CHALMERS
UNIVERSITY OF TECHNOLOGY

Measurement and Characterisation of Solar Energy Availability for Low-Power Energy Harvesting Devices

Master of Science Thesis

Rani El-Maalouf

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis EX061/2016

Measurement and Characterisation of Solar Energy Availability for Low-Power Energy Harvesting Devices

Rani El-Maalouf

Department of Signals and Systems
Division of Signal Processing
Chalmers University of Technology
Gothenburg, Sweden 2016

Measurement and Characterisation of
Solar Energy Availability for Low-Power
Energy Harvesting Devices
Rani El-Maalouf

© Rani El-Maalouf, 2016.

Supervisor: Ayca Ozcelikkale, Department of Signals and Systems
Examiner: Tomas Mckelvey, Department of Signals and Systems

Master's Thesis EX061/2016
Department of Signals and Systems
Division of Signal Processing
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Abstract

Many of today's smaller devices can be liberated from their dependence of a constant power source by solar power, allowing for a more efficient use of current energy resources. The aim of this thesis is to measure the available light energy in an area and use this data for characterisation and prediction. This allows for better insight into what devices can be powered by solar energy, and the requirements for solar panel and battery.

For measurements, a device was constructed by connecting light sensors to a micro-controller. The micro-controller would then process the signals received from the light sensors and store this data on a micro SD card. Measurements were taken at various offices in the Department of Signals and Systems in Chalmers University of Technology.

Once the measurements were taken, they were used to obtain a profile for the character of sunlight. Each day contained a total of 48 time-slots, where each time-slot represents the mean value of the past half hour. By calculating the mean value for all days in each time-slot, a profile was constructed that gave a reasonable estimate for the character of sunlight. It was concluded that, while the mean gave an illustration as to how the character of available light energy typically changed during a day, values representing how the light varied for each time-slot were also necessary.

For determining the requirements of low-power devices, variations of solar panel sizes, intervals for active periods and initial battery charges were considered. It was concluded that the estimated available energy provided sufficient energy for indefinite use of a low-power wireless sensor during the month of May. This had a strong dependence on the size of the solar panel and the frequency of active periods chosen for the Machine-to-Machine (M2M) model. It was also established that an initial charge of the battery was required in order to avoid initial down-time.

Four algorithms, Exponentially Weighted Moving Average (EWMA), Weather Conditioned Moving Average (WCMA), Pro-Energy and a Weighted Min-Max Profile algorithm, were chosen for prediction. Min-Max was a proposed algorithm which was inspired by Pro-Energy, and was designed to require less memory and computational resources. All four algorithms provided the ability to perform a prediction for the next time-slot, but only Pro-Energy and Min-Max allowed for predictions of larger prediction horizons. It was concluded that the Min-Max algorithm provided most benefits when constraints were present. However, it was also determined that this algorithm was not preferred when larger prediction horizons, with highly varying weather conditions, were present.

Keywords: characterisation, low-power, machine-to-machine, measurement, solar irradiation, available, light

Acknowledgements

I would like to express my immense gratitude to my supervisor, Ayca Ozcelikkale, for her invaluable support, guidance and feedback throughout the work of this thesis. I would also like to thank Maryam Fatemi, Xinlin Zhang and Nikolaos Kolomvakis for allowing me to use their offices when collecting measurement data, and my examiner, Tomas McKelvey, for his patience and understanding.

Rani El-Maalouf, Gothenburg, June 2016

Contents

List of Figures

List of Tables

1 Introduction

The Intenet of Things (IoT), as described by Kevin Ashton, imagines a futuristic world with seamlessly connected devices [2]. The idea behind the IoT is to have a large amount of devices connected and exchanging data with one another, while allowing for open access to the shared information [3]. If this idea is implemented on a large scale, such as an entire city, it would allow for a city to become "smart". For example, in a smart city, the information from the sensors of other vehicles inside the city is shared [4], allowing the detection of movements that cannot be detected by the sensors of an individual vehicle. This would in turn help improve and make the implementation of autonomous driving easier and safer. Another example is a smart city using a Wireless Sensor Network (WSN), consisting of light and movement measurements. This would allow for adaptive lighting of the streets, based on movements in the streets when it is dark.

The possibilities for a smart city are immense, when considering how much can be done by having all electronic devices connected to one big network. In order to make the implementation of a smart city a reality, it is important to make efficient use of the available resources. Powering a WSN with light energy shows potential by extending the battery life time [5], [6]. However, having sensors that independently harvest all of their required energy through light would make for an independent WSN, allowing a vast number of sensors to be placed in any location that meet the required conditions.

Determining the available energy can be achieved by conducting measurements in an area. Light measurement devices can be costly [7], and devices that can both measure and log irradiance measurements even more so [8], [9]. Budget-friendly light measurement loggers do of course exist [10], such a device may however require additional purchases in the form of hardware and software in order to function as intended.

The collected data must also be analysed in order to determine whether the available energy is sufficient. Considering the fact that no energy from solar irradiance can be harvested during the evening, it is important to be able to decide the conditions for when the device is expected to function. An algorithm for predicting the available energy is required if the device needs to schedule its transmissions in order to avoid any downtime. This part plays a crucial role in the WSN, since the features of a smart city depend on having the necessary data available at the required moment. A prediction algorithm must not lead to a large increase in the consumption of energy, since that would ultimately not improve the battery life of the device.

Common prediction algorithms used, such as the Exponentially Weighted Moving Average (EWMA) [11] and the Weather Conditioned Moving Average (WCMA) [12], show simple prediction algorithms that would allow for an estimate as to whether the available energy in the future is sufficient for the next time-slot or not. For these algorithms, it is common to discretise one day into N time-slots, and perform predictions for time-slot n+1 at the end of time-slot n. Another algorithm is Pro-Energy [13], which is a promising algorithm that allows for predictions using longer prediction horizons. Predicting over a longer prediction horizon allows for the prediction of time-slots after the next one.

This thesis aims to provide a method of constructing a light measurement device with logging capabilities using popular and less costly equipment. After collecting sufficient measurement data of the solar irradiance in an environment, a profile will be constructed for illustrating how the light energy changes in one day. Some commonly used prediction algorithms, as well as a proposed solution, will be examined and compared. Finally, the favourable conditions of the proposed algorithm will be presented and discussed.

2
Hardware

In this chapter, the different choices for the hardware that were used for this project will be presented. Aside from examining these choices, a discussion will also take place where factors regarding the accuracy of the chosen hardware will be investigated.

2.1 Micro-controller

A very widely supported micro-controller is the ATmega328P, which is used in the Arduino Uno board. Seeing how Arduino is an open-source project, it follows that all of its software and libraries are also open-source. This, combined with its simplicity in implementation has allowed it to become very popular among hobbyists, since the Arduino boards allow people with non-engineering degrees to easily create their own Do It Yourself (DIY) projects by following guides. [14]

Another widely supported micro-controller is the ATtiny85, which is used to "shrinkify" Arduino projects. Among the reasons for shrinking a project could be decreased power consumption or requirements for less space.

2.1.1 ATtiny85

The ATtiny85 contains an EEPROM (Electrically Erasable Programmable Read-Only Memory) of 512 bytes that can be used to store logged data. If each log entry would consume 1 byte, this would roughly translate to approximately 8.5 hours of data using a sampling frequency of 1 sample/minute, or 3.5 days using 1 sample every 10 minutes.

Micro-controllers usually require programmers to flash software. However, in this case it is possible to use an Arduino Uno as a programmer by using an online guide and a library that is provided by the High-Low tech group, which is a research group at the MIT Media Lab [15]. Once the necessary software has been uploaded to the ATtiny, it will be able to save information into its EEPROM. However, it is crucial that no additional software is flashed onto the ATtiny before extracting data from EEPROM, seeing how this resets the memory. By flashing all the necessary software, and using feedback through the serial port, it is possible to reset, log and extract information without any risk of the ATtiny resetting its memory unless directly commanded to do so. The setup used to flash and extract data from the ATtiny using an Arduino Uno is displayed in Figure 2.1 below.

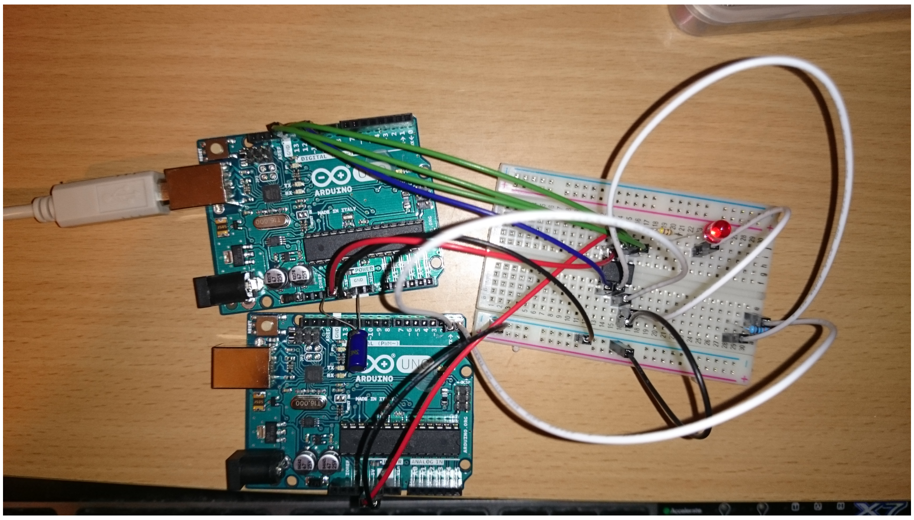

Figure 2.1: A picture of how the Arduinos are connected to the ATtiny for flashing and extracting information.

Figure 2.1 shows two Arduino Uno's connected to an ATtiny85. The upper Arduino in the Figure is used to flash software onto the ATtiny, whereas the bottom Arduino is used to read information from the ATtiny and forward it to the PC's serial port.

When extracting data from the ATtiny, it is important to keep in mind that two factors could cause the data to become corrupt.

- Non-matching baud rate between the port window in the PC and the one defined for the ATtiny.
- Software on the micro-controller of the Arduino Uno interfering with the data sent form the ATtiny.

An example of the corrupt data on the PC can be seen in Figure 2.2.

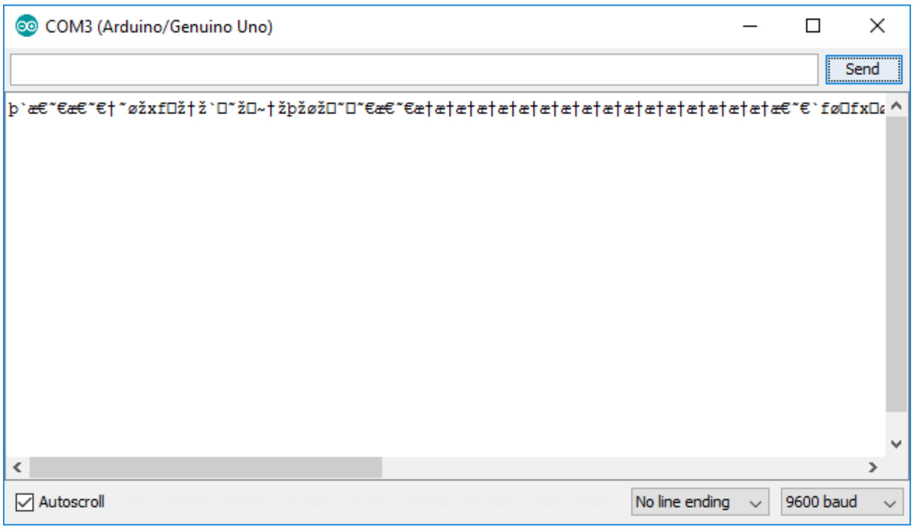

Figure 2.2: A print-screen of what the information from the serial port looks like when the data is corrupt.

These issues can easily be prevented by flashing the bare minimum software onto the Arduino board that is used to forward the received data from the ATtiny to the PC, and using the correct baud rate.

2.1.2 Arduino Uno

The Arduino Uno contains an EEPROM of 1024 bytes, allowing it to store twice the amount of data as the ATtiny85. Flashing software onto the Arduino is also somewhat simpler, since Arduino's homepage offers a free client to write, compile and upload code onto the Arduino Uno. [16] The Arduino Uno is connected to the PC by a USB type-B socket, which means that no other software, or external library is required to communicate with the Arduino.

2.1.3 Choice of micro-controller

The choice of micro-controller for this project depends on two factors; sampling frequency and duration. If no external memory storage is used, then a trade-off will have to take place between these two factors. Also, considering the fact that the data from the light sensors will most likely use more than one byte per sample, which will be discussed in more detail below, it is clear that an external storage solution would provide some convenience.

