGridEnergy dailyirrEnergy
```

```
%% Yields
% Reference yield
Agora.dailyRefyield = Agora.daily_Solarenergy./(1*19.83); %The units are h/d
% Array yield
Agora.dailyArrayyield = Agora.daily_Arrayenergy./2.6; %Where 2.6kW is the rated power
% Final yield
Agora.dailyFinalyield = Agora.daily_Gridenergy./2.6;
%% Performance Ratio and efficiencies
% Performance ratio
Agora.PR = Agora.dailyFinalyield ./Agora.dailyRefyield;
%Inverter efficiency
Agora.InvEff = Agora.daily_Gridenergy./Agora.daily_Arrayenergy;
%Array efficiency.
Agora.ArrayEff = Agora.daily_Arrayenergy./(Agora.daily_Solarenergy);
%System efficiency
Agora.SysEff = Agora.daily_Gridenergy./(Agora.daily_Solarenergy);
%Capture loss
Agora.LC = Agora.dailyRefyield- Agora.dailyArrayyield;
%System loss
Agora.LS = Agora.dailyRefyield - Agora.dailyFinalyield;

%
Agora.PR(isnan(Agora.PR))=0;
Agora.InvEff(isnan(Agora.InvEff))=0;
Agora.SysEff(isnan(Agora.SysEff))=0;
Agora.ArrayEff(isnan(Agora.ArrayEff))=0;

%% Saving the variables in .mat file
Agora3year.PR=Agora.PR;
Agora3year.InvEff=Agora.InvEff;
Agora3year.SysEff=Agora.SysEff;
Agora3year.ArrayEff=Agora.ArrayEff;
Agora3year.daily_avgCelltemp=Agora.daily_avgCelltemp;
Agora3year.daily_avgAmbtemp=Agora.daily_avgAmbtemp;
Agora3year.day=Agora.day;
```

```
Agora3year.daily_avgIrradiance=Agora.daily_avgIrradiance;
Agora3year.dailyRefyield=Agora.dailyRefyield;
Agora3year.dailyFinalyield=Agora.dailyFinalyield;
Agora3year.dailyArrayyield=Agora.dailyArrayyield;
Agora3year.LS=Agora.LS;
Agora3year.LC=Agora.LC;
Agora3year.data=Agora.data;
Agora3year.time2 = Agora.time2;
Agora3year.daily_Gridenergy=Agora.daily_Gridenergy;
Agora3year.daily_Arrayenergy=Agora.daily_Arrayenergy;
Agora3year.daily_Solarenergy=Agora.daily_Solarenergy;
save Agora3year.mat Agora3year p
```

code/Matcode/AgoraPerformanceIndices.m

```
%Written by Dereck Mutungi
% This program is used to calculate the grid and array energies per day of
% subsystem 1 and 2 of the Sjarvi system. Yields, PR and efficiencies are
% also calculated.
%%
clear all; close all; clc;
% Load the complete Year cycle matrices. See Filtered_by_G_I.m
load SjarviFiltered.mat
% The datafiltered1 is data with zero irradiance eliminated.
% datafiltered1 has a minimum cut-in irradiance of 10W/m2
% The energy to the grid per day is calculated as the sum of the product
% of power and recording interval over the entire day. For the array
% energy, the summation of voltage, current and recording period
% area of the array is 37.95 sq m
%%
q=1;
for j=1:length(Sjarvi.Yearfiltered1)
      l=1; t=1;
     [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Sjarvi.Yearfiltered1{j}(1,1));
     Sjarvi.daily_Gridenergy1{j}(1) = (Sjarvi.datafiltered1{j}(1,1)* 0.25);
```

```
Sjarvi.daily_Gridenergy2{j}(1) = (Sjarvi.datafiltered1{j}(1,2)* 0.25)
;
Sjarvi.daily_Arrayenergy1{j}(1)= (Sjarvi.datafiltered1{j}(1,3)...
    *Sjarvi.datafiltered1{j}(1,4)* 0.25);
Sjarvi.daily_Arrayenergy2{j}(1)= (Sjarvi.datafiltered1{j}(1,5)...
    *Sjarvi.datafiltered1{j}(1,6)* 0.25);
Sjarvi.daily_Solarenergy{j}(1)= (Sjarvi.datafiltered1{j}(1,9)...
    *37.95* 0.25);
Sjarvi.dayfiltered1{j}(1)=Sjarvi.Yearfiltered1{j}(1);
for x = 2 : length(Sjarvi.datafiltered1{j})
%Converting the date into a vector
[Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Sjarvi.Yearfiltered1{j}(x
,1));
   if (D(x-1)== D(x)) && (M(x-1) == M(x))
       if ((MN(x)-MN(x-1))== 15)||((MN(x)-MN(x-1))== -45)
     Sjarvi.daily_Gridenergy1{j}(1) = Sjarvi.daily_Gridenergy1{j}(1)
+...
         (Sjarvi.datafiltered1{j}(x,1)* 0.25);
     Sjarvi.daily_Gridenergy2{j}(1) = Sjarvi.daily_Gridenergy2{j}(1)
+...
      (Sjarvi.datafiltered1{j}(x,2)* 0.25);
     Sjarvi.daily_Arrayenergy1{j}(1)= Sjarvi.daily_Arrayenergy1{j}(1)
+...
      (Sjarvi.datafiltered1{j}(x,3)*Sjarvi.datafiltered1{j}(x,4)* 0.25)
;
     Sjarvi.daily_Arrayenergy2{j}(1)= Sjarvi.daily_Arrayenergy2{j}(1)
+...
      (Sjarvi.datafiltered1{j}(x,5)*Sjarvi.datafiltered1{j}(x,6)* 0.25)
;
     Sjarvi.daily_Solarenergy{j}(1)= Sjarvi.daily_Solarenergy{j}(1)+...
      (Sjarvi.datafiltered1{j}(x,9)*37.95* 0.25);
Sjarvi.dayfiltered1{j}(1)=Sjarvi.Yearfiltered1{j}(x);
  elseif ((MN(x)-MN(x-1))== 30)||((MN(x)-MN(x-1))== -30)
     Sjarvi.daily_Gridenergy1{j}(1) = Sjarvi.daily_Gridenergy1{j}(1)
+...
         (Sjarvi.datafiltered1{j}(x,1)* 0.5);
```

```
        Sjarvi.daily_Gridenergy2{j}(1) = Sjarvi.daily_Gridenergy2{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,2)* 0.5);
        Sjarvi.daily_Arrayenergy1{j}(1)= Sjarvi.daily_Arrayenergy1{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,3)*Sjarvi.datafiltered1{j}(x,4)* 0.5);
        Sjarvi.daily_Arrayenergy2{j}(1)= Sjarvi.daily_Arrayenergy2{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,5)*Sjarvi.datafiltered1{j}(x,6)* 0.5);
         Sjarvi.daily_Solarenergy{j}(1)= Sjarvi.daily_Solarenergy{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,9)*37.95* 0.5);
     Sjarvi.dayfiltered1{j}(1)=Sjarvi.Yearfiltered1{j}(x);
        elseif ((MN(x)-MN(x-1))== 10)||((MN(x)-MN(x-1))== -50)
        Sjarvi.daily_Gridenergy1{j}(1) = Sjarvi.daily_Gridenergy1{j}(1)
    +...
            (Sjarvi.datafiltered1{j}(x,1)* (1/6));
        Sjarvi.daily_Gridenergy2{j}(1) = Sjarvi.daily_Gridenergy2{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,2)* (1/6));
        Sjarvi.daily_Arrayenergy1{j}(1)= Sjarvi.daily_Arrayenergy1{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,3)*Sjarvi.datafiltered1{j}(x,4)* (1/6)
    );
        Sjarvi.daily_Arrayenergy2{j}(1)= Sjarvi.daily_Arrayenergy2{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,5)*Sjarvi.datafiltered1{j}(x,6)* (1/6)
    );
         Sjarvi.daily_Solarenergy{j}(1)= Sjarvi.daily_Solarenergy{j}(1)
    +...
         (Sjarvi.datafiltered1{j}(x,9)*37.95* (1/6));
         Sjarvi.dayfiltered1{j}(1)=Sjarvi.Yearfiltered1{j}(x);
        else
        Sjarvi.daily_Gridenergy1{j}(1) = Sjarvi.daily_Gridenergy1{j}(1)
    +...
            (Sjarvi.datafiltered1{j}(x,1)* 0);
```

```
        Sjarvi.daily_Gridenergy2{j}(1) = Sjarvi.daily_Gridenergy2{j}(1) +...
         (Sjarvi.datafiltered1{j}(x,2)* 0);
        Sjarvi.daily_Arrayenergy1{j}(1)= Sjarvi.daily_Arrayenergy1{j}(1) +...
         (Sjarvi.datafiltered1{j}(x,3)*Sjarvi.datafiltered1{j}(x,4)* 0);
        Sjarvi.daily_Arrayenergy2{j}(1)= Sjarvi.daily_Arrayenergy2{j}(1) +...
         (Sjarvi.datafiltered1{j}(x,5)*Sjarvi.datafiltered1{j}(x,6)* 0);
         Sjarvi.daily_Solarenergy{j}(1)= Sjarvi.daily_Solarenergy{j}(1) +...
         (Sjarvi.datafiltered1{j}(x,9)*37.95*0);
         Sjarvi.dayfiltered1{j}(1)=Sjarvi.Yearfiltered1{j}(x);
        end

      else
           Sjarvi.dayfiltered(q)=Sjarvi.dayfiltered1{j}(1);
           Sjarvi.dailyGridenergy1(q)=Sjarvi.daily_Gridenergy1{j}(1)/1000;
           Sjarvi.dailyGridenergy2(q)=Sjarvi.daily_Gridenergy2{j}(1)/1000;
           Sjarvi.dailyArrayenergy1(q)=Sjarvi.daily_Arrayenergy1{j}(1)/1000;
           Sjarvi.dailyArrayenergy2(q)=Sjarvi.daily_Arrayenergy2{j}(1)/1000;
           Sjarvi.dailySolarenergy(q)= Sjarvi.daily_Solarenergy{j}(1)/1000;
          %initialising the next day energies.
        l=l+1;
        q=q+1;
        Sjarvi.daily_Gridenergy1{j}(1) = (Sjarvi.datafiltered1{j}(x,1)*0.25);
        Sjarvi.daily_Gridenergy2{j}(1) = (Sjarvi.datafiltered1{j}(x,2)*0.25);
        Sjarvi.daily_Arrayenergy1{j}(1)= (Sjarvi.datafiltered1{j}(x,3)...
         *Sjarvi.datafiltered1{j}(x,4)* 0.25);
        Sjarvi.daily_Arrayenergy2{j}(1)= (Sjarvi.datafiltered1{j}(x,5)...
         *Sjarvi.datafiltered1{j}(x,6)* 0.25);
```

```
        Sjarvi.daily_Solarenergy{j}(1)=(Sjarvi.datafiltered1{j}(x,9)*
    37.95*0.25);
        Sjarvi.dayfiltered1{j}(1)=Sjarvi.Yearfiltered1{j}(x);
        continue;
      end
    end
Sjarvi.dailyGridenergy1(q)=Sjarvi.daily_Gridenergy1{j}(1)/1000;
Sjarvi.dailyGridenergy2(q)=Sjarvi.daily_Gridenergy2{j}(1)/1000;
Sjarvi.dailyArrayenergy1(q)=Sjarvi.daily_Arrayenergy1{j}(1)/1000;
Sjarvi.dailyArrayenergy2(q)=Sjarvi.daily_Arrayenergy2{j}(1)/1000;
Sjarvi.dailySolarenergy(q)= Sjarvi.daily_Solarenergy{j}(1)/1000;
Sjarvi.dayfiltered(q)=Sjarvi.Yearfiltered1{j}(x);
end
% To free up more memory, get rid of the unnecessary variables
clear D H M MN S Y j l p q x

%% Yields
% Reference yield
Sjarvi.dailyRefyield = Sjarvi.dailySolarenergy./(1*37.95); %The units are
    h/d
% Array yield
Sjarvi.dailyArrayyield1 = Sjarvi.dailyArrayenergy1./5.13;
Sjarvi.dailyArrayyield2 = Sjarvi.dailyArrayenergy2./1.14;
% Final yield
Sjarvi.dailyFinalyield1 = Sjarvi.dailyGridenergy1./5.13;
Sjarvi.dailyFinalyield2 = Sjarvi.dailyGridenergy2./1.14;
%% Performance Ratio and efficiencies
% Performance ratio
Sjarvi.PR1= Sjarvi.dailyFinalyield1 ./Sjarvi.dailyRefyield;
Sjarvi.PR2= Sjarvi.dailyFinalyield2 ./Sjarvi.dailyRefyield;
%Inverter efficiency
Sjarvi.InvEff1 = Sjarvi.dailyGridenergy1./Sjarvi.dailyArrayenergy1;
Sjarvi.InvEff2 = Sjarvi.dailyGridenergy2./Sjarvi.dailyArrayenergy2;
%Array efficiency.
%For the array efficiency, the total array area is split so as to get the
%in-plane solar energy on each sub-system
area1=(37.95/33)*27;
```