Due to the Arduino's popularity, it is also possible to buy a large variety of modules, such as an SD-card reader. Doing so would allow the Arduino Uno to take advantage of several gigabytes worth of storage onto an external memory device. This additional module would also allow for several months of data sampling at a high sampling frequency, without adopting an inconvenient trade-off between sampling frequency and duration.

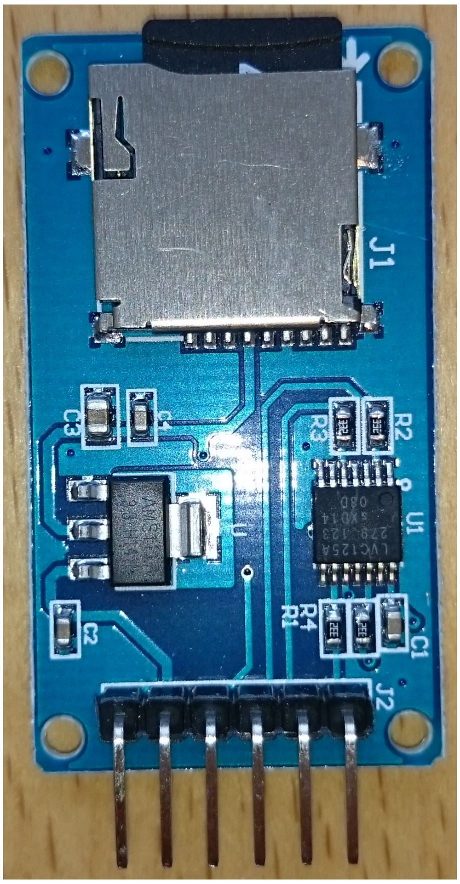

Figure 2.3: A picture of an SD-card module that can be used with an Arduino Uno.

Figure 2.3 shows what the SD-card module that is used for this project looks like. This module can hold a micro SD-card that can store up to 32 GB [17].

2.2 Sensor

There are mainly two different kinds of sensors that can be used to measure light, those that convert light to a voltage, and those that convert light to a frequency. While both sensors could be used to measure light, there are some differences that would make a light to frequency sensor preferable.

2.2.1 Light to voltage

Converting light to a voltage would mean that the available resolution of the measured data is only as good as the voltage meter. Whatever inaccuracies follow as a consequence would have to be included on top of whatever variations that follow from the sensor. Aside from the already mentioned factors, small light to voltage sensors that can measure more than 1 mW/cm^2 of light are difficult to find.

2.2.2 Light to frequency

For a light to frequency sensor, the resolution's bottleneck lies in the accuracy of the frequency output from sensor. This means that there would be no need to consider the accuracy of any voltage metre, since the frequency is measured by interrupts. In this case, it is crucial for the micro-controller to be able to be able to handle the frequency of interrupts, which could reach up to 1 MHz per sensor. Saving these frequencies would clearly require more than one byte per sample, considering the fact that one byte can only store an integer value of up to 255, whereas a maximum frequency of 1 MHz contains up to 7 digits.

2.2.2.1 TSL235

The form-factor of this sensor makes it very simple to use for implementation. It also has the ability to measure up to $1mW/cm^2$ with a frequency of up to 1 MHz. [18]

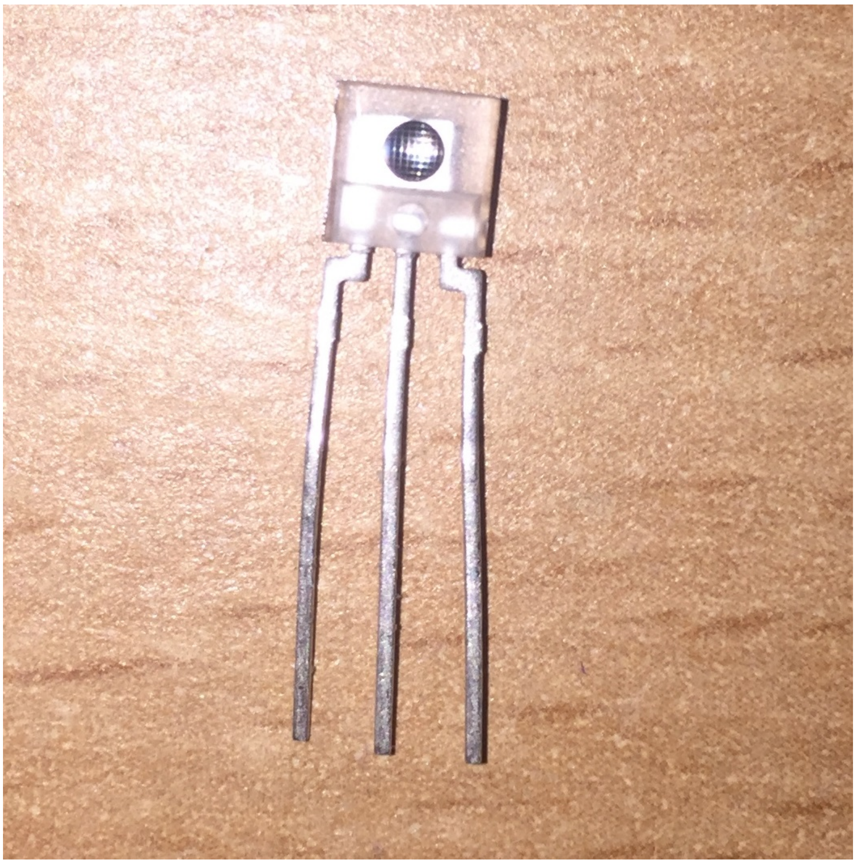

Figure 2.4: A picture of a TSL235, light to frequency sensor.

Figure 2.4 shows an image of a TSL235. It can be seen that its connectors are very easy to use, making application simple.

2.2.2.2 TSL230BRD

This sensor is programmable with three different sensitivity levels, while simultaneously allowing for a reduced maximum frequency by using a “divide-by” factor. The three available sensitivity levels can measure up to 1, 10 and 100 mW/cm^2. [19]

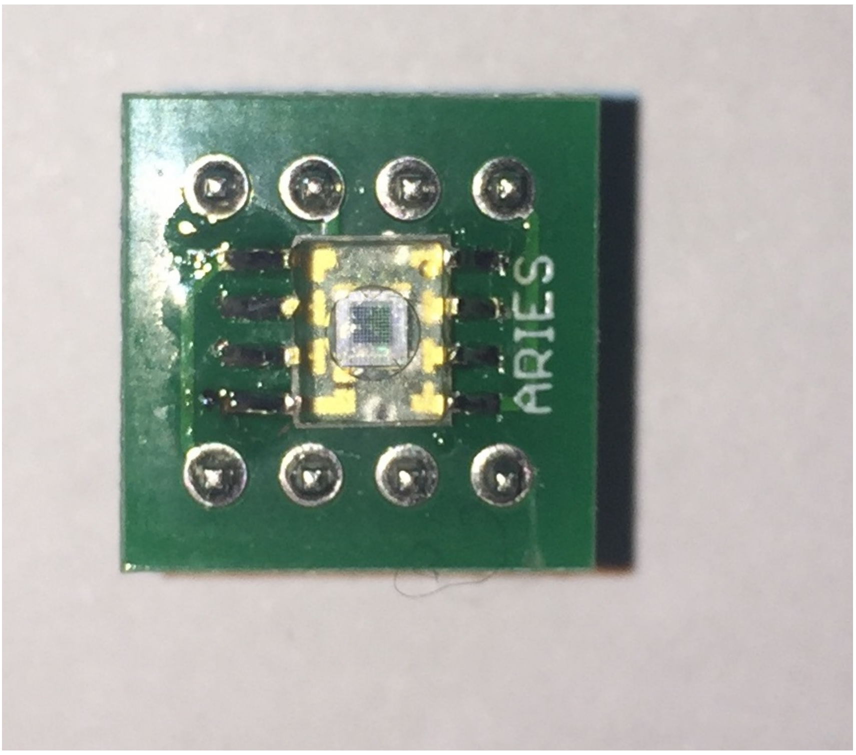

Figure 2.5: A picture of a TSL230BRD, light to frequency sensor.

Figure 2.5 shows a TSL230BRD soldered onto an SOIC to DIP adapter. The manufacturing of this sensor in DIP sockets ceased some time ago, making it only possible to purchase it with an SOIC socket. In order to be able to use this sensor on a breadboard, an adapter was required for compatibility.

2.2.3 Choice of sensor

According to some values taken from a study that included measurements of light, a sensor must at least be able to measure 40 mW/cm^2 [20]. Aside from the fact that a light to frequency sensor would give measurements with less variations, locating a sensor that turns light to voltage while also meeting the requirement of measuring up to 100 mW/cm^2 proved to be difficult. For these reasons, the chosen sensor for this project will be TSL230BRD.

Note that while a light to frequency sensor may be better suited for this project, one should not completely render light to voltage sensors useless. One area of use that a light to voltage sensor excels at when compared to a light to frequency sensor, is detecting a change in light, such as a blink. This would cause the sensor to send a single interrupt, whereas a light to frequency sensor would send multiple interrupts.

2.3 Soldering

The TSL230BRD sensor is moisture sensitive, which means that there are special conditions that must be met for optimal performance. [19] This means that soldering the sensor onto an adapter must be done very carefully. Sensors like these can be reflow-soldered or soldered in a normal fashion if done carefully. While it is preferable to reflow-solder these components, having access to such equipment is not always possible. For this reason, the sensors will be soldered by using soldering tin and soldering iron.

When soldering, it is crucial to ensure that the tip of the soldering iron does not come in direct contact with the sensor. It is also important that contact between the soldering iron and the pins of the sensor remain minimal and not occur for more than a few seconds. Only three measuring boards could be used simultaneously when performing tests for the sensors. Each board can carry two separate sensors, which means that a total of six sensors can be tested at a time. Three separate tests were performed in order to evaluate the performance of the sensors, which can be seen in Figures 2.6 through 2.8.

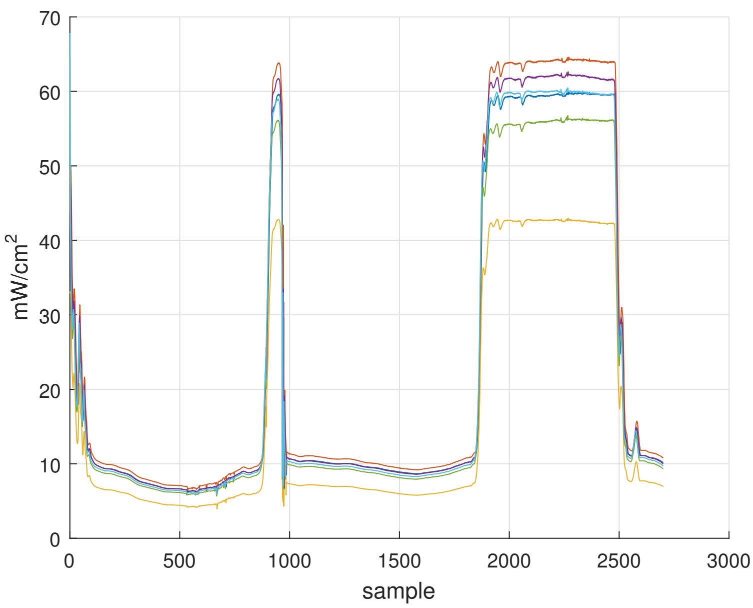

Figure 2.6: A plot showing the results from the first test.

Figure 2.6 shows the values of the first set of six sensors that were tested in an outdoor environment. It is clear that one sensor has a very large bias, which is highly unreasonable. It is safe to assume that this sensor has sustained too much damage when compared to the others.

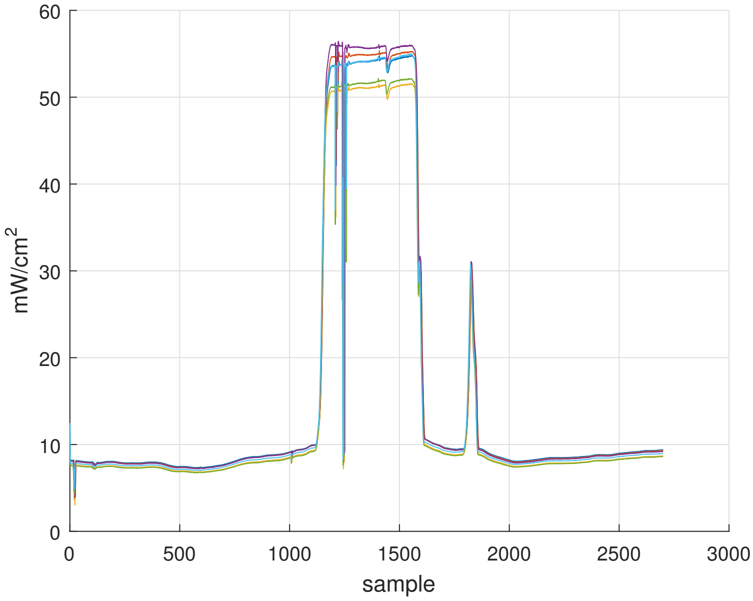

Figure 2.7: A plot showing the results from the second test.