```
area2=37.95-area1;
Sjarvi.ArrayEff1 = Sjarvi.dailyArrayenergy1...
    ./(Sjarvi.dailySolarenergy*(area1/37.95));
Sjarvi.ArrayEff2 = Sjarvi.dailyArrayenergy2...
    ./(Sjarvi.dailySolarenergy*(area2/37.95));
%System efficiency
Sjarvi.SysEff1 = (Sjarvi.dailyGridenergy1)...
    ./(Sjarvi.dailySolarenergy*(area1/37.95));
Sjarvi.SysEff2 = (Sjarvi.dailyGridenergy2)...
    ./(Sjarvi.dailySolarenergy*(area2/37.95));
%Capture loss
Sjarvi.LC1 = Sjarvi.dailyRefyield- Sjarvi.dailyArrayyield1;
Sjarvi.LC2 = Sjarvi.dailyRefyield- Sjarvi.dailyArrayyield2;
%System loss
Sjarvi.LS1 = Sjarvi.dailyRefyield - Sjarvi.dailyFinalyield1;
Sjarvi.LS2 = Sjarvi.dailyRefyield - Sjarvi.dailyFinalyield2;
%
Sjarvi.PR1(isnan(Sjarvi.PR1))=[];
Sjarvi.PR2(isnan(Sjarvi.PR2))=[];
Sjarvi.SysEff1(isnan(Sjarvi.SysEff1))=[];
Sjarvi.SysEff2(isnan(Sjarvi.SysEff2))=[];
Sjarvi.ArrayEff1(isnan(Sjarvi.ArrayEff1))=[];
Sjarvi.ArrayEff2(isnan(Sjarvi.ArrayEff2))=[];
Sjarvi.InvEff1(isnan(Sjarvi.InvEff1))=0;
Sjarvi.InvEff2(isnan(Sjarvi.InvEff2))=0;
Sjarvi.InvEff1(isinf(Sjarvi.InvEff1))=0;
Sjarvi.InvEff2(isinf(Sjarvi.InvEff2))=0;

%% Removing the hour, minute and second component of the matlab dates
%%Assigned to the day and dayfiltered.
for x = 1 : length (Sjarvi.day)
    [Y M D]= datevec(Sjarvi.day(x));
    Sjarvi.day(x)=datenum(Y,M,D);
end
for x= 1:length(Sjarvi.dayfiltered)
    [Y M D]= datevec(Sjarvi.dayfiltered(x));
    Sjarvi.day_filtered(x)=datenum(Y,M,D);
```

```
end
day=0;
%Filtered average daily irradiance
for x = 1: length(Sjarvi.day)
    day =find(Sjarvi.day_filtered==Sjarvi.day(x));
    Sjarvi.filteredAvgIrradiance(day) = Sjarvi.avgIrradiance(x);
    Sjarvi.filteredAmbtemp(day) = Sjarvi.dailyAmbtemp(x);
    Sjarvi.filteredCelltemp(day) = Sjarvi.dailyCelltemp(x);
end
%For the full reported cycle, the days with no values will be assigned
%empty matrices;
Sjarvi.Full.avgIrradiance=Sjarvi.avgIrradiance;
Sjarvi.Full.dailyAmbtemp = Sjarvi.dailyAmbtemp;
Sjarvi.Full.dailyCelltemp = Sjarvi.dailyCelltemp;
Sjarvi.Full.day=Sjarvi.day;
Sjarvi.day_filtered(1577)=734507;
for x = 1: length(Sjarvi.day)
    day =find(Sjarvi.day_filtered==Sjarvi.day(x));
    if ((isempty(day)==0))
        Sjarvi.Full.PR1(x)=Sjarvi.PR1(day);
        Sjarvi.Full.PR2(x)=Sjarvi.PR2(day);
        Sjarvi.Full.InvEff1(x)=Sjarvi.InvEff1(day);
        Sjarvi.Full.InvEff2(x)=Sjarvi.InvEff2(day);
        Sjarvi.Full.SysEff1(x)=Sjarvi.SysEff1(day);
        Sjarvi.Full.SysEff2(x)=Sjarvi.SysEff2(day);
        Sjarvi.Full.ArrayEff1(x)=Sjarvi.ArrayEff1(day);
        Sjarvi.Full.ArrayEff2(x)=Sjarvi.ArrayEff2(day);
        Sjarvi.Full.InvEff1(x)=Sjarvi.InvEff1(day);
        Sjarvi.Full.InvEff2(x)=Sjarvi.InvEff2(day);
        Sjarvi.Full.Finalyield1(x)=Sjarvi.dailyFinalyield1(day);
        Sjarvi.Full.Finalyield2(x)=Sjarvi.dailyFinalyield2(day);
        Sjarvi.Full.Arrayyield1(x)=Sjarvi.dailyArrayyield1(day);
        Sjarvi.Full.Arrayyield2(x)=Sjarvi.dailyArrayyield2(day);
        Sjarvi.Full.Refyield(x)=Sjarvi.dailyRefyield(day);
    end
    %To complete the full cycle.
    if (x==2287)
```

```
            Sjarvi.Full.PR1(x)=0;
            Sjarvi.Full.PR2(x)=0;
            Sjarvi.Full.InvEff1(x)=0;
            Sjarvi.Full.InvEff2(x)=0;
            Sjarvi.Full.SysEff1(x)=0;
            Sjarvi.Full.SysEff2(x)=0;
            Sjarvi.Full.ArrayEff1(x)=0;
            Sjarvi.Full.ArrayEff2(x)=0;
            Sjarvi.Full.InvEff1(x)=0;
            Sjarvi.Full.InvEff2(x)=0;
            Sjarvi.Full.Finalyield1(x)=0;
            Sjarvi.Full.Finalyield2(x)=0;
            Sjarvi.Full.Arrayyield1(x)=0;
            Sjarvi.Full.Arrayyield2(x)=0;
            Sjarvi.Full.Refyield(x)=0;
      end
end
save performanceParameters.mat Sjarvi area1 area2

Sjarvi7year = Sjarvi.Full;

save 7yearcycle.mat Sjarvi7year
```

code/Matcode/SjarviPerformanceIndices.m

```
% Written by Dereck Mutungi
% This program is used to get the monthly averages of the data collected.
clear all; close all; clc;
%Load necessary Agora3year matrix to plot graphs
load Agora3year.mat
%%
[Y(1) M(1) D(1) H(1) MN(1) S(1)] =  datevec(Agora3year.day(1));
  l=1;
  PR(1)=Agora3year.PR(1);
  count(1)=1;
  month(1)=0;
  ArrayEff(1)=Agora3year.ArrayEff(1);
  SysEff(1)=Agora3year.SysEff(1);
```

```
  Arrayyield(l) = Agora3year.dailyArrayyield(1);
  Refyield(l) = Agora3year.dailyRefyield(1);
  Finalyield(l) = Agora3year.dailyFinalyield(1);
  Irradiance(l) = Agora3year.daily_avgIrradiance(1);
  InvEff(l)= Agora3year.InvEff(1);
for j=2:p

    [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Agora3year.day(j));
     if (M(j-1)== M(j)) && (Y(j-1) == Y(j))
      month(l)=M(j);
      count(l)=count(l)+1;
      PR(l)=PR(l)+Agora3year.PR(j);
      ArrayEff(l)=ArrayEff(l)+Agora3year.ArrayEff(j);
      SysEff(l)=SysEff(l)+Agora3year.SysEff(j);
      Arrayyield(l) =  Arrayyield(l)+ Agora3year.dailyArrayyield(j);
      Refyield(l) = Refyield(l)+ Agora3year.dailyRefyield(j);
      Finalyield(l) = Finalyield(l)+Agora3year.dailyFinalyield(j);
      Irradiance(l) = Irradiance(l)+Agora3year.daily_avgIrradiance(j);
      InvEff(l)= InvEff(l)+Agora3year.InvEff(j);
     else
        meanPR(l)=PR(l)/count(l);
        meanArrayEff(l)=ArrayEff(l)/count(l);
        meanSysEff(l)=SysEff(l)/count(l);
        meanArrayyield(l) = Arrayyield(l)/count(l);
        meanFinalyield(l) = Finalyield(l)/count(l);
        meanRefyield(l) = Refyield(l)/count(l);
        meanIrradiance(l) = Irradiance(l)/count(l);
        meanInvEff(l) = InvEff(l)/count(l);
        l=l+1;
        month(l)=M(j);
        count(l)=1;
        PR(l)= Agora3year.PR(j);
        ArrayEff(l)=Agora3year.ArrayEff(j);
        SysEff(l)=Agora3year.SysEff(j);
        Arrayyield(l) = Agora3year.dailyArrayyield(j);
        Refyield(l) = Agora3year.dailyRefyield(j);
        Finalyield(l) = Agora3year.dailyFinalyield(j);
```

```
            Irradiance(l) = Agora3year.daily_avgIrradiance(j);
            InvEff(l)= Agora3year.InvEff(j);
        end
end
meanPR(l)=PR(l)/count(l);
meanArrayEff(l)=Agora3year.ArrayEff(l)/count(l);
meanSysEff(l)=Agora3year.SysEff(l)/count(l);
meanArrayyield(l) = Arrayyield(l)/count(l);
meanFinalyield(l) = Finalyield(l)/count(l);
meanRefyield(l) = Refyield(l)/count(l);
meanIrradiance(l) = Irradiance(l)/count(l);
meanInvEff(l) = InvEff(l)/count(l);

clear H M MN S Y count j i D l
%% Stacked monthly yields graph.
for i=1:length(PR)
    Tot{i}=[Finalyield(i) Arrayyield(i)-Finalyield(i) Refyield(i)-
    Arrayyield(i)];

end
Total=vertcat(Tot{1:length(Tot)});
bar(Total,'stack');
grid on
xlim([0,37]);
set(gca,'XTick',0:6:36)
ylabel('Yields (h/m)');
xlabel('Month');
title ({'Stacked graph showing the final yield,capture loss and system
    loss';...
     'Agora'});
legend('Y_f','L_s','L_c');

%%
monthlyArrayEff=meanArrayEff';
monthlyPR=meanPR';
monthlySysEff=meanSysEff';
monthlyArrayyield=meanArrayyield';
```

```
monthlyFinalyield=meanFinalyield';
monthlyRefyield=meanRefyield';
monthlyIrradiance=meanIrradiance';
monthlyInvEff = meanInvEff';

save Agoramonthlyvalues.mat monthlyArrayEff monthlyPR monthlySysEff...
monthlyArrayyield monthlyFinalyield monthlyRefyield monthlyIrradiance...
monthlyInvEff

clear p j irr i h a ArrayEff Arrayyield Finalyield Irradiance PR Refyield
    ...
     SysEff meanArrayEff meanPR meanSysEff meanArrayyield meanFinalyield
    ...
     meanRefyield meanIrradiance InvEff meanInvEff
%%
%weekly average of the data
l=1;
PR(l)=Agora3year.PR(1);
count(l)=1;
ArrayEff(l)=Agora3year.ArrayEff(1);
SysEff(l)=Agora3year.SysEff(1);
Arrayyield(l) = Agora3year.dailyArrayyield(1);
Refyield(l) = Agora3year.dailyRefyield(1);
Finalyield(l) = Agora3year.dailyFinalyield(1);
Irradiance(l) = Agora3year.daily_avgIrradiance(1);
InvEff(l)= Agora3year.InvEff(1);
for j=1:length(Agora3year.time2)
    if (mod(j,7)~=0)
       count(l)=count(l)+1;
       PR(l)=PR(l)+Agora3year.PR(j);
       ArrayEff(l)=ArrayEff(l)+Agora3year.ArrayEff(j);
       SysEff(l)=SysEff(l)+Agora3year.SysEff(j);
       Arrayyield(l) =  Arrayyield(l)+ Agora3year.dailyArrayyield(j);
       Refyield(l) = Refyield(l)+ Agora3year.dailyRefyield(j);
       Finalyield(l) = Finalyield(l)+Agora3year.dailyFinalyield(j);
       Irradiance(l) = Irradiance(l)+Agora3year.daily_avgIrradiance(j);
       InvEff(l)= InvEff(l)+ Agora3year.InvEff(j);
```

```
    else
       meanPR(l)=PR(l)/count(l);
          meanArrayEff(l)=ArrayEff(l)/count(l);
          meanSysEff(l)=SysEff(l)/count(l);
          meanArrayyield(l) = Arrayyield(l)/count(l);
          meanFinalyield(l) = Finalyield(l)/count(l);
          meanRefyield(l) = Refyield(l)/count(l);
          meanIrradiance(l) = Irradiance(l)/count(l);
          meanInvEff(l) = InvEff(l)/count(l);
          l=l+1;
          count(l)=1;
          PR(l)= Agora3year.PR(j);
          ArrayEff(l)=Agora3year.ArrayEff(j);
          SysEff(l)=Agora3year.SysEff(j);
          Arrayyield(l) = Agora3year.dailyArrayyield(j);
          Refyield(l) = Agora3year.dailyRefyield(j);
          Finalyield(l) = Agora3year.dailyFinalyield(j);
          Irradiance(l) = Agora3year.daily_avgIrradiance(j);
          InvEff(l)= Agora3year.InvEff(j);
    end
end
meanPR(l)=PR(l)/count(l);
meanArrayEff(l)=Agora3year.ArrayEff(l)/count(l);
meanSysEff(l)=Agora3year.SysEff(l)/count(l);
meanArrayyield(l) = Arrayyield(l)/count(l);
meanFinalyield(l) = Finalyield(l)/count(l);
meanRefyield(l) = Refyield(l)/count(l);
meanIrradiance(l) = Irradiance(l)/count(l);
meanInvEff(l) = InvEff(l)/count(l);

weeklyArrayEff=meanArrayEff';
weeklyPR=meanPR';
weeklySysEff=meanSysEff';
weeklyArrayyield=meanArrayyield';
weeklyFinalyield=meanFinalyield';
weeklyRefyield=meanRefyield';
weeklyIrradiance=meanIrradiance';
```

```
weeklyInvEff=meanInvEff';

clear p j irr i h a ArrayEff Arrayyield Finalyield Irradiance PR Refyield ...
    SysEff meanArrayEff meanPR meanSysEff meanArrayyield meanFinalyield ...
    meanRefyield meanIrradiance meanInvEff InvEff
% Weekly values to be used in ARIMA model value and degradation
% calculations.
save Agoraweeklyvalues.mat weeklyArrayEff weeklyPR weeklySysEff...
weeklyArrayyield weeklyFinalyield weeklyRefyield weeklyIrradiance ...
weeklyInvEff
```

code/Matcode/monthlyWeeklyAveragesAgora.m

```
%Written by Dereck Mutungi

% This program is used to get the monthly averages of the data collected.
% Weekly averages were also calculated and a stack graph plotted for the
% monthly yield values.
%%
clear all; close all; clc;
%Load necessary 7 year Sjarvi data matrix to plot graphs.See
%GridArrayenergies.m
load 7yearcycle.mat
%%
[Y(1) M(1) D(1) H(1) MN(1) S(1)] =  datevec(Sjarvi7year.day(1));
   l=1;
   PR1(1)=Sjarvi7year.PR1(1);
   count(1)=1;
   month(1)=0;
   ArrayEff1(1)=Sjarvi7year.ArrayEff1(1);
   SysEff1(1)=Sjarvi7year.SysEff1(1);
   Arrayyield1(1) = Sjarvi7year.Arrayyield1(1);
   Refyield(1) = Sjarvi7year.Refyield(1);
   Finalyield1(1) = Sjarvi7year.Finalyield1(1);
   Irradiance(1) = Sjarvi7year.avgIrradiance(1);
   PR2(1)=Sjarvi7year.PR2(1);
```

```
   ArrayEff2(1)=Sjarvi7year.ArrayEff2(1);
   SysEff2(1)=Sjarvi7year.SysEff2(1);
   Arrayyield2(1) = Sjarvi7year.Arrayyield2(1);
   Finalyield2(1) = Sjarvi7year.Finalyield2(1);
   InvEff1(1) = Sjarvi7year.InvEff1(1);
   InvEff2(1) = Sjarvi7year.InvEff2(1);
for j=2:length(Sjarvi7year.day)

     [Y(j) M(j) D(j) H(j) MN(j) S(j)] =  datevec(Sjarvi7year.day(j));
      if (M(j-1)== M(j)) && (Y(j-1) == Y(j))
       month(1)=M(j);
       count(1)=count(1)+1;
       PR1(1)=PR1(1)+Sjarvi7year.PR1(j);
       ArrayEff1(1)=ArrayEff1(1)+Sjarvi7year.ArrayEff1(j);
       SysEff1(1)=SysEff1(1)+Sjarvi7year.SysEff1(j);
       Arrayyield1(1) =  Arrayyield1(1)+ Sjarvi7year.Arrayyield1(j);
       Refyield(1) = Refyield(1)+ Sjarvi7year.Refyield(j);
       Finalyield1(1) = Finalyield1(1)+Sjarvi7year.Finalyield1(j);
       Irradiance(1) = Irradiance(1)+Sjarvi7year.avgIrradiance(j);
       PR2(1)=PR2(1)+Sjarvi7year.PR2(j);
       ArrayEff2(1)=ArrayEff2(1)+Sjarvi7year.ArrayEff2(j);
       SysEff2(1)=SysEff2(1)+Sjarvi7year.SysEff2(j);
       Arrayyield2(1) =  Arrayyield2(1)+ Sjarvi7year.Arrayyield2(j);
       Finalyield2(1) = Finalyield2(1)+Sjarvi7year.Finalyield2(j);
      else
          meanPR1(1)=PR1(1)/count(1);
          meanArrayEff1(1)=ArrayEff1(1)/count(1);
          meanSysEff1(1)=SysEff1(1)/count(1);
          meanArrayyield1(1) = Arrayyield1(1)/count(1);
          meanFinalyield1(1) = Finalyield1(1)/count(1);
          meanRefyield(1) = Refyield(1)/count(1);
          meanIrradiance(1) = Irradiance(1)/count(1);
          meanPR2(1)=PR2(1)/count(1);
          meanArrayEff2(1)=ArrayEff2(1)/count(1);
          meanSysEff2(1)=SysEff2(1)/count(1);
          meanArrayyield2(1) = Arrayyield2(1)/count(1);
          meanFinalyield2(1) = Finalyield2(1)/count(1);
```

```
            meanInvEff1(l)=InvEff1(l)/count(l);
            meanInvEff2(l)=InvEff2(l)/count(l);
            l=l+1;
            month(l)=M(j);
            count(l)=1;
            PR1(l)= Sjarvi7year.PR1(j);
            ArrayEff1(l)=Sjarvi7year.ArrayEff1(j);
            SysEff1(l)=Sjarvi7year.SysEff1(j);
            Arrayyield1(l) = Sjarvi7year.Arrayyield1(j);
            Refyield(l) = Sjarvi7year.Refyield(j);
            Finalyield1(l) = Sjarvi7year.Finalyield1(j);
            Irradiance(l) = Sjarvi7year.avgIrradiance(j);
            PR2(l)= Sjarvi7year.PR2(j);
            ArrayEff2(l)=Sjarvi7year.ArrayEff2(j);
            SysEff2(l)=Sjarvi7year.SysEff2(j);
            Arrayyield2(l) = Sjarvi7year.Arrayyield2(j);
            Finalyield2(l) = Sjarvi7year.Finalyield2(j);
            InvEff1(l) = Sjarvi7year.InvEff1(j);
            InvEff2(l) = Sjarvi7year.InvEff2(j);
        end
end
meanPR1(l)=PR1(l)/count(l);
meanArrayEff1(l)=Sjarvi7year.ArrayEff1(l)/count(l);
meanSysEff1(l)=Sjarvi7year.SysEff1(l)/count(l);
meanArrayyield1(l) = Arrayyield1(l)/count(l);
meanFinalyield1(l) = Finalyield1(l)/count(l);
meanRefyield(l) = Refyield(l)/count(l);
meanIrradiance(l) = Irradiance(l)/count(l);
meanPR2(l)=PR2(l)/count(l);
meanArrayEff2(l)=ArrayEff2(l)/count(l);
meanSysEff2(l)=SysEff2(l)/count(l);
meanArrayyield2(l) = Arrayyield2(l)/count(l);
meanFinalyield2(l) = Finalyield2(l)/count(l);
meanInvEff1(l)=InvEff1(l)/count(l);
meanInvEff2(l)=InvEff2(l)/count(l);

clear H M MN S Y count j i D l
```

```
%%

% To complete the full cycle, the missing months are given a zero value.
% This does not affect the trending since, there will be an irradiance
% condition used to filter.
n=16; n2=28;
month=[month(1:n-1),4,5,month(n:n2-1),6,month(n2:end)];
meanPR1=[meanPR1(1:n-1),NaN,NaN,meanPR1(n:n2-1),NaN,meanPR1(n2:end)];
meanArrayEff1=[meanArrayEff1(1:n-1),NaN,NaN,meanArrayEff1(n:n2-1),...
    NaN,meanArrayEff1(n2:end)];
meanSysEff1=[meanSysEff1(1:n-1),NaN,NaN,meanSysEff1(n:n2-1),...
    NaN,meanSysEff1(n2:end)];
meanArrayyield1=[meanArrayyield1(1:n-1),NaN,NaN,meanArrayyield1(n:n2-1),...
    NaN,meanArrayyield1(n2:end)];
meanFinalyield1=[meanFinalyield1(1:n-1),NaN,NaN,meanFinalyield1(n:n2-1),...
    NaN,meanFinalyield1(n2:end)];
meanRefyield =[meanRefyield(1:n-1),NaN,NaN,meanRefyield(n:n2-1),...
    NaN,meanRefyield(n2:end)];
meanIrradiance=[meanIrradiance(1:n-1),NaN,NaN,meanIrradiance(n:n2-1),...
    NaN,meanIrradiance(n2:end)];
meanPR2=[meanPR2(1:n-1),NaN,NaN,meanPR2(n:n2-1),NaN,meanPR2(n2:end)];
meanArrayEff2=[meanArrayEff2(1:n-1),NaN,NaN,meanArrayEff2(n:n2-1),...
    NaN,meanArrayEff2(n2:end)];
meanSysEff2=[meanSysEff2(1:n-1),NaN,NaN,meanSysEff2(n:n2-1),...
    NaN,meanSysEff2(n2:end)];
meanArrayyield2=[meanArrayyield2(1:n-1),NaN,NaN,meanArrayyield2(n:n2-1),...
    NaN,meanArrayyield2(n2:end)];
meanFinalyield2=[meanFinalyield2(1:n-1),NaN,NaN,meanFinalyield2(n:n2-1),...
    NaN,meanFinalyield2(n2:end)];
Finalyield1=[Finalyield1(1:n-1),NaN,NaN,Finalyield1(n:n2-1),...
    NaN,Finalyield1(n2:end)];
Finalyield2=[Finalyield2(1:n-1),NaN,NaN,Finalyield2(n:n2-1),...
    NaN,Finalyield2(n2:end)];
```

```
Arrayyield1=[Arrayyield1(1:n-1),NaN,NaN,Arrayyield1(n:n2-1),...
    NaN,Arrayyield1(n2:end)];
Arrayyield2=[Arrayyield2(1:n-1),NaN,NaN,Arrayyield2(n:n2-1),...
    NaN,Arrayyield2(n2:end)];
Refyield  =[Refyield(1:n-1),NaN,NaN,Refyield(n:n2-1),...
    NaN,Refyield(n2:end)];
meanInvEff1  =[meanInvEff1(1:n-1),NaN,NaN,meanInvEff1(n:n2-1),...
    NaN,meanInvEff1(n2:end)];
meanInvEff2  =[meanInvEff2(1:n-1),NaN,NaN,meanInvEff2(n:n2-1),...
    NaN,meanInvEff2(n2:end)];

month=month';
monthlySjarviPR1=meanPR1';
monthlySjarviArrayEff1=meanArrayEff1';
monthlySjarviSysEff1=meanSysEff1';
monthlySjarviArrayyield1=meanArrayyield1';
monthlySjarviFinalyield1=meanFinalyield1';
monthlySjarviRefyield1=meanRefyield';
monthlySjarviIrradiance=meanIrradiance';
monthlySjarviPR2=meanPR2';
monthlySjarviArrayEff2=meanArrayEff2';
monthlySjarviSysEff2=meanSysEff2';
monthlySjarviArrayyield2=meanArrayyield2';
monthlySjarviFinalyield2=meanFinalyield2';
monthlySjarviInvEff1 = meanInvEff1';
monthlySjarviInvEff2 = meanInvEff2';

save Sjarvimonthlyvalues.mat monthlySjarviPR1 monthlySjarviArrayEff1...
    monthlySjarviSysEff1 monthlySjarviArrayyield1
   monthlySjarviFinalyield1...
    monthlySjarviRefyield1 monthlySjarviIrradiance monthlySjarviPR2...
    monthlySjarviArrayEff2 monthlySjarviSysEff2 monthlySjarviArrayyield2
   ...
    monthlySjarviFinalyield2 monthlySjarviInvEff1 monthlySjarviInvEff2
%% Plotting of stacked graphs
for i=1:length(Finalyield1)
    Tot1{i}=[Finalyield1(i) Arrayyield1(i)-Finalyield1(i)...
```