Figure 2.7 shows the values of the second set of six sensors that were tested in an outdoor environment. Here it can be seen that their deviation from one another is very small (up to 5 mW/cm^2).

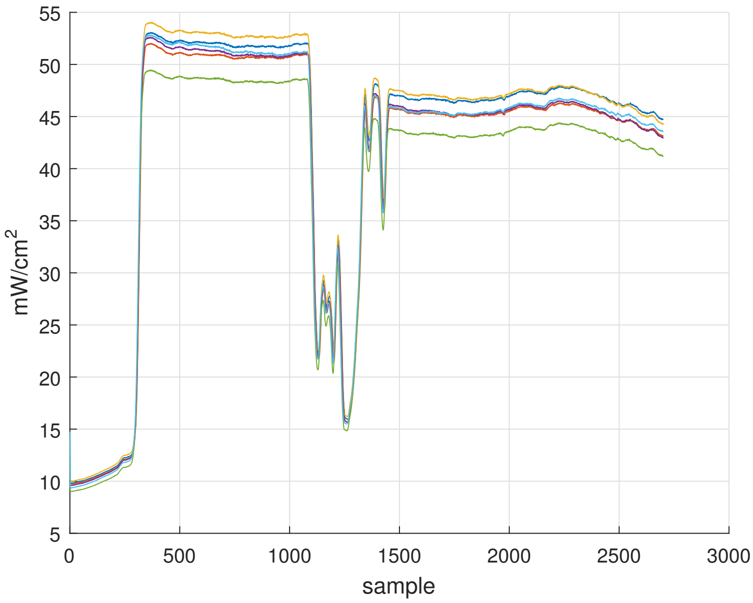

Figure 2.8: A plot showing the results from the third test.

Figure 2.8 shows the values of the third set of six sensors that were tested in an outdoor environment. Here the deviations are similar to those of Figure 2.7. Although some sensors have sustained minor damage, they are not so damaged that they produce inaccurate or highly unreasonable results.

Apart from the single sensor in the first test seen in Figure 2.6, all the sensors give reasonable measurement values with an acceptable bias of at most 10% from one another. According to the data-sheet of the TSL230BRD [19], an absolute frequency tolerance of 10% can be expected. Although some sensors may have sustained minor damage, it is to be expected that their deviations from one another can fall inside this tolerance level. Seeing how a maximum tolerance of 10% should already be expected from the sensors, this is also the value that will be used in order to assess whether the measurement values indicate faulty sensors. It is worth mentioning that if a 10% tolerance is to be expected for each sensor, then this adds up to a maximum deviation of 20% between two sensors. However, when searching for indications of a faulty sensor, a 10% tolerance will be calculated using the values from the sensor showing values closest to the middle of all six sensors used for each test. A 20% tolerance will only be investigated if a sensor displays values outside of a 10% tolerance of the chosen sensor.

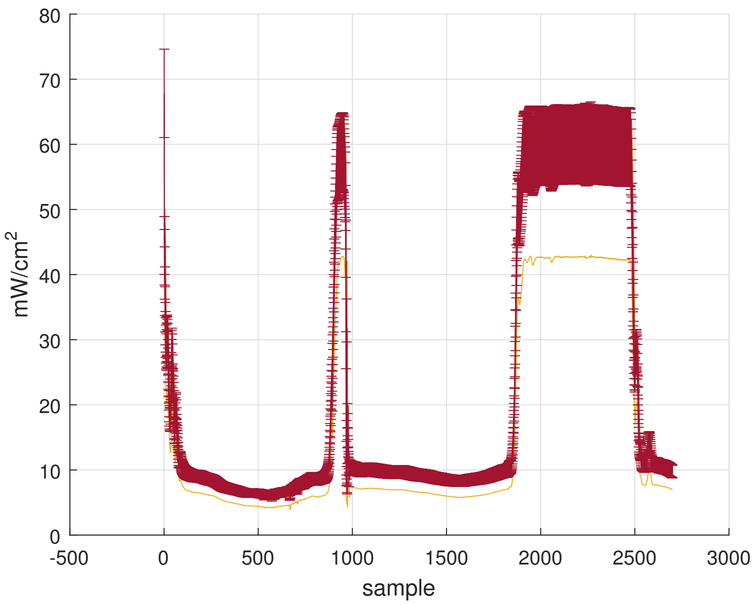

Figure 2.9: A plot showing tolerance of 10% for the measurements from the first test.

Figure 2.9 shows the tolerance range 10% in red for Figure 2.6. It can be seen that the sensor which was previously assumed to be too damaged clearly displays most of its measurement values outside the tolerance of 10%. However, the measurement values of all other sensors are located within the tolerance, indicating functionality.

Note here that the 10% tolerance values were calculated from the sensor values in the middle while ignoring the sensor values with a large deviation from the rest. Here it is desirable to further investigate if the lowest values of the chosen sensor values are smaller than the highest values of the sensor outside of the 10% tolerance zone.

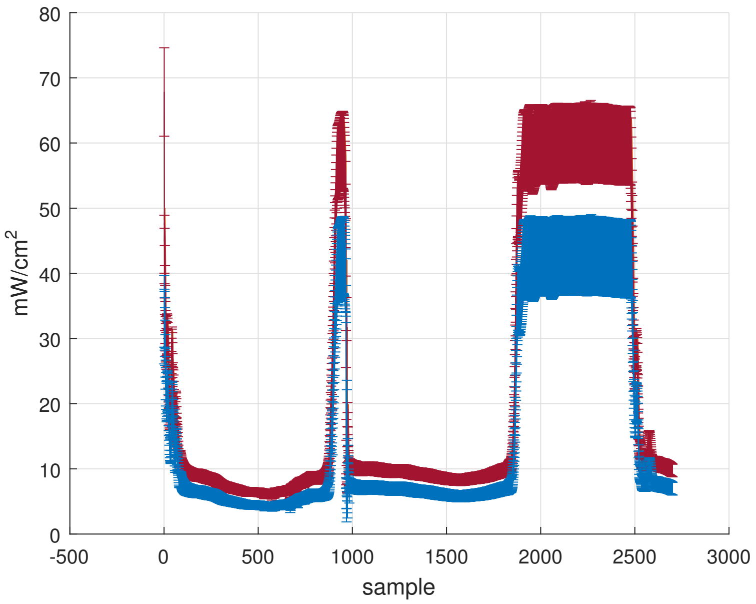

Figure 2.10: A plot showing tolerance of 10% for the measurements from the first test, as well as a 10% tolerance for the sensor showing values outside of the tolerance range shown in Figure 2.9.

Figure 2.10 shows the tolerance range 10% in red for Figure 2.6, and a 10% tolerance range for the yellow values outside of the tolerance range displayed in red. Here it can be seen that the maximum value inside the blue 10% tolerance is still smaller than the lowest values of the red 10% tolerance. This further strengthens the argument that this sensor is far too damaged to be used for measurement.

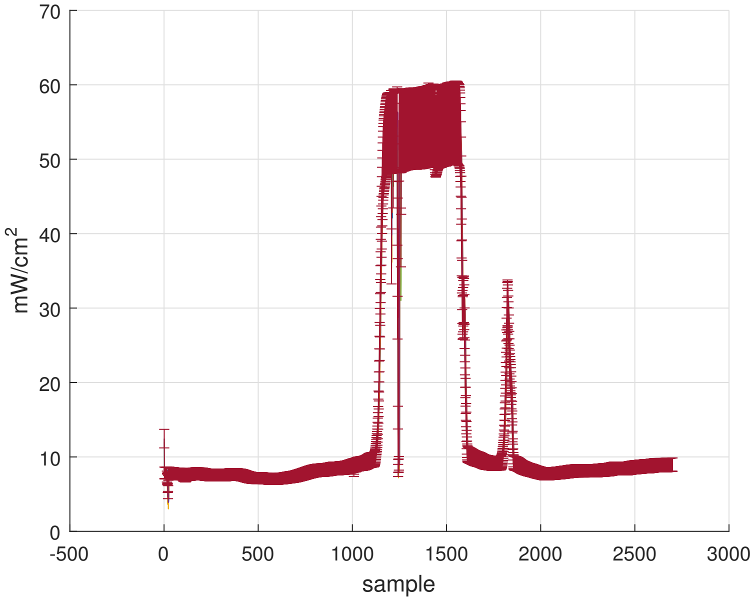

Figure 2.11: A plot showing tolerance of 10% for the measurements from the second test.

Figure 2.11 shows the tolerance range 10% in red for Figure 2.7. Here, all sensors display their measurement values within a tolerance of 10%, indicating that they are functional.

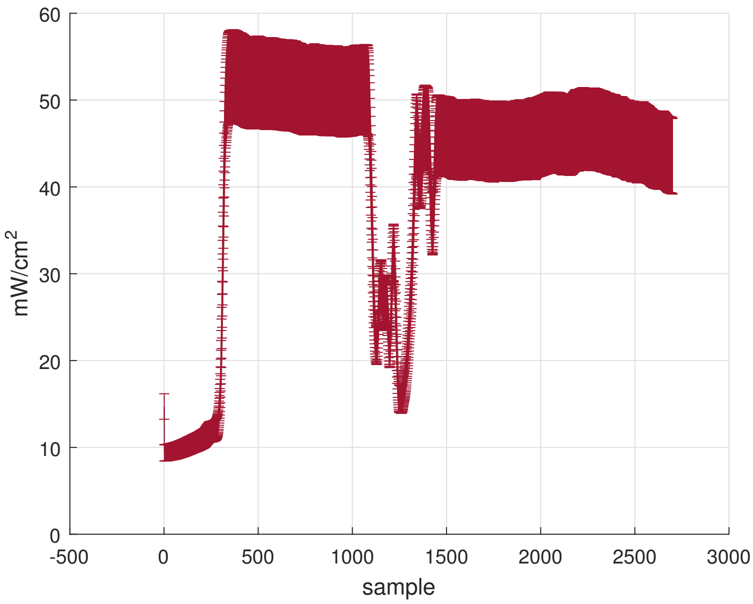

Figure 2.12: A plot showing tolerance of 10% for the measurements from the third test.

Figure 2.12 shows the tolerance range 10% in red for Figure 2.8. Here it can also be seen that all sensors are displaying their measurement values within a 10% tolerance, which indicates that they are also functional.

2.4 Discussion

This section will discuss the various errors that occur during measurements, and what these errors depend on. There are two different kinds of errors; those that depend on time, such as jitter, and those that depend on values, such as variations in sensors and faulty measurements.

2.4.1 Power supply and jitter

According to the data-sheet of the ATmega328, a minimum voltage of 4.5 V is required for it to operate at 20 MHz. If the micro-controller is to operate at a frequency of 16 MHz, then more than 3.6 V is required. [21] This means that a constant supply of at least 4.5 V must be given to the micro-controller in order to obtain a correct sampling frequency.

The voltage regulator is bypassed if a power source does not come from the USB port, or the power outlet of the Arduino Uno. This brings a consequence when using AA or AAA batteries, seeing how this method of supplying power would be done through the 5V and ground pins. Doing this however, would result in a large amount of jitter since the power requirements are not met. When testing this method of power supply, jitter was no issue when collecting samples for minutes. However, when collecting samples over the course of three days, jitter reached values over 35 percent, since the batteries' voltage supplied less than 4.5 V for a time before the test period was complete.

Considering the feasibility of using batteries in this manner, it is clear that another method of supplying power to the micro-controller is required. One other option is using a power bank, which is commonly used for smartphones in order to charge them on the go. Using such a device would allow for supplying power through the USB port, which is beneficial. A problem with using such a device however, is that they are tailored for smartphones, meaning that they switch off once a phone is fully charged in order to conserve power. This issue leads to somewhat different complications, seeing how the Arduino Uno would power off and on again repeatedly.

If a power outlet is used directly with the aid of an adapter that is commonly used for smartphones, all of the above issues related to using batteries would be eliminated. A constant power supply would be available at all times, which is regulated after the Arduino Uno's needs before reaching the board. The only downside of using this method of supplying power is that it is not quite as mobile as using a battery. Among the limitations would be difficulties in supplying power to locations that are not close to a power outlet. This limitation can still prove difficult in an office where power outlets are not difficult to find, since a USB-cord might not always be long enough. Worth mentioning here, is that it is also possible to power an Arduino Uno by using a laptop or other computing device with a USB-port, which would provide the same results as using a power outlet.

Despite the limitations of supplying power directly from a power outlet, this method still proved the easiest to implement due to its simplicity. This method provides a jitter of less than two percent, which is a significant improvement over 35 percent.