```
        Refyield(i)-Arrayyield1(i)];
    Tot2{i}=[Finalyield2(i) Arrayyield2(i)-Finalyield2(i)...
        Refyield(i)-Arrayyield2(i)];
end
subplot(2,1,1)
Total1=vertcat(Tot1{1:length(Tot1)});
bar(Total1,'stack');
grid on
xlim([0,85]);
set(gca,'XTick',0:6:84)
ylabel('Yields (h/m)');
xlabel('Month');
title({'Stacked graph showing the final yield,capture loss and system
    loss';...
    'Sjarvi:Subsystem 1'});
legend('Y_f','L_s','L_c');
subplot(2,1,2)
Total2=vertcat(Tot2{1:length(Tot2)});
bar(Total2,'stack');
grid on
xlim([0,85]);
set(gca,'XTick',0:6:84)
ylabel('Yields (h/m)');
xlabel('Month');
title ({'Stacked graph showing the final yield,capture loss and system
    loss';...
    'Sjarvi:Subsystem 2'});
legend('Y_f','L_s','L_c');


clear p j irr i h a ArrayEff1 Arrayyield1 Finalyield1 Irradiance PR1...
    Refyield SysEff1 meanArrayEff1 meanPR1 meanSysEff1 meanArrayyield1...
    meanFinalyield1 meanRefyield meanIrradiance ArrayEff2 Arrayyield2...
    Finalyield2 PR2 Refyield SysEff2 meanArrayEff2 meanPR2 meanSysEff2...
    meanArrayyield2 meanFinalyield2

%%
```

```
%weekly average of the data

for daycount=732678: 735234
    pos = find(daycount==Sjarvi7year.day);
    if (length(pos)<1)
        position = find((daycount-1)==Sjarvi7year.day);
        Sjarvi7year.day=[Sjarvi7year.day(1:position),daycount,...
         Sjarvi7year.day(position+1:end)];
     Sjarvi7year.ArrayEff1=[Sjarvi7year.ArrayEff1(1:position),0,...
         Sjarvi7year.ArrayEff1(position+1:end)];
     Sjarvi7year.SysEff1=[Sjarvi7year.SysEff1(1:position),0,...
         Sjarvi7year.SysEff1(position+1:end)];
     Sjarvi7year.Arrayyield1=[Sjarvi7year.Arrayyield1(1:position),0,...
         Sjarvi7year.Arrayyield1(position+1:end)];
     Sjarvi7year.Finalyield1=[Sjarvi7year.Finalyield1(1:position),0,...
         Sjarvi7year.Finalyield1(position+1:end)];
     Sjarvi7year.PR1=[Sjarvi7year.PR1(1:position),0,...
         Sjarvi7year.PR1(position+1:end)];
     Sjarvi7year.Refyield=[Sjarvi7year.Refyield(1:position),0,...
         Sjarvi7year.Refyield(position+1:end)];
     Sjarvi7year.avgIrradiance=[Sjarvi7year.avgIrradiance(1:position)
    ,0,...
         Sjarvi7year.avgIrradiance(position+1:end)];
     Sjarvi7year.ArrayEff2=[Sjarvi7year.ArrayEff2(1:position),0,...
         Sjarvi7year.ArrayEff2(position+1:end)];
     Sjarvi7year.SysEff2=[Sjarvi7year.SysEff2(1:position),0,...
         Sjarvi7year.SysEff2(position+1:end)];
     Sjarvi7year.Arrayyield2=[Sjarvi7year.Arrayyield2(1:position),0,...
         Sjarvi7year.Arrayyield2(position+1:end)];
     Sjarvi7year.Finalyield2=[Sjarvi7year.Finalyield2(1:position),0,...
         Sjarvi7year.Finalyield2(position+1:end)];
     Sjarvi7year.PR2=[Sjarvi7year.PR2(1:position),0,...
         Sjarvi7year.PR2(position+1:end)];
     Sjarvi7year.InvEff1=[Sjarvi7year.InvEff1(1:position),0,...
         Sjarvi7year.InvEff1(position+1:end)];
     Sjarvi7year.InvEff2=[Sjarvi7year.InvEff2(1:position),0,...
         Sjarvi7year.InvEff2(position+1:end)];
```

```
    end
end
  l=1;
PR1(l)=Sjarvi7year.PR1(1);
  count(l)=1;
  ArrayEff1(l)=Sjarvi7year.ArrayEff1(1);
  SysEff1(l)=Sjarvi7year.SysEff1(1);
  Arrayyield1(l) = Sjarvi7year.Arrayyield1(1);
  Refyield(l) = Sjarvi7year.Refyield(1);
  Finalyield1(l) = Sjarvi7year.Finalyield1(1);
  Irradiance(l) = Sjarvi7year.avgIrradiance(1);
  PR2(l)=Sjarvi7year.PR2(1);
  ArrayEff2(l)=Sjarvi7year.ArrayEff2(1);
  SysEff2(l)=Sjarvi7year.SysEff2(1);
  Arrayyield2(l) = Sjarvi7year.Arrayyield2(1);
  Finalyield2(l) = Sjarvi7year.Finalyield2(1);
  InvEff1(l) = Sjarvi7year.InvEff1(1);
  InvEff2(l) = Sjarvi7year.InvEff2(1);
for j=2:length(Sjarvi7year.day)
    if (mod(j,7)~=0)
      count(l)=count(l)+1;
      PR1(l)=PR1(l)+Sjarvi7year.PR1(j);
      ArrayEff1(l)=ArrayEff1(l)+Sjarvi7year.ArrayEff1(j);
      SysEff1(l)=SysEff1(l)+Sjarvi7year.SysEff1(j);
      Arrayyield1(l) =  Arrayyield1(l)+ Sjarvi7year.Arrayyield1(j);
      Refyield(l) = Refyield(l)+ Sjarvi7year.Refyield(j);
      Finalyield1(l) = Finalyield1(l)+Sjarvi7year.Finalyield1(j);
      Irradiance(l) = Irradiance(l)+Sjarvi7year.avgIrradiance(j);
      PR2(l)=PR2(l)+Sjarvi7year.PR2(j);
      ArrayEff2(l)=ArrayEff2(l)+Sjarvi7year.ArrayEff2(j);
      SysEff2(l)=SysEff2(l)+Sjarvi7year.SysEff2(j);
      Arrayyield2(l) =  Arrayyield2(l)+ Sjarvi7year.Arrayyield2(j);
      Finalyield2(l) = Finalyield2(l)+Sjarvi7year.Finalyield2(j);
      InvEff1(l) = InvEff1(l)+Sjarvi7year.InvEff1(j);
      InvEff2(l) = InvEff2(l)+Sjarvi7year.InvEff2(j);
     else
         meanPR1(l)=PR1(l)/count(l);
```

```
            meanArrayEff1(l)=ArrayEff1(l)/count(l);
            meanSysEff1(l)=SysEff1(l)/count(l);
            meanArrayyield1(l) = Arrayyield1(l)/count(l);
            meanFinalyield1(l) = Finalyield1(l)/count(l);
            meanRefyield(l) = Refyield(l)/count(l);
            meanIrradiance(l) = Irradiance(l)/count(l);
            meanPR2(l)=PR2(l)/count(l);
            meanArrayEff2(l)=ArrayEff2(l)/count(l);
            meanSysEff2(l)=SysEff2(l)/count(l);
            meanArrayyield2(l) = Arrayyield2(l)/count(l);
            meanFinalyield2(l) = Finalyield2(l)/count(l);
            meanInvEff1(l)=InvEff1(l)/count(l);
            meanInvEff2(l)=InvEff2(l)/count(l);
            l=l+1;
            count(l)=1;
            PR1(l)= Sjarvi7year.PR1(j);
            ArrayEff1(l)=Sjarvi7year.ArrayEff1(j);
            SysEff1(l)=Sjarvi7year.SysEff1(j);
            Arrayyield1(l) = Sjarvi7year.Arrayyield1(j);
            Refyield(l) = Sjarvi7year.Refyield(j);
            Finalyield1(l) = Sjarvi7year.Finalyield1(j);
            Irradiance(l) = Sjarvi7year.avgIrradiance(j);
            PR2(l)= Sjarvi7year.PR2(j);
            ArrayEff2(l)=Sjarvi7year.ArrayEff2(j);
            SysEff2(l)=Sjarvi7year.SysEff2(j);
            Arrayyield2(l) = Sjarvi7year.Arrayyield2(j);
            Finalyield2(l) = Sjarvi7year.Finalyield2(j);
            InvEff1(l) = Sjarvi7year.InvEff1(j);
            InvEff2(l) = Sjarvi7year.InvEff2(j);
        end
end
meanPR1(l)=PR1(l)/count(l);
meanArrayEff1(l)=Sjarvi7year.ArrayEff1(l)/count(l);
meanSysEff1(l)=Sjarvi7year.SysEff1(l)/count(l);
meanArrayyield1(l) = Arrayyield1(l)/count(l);
meanFinalyield1(l) = Finalyield1(l)/count(l);
meanRefyield(l) = Refyield(l)/count(l);
```

```
meanIrradiance(l) = Irradiance(l)/count(l);
meanPR2(l)=PR2(l)/count(l);
meanArrayEff2(l)=ArrayEff2(l)/count(l);
meanSysEff2(l)=SysEff2(l)/count(l);
meanArrayyield2(l) = Arrayyield2(l)/count(l);
meanFinalyield2(l) = Finalyield2(l)/count(l);
meanInvEff1(l)=InvEff1(l)/count(l);
meanInvEff2(l)=InvEff2(l)/count(l);

weeklySjarviPR1=meanPR1';
weeklySjarviArrayEff1=meanArrayEff1';
weeklySjarviSysEff1=meanSysEff1';
weeklySjarviArrayyield1=meanArrayyield1';
weeklySjarviFinalyield1=meanFinalyield1';
weeklySjarviRefyield=meanRefyield';
weeklySjarviIrradiance=meanIrradiance';
weeklySjarviPR2=meanPR2';
weeklySjarviArrayEff2=meanArrayEff2';
weeklySjarviSysEff2=meanSysEff2';
weeklySjarviArrayyield2=meanArrayyield2';
weeklySjarviFinalyield2=meanFinalyield2';
weeklySjarviInvEff1 = meanInvEff1';
weeklySjarviInvEff2 = meanInvEff2';
clear p j irr i h a ArrayEff1 Arrayyield1 Finalyield1 Irradiance PR1 Refyield ...
    SysEff1 meanArrayEff1 meanPR1 meanSysEff1 meanArrayyield1 meanFinalyield1 ...
    meanRefyield meanIrradiance ArrayEff2 Arrayyield2 Finalyield2 PR2 Refyield ...
    SysEff2 meanArrayEff2 meanPR2 meanSysEff2 meanArrayyield2 meanFinalyield2 ...
% Weekly values will be used in getting ARIMA model values. Also
% degradation calculations.
save Sjarviweeklyvalues.mat weeklySjarviArrayEff1 weeklySjarviPR1 ...
    weeklySjarviSysEff1 weeklySjarviArrayyield1 weeklySjarviFinalyield1
    ...
    weeklySjarviRefyield weeklySjarviIrradiance weeklySjarviArrayEff2 ...
```

```
    weeklySjarviPR2 weeklySjarviSysEff2 weeklySjarviArrayyield2 ...
    weeklySjarviFinalyield2 weeklySjarviInvEff1 weeklySjarviInvEff2
```

code/Matcode/monthlyAverages.m

The same format was used to plot performance indices for Saarijärvi data. Therefore there is no need to replicate the code.

```
%Written by Derek Mutungi

% This code plots different scatter plots using performance indices for
%Agora.
%%
clear all; close all; clc;
%Load necessary Agora matrix to plot graphs. See AgoraPerformanceIndices.
    m
load Agora3year.mat
%Setting constants
xtag ='Time (month/year)';
n =length(Agora3year.day);
%% Graphs
% Graph of yields against time
figure (1);
subplot(1,2,1);
plot(Agora3year.day,Agora3year.dailyArrayyield,'.');
hold on
grid on
title('Array yield');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
ylabel('Yield (h/d)');
datetick('x','mm/yy');
ylim([0,9]);
subplot(1,2,2);
plot(Agora3year.day,Agora3year.dailyFinalyield,'*r')
hold on
grid on
```

```
title('Final yield');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
ylabel('Yield (h/d)');
datetick('x','mm/yy');
ylim([0,9]);
figure (2);
plot(Agora3year.day,Agora3year.dailyRefyield,'.g')
hold on
grid on
title('Reference yield');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
ylabel('Yield (h/d)');
datetick('x','mm/yy');
ylim([0,9]);
%
%Graphs of efficiencies with time
figure (3);
subplot(1,2,1);
plot(Agora3year.day,Agora3year.PR,'.')
hold on
grid on
title('Performance Ratio');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
ylabel('Performance Ratio');
datetick('x','mm/yy');
ylim([0,1]);
subplot(1,2,2);
plot(Agora3year.day,Agora3year.InvEff,'.g')
hold on
grid on
title('Inverter efficiency');
```

```
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
datetick('x','mm/yy');
ylabel('Inverter efficiency, \eta_{inv}');
ylim([0,1]);

figure(4);
plot(Agora3year.day,Agora3year.LC,'*r')
hold on
grid on
title('Capture Loss');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
datetick('x','mm/yy');
ylabel('Capture Loss,L_C (h/d)');
ylim([-inf,6]);

figure(5)
subplot(1,2,1)
plot(Agora3year.day,Agora3year.SysEff,'.g')
hold on
grid on
title('System efficiency');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
datetick('x','mm/yy');
ylabel('System efficiency, \eta_{sys}');
ylim([0,0.16]);
subplot(1,2,2)
plot(Agora3year.day,Agora3year.ArrayEff,'.g')
hold on
grid on
title('Array efficiency');
set(gca,'XTick',Agora3year.day(1):100:Agora3year.day(n));
```

```
xlim([Agora3year.day(1),Agora3year.day(n)]);
xlabel(xtag);
datetick('x','mm/yy');
ylabel('Array efficiency, \eta_{array}');
ylim([0,0.16]);
%%
%Graphs of affect of ambient and cell temperatures to Performance Ratio,
%Capture Loss, System efficiency
figure (6)
subplot(1,2,1);
plot(Agora3year.daily_avgCelltemp,Agora3year.SysEff,'.r')
hold on
grid on
title('System efficiency as a function of cell temperature');
xlabel('Cell temperature(^oC)');
ylabel('System efficiency, \eta_{sys}');
ylim([0,0.16]);
xlim([-30,40]);
subplot(1,2,2);
plot(Agora3year.daily_avgAmbtemp,Agora3year.SysEff,'.b')
hold on
grid on
title('System efficiency as a function of ambient temperature');
xlabel('Ambient temperature(^oC)');
ylabel('System efficiency, \eta_{sys}');
ylim([0,0.16]);
xlim([-30,40]);

figure (7)
subplot(1,2,1);
plot(Agora3year.daily_avgCelltemp,Agora3year.ArrayEff,'.r')
hold on
grid on
title('Array efficiency as a function of cell temperature');
xlabel('Cell temperature(^oC)');
ylabel('Array efficiency, \eta_{array}');
ylim([0,0.16]);
```

```
xlim([-30,40]);
subplot(1,2,2);
plot(Agora3year.daily_avgAmbtemp,Agora3year.ArrayEff,'.b')
hold on
grid on
title('Array efficiency as a function of ambient temperature');
xlabel('Ambient temperature(^oC)');
ylabel('Array efficiency, \eta_{array}');
ylim([0,0.16]);
xlim([-30,40]);


figure(8)
subplot(1,2,1);
plot(Agora3year.daily_avgCelltemp,Agora3year.PR,'.r')
hold on
grid on
title('Performance Ratio as a function of cell temperature: Agora');
xlabel('Cell temperature(^oC)');
ylabel('Performance Ratio');
ylim([0,1]);
xlim([-30,40]);
subplot(1,2,2);
plot(Agora3year.daily_avgAmbtemp,Agora3year.PR,'.b')
hold on
grid on
title('Performance Ratio as a function of ambient temperature');
xlabel('Ambient temperature(^oC)');
ylabel('Performance Ratio');
ylim([0,1]);
xlim([-30,40]);

%%
% Capture loss as a function of temperature
figure(9)
subplot(1,2,1);
plot(Agora3year.daily_avgCelltemp,Agora3year.LC,'.r')
```

```
hold on
grid on
title('Capture loss as a function of cell temperature');
xlabel('Cell temperature(^oC)');
ylabel('Capture loss (h/d)');
ylim([0,6]);
xlim([-30,40]);
subplot(1,2,2);
plot(Agora3year.daily_avgAmbtemp, Agora3year.LC, '.b')
hold on
grid on
title('Capture loss as a function of ambient temperature');
xlabel('Ambient temperature(^oC)');
ylabel('Capture loss (h/d)');
ylim([0,6]);
xlim([-30,40]);

figure(10)
subplot(1,2,1);
plot(Agora3year.daily_avgIrradiance , Agora3year.LC, '.r')
hold on
grid on
title('Capture loss as a function of irradiance');
ylabel('Capture loss (h/d)');
xlabel('Average Irradiance (W/m^2 d)');
ylim([0,6]);
xlim([0,600]);
subplot(1,2,2);
plot(Agora3year.daily_avgIrradiance , Agora3year.LS, '.b')
hold on
grid on
title('System loss as a function of irradiance');
ylabel('System loss (h/d)');
xlabel('Average Irradiance (W/m^2 d)');
ylim([0,6]);
xlim([0,600]);

```

```
figure (11)
plot(Agora3year.daily_avgIrradiance ,Agora3year.ArrayEff ,'.b')
hold on
grid on
title('Array efficiency as a function of average irradiance');
ylabel('Array efficiency \eta_{array}');
xlabel('Average Irradiance (W/m^2 d)');
ylim([0,0.15]);
xlim([0,600]);


figure (12)
x1 = Agora3year.day(1):Agora3year.day(n);
y1 = Agora3year.LC;
x2 = Agora3year.day(1):Agora3year.day(n);
y2 = Agora3year.daily_avgCelltemp;
hl1 = line(x1,y1,'Color','r');
ax1 = gca;
set(ax1,'XColor','r','YColor','r')
ax2 = axes('Position',get(ax1,'Position') ,...
           'XAxisLocation','top' ,...
           'YAxisLocation','right' ,...
           'Color','none' ,...
           'XColor','k','YColor','k');
 hl2 = line(x2,y2,'Color','k','Parent',ax2);
xlimits = get(ax1,'XLim');
ylimits = get(ax1,'YLim');
xinc = (xlimits(2)-xlimits(1))/5;
yinc = (ylimits(2)-ylimits(1))/5;
set(ax1,'XTick',[xlimits(1):xinc:xlimits(2)] ,...
        'YTick',[ylimits(1):yinc:ylimits(2)]);
set(get(ax1,'Ylabel'),'String','Capture loss(h/d)')
set(get(ax2,'Ylabel'),'String','Cell temperature (^oC)')
set(get(ax1,'Xlabel'),'String','Time (month/year)')
set(get(ax2,'Xlabel'),'String','Time (month/year)')
datetick(ax1,'x','mm/yy');
datetick(ax2,'x','mm/yy');
```

code/Matcode/AgoraGraphs.m

10-minute average data values were used to find the behaviour of array efficiency, PR, normalised array power at different cell temperature and irradiance levels.

```
%Written by Dereck Mutungi

% This program is used to derive the array efficiency for Agora and
% Saarijarvi systems. The normalised array will be used with the
% instantaneous irradiance. Taking 2012 as the base year, the PR,
% irradiance, array efficiency, inverter efficiency were calculated at 10
% minute intervals.
close all; clear all; clc;
%% Loading the required matrices
load AgoraFiltered.mat
load SjarviFiltered.mat
load ArrayEffIrrAllsystems.mat
load ArrayEffTempAllsystems.mat
load PRIrrAllSystems.mat
load NewNormAETemp.mat
load Agoratemp.mat
%% Extracting the temperature of the cell (module) for the reporting
    period
%% Selecting data in that period
p=0;
for j=1:k
    Agora1.val{j}(isnan(Agora1.val{j}))=0;
    % To ensure no NaN values are left
    % Condition to select data between 1/1/2010 and 31/12/2012
    if ((Agora1.time{j}(1,1)>=734139) && (Agora1.time{j}(1,1)<735235))
    p=p+1;
    Agora1.data{p}=Agora1.val{j};
        else
        continue
    end
end
%This condition is to check if the file has got more than two record
    %values.

for j2=1:p
```

```
    if  (length(Agora1.data{j2})~=2)
       Agora.celltemp{j2}=Agora1.data{j2};

    else
        Agora.celltemp{j2}(1) = (Agora1.data{j2}(1,1));

    end

end

%% Taking a base year of 2012 for comparison between both systems
% Agora system
q = 0; l=0; Irr(10)=0; Agora.Rated_power = 2600; G_ref = 1000;
for j=1:length(Agora.Timefiltered)

    Agora.datafiltered{j}(isnan(Agora.datafiltered{j}))=0;
 for x = 1 : length(Agora.datafiltered{j})
  % checks for cell size since some days have no data thus size mismatch
   if ((length(Agora.Timefiltered{j}))==(length(Agora.datafiltered{j})))
    %Converting the date into a vector
   [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Agora.Timefiltered{j}(x,1)
   );
         if (Y(x)==2012)
             q =q+1;
             Irr(q) = Agora.datafiltered{j}(x,1);
             AP(q)=(Agora.datafiltered{j}(x,2)*...
                 Agora.datafiltered{j}(x,7))/1000;
               temp(q)=Agora.celltemp{j}(x,1);
             GP(q) = Agora.datafiltered{j}(x,6);
             if (q==10)
                 l=l+1;
                 Agora.irradiance(l) = mean(Irr);
                  Agora.ArrayPower(l)=mean(AP);
                  Agora.modtemp(l)=mean(temp);
                   Agora.GridPower(l)=mean(GP);
                 Irr =0; q=0; AP =0; temp =0; GP =0;
```

```
            end



        else
        continue;
        end
    end

 end
if ((q>1)&&(q<10))
     l=l+1;
    Agora.irradiance(l) = mean(Irr);
    Agora.ArrayPower(l)=mean(AP);
    Agora.modtemp(l)=mean(temp);
    Agora.GridPower(l)=mean(GP);
     Irr =0; q=0; AP =0; temp =0; GP =0;
     end
    q=0;
end
clear D H M MN S Y l k p q j2 j temp Irr AP x
 %The area of the array is 19.83 sq m
 Agora.Solarpower = Agora.irradiance.*19.83; %Wh
 Agora.arrayEff = Agora.ArrayPower./Agora.Solarpower;
AgoraRefYield = (Agora.irradiance./G_ref)';
AgoraArrayYield = (Agora.ArrayPower./Agora.Rated_power)';
AgoraFinalYield = (Agora.GridPower./Agora.Rated_power)';
AgoraInvEff = (Agora.GridPower./Agora.ArrayPower)';
AgoraPR = AgoraFinalYield./AgoraRefYield;
AgoraInvEff(isnan(AgoraInvEff))=0;
AgoraPR(isnan(AgoraPR))=0;
%%
%Saarijarvi system
l=0; Sjarvi.Rated_power1=5130; Sjarvi.Rated_power2=1140;

for j=1:length(Sjarvi.Yearfiltered1)
     for x = 1 : length(Sjarvi.datafiltered1{j})
```

```
        [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Sjarvi.Yearfiltered1{j}(x,1));
         if (Y(x)==2012)
        l=l+1;
  % Extracting the power values from the raw data
        Sjarvi.daily_ArrayPower1(l)= Sjarvi.datafiltered1{j}(x,3)*...
           Sjarvi.datafiltered1{j}(x,4);
        Sjarvi.daily_ArrayPower2(l)= Sjarvi.datafiltered1{j}(x,5)*...
           Sjarvi.datafiltered1{j}(x,6);
         Sjarvi.daily_GridPower1(l)= Sjarvi.datafiltered1{j}(x,1);
        Sjarvi.daily_GridPower2(l)= Sjarvi.datafiltered1{j}(x,2);
        Sjarvi.irradiance(l)= Sjarvi.datafiltered1{j}(x,9);
        Sjarvi.modtemp(l) = Sjarvi.datafiltered1{j}(x,7);
 %The area of the sub-array  1 is 31.05 sq m  while the other is 6.9 sq m
         Sjarvi.daily_Solarpower1(l) = Sjarvi.irradiance(l).*31.05; %Wh
         Sjarvi.daily_Solarpower2(l) = Sjarvi.irradiance(l).*6.9; %Wh
         Sjarvi.arrayEff1(l) = Sjarvi.daily_ArrayPower1(l)./...
           Sjarvi.daily_Solarpower1(l);
         Sjarvi.arrayEff2(l) = Sjarvi.daily_ArrayPower2(l)./...
           Sjarvi.daily_Solarpower2(l);
         end
    end
end

clear D H M MN S Y  j l x
%Calculating the performance indices
SjarviRefYield = (Sjarvi.irradiance./G_ref)';
SjarviArrayYield1 = (Sjarvi.daily_ArrayPower1./Sjarvi.Rated_power1)';
SjarviArrayYield2 = (Sjarvi.daily_ArrayPower2./Sjarvi.Rated_power2)';
SjarviFinalYield1 = (Sjarvi.daily_GridPower1./Sjarvi.Rated_power1)';
SjarviFinalYield2 = (Sjarvi.daily_GridPower2./Sjarvi.Rated_power2)';
SjarviInvEff1 = (Sjarvi.daily_GridPower1./Sjarvi.daily_ArrayPower1)';
SjarviInvEff2 = (Sjarvi.daily_GridPower2./Sjarvi.daily_ArrayPower2)';
SjarviPR1 = (SjarviFinalYield1./SjarviRefYield);
SjarviPR2 = (SjarviFinalYield2./SjarviRefYield);
SjarviInvEff1(isnan(SjarviInvEff1))=0;
```