2.4.2 Capacitors

According to the data sheet of the TSL230, the power supply lines should be decoupled by a 0.01 to 0.1 μF capacitor for optimal performance. While it is possible to obtain feasible values when collecting samples without using capacitors, it is still beneficial to use them.

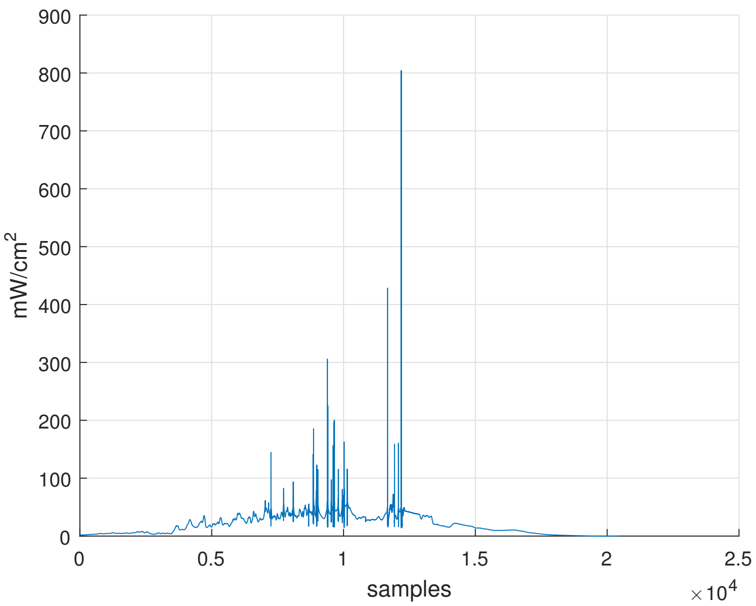

Figure 2.13: A plot showing the effects of not using a capacitor.

Figure 2.13 shows the effects of not using a capacitor when measuring light. It is known that the solar constant is 1360 W/m^2 and the solar irradiation inside the atmosphere can only reach $\approx$ 1000 W/m^2 [22], which translates to 136 mW/cm^2 and 100 mW/cm^2 respectively. For this reason, it is highly unreasonable that values exceeding 300 mW/cm^2 are measured. While these large spikes can easily be removed by filtering them, some data can still be lost when filtering. The reason for this being that filtering would attenuate all sample changes that are too quick, but this could result in losing some data where these changes were not a consequence of not using a capacitor, but a result of what actually happens in an environment.

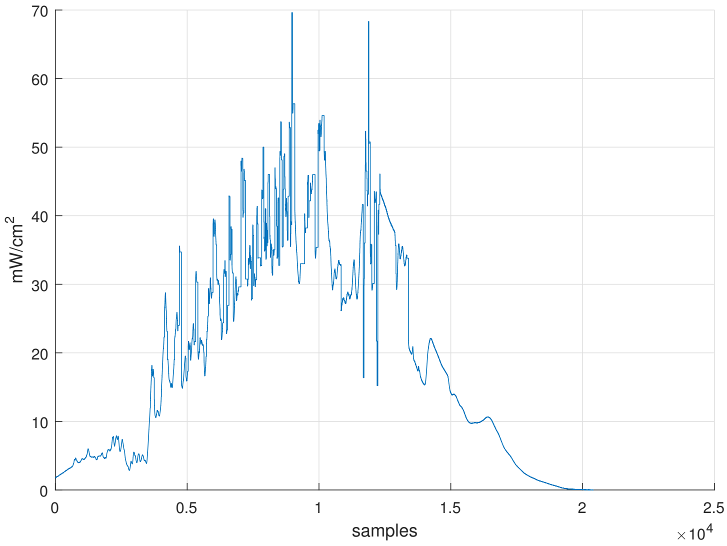

Figure 2.14: A plot showing what a filtered version of Figure 2.13 could look like.

Figure 2.14 illustrates what a filtered result may look like. While it is clear that this is a significant improvement over that of Figure 2.13, it can be seen that some values are still too high. For the test environment, it is unrealistic that values should exceed 50 mW/cm^2 for indoor conditions, when compared to the values obtained in [20]. This argument can be strengthened further by taking note that all values above 60 mW/cm^2 look like spiked values.

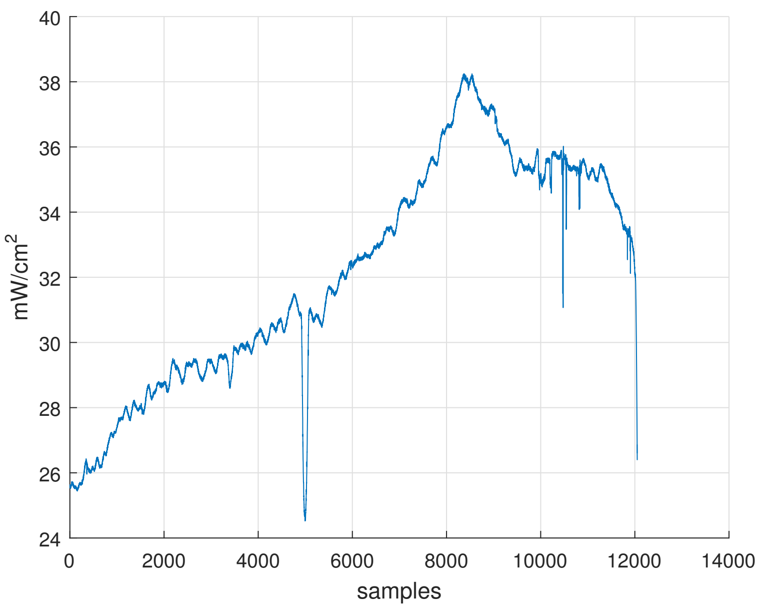

Figure 2.15: A plot showing the results of using a capacitor for some test samples.

Figure 2.15 shows how some test samples look like when using a capacitor. It is clear that no unreasonable spikes can be seen in this test data even though it is unfiltered. The benefits of using capacitors for optimal results can clearly be seen, which is why they will be used in this project.

2.4.3 Shadow

When constructing a measurement device, it is important that the wires do not cast a shadow over the sensors in order to obtain correct values. Having a shadow cast on a sensor could result in large errors as opposed to what the data would look like if no shadow was cast.

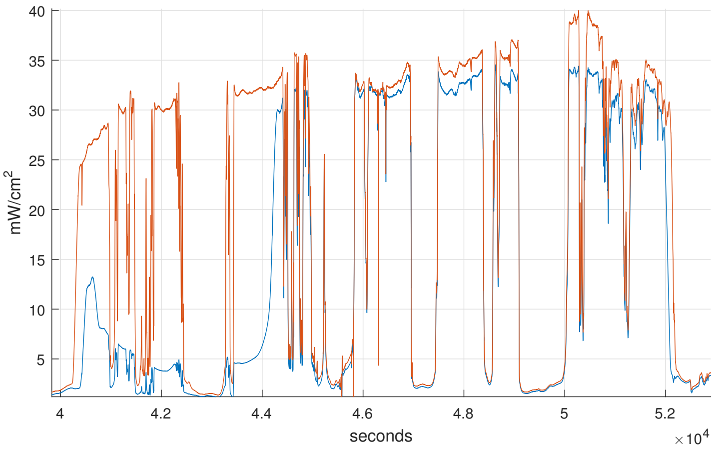

Figure 2.16: A plot the effect of shadow from the wiring.

Figure 2.16 shows the possible impact of shadow from wiring. It is clear that, although there may be some bias between the sensors, it can be seen that at some samples the deviation from the two is large. Aside from a large deviation, it can also be seen that the character of the sensor with a shadow cast over it is not similar to that of the one that does not have a shadow cast over it. This means that this issue cannot be resolved by adding a bias to the values. In order to fix this issue, a minimum amount of wires must be used in the vicinity of the sensors.

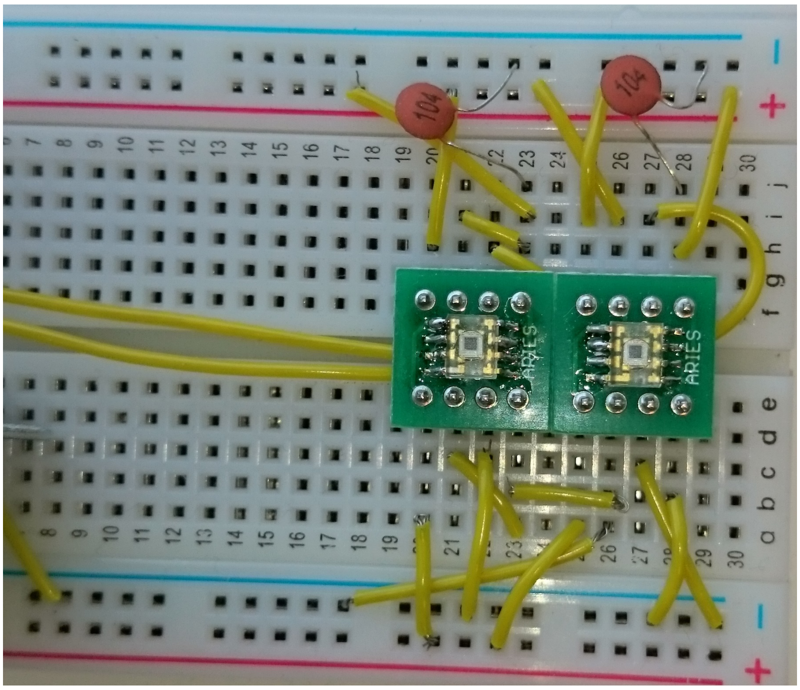

Figure 2.17: An image showing how the wiring around the measurement sensors looks like.

Figure 2.17 shows acceptable wiring around the measurement sensors in order to minimize the shadow cast.

3

Characterisation

This chapter will focus on defining the important factors for characterising the available energy. Furthermore, the requirements for powering a device to be used for machine-to-machine communication will also be determined.

3.1 Available energy

The energy requirements for this project may be harvested from small solar panels. For this reason, the unit used for irradiance is mW/cm^2 as opposed to W/m^2 which is used for larger solar panels. When measuring the intensity of sun light in an environment, it is crucial the sampling frequency is not too slow. If the chosen frequency is slower than necessary, some data could be missed.

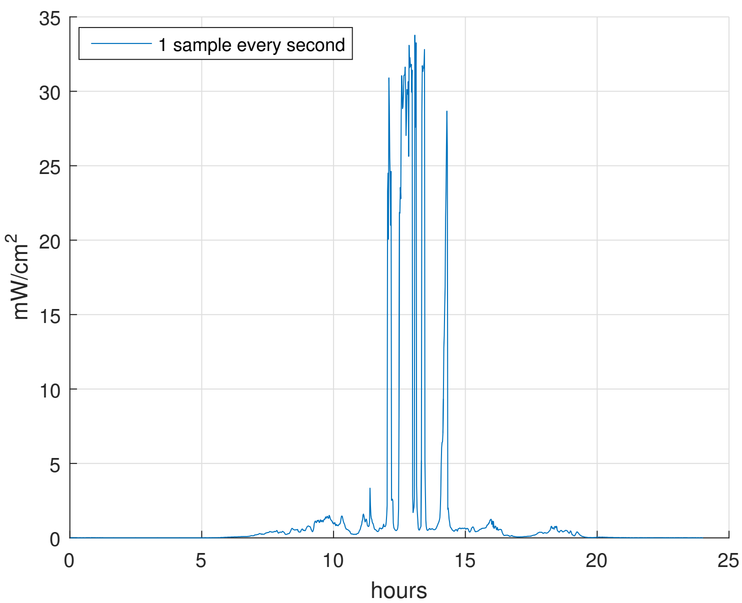

Figure 3.1: A plot of test data sampled every second.

Figure 3.1 shows the test sample data that was used to examine the effect of sampling. It can be seen that the test data is taken over the course of a day with high variability in weather conditions, which should help in determining whether a sampling frequency is high enough. In order to see the true effect of the sampling rate, only the hours between 11 and 15 will be shown in the following figures, since that is the time where large variations occur in the test data.

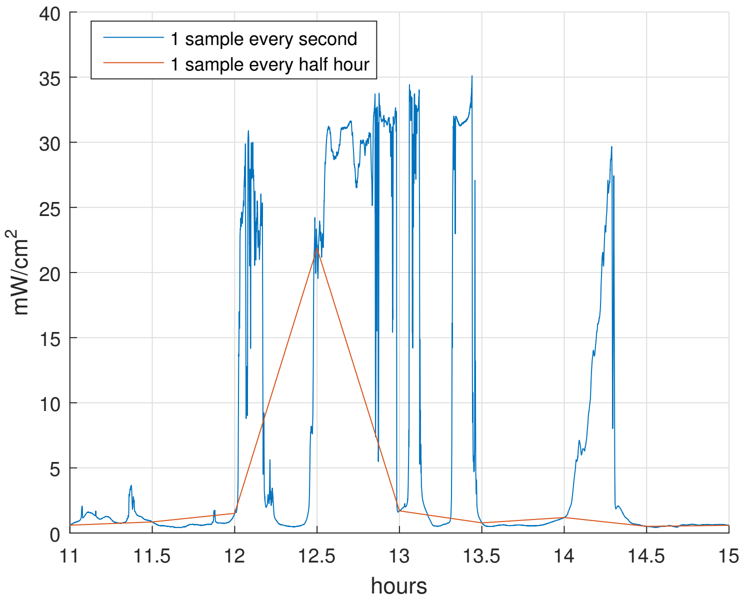

Figure 3.2: A plot showing the effect of one sample taken every half hour.

Figure 3.2 shows how a sample every half hour compares to that of a sample taken every second. Here, it can clearly be seen how large portions of data are missed with this sampling rate.

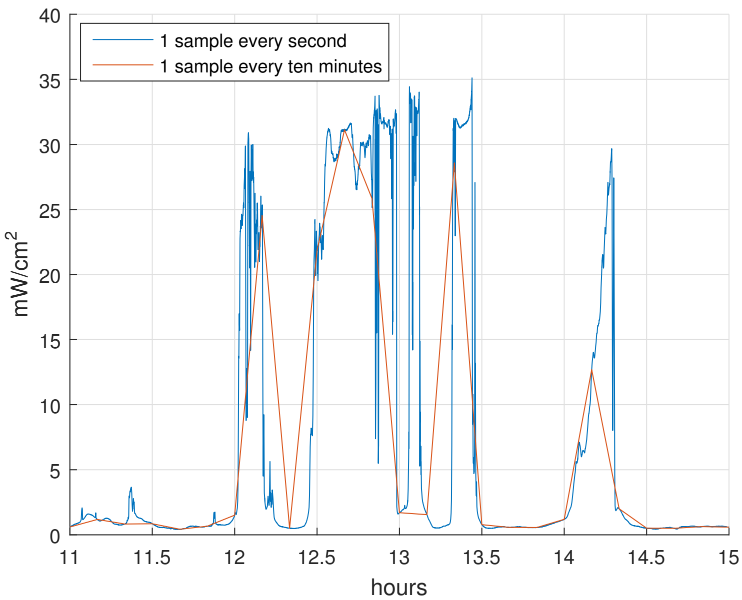

Figure 3.3: A plot showing the effect of one sample taken every ten minutes.

Figure 3.3 shows how a sample every ten minutes compares to that of a sample taken every second. While improvements can be seen when compared to collecting one sample every half hour, it is still clear that large portions of data are still missed with this sampling frequency.

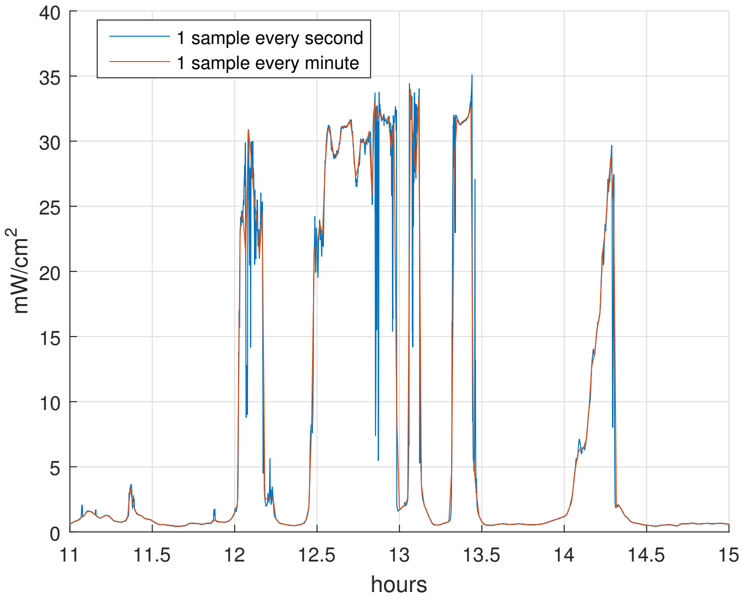

Figure 3.4: A plot showing the effect of one sample taken every minute.

Figure 3.4 shows how a sample every minute compares to that of a sample taken every second. Collecting one sample every minute yields somewhat more satisfactory results. However, it can be seen that some small portions of data are missed during quicker variations in solar irradiance. Two such occasions can be observed, one being directly after the 12th hour and another being slightly before the 13th hour.

3.2 Choice of sampling frequency

Seeing how collecting one sample every half hour or ten minutes could lead to a severe underestimation of the available energy, these two sampling frequencies will not be considered.

Collecting one sample every minute results in very few errors, and provides a somewhat accurate description of how the light energy in the environment changes. However, during quicker changes in weather conditions, some data might still be missed. While it may still be possible to use this sampling frequency, it can be argued that slight over-estimations made may prove unbeneficial. On the other hand, a higher sampling frequency results in higher energy consumption of the measuring unit, which may result in shorter battery life. Also, a high sampling frequency would require a larger memory for storing all the collected data. This means that it may still be beneficial to use a slower sampling frequency if the battery life of the device must be maximized, or if the available memory is limited.

In this project, a constant power supply from a PC or a power outlet is used, as well as micro SD cards capable of storing several GB worth of data. This gives no reason to favor a slower sampling frequency over a quick one. With these arguments in mind, the chosen sampling frequency for this project will be that of 1 sample taken every second.

3.3 Calculating solar irradiance

Calculating the sum of all of the collected samples, multiplied by the time of each sample provides the available energy in Joules per area unit, seeing how the solar irradiance is measured in Watts per area unit. Since one sample is taken each second, and 1 $W = 1\ J/s$, calculating the available energy from the measurement data can easily be done by equation (3.1).

$$E_{available} = \sum_{i=1}^{N} e_i \tag{3.1}$$

where e_i is the i^{th} sample of the data-set, and N is the total number of samples. When harvesting solar energy with solar panels, three factors must be taken into account:

- Efficiency
- Size
- Angle

A solar panel is not 100% effective, which limits the amount of energy that may be harvested from solar irradiance. Also the amount of energy harvested is largely dependent on the size of the solar panel, since solar irradiance is defined as energy per area unit. Taking only the efficiency and solar panel area, the harvested energy may be calculated as

$$\begin{aligned} E_{harvested} &= \eta \cdot A \cdot E_{available} \\ &= \eta \cdot A \cdot \sum_{i=1}^{N} e_i \end{aligned} \tag{3.2}$$

where η is the efficiency and A is the area of the solar panel. The amount of harvestable energy with respect to the angle between the panel and the solar irradiation can be modeled by multiplying the irradiation with a cosine function [22].

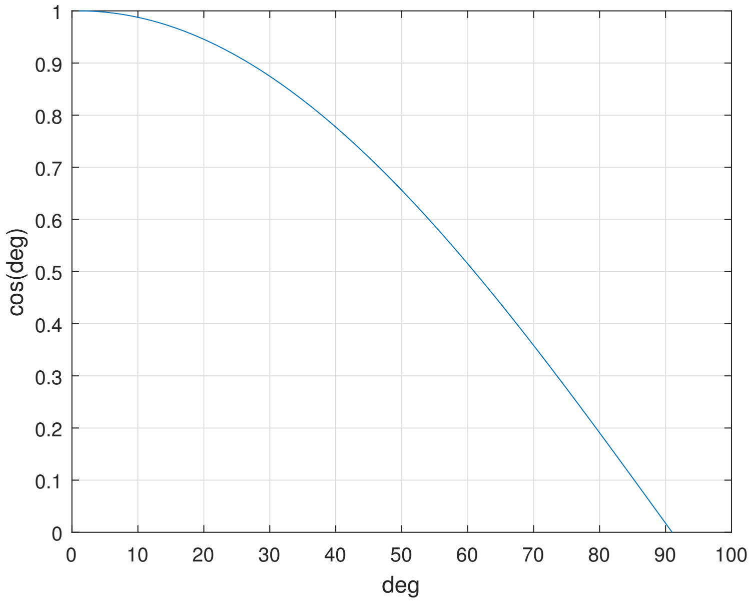

Figure 3.5: A plot showing the dependency between the angle and the solar irradiation.

Figure 3.5 shows how the amount of harvestable energy deteriorates as the angle of the panel with respect to the direction of incoming light increases. While it will not always be realistically possible to maintain an angle of 0 degrees between the solar irradiation and panel when harvesting energy for small devices, it is worth examining the placement of the device in order to maximize harvest.

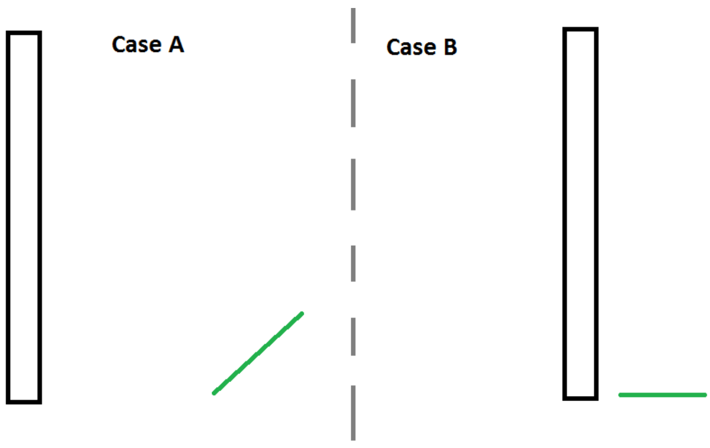

Figure 3.6: An image illustrating two different cases for the placement of a solar panel, as seen from the side.

Figure 3.6 illustrates two possible cases for placing the solar panel, where the black rectangle is the window and the green line is the solar panel. Note that Case A has an angle of approximately 45 degrees from the window, whereas the panel has an angle of 90 degrees from the window. For comparison, each case was tested with measurements from 10 days each. The measurements were taken from an office with a window facing North East over the course of 24 hours per day, with each day starting from midnight. Using a solar panel that is 55 mm wide and 70 mm long, with an efficiency of 17% [23], the minimum and maximum harvest for each case in an entire day can be obtained using Equation (3.2)

Table 3.1: Least and most harvested energy for different placement cases.

	Case A	Case B
Minimal harvest	316.4	640
Maximal harvest	678	1197

It can clearly be seen from Table 3.1 that the estimated energy harvested in case B is almost double that of case A. In order to characterise the energy of light in the office, 48 time-slots will be used for one entire day. This translates to one time-slot every half hour, where each time-slot represents the average solar irradiance for the past half hour. Furthermore, the average over ten days of the measured irradiance is used for constructing a profile, in order to reduce the effect of variance.

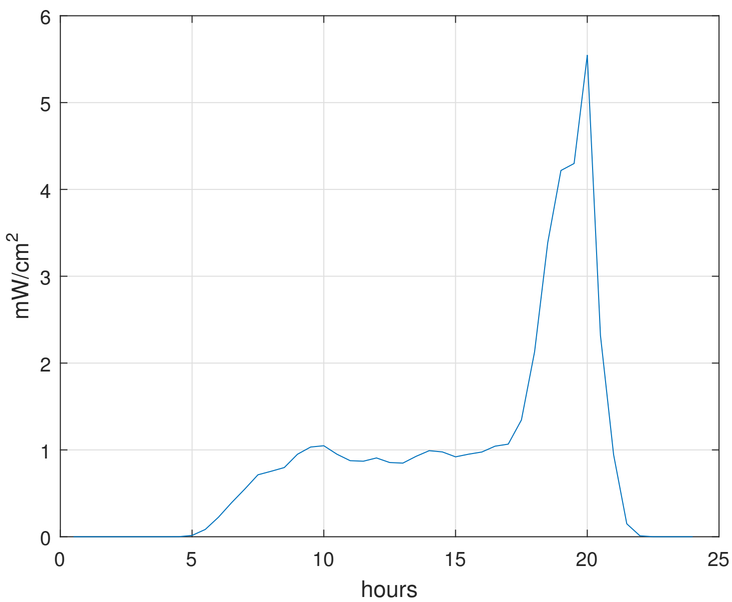

Figure 3.7: A plot showing the character of the sunlight in an office facing North East for case A.

Figure 3.7 shows a plot with a profile of the sunlight for case A, in an office located in Gothenburg facing North East. At first, sunrise occurs slightly after 5 AM, with the sun fully up at around 8. Between 8 AM and approximately 5 PM, the level of sunlight remains mostly the same. After 5 PM, the sun is no longer shining in from behind the office, but above it. Between the times of 5 PM and approximately 9 PM, is when the sun shines into the office with the peak being around 8 PM.