```
SjarviInvEff2(isnan(SjarviInvEff2))=0;
SjarviInvEff1(isinf(SjarviInvEff1))=0;
SjarviInvEff2(isinf(SjarviInvEff2))=0;
SjarviPR1(isnan(SjarviPR1))=0;
SjarviPR2(isnan(SjarviPR2))=0;

%Normalised array efficiency graph using the manufacturers values of the
%modules NAPs = 0.131 and HIT = 0.165
Agora.normArrayEff = Agora.arrayEff./0.131;
Sjarvi.normArrayEff1 = Sjarvi.arrayEff1./0.165;
Sjarvi.normArrayEff2 = Sjarvi.arrayEff2./0.165;


% Declaring variables for use in the curve fitting tool
AgoranormAE = Agora.normArrayEff';
AgoraAE = Agora.arrayEff';
AgoraIrr = Agora.irradiance';
SjarvinormAE1 = Sjarvi.normArrayEff1';
SjarviAE1 = Sjarvi.arrayEff1';
SjarvinormAE2 = Sjarvi.normArrayEff2';
SjarviAE2 = Sjarvi.arrayEff2';
SjarviIrr = Sjarvi.irradiance';
AgoraCelltemp = Agora.modtemp';
SjarviCelltemp = Sjarvi.modtemp';
%AgoraSmoothAEL = AgoraAELoessSmooth.y;
AgoraSmoothAERL = AgoraAERLoessSmooth.y;
Sjarvi1SmoothAEL = SjarviAE1LoessSmooth.y;
Sjarvi1SmoothAERL = SjarviAE1RLoessSmooth.y;
Sjarvi2SmoothAEL = SjarviAE2LoessSmooth.y;
Sjarvi2SmoothAERL = SjarviAE2RLoessSmooth.y;
AgoraSmoothAE_T =AgoraAETempSmooth.y;
SjarviSmoothAE_T1 = SjarviAE1TempSmooth.y;
SjarviSmoothAE_T2 = SjarviAE2TempSmooth.y;
AgoraPRSmooth =PRIrrAgora.y;
Sjarvi1PRSmooth=PRIrrSjarvi1.y;
Sjarvi2PRSmooth=PRIrrSjarvi2.y;
AgoraNormSmoothAE_T=NormAgoraAETemp.y;
```

```
Sjarvi1NormSmoothAE_T =NormSjarviAETemp1.y;
Sjarvi2NormSmoothAE_T =NormSjarviAETemp2.y;
%% Plots
figure (1)
plot(AgoraIrr,AgoranormAE,'.b',SjarviIrr,SjarvinormAE1,'.g',...
    SjarviIrr, SjarvinormAE2,'.r' )
hold on
grid on
title('Comparison of normalized array efficiencies and irradiance');
ylabel('Normalized array efficiency');
xlabel('Irradiance (W/m^2)');
ylim([0,1.2]);
xlim([0,1400]);
legend('Agora','Sjarvi 1','Sjarvi 2');
```

code/Matcode/NormalisedArrayEffFull.m

```
%Written by Dereck Mutungi

% This program is used to plot histograms of DC and AC power for the
    Agora
% and Saarijarvi systems. Also used to plot the inverter efficiency at
% partial loads. In addition, the irradiance histogram was plotted.
%%
clc; clear;close all;
%% Load the Agora matrix. See Filtered_by_G_I.m
% This matrix also contains data filtered by irradiance and current. Plus
% original data.
load AgoraFiltered.mat
load SjarviFiltered.mat
load Smoothed_InvEff_PartialLoad.mat
%% Selecting the date and adding the power and getting the energy per day
%% in Wh

r1 = 5380; % rating of inverter input power in W
r2 = 1610; % rating of inverter input power in W
q=1; l=0;
```

```
for j=1:length(Sjarvi.Yearfiltered1)
    for x = 1 : length(Sjarvi.datafiltered1{j})
        l=l+1;
  % Extracting the power values from the raw data
      Sjarvi.daily_GridPower1(l) = Sjarvi.datafiltered1{j}(x,1);
       Sjarvi.daily_GridPower2(l) = Sjarvi.datafiltered1{j}(x,2);
       Sjarvi.daily_ArrayPower1(l)= Sjarvi.datafiltered1{j}(x,3)*...
           Sjarvi.datafiltered1{j}(x,4);
       Sjarvi.daily_ArrayPower2(l)= Sjarvi.datafiltered1{j}(x,5)*...
           Sjarvi.datafiltered1{j}(x,6);
       Sjarvi.irradiance(l)= Sjarvi.datafiltered1{j}(x,9);
       SPartialLoad1{j}(x) = Sjarvi.daily_ArrayPower1(l)/r1;
       SInvEff1{j}(x) = Sjarvi.daily_GridPower1(l) ./...
           Sjarvi.daily_ArrayPower1(l);
       SPartialLoad2{j}(x) = Sjarvi.daily_ArrayPower2(l)/r2;
       SInvEff2{j}(x) = Sjarvi.daily_GridPower2(l)./...
           Sjarvi.daily_ArrayPower2(l);
    end
end
%%
%%
l=0;
r3 = 2700; % W

for j=1:length(Agora.Timefiltered)

    Agora.datafiltered{j}(isnan(Agora.datafiltered{j}))=0;
 for x = 1 : length(Agora.datafiltered{j})
 % checks for cell size since some days have no data thus size mismatch
   if ((length(Agora.Timefiltered{j}))==(length(Agora.datafiltered{j})))
    %Converting the date into a vector
      [Y(x) M(x) D(x) H(x) MN(x) S(x)] =  datevec(Agora.Timefiltered{j}(x
    ,1));
            if (Y(x)==2010)
                l =l+1;
                Agora.daily_GridPower(l) = Agora.datafiltered{j}(x,6);
                Agora.daily_ArrayPower(l)= (Agora.datafiltered{j}(x,2)*...
```

```
            Agora.datafiltered{j}(x,7))/1000;
        Agora.irradiance(l) = Agora.datafiltered{j}(x,1);
            APartialLoad(l) = Agora.daily_ArrayPower(l)/r3;
        AInvEff(l) = Agora.daily_GridPower(l) /Agora.daily_ArrayPower(l);
          end
    else
        continue;
     end
end

end
clear D H M MN S Y  j l x




% Ratio of the array energy (input energy to the inverter) and the rated
% inverter energy for the same period
PartialLoad = APartialLoad;
InvEff = AInvEff;



% Ratio of the array energy (input energy to the inverter) and the rated
% inverter energy for the same period
PartialLoad1 = cell2mat(SPartialLoad1);
InvEff1 = cell2mat(SInvEff1);
PartialLoad2 = cell2mat(SPartialLoad2);
InvEff2 = cell2mat(SInvEff2);


figure (1)
plot(PartialLoad, InvEff,'.b',PartialLoad1, InvEff1,'.g',PartialLoad2,...
    InvEff2,'.r' )
hold on
grid on
title('Comparison of the inverter efficiencies at partial load');
```

```
ylabel('Inverter efficiency, \eta _{inv}');
xlabel('Partial load');
ylim([0,1.2]);
xlim([0,1]);
legend('Agora','Sjarvi 1','Sjarvi 2');
%%
figure(2)
x1 =Sjarvi.daily_GridPower1./5130;
y1 = Sjarvi.daily_ArrayPower1./5130;
x2 = Sjarvi.daily_GridPower2./1140;
y2 = Sjarvi.daily_ArrayPower2./1140;
topEdge = 1; % defining the  upper limit of the irradiance
botEdge = 0;    % defining the lower limit of the irradiance
numBins =10;    % defining the number of bins

binEdges =  linspace(botEdge, topEdge, numBins+1);

[h1, whichBin1] = histc(y1,binEdges);  %h shows the count per bin
[h2, whichBin2] = histc(x1,binEdges);
%for i = 1 : numBins+1
 %   flagBinMembers = (whichBin == i);
  %  binMembers = y(flagBinMembers);
  %  binSum(i)= sum(binMembers);
%end;
relativefreq1 = h1/sum(h1);
relativefreq2 = h2/sum(h2);
bar(binEdges, [relativefreq1;relativefreq2]',1)
grid on
hold on
ylabel('Fraction of total energy produced');
xlabel('Normalised Power');
title('Histogram of normalised power: Saarijarvi subsystem 1');
%%
figure(3)
[h3, whichBin1] = histc(y2,binEdges);  %h shows the count per bin
[h4, whichBin2] = histc(x2,binEdges);
%for i = 1 : numBins+1
```

```
 %   flagBinMembers = (whichBin == i);
  %  binMembers = y(flagBinMembers);
  %  binSum(i)= sum(binMembers);
%end;
relativefreq1 = h3/sum(h3);
relativefreq2 = h4/sum(h4);
bar(binEdges, [relativefreq1;relativefreq2]',1)
grid on
hold on
ylabel('Fraction of total energy produced');
xlabel('Normalised Power');
title('Histogram of normalised power: Saarijarvi subsystem 2');
%%
figure(4)
x1 =Agora.daily_GridPower./2600;
x2 = Agora.daily_ArrayPower./2600;
%y = ;
topEdge = 1; % defining the  upper limit of the irradiance
botEdge = 0;    % defining the lower limit of the irradiance
numBins =10;    % defining the number of bins

binEdges =  linspace(botEdge, topEdge, numBins+1);

[h1, whichBin1] = histc(x1,binEdges);  %h shows the count per bin
[h2, whichBin2] = histc(x2,binEdges);
%for i = 1 : numBins+1
 %   flagBinMembers = (whichBin == i);
  %  binMembers = y(flagBinMembers);
  %  binSum(i)= sum(binMembers);
%end;
relativefreq1 = h1/sum(h1);
relativefreq2 = h2/sum(h2);
bar(binEdges, [relativefreq2;relativefreq1]',1)
grid on
hold on
ylabel('Fraction of total energy produced');
xlabel('Normalised Power');
```

```
title('Histogram of normalised power distribution: Agora');

%%
figure(5)
x3 =Sjarvi.irradiance./1000;

topEdge = 1.2; % defining the  upper limit of the irradiance
botEdge = 0;     % defining the lower limit of the irradiance
numBins =12;     % defining the number of bins

binEdges =  linspace(botEdge, topEdge, numBins+1);

[h5, whichBin1] = histc(x3,binEdges);  %h shows the count per bin

%for i = 1 : numBins+1
 %    flagBinMembers = (whichBin == i);
  %   binMembers = y(flagBinMembers);
  %   binSum(i)= sum(binMembers);
%end;
relativefreq5 = h5./sum(h5);
bar(binEdges, relativefreq5 ,1)
grid on
hold on
ylabel('Fraction of total solar irradiation');
xlabel('{G_I }/{G_{ref}}');
title('Histogram of normalised distribution of in plane irradiation:
    Saarijarvi');
%%
figure(6)
x3 =Agora.irradiance./1000;

topEdge = 1.2; % defining the  upper limit of the irradiance
botEdge = 0;     % defining the lower limit of the irradiance
numBins =12;     % defining the number of bins

binEdges =  linspace(botEdge, topEdge, numBins+1);

```

```
[h3, whichBin1] = histc(x3,binEdges);  %h shows the count per bin

%for i = 1 : numBins+1
 %    flagBinMembers = (whichBin == i);
  %   binMembers = y(flagBinMembers);
  %   binSum(i)= sum(binMembers);
%end;
relativefreq3 = h3./sum(h3);
bar(binEdges, relativefreq3 ,1)
grid on
hold on
ylabel('Fraction of total solar irradiation');
xlabel('{G_I }/{G_{ref}}');
title('Histogram of normalised distribution of in plane irradiation:
    Agora');
%%
%%Graphs of the smoothed out values of the inverter efficiency and the
%%partial load. The method used to smooth out the data was  Lowess
figure (7)
AP = AgoraSmootheInvPart.x;
AIE = AgoraSmootheInvPart.y;
SP1 = Sjarvi1smoothedInvPArt.x;
SIE1 = Sjarvi1smoothedInvPArt.y;
SP2 = Sjarvi2smoothedInvPart.x;
SIE2 = Sjarvi2smoothedInvPart.y;
plot(AP,AIE,'.b',SP1,SIE1,'.g',SP2, SIE2,'.r' )
hold on
grid on
title('Comparison of the inverter efficiencies at partial load (smoothed
    values)');
ylabel('Inverter efficiency, \eta _{inv}');
xlabel('Partial load');
ylim([0,1.2]);
xlim([0,1]);
legend('Agora','Sjarvi 1','Sjarvi 2');
```

code/Matcode/HistogramsInvSizing.m

# B R code

## B.1 Agora R code