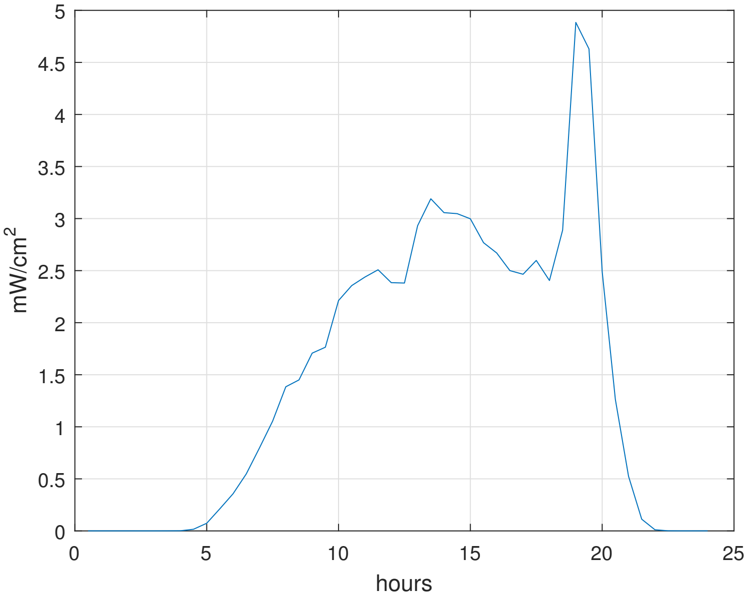

Figure 3.8: A plot showing the character of the sunlight in an office facing North East for case B.

Figure 3.8 shows a plot with a profile of the sunlight for case B. While some similarities can be seen here, it can also be observed that the spike in energy harvest that occurs at around 8 PM is slightly lower here, which is most likely depends on the placement of the panel in case B. Despite smaller peak values towards the end of the day, this different placement still results in a somewhat higher estimate of energy harvested overall. This means that even though the available energy may be higher at certain times, it is still beneficial to place it closer to the window at a non-optimal angle for peak hours, since more energy can be harvested during non-peak hours.

It is worth mentioning that while the profile constructed with the mean of all measurements gives an idea as to how the character of sunlight shines in on the office, it will not always give an accurate description for the character. The reason for this being that variations in weather conditions cannot be accurately displayed with a simple mean. Such a profile can however still provide reasonable results when performing predictions. The Weather Conditioned Moving Average (WCMA) algorithm uses a the mean of the past D days in order to maintain an updated profile of what the weather conditions look like in an environment [12]. Another possibility is to weight a profile of the sunlight based on weather conditions, such as one weight for cloudy weather, another weight for sunny weather and so on, combining these profiles depending on the weather conditions of the day. This is done for the Pro-Energy algorithm, where it saves D amount of profiles and uses a weighted sum of these profile measurements in order to perform its predictions [13]. The performance of these algorithms will be discussed in more detail in Chapter 5.

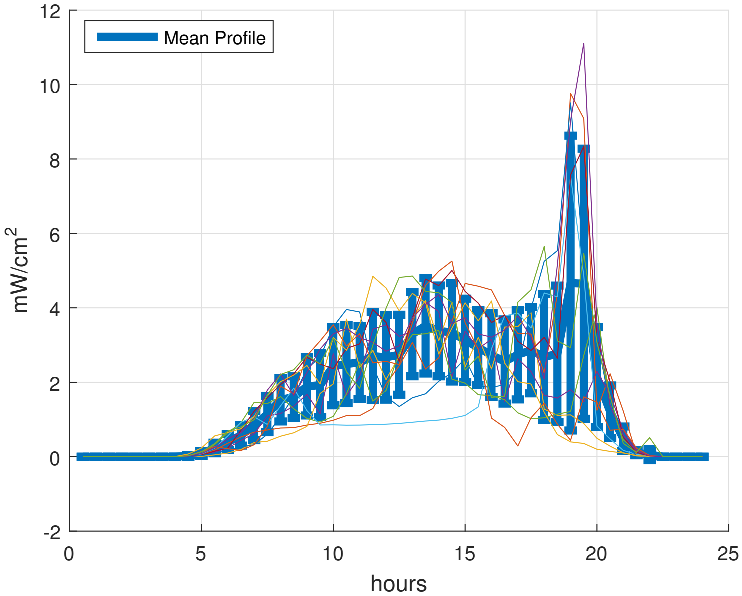

Figure 3.9: A plot of the constructed profile in case B, with the standard deviation of each time-slot.

Figure 3.9 shows a plot of the profile constructed in Figure 3.8, with the standard deviation for each time-slot, and all of the ten days that were used in its construction. This figure gives a clear representation as to the available irradiance for each time-slot in an entire day, and how much each time-slot typically varies.

While it is obvious that the profile in Figure 3.8 cannot completely represent what future days look like, it can be used in order to gain an understanding as to when the most energy can be harvested for multiple days. For this reason alone, the mean profile shown in Figure 3.8 has achieved its purpose. However, if this profile is to be used for something else, such as prediction, it will not be enough if used without any other algorithms or data to aid it, due to high variations in weather conditions, as can be seen in Figure 3.9.

So far, only an office located in Gothenburg, facing North East has been investigated. The office that has been examined shall from here on be denoted as office A. An office across the hall from office A, denoted office B, was also examined. This office had a window facing the opposite direction (South West). It was observed that a somewhat larger estimate of energy could be harvested for this office.

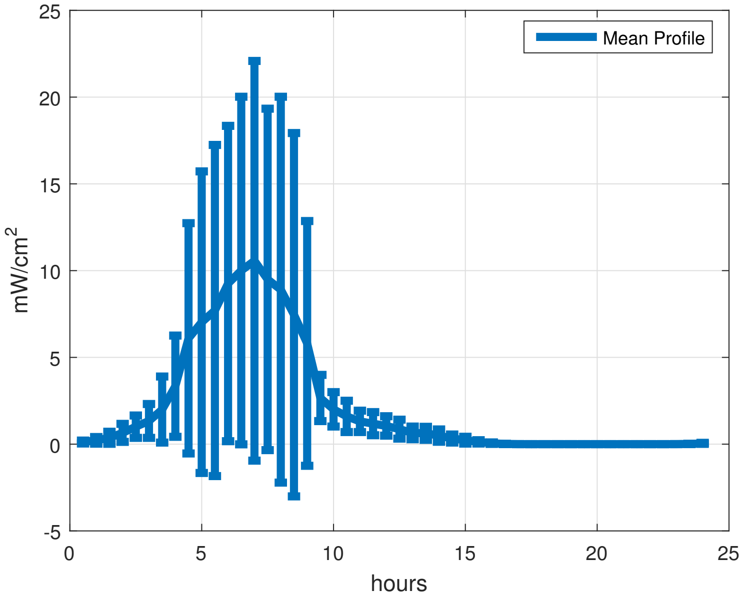

Figure 3.10: A plot of the constructed profile in office B, with the standard deviation of each time-slot.

Figure 3.10 shows the mean profile and standard deviation of each time-slot for office B, which is located across office A. Here, the profile was constructed in the same manner as was previously done, namely by calculating the mean over ten days worth of measurements for each time-slot. Seeing how the duration of direct sunlight is longer in this office during the hours leading up to noon, frequent blocking of the sunlight with the aid of blinds was observed. Naturally, this would mean considerably less harvestable energy when the blinds are closed and higher variations of irradiance, making prediction somewhat more difficult. Over the course of ten days, an estimated minimum and maximum of approximately 322 and 4607 J was harvested respectively. For comparison, data was also collected from another office, with a window facing the same direction as that of office B. This third office will be denoted office C, and is located one floor beneath office B. Worth mentioning here is that office A and B are located on the seventh floor, and office C is located on the sixth floor.

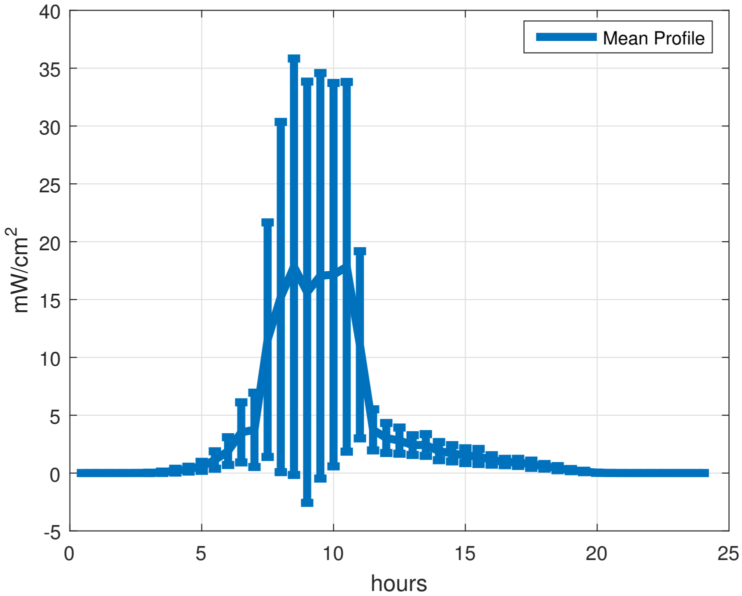

Figure 3.11: A plot of the constructed profile in office C, with the standard deviation of each time-slot.

Figure 3.11 shows the mean profile and standard deviation of each time-slot for office C, which was obtained in the same manner as for office B. While it is clear that both offices have a peak at a similar hour, it can also be observed that the peak in energy harvest lasted for longer hours in office C. Over the course of ten days, an estimated minimum and maximum of approximately 500 and 4404 J was harvested respectively. The minimum value for this office is larger than that of office B, and the maximum value, smaller.

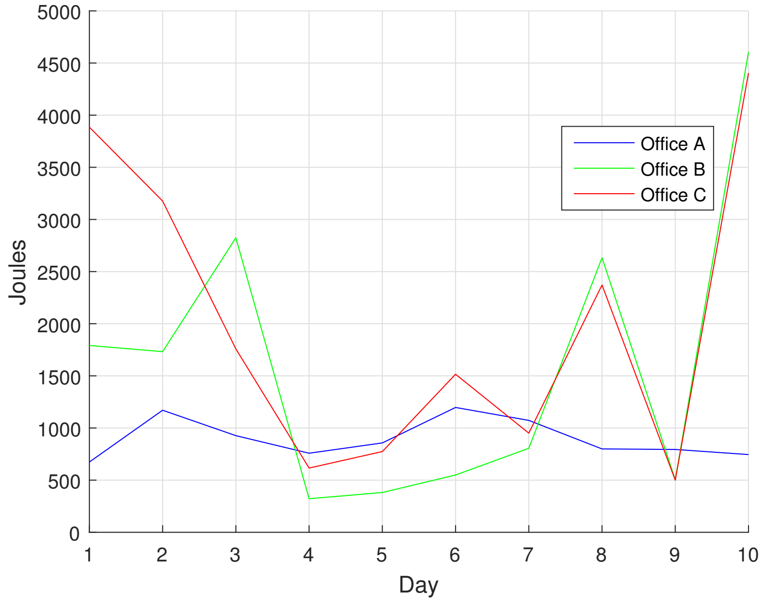

Figure 3.12: A plot comparing the harvested energy over ten days for all three offices.

Figure 3.12 shows how the estimate of harvested energy varies for each of the three offices. While one may expect the harvested energy to be somewhat similar for offices B and C, it is clear that this is not always the case. This indicates that the duration of using the blinds varies between offices, and suggests a random behaviour.

Finally, it is also worth looking into how much energy can be harvested from an area with minimal solar irradiation.

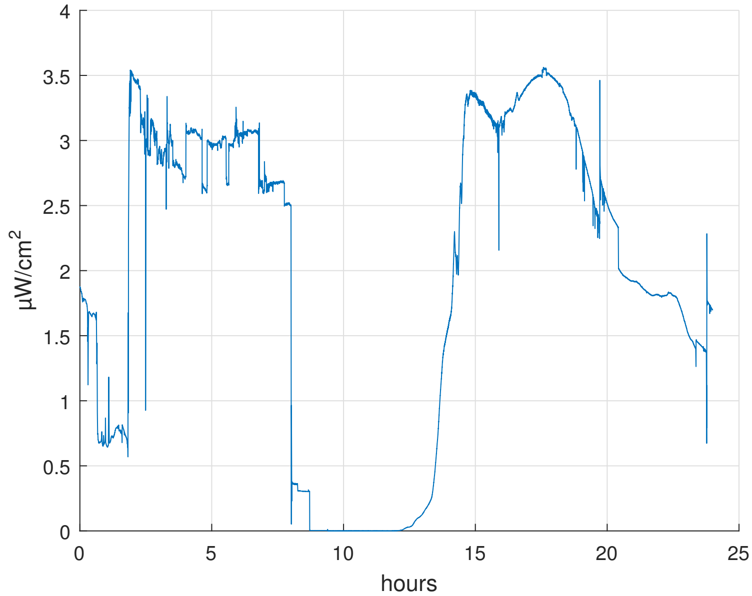

Figure 3.13: A plot of the solar irradiance over the course of 24 hours from an office with a dimmed window.

Figure 3.13 shows the available irradiance from a room where the blinds of the window are always closed, for 24 hours starting from 1 AM. Using the same parameters for the solar panel as was previously used, a total estimated harvest of approximately 1.1 J can be obtained using Equation (3.2).

4

Machine-to-Machine Communications and Energy Consumption

This section presents a model that will be used throughout this project in order to determine the necessary energy requirements. This model is a standard that can be used to calculate energy consumption for small devices using machine-to-machine (M2M) communication.

4.1 Power consumption model and energy requirements

In order to determine how much energy can be used, a model for power consumption using M2M communications will be used from [1], which is also listed below for convenience.

Table 4.1: Model parameters for power consumption. [1]

	Power consumption for year	2015 (mW)	2020 (mW)
P_{tx}	Transmission	500	300
P_{act}	Active period	150	100
P_{clock}	Accurate clock	10	10
P_{base}	Base	0.015	0.01
P_{sleep}	Sleep	0.03	0

Table 4.2: Common model parameters. [1]

	Parameter	Value
n	Number of cycles	2 - 300
t	Data reporting period	768 s
l	Packet size	8000 bits
t_{DRX}	Length of DRX cycle	2.56 - 384 s
t_{tx}	Time spend in tx period	50 ms
t_{act}	Time spent in active period	10 ms
t_{sync}	Time to obtain sync	10 ms
C_{bat}	Battery capacity	6500 J

It is worth mentioning here that the shortest cycle of 2.56 seconds is a value that is assumed for the model, where some time between each active periods is required. This model assumes a data transfer rate of 160 kbits/s. A linear relation is assumed between the transmission time and package size, according to

$$t_{tx} = \frac{l}{160000} \tag{4.1}$$

where l is the data package size in bits. During transmission, the energy consumption can be calculated as

$$E_{tx} = t_{sync} \cdot (P_{act} + P_{clock}) + t_{tx} \cdot (P_{tx} + P_{clock}) \tag{4.2}$$

Energy consumption for the active period can be calculated as

$$E_{act} = n \cdot (t_{act} + t_{sync}) \cdot (P_{act} + P_{clock}) \tag{4.3}$$

When the device is in its nonactive period, the energy consumption is

$$E_{sleep} = (t - t_{tx} - n(t_{sync} + t_{act})) \cdot P_{sleep} \tag{4.4}$$

Apart form the previously mentioned power consumption equations, a base power consumption, that is constantly consumed, is denoted as

$$E_{base} = P_{base} \cdot t \tag{4.5}$$

The total energy consumption is obtained by calculating the sum of equations 4.2 through 4.5

$$E = E_{tx} + E_{act} + E_{sleep} + E_{base} \tag{4.6}$$

Inserting Equations (4.2) - (4.5) into Equation (4.6) yields

$$\begin{aligned} E = {} & t_{sync} \cdot (P_{act} + P_{clock}) + t_{tx} \cdot (P_{tx} + P_{clock}) \\ & + n \cdot (t_{act} + t_{sync}) \cdot (P_{act} + P_{clock}) \\ & + (t - t_{tx} - n(t_{sync} + t_{act})) \cdot P_{sleep} + P_{base} \cdot t \end{aligned} \tag{4.7}$$

According to the model used, the maximum amount of cycles possible in a 768 second period is 300, which is due to the fact that one cycle cannot be shorter than 2.56 seconds, as explained earlier. Note that only one transmission is performed over the period of 768 seconds. What is meant by 300 cycles here, is the amount of times that the device wakes up in order to receive data. This translates to 112.5 transmissions and 33750 active periods over the course of 24 hours. In order to obtain a more accurate maximum consumption of the model, 113 transmissions will be used, with 150 active periods instead of 300 for the last 768 second period. This will give a slightly more accurate estimate, while minimising overestimation of the consumed energy. For an entire day, this will give a maximum consumption of

$$E_{max} \approx 114.95\ J \tag{4.8}$$

The required energy to power the device from the model for one day without sending any transmissions can be calculated by summing the energy requirement for the base consumption and the sleep consumption, which yields

$$E_{min} \approx 3.89\ J \tag{4.9}$$

While this gives an idea as to the necessary minimum requirements for the device from the model to function, it is desirable to have a minimum value that includes at least one transmission, since anything else would make the device somewhat meaningless. Using a single cycle for a period of 24 hours, the minimum energy required to send one transmission and have one active period in one day is

$$E_{min} \approx 3.92\ J \tag{4.10}$$

4.2 Consumption of available energy

For this project, it will be assumed that the devices requiring power will have access to a limited amount of power from a battery. It is also assumed that no energy is lost when recharging the battery from the available energy. Furthermore, only the office with the least estimated harvest in energy, being office A, will be used in order to obtain the minimum requirements.

When comparing the estimated minimum harvest values from the office A, seen in Table 3.1, with the maximum consumption for the model, seen in Equation (4.8), it is clear that using a 55x70 mm solar panel produces a large excess of energy. An excess of energy means that the available energy for use is larger than what is required. This means that it would either be possible to reduce the panel size, change the placement of the device, or supply power for more energy consuming devices. This section will examine how the panel and battery size affect the device model. Furthermore, the effect of the active periods' frequency (parameter n) will also be examined. Worth mentioning is that the efficiency is 17% for all panel sizes used below.

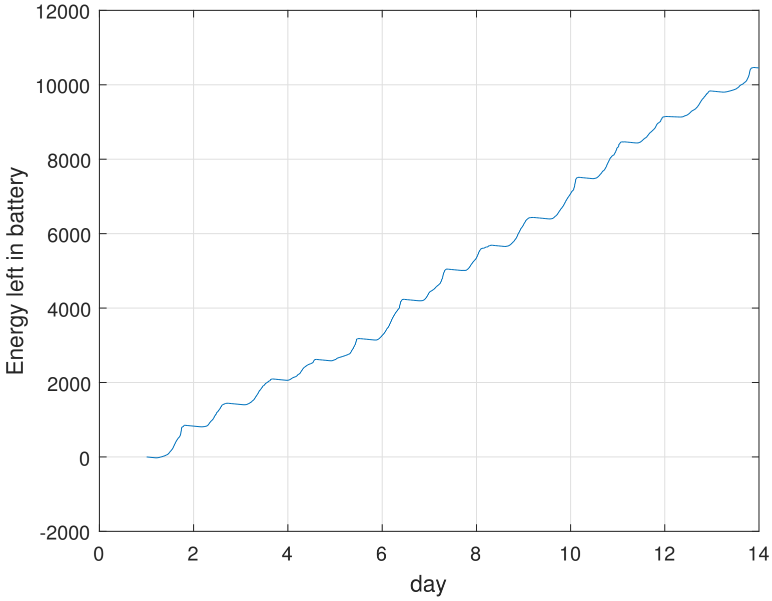

Figure 4.1: A plot of the estimated energy harvest when using a 55x70 mm solar panel.

Figure 4.1 shows the estimated energy harvest for office A, if a 55x70 mm solar panel was used over the course of 14 days. According to the estimate of Tirronen et al [1], the use of a 6500 J battery, with $n = 300$, in a 768 second period would allow for a battery life of 1.9 months. The figure shows the energy of the battery reaching a larger value than this within ten days, which suggests that it would be possible to charge a 6500 J battery to full capacity.

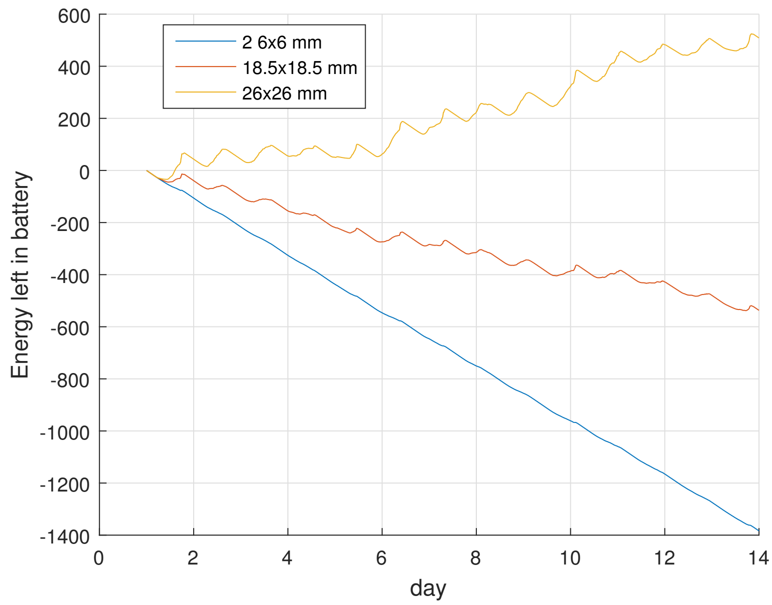

Figure 4.2: A plot of the estimated energy harvest when using different solar panels.

Figure 4.2 shows the estimated energy harvest for office A if a 26x26, 18.5x18.5 and two 6x6 mm solar panels [24] were used over the course of 14 days. Here, it can be observed that an excess of energy is still available when using a 26x26 mm panel. However, it can also be seen that days with a low harvest may cause a decline in available energy (day 4 and 5). This means that, although the battery will most likely be able to supply the device for a full day, it will still be possible for the device to lose power during days with a low harvest. The smaller panels here do not provide an increase in estimate of stored energy for the battery, indicating that the smaller panels are insufficient for the maximum consumption case. This is however not the case if less cycles are used.

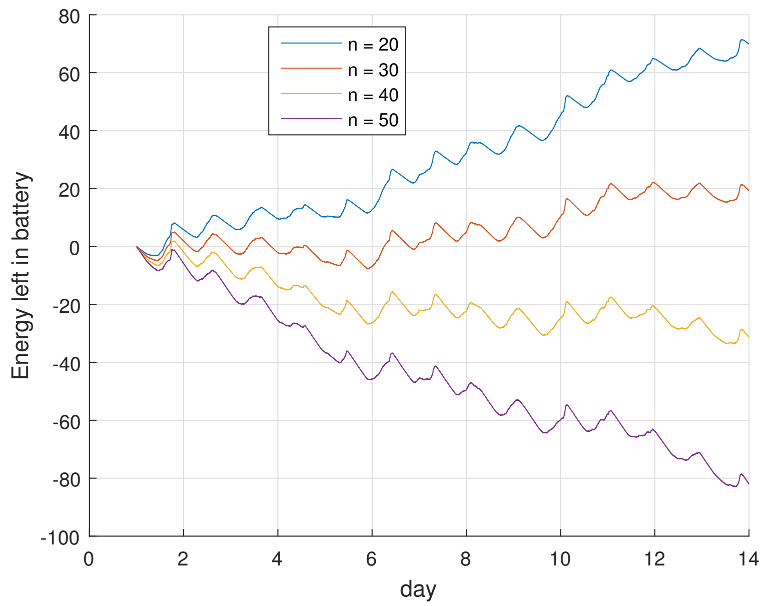

Figure 4.3: A plot of the estimated energy harvest when using 2 6x6 mm solar panels with varying cycles in a 768 second period.

Figure 4.3 shows the estimated energy harvest for office A, if two 6x6 mm solar panels over the course of 14 days were used while varying the parameter n. Here it is estimated that very small solar cells can be used while still providing the ability to supply enough power for the device and recharge its battery. Reducing the amount of cycles over the period of 768 seconds results in obvious improvements for battery life, allowing for smaller solar panels to be used while still maintaining the ability to increase the energy stored in the battery. An issue that may arise however, is how much energy the battery must be able to store. Insufficient measurements have been conducted throughout the duration of this project in order to obtain a proper estimate, which is why no suggestions will be made as to the recommended, total battery capacity.

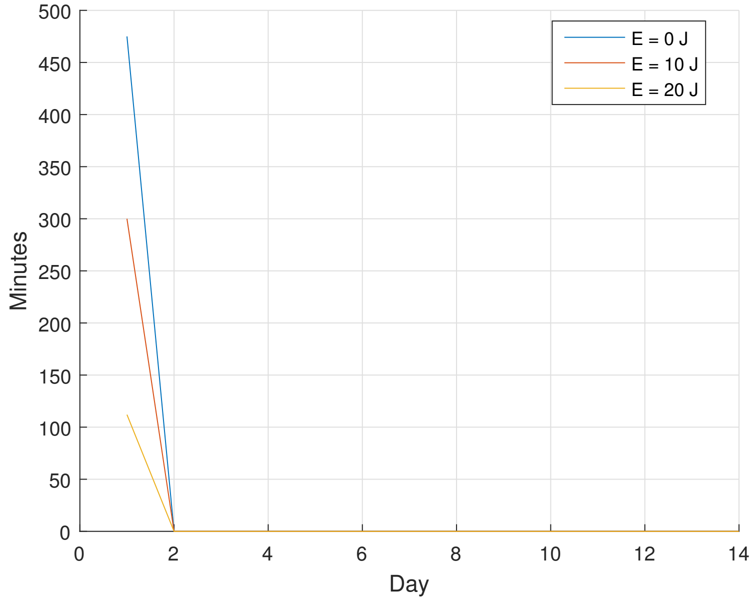

Figure 4.4: A plot illustrating how initial charge of the battery affects the estimated down-time of the device for a 55x70mm solar panel.