```
#This program was written by D mutungi
#It is used to check for seasonality of Agora data and calculate the
#ARIMA values
library(R.matlab)
data <- readMat(file.choose())
str(data)
summary(data)

#Putting the performance ratio into a time series
Prtimeseries <- ts(data$weeklyPR, frequency=52, start=c(2010,1))
plot.ts(Prtimeseries, type='o')
PRcomponents <- decompose(Prtimeseries, type=c("multiplicative"))
plot(PRcomponents)

#Loading the forecast package and finding ARIMA model
library(forecast)
auto.arima(Prtimeseries)

#Assigning the PR raw data with corresponding ARIMA model
fit<-arima(Prtimeseries, order=c(2,0,2),
seasonal=list(order=c(0,0,1), period=52),include.mean=FALSE)



plot(Prtimeseries,col="red", type="o")
lines(fitted(fit),col="blue", type="o", pch=22, lty=2)


#Writing the raw and ARIMA values to a mat file to be used in other
    software
pr <-Prtimeseries
pr_arima<-fitted(fit)

```

```
RawPRsys <- as.matrix(pr)
ArimaPRsys <- as.matrix(pr_arima)

AgoraPRwk <- paste(tempfile(), ".mat", sep="")

writeMat("AgoraPRwk.mat",RawPRsys=RawPRsys, ArimaPRsys=ArimaPRsys)
data1 <- readMat("AgoraPRwk.mat")
print(data1)

unlink(AgoraPRwk)
```

code/RCode/PRseasonalityAgora.R

```
#This program is written by D mutungi
#It is used to check for seasonality of array efficiency Agora data and
#to calculate the ARIMA values of the same.
library(R.matlab)
data <- readMat(file.choose())
str(data)
summary(data)

#Replacing NaN with zero values. This is done to avoid unnecessary errors
    when exporting the files to other software.
data$weeklyArrayEff <- replace(data$weeklyArrayEff,data$weeklyArrayEff ==
    0, NaN)

#Setting the array efficiency as a time series
ArrayEffts <- ts(data$weeklyArrayEff, frequency=52, start=c(2010,1))

#Plot for visual inspection of the data.
plot.ts(ArrayEffts, type='o')

#Loading the forecast package and finding ARIMA model
library(forecast)
auto.arima(ArrayEffts)

#Using the ARIMA models to calculate a new dataset.
fit<-arima(ArrayEffts, order=c(1,0,0),
```

```r
seasonal=list(order=c(1,0,1), period=52),include.mean=FALSE)

#Plotting the two data sets for a visual comparison.
plot(ArrayEffts,col="red", type="o")
lines(fitted(fit),col="blue", type="o", pch=22, lty=2)


#Saving the data as a mat file.
AE <-ArrayEffts
AE_arima<-fitted(fit)


RawAEsys <- as.matrix(AE)
ArimaAEsys <- as.matrix(AE_arima)


AgoraAEwk <- paste(tempfile(), ".mat", sep="")

writeMat("AgoraAEwk.mat",RawAEsys=RawAEsys, ArimaAEsys=ArimaAEsys)
data1 <- readMat("AgoraAEwk.mat")
print(data1)

unlink(AgoraAEwk)
```

code/RCode/AgoraAEseasonal.R

### B.2 Saarijärvi R code

```r
#This program is written by D mutungi
#It is used to check for seasonality of PR Sjarvi data and calculate the
#ARIMA values
library(R.matlab)
data <- readMat(file.choose())
str(data)
summary(data)
#Replacing NaN with zero values. This is done to avoid unnecessary errors
    when exporting the files to other software.
```

```r
data$weeklySjarviPR1 <- replace(data$weeklySjarviPR1,data$weeklySjarviPR1
    == 0, NaN)
data$weeklySjarviPR2 <- replace(data$weeklySjarviPR2,data$weeklySjarviPR2
    == 0, NaN)

#Setting the performance ratio as a time series
Prtimeseries <- ts(data$weeklySjarviPR1, frequency=52, start=c(2006,1))
Prtimeseries1 <- ts(data$weeklySjarviPR2, frequency=52, start=c(2006,1))

#Plot for visual inspection of the data.
plot.ts(Prtimeseries, type='o')
plot.ts(Prtimeseries1, type='o')
PRcomponents <- decompose(Prtimeseries, type=c("multiplicative"))
plot(PRcomponents)

#Loading the forecast package and finding ARIMA model
library(forecast)
auto.arima(Prtimeseries)
auto.arima(Prtimeseries1)

#Using the ARIMA models to calculate a new dataset.
fit<-arima(Prtimeseries, order=c(5,0,0),
seasonal=list(order=c(1,0,0), period=52),include.mean=FALSE)
fit1<-arima(Prtimeseries1, order=c(3,0,0),include.mean=FALSE)

#Plotting the two data sets for a visual comparison.
plot(Prtimeseries,col="red", type="o")
lines(fitted(fit),col="blue", type="o", pch=22, lty=2)
plot(Prtimeseries1,col="red", type="o")
lines(fitted(fit1),col="blue", type="o", pch=22, lty=2)

#Saving the data as a mat file.
pr1 <-Prtimeseries
pr1_arima<-fitted(fit)
pr2 <-Prtimeseries1
pr2_arima<-fitted(fit1)
```

```
RawPRsys1 <- as.matrix(pr1)
ArimaPRsys1 <- as.matrix(pr1_arima)
RawPRsys2 <- as.matrix(pr2)
ArimaPRsys2 <- as.matrix(pr2_arima)


SjarviPRwk <- paste(tempfile(), ".mat", sep="")

writeMat("SjarviPRwk.mat",RawPRsys1=RawPRsys1, ArimaPRsys1=ArimaPRsys1,
RawPRsys2=RawPRsys2, ArimaPRsys2=ArimaPRsys2)
data1 <- readMat("SjarviPRwk.mat")
print(data1)

unlink(SjarviPRwk)










pp<-1:length(nw_sub)
res<- lm( (nw_sub)~pp)
plot(pp,(nw_sub))
abline(res)
nwa <- ts(nw_sub, frequency=6, start=c(2006,1))
auto.arima(nwa)
fit<-arima(nwa, order=c(0,1,1),
seasonal=list(order=c(1,0,0), period=6),include.mean=FALSE)

plot(nwa,col="red", type ="o")
lines(fitted(fit),col="blue")
```

```
plot(decompose(nwa))
```

code/RCode/PRseasonality.R

```
#This program is written by D mutungi
#It is used to check for seasonality of Array effiency Sjarvi data
#and calculate the ARIMA values of the same.
library(R.matlab)
data <- readMat(file.choose())
str(data)
summary(data)

#Replacing NaN with zero values. This is done to avoid unnecessary errors
     when exporting the files to other software.
data$weeklySjarviArrayEff1 <- replace(data$weeklySjarviArrayEff1,data$
    weeklySjarviArrayEff1 == 0, NaN)
data$weeklySjarviArrayEff2 <- replace(data$weeklySjarviArrayEff2,data$
    weeklySjarviArrayEff2 == 0, NaN)

 #Setting the array efficiency as a time series
ArrayEff1ts <- ts(data$weeklySjarviArrayEff1, frequency=52, start=c
    (2006,1))
ArrayEff2ts <- ts(data$weeklySjarviArrayEff2, frequency=52, start=c
    (2006,1))

#Plot for visual inspection of the data.
plot.ts(ArrayEff1ts, type='o')
plot.ts(ArrayEff2ts, type='o')

#Loading the forecast package and finding ARIMA model
library(forecast)
auto.arima(ArrayEff1ts)
auto.arima(ArrayEff2ts)

#Using the ARIMA models to calculate a new dataset.
fit<-arima(ArrayEff1ts, order=c(1,0,0),
seasonal=list(order=c(1,0,0), period=52),include.mean=FALSE)
fit1<-arima(ArrayEff2ts, order=c(3,0,0),include.mean=FALSE)
```

```
#Plotting the two data sets for a visual comparison.
plot(ArrayEff1ts,col="red", type="o")
lines(fitted(fit),col="blue", type="o", pch=22, lty=2)
plot(ArrayEff2ts,col="red", type="o")
lines(fitted(fit1),col="blue", type="o", pch=22, lty=2)

 #Saving the data as a mat file.
AE1 <-ArrayEff1ts
AE1_arima<-fitted(fit)
AE2 <-ArrayEff2ts
AE2_arima<-fitted(fit1)


RawAEsys1 <- as.matrix(AE1)
ArimaAEsys1 <- as.matrix(AE1_arima)
RawAEsys2 <- as.matrix(AE2)
ArimaAEsys2 <- as.matrix(AE2_arima)


SjarviAEwk <- paste(tempfile(), ".mat", sep="")

writeMat("SjarviAEwk.mat",RawAEsys1=RawAEsys1, ArimaAEsys1=ArimaAEsys1,
RawAEsys2=RawAEsys2, ArimaAEsys2=ArimaAEsys2)
data1 <- readMat("SjarviAEwk.mat")
print(data1)

unlink(SjarviAEwk)
```

code/RCode/SjarviAEseasonality.R

# C Graphical results

## C.1 Time dependent

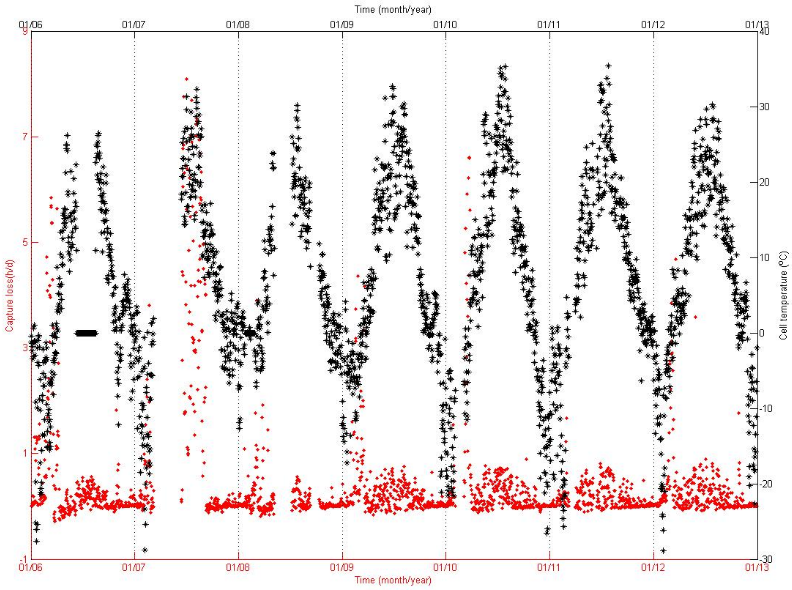
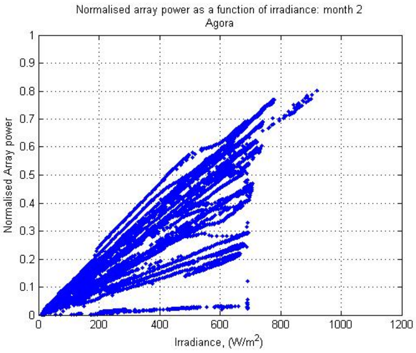


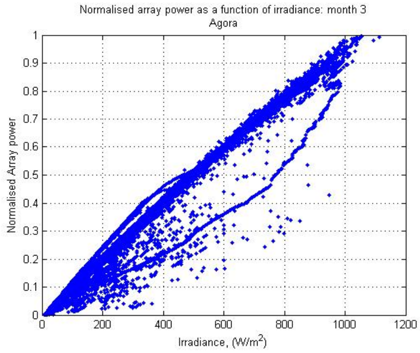

Figure 44: Capture loss as a function of time; Saarijärvi system

## C.2 Cut-in irradiance

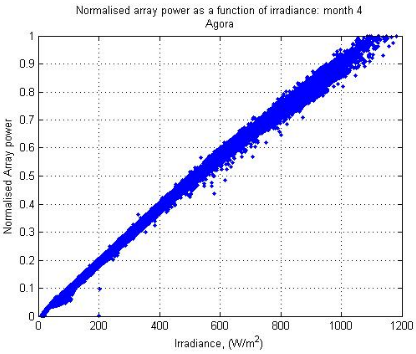
The month of January had no detectable irradiance hence there is no plot for the Agora site. The month number is equivalent to the month of the year. Below are the scatter plots for Agora:


Normalised array power as a function of irradiance: month 2
Agora
Normalised Array power
Irradiance, (W/m$^2$)

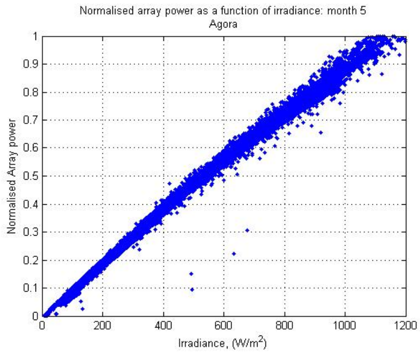
(a)


Normalised array power as a function of irradiance: month 3
Agora
Normalised Array power
Irradiance, (W/m$^2$)

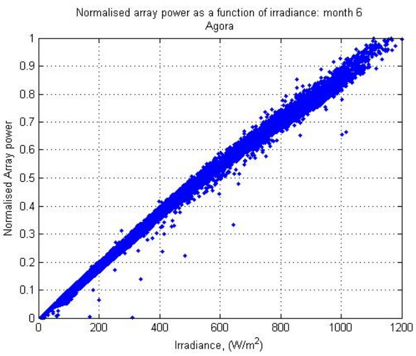
(b)


Normalised array power as a function of irradiance: month 4
Agora
Normalised Array power
Irradiance, (W/m$^2$)

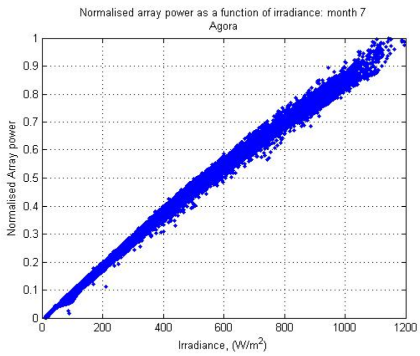
(c)


Normalised array power as a function of irradiance: month 5
Agora
Normalised Array power
Irradiance, (W/m$^2$)

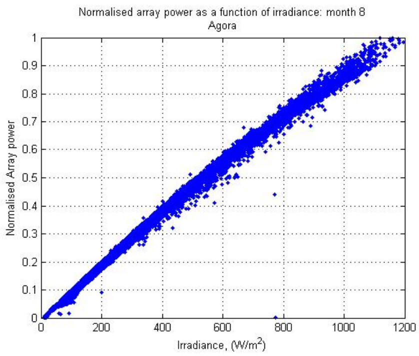
(d)


Normalised array power as a function of irradiance: month 6
Agora
Normalised Array power
Irradiance, (W/m$^2$)

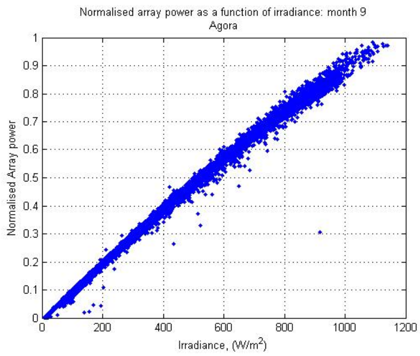
(e)


Normalised array power as a function of irradiance: month 7
Agora
Normalised Array power
Irradiance, (W/m$^2$)

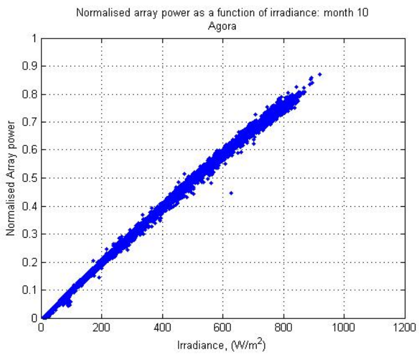
(f)


Normalised array power as a function of irradiance: month 8
Agora
Normalised Array power
Irradiance, (W/m$^2$)

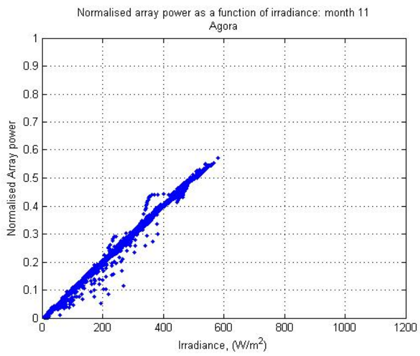
(g)


Normalised array power as a function of irradiance: month 9
Agora
Normalised Array power
Irradiance, (W/m$^2$)

(h)


Normalised array power as a function of irradiance: month 10
Agora
Normalised Array power
Irradiance, (W/m$^2$)

(i)


Normalised array power as a function of irradiance: month 11
Agora
Normalised Array power
Irradiance, (W/m$^2$)

(j)

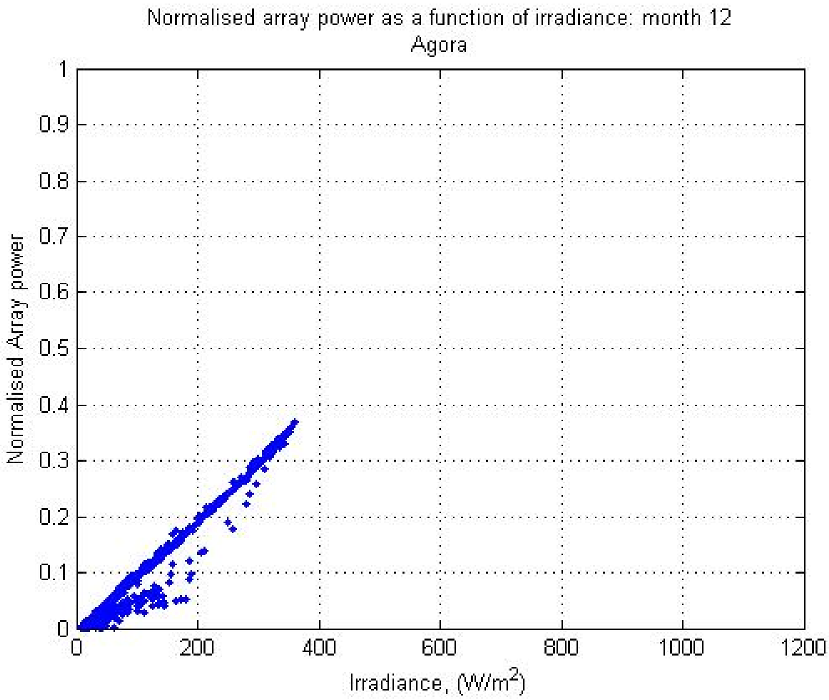
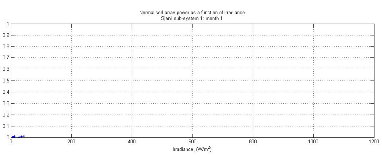
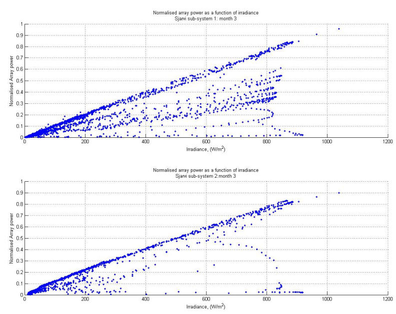


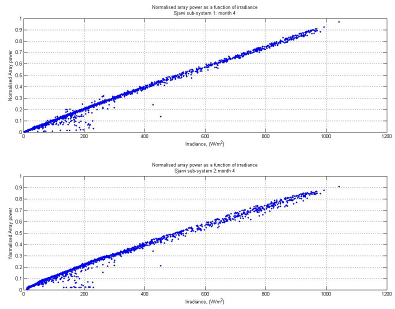

Figure 45

Scatter plots of normalised array power to irradiance for Saarijärvi system. The month of February had no detected irradiance or array power output.

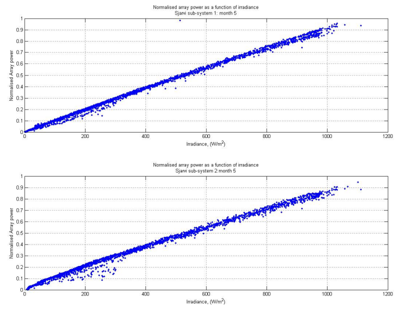


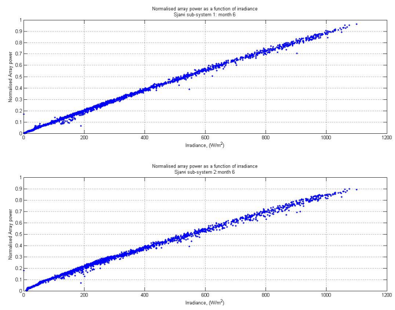

(a) (b)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 4
Normalised Array power
Irradiance, (W/m2)
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 4
Normalised Array power
Irradiance, (W/m2)

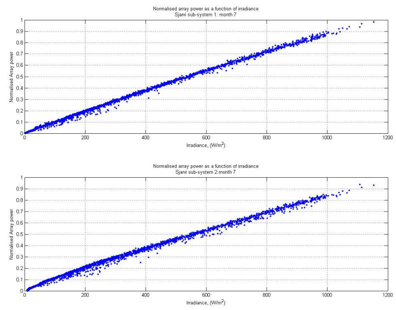
(c)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 5
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 5

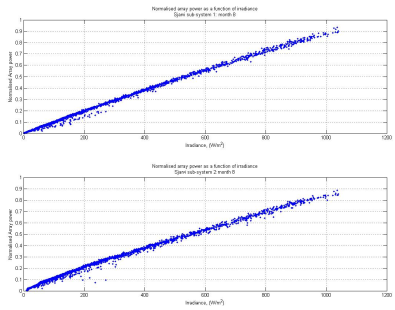
(d)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 6
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 6

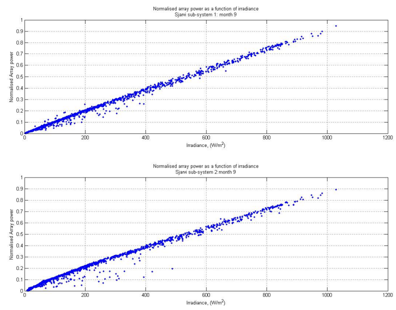
(e)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 7
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 7

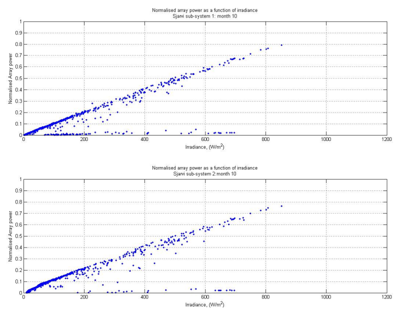
(f)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 8
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 8

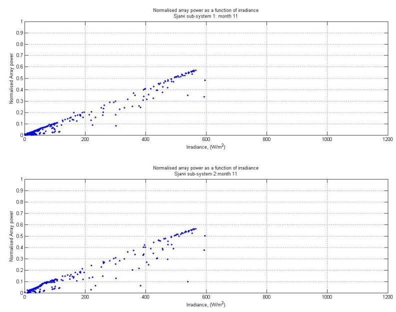
(g)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 9
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 9

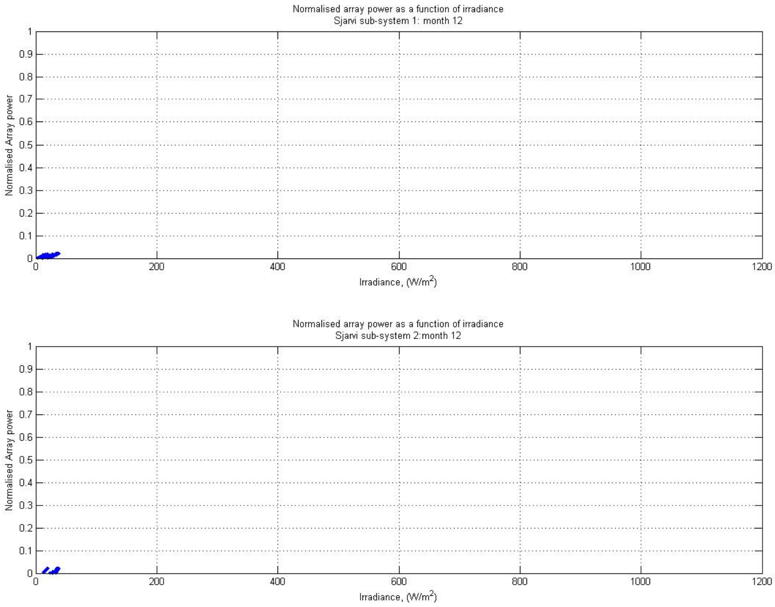
(h)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 10
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 10

(i)


Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 11
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 11

(j)



Normalised array power as a function of irradiance
Sjarvi sub-system 1: month 12
Normalised Array power
Irradiance, (W/m2)
Normalised array power as a function of irradiance
Sjarvi sub-system 2:month 12
Normalised Array power
Irradiance, (W/m2)


Figure 46