Figure 4.4 shows how the estimated down-time of the device decreases, as the initial battery charge increases when using a 55x70 solar panel, and a cycle of $n = 300$. Considering the results from Figure 4.1, it is reasonable that the only down-time that will occur is in the first day, since no initial charge of the battery was assumed.

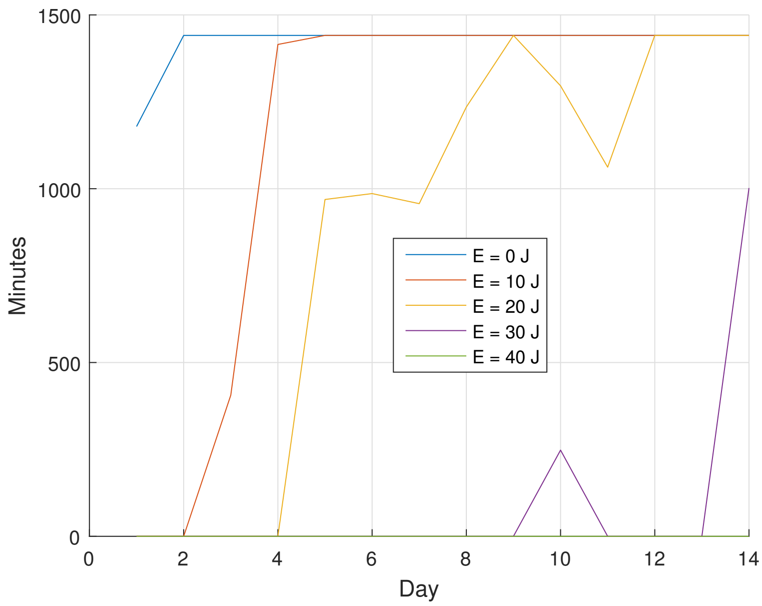

Figure 4.5: A plot illustrating how initial charge of the battery affects the estimated down-time of the device for two 6x6 solar panels.

Figure 4.5 shows how the estimated down-time of the device decreases as the initial battery charge increases, when using two 6x6 solar panels and a cycle of $n = 40$. If the results from Figure 4.3 are taken into consideration, it can be seen that the estimated energy remaining in the battery stays at a relative constant level after some days for $n = 40$. A negative level can be countered by adding a sufficient initial charge. Figure 4.5 shows that an estimated initial charge of 40 J in the battery is sufficient to allow the device to function for 14 days without any down-time during the month of May.

In the case of a large excess in energy, it may be preferable to increase the amount of transmissions in one day instead of adding constraints on the hardware. A device may require instructions for each of its transmissions, meaning that it would need to increase its transmission frequency. Another instance is if a device only transmits data, only requiring instructions every 768 seconds instead. Replacing some active periods with transmissions in the model, would give an estimate of the impact that this change would have on the overall energy consumption.

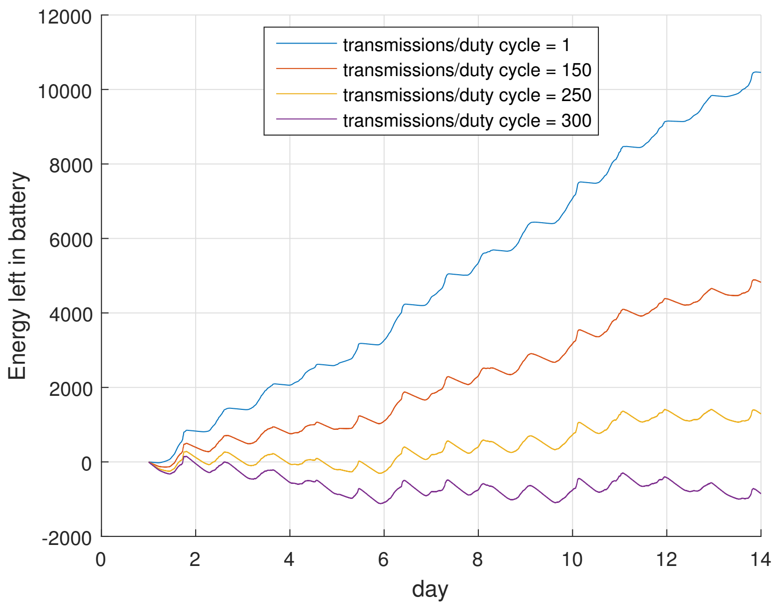

Figure 4.6: A plot illustrating the impact of an increase in transmissions, on the estimated battery life when using a 55x70 mm solar panel.

Figure 4.6 shows how an increase in transmissions every 768 seconds impacts the estimated battery life when using a 55x70 mm solar panel. It can be seen that if new instructions are required for each transmission, an excess in the estimated, available energy is still present. However, if a device's sole purpose is to transmit data, the estimated, available energy in office A would no longer produce an excess amount of energy.
Finally, consider the harvested energy from Figure 3.13 in section 3.3, where the estimated, total, harvested energy is approximately 1.1 J. When this harvest estimate is compared to the minimum requirements of Equation (4.10), it is clear that the estimated harvest in this case is somewhat lacking, meaning that some solar

irradiance is required in order for this model to work.

5

Prediction

This chapter will present the different algorithms that will be evaluated in this project, as well as a new one that was proposed. The performance of the proposed algorithm will also be compared to the other three algorithms.

5.1 Prediction algorithms

This section will present and explain the equations used for each individual algorithm.

5.1.1 Exponentially Weighted Moving Average

The Exponentially Weighted Moving Average (EWMA) algorithm uses exponentially decaying weights for maintaining a historical average for each prediction slot. One day is divided into N slots, with each slot updated according to

$$E(n+1) = \alpha \cdot E(n) + (1-\alpha) \cdot e(n+1) \tag{5.1}$$

where $E(n+1)$ is the predicted energy value for slot $n+1$, $e(n+1)$ is the weighted average of all previous days for slot $n+1$, and α is a weighting factor. [11]
The EWMA algorithm assumes that the energy generated during a time-slot for the previous day should be somewhat similar to the next. Seeing how the historical values decay at an exponential rate, the predicted values are mostly affected by a few days prior to the current one. This means that seasonal variations and the hours of which the sun is usually shining during the current day are easily taken into account.

5.1.2 Weather Conditioned Moving Average

The Weather Conditioned Moving Average (WCMA) algorithm has its foundations on EWMA. This algorithm stores N energy values for D past days, and uses a weighting factor similar to the EWMA algorithm. What is unique about the WCMA algorithm however, is the inclusion of a unique factor called GAP_k [12] as seen below

$$E(d, n+1) = \alpha \cdot E(d, n) + GAP_k \cdot (1-\alpha) \cdot M_D(d, n+1) \tag{5.2}$$

Where E(d,n) is the predicted energy value for sample n of day d and $M_D(d, n+1)$ is the mean of D past days at time-slot $n+1$ of the day, as seen in Equation (5.3)

$$M_D(d,n) = \frac{\sum_{i=d-1}^{d-D} E(i,n)}{D} \tag{5.3}$$

The factor GAP_k allows the WCMA algorithm to take the solar conditions of the current day relative to the previous days into account. This is done by calculating the quotient between the energy samples and the average energy of the previous D days for the K past samples, which is shown in Equation (5.4)

$$v_k = \frac{E(d, n-K+k-1)}{M_D(d, n-K+k-1)} \tag{5.4}$$

The quotients v are then stored in a vector V

$$V = [v_1, v_2, ..., v_K] \tag{5.5}$$

These quotients of vector V are then scaled depending on how recent those samples are, which is done by multiplying the k^{th} quotient with a weight. These weights are stored in vector P

$$P = [p_1, p_2, ..., p_K] \tag{5.6}$$

where,

$$p_k = \frac{k}{K} \tag{5.7}$$

The weighting factor GAP_k is then calculated as

$$GAP_k = \frac{V \cdot P}{\sum P} \tag{5.8}$$

5.1.3 Pro-Energy

Unlike EWMA and WCMA, Pro-Energy is an algorithm that allows for predictions for longer prediction horizons

$$\hat{E}_{n+i} = \gamma_i \cdot C_n + (1-\gamma_i) \cdot WP_{n+i} \tag{5.9}$$

$\hat{E}_{n+i}$ is the predicted energy at time-slot $n+i$ for the current day, C_n is the harvested energy during last time-slot for the current day, WP_{n+i} is a weighted profile for time-slot $n+i$, and γ_i is the correlation factor for the prediction of time-slot $n+i$ [13]. Here, $\gamma_i = \alpha$ when $i = 1$. For values of $i > 1$, γ_i is determined by

$$\gamma_n = \begin{cases} \alpha \cdot (1 - \frac{i-1}{G}) & \text{if } i \leq G \\ 0 & \text{if } i \geq G \end{cases} \quad \forall i \in 1 \leq i \leq F \tag{5.10}$$

where α is a weighting factor similar to the ones in EWMA and WCMA, i is the i^{th} time-slot after n, G is the number of time-slots in the future which show a correlation above a given threshold with time-slot n, and F is the number of future time-slots for which Pro-Energy is delivering energy predictions. The weighted profile, WP_{n+i} is calculated as

$$WP_{n+i} = \frac{1}{P-1}\sum_{j=0}^{P} w_j \cdot E^{d_j}_{n+i} \tag{5.11}$$

where P is the amount of stored profiles and w_j is

$$w_j = 1 - \frac{MAE_k(E^{d_j}, C)}{\sum_{j=0}^{P} MAE_k(E^{d_j}, C)} \tag{5.12}$$

Here, $MAE_k(E^{d_j}, C)$ represents the Mean Absolute Error (MAE) between E^{d_j} and C, where E^{d_j} is the j^{th} stored profile.

5.1.4 Proposed Approach: Weighted Min-Max Profile

This algorithm is inspired by the weighted profiling of Pro-Energy. In this case, the past D days are stored, and two profiles consisting of the minimum and maximum values for each time-slot are constructed. For example, if the minimum value for time-slot 24 was 1 and the maximum value was 10, then these are the values that are stored in the minimum and maximum profile respectively. The aim here is to construct a simplified version of the Pro-Energy algorithm, while still maintaining the ability to perform predictions over longer prediction horizons.

$$\hat{E}(n+i) = E(n) \cdot \alpha^i + (1-\alpha^i) \cdot w_n \cdot P_{n+i} \tag{5.13}$$

where $E(n+i)$ is the predicted energy for time-slot $n+i$, $E(n)$ is the measured energy in time-slot n, α is a weighting factor similar to that of WCMA, w_n is the weight of the profiles obtained from time-slot n and P_{n+i} is the minimum and maximum profile value for time-slot $n+i$.

The matrix P is denoted as

$$P_{n+i} = \begin{bmatrix} E(n+i)_{min} & E(n+i)_{max} \end{bmatrix}^T \tag{5.14}$$

and the vector w is denoted as

$$w_n = \begin{bmatrix} w_{min} & w_{max} \end{bmatrix} \tag{5.15}$$

It is also possible to determine the weights by calculating the the difference between the measured time-slot $E(n)$ and the profiles at same time-slot P_n, which is done according to

$$dist_{min} = \frac{1}{K}\left(\sum_{k=1}^{K} E(n-k)_{max} - E(n-k)\right)$$
$$dist_{max} = \frac{1}{K}\left(\sum_{k=1}^{K} E(n-k) - E(n-k)_{min}\right)$$

where K is the amount of past time-slots, $dist_{min}$ and $dist_{max}$ are the average distances between the past K measured time-slots and the minimum and maximum profiles respectively. The weights are then calculated as

$$w_n = \left[\frac{dist_{min}}{E(n)_{max} - E(n)_{min}} \frac{dist_{max}}{E(n)_{max} - E(n)_{min}}\right] \tag{5.16}$$

This means that each weight is based on how similar the minimum and maximum profiles are at the current time-slot. These weights are then updated for each time-slot in order to keep them up to date.

Note that when performing predictions for time-slots where $i > 1$, it is important to place a larger emphasis on the profile instead of the measurement, since the value of the measurement is less accurate for these time-slots. By setting the weight α raised to the power of i, this would allow for a reduced impact of the measurement from time-slot n for longer prediction horizons.

5.2 Performance of algorithms

When conducting the performance test for the previously mentioned algorithms, no parameter tuning will be performed. The idea here is to compare how these prediction algorithms work with one another, and not how each algorithm can be improved by tuning. The parameter α decides how much emphasis shall be placed on the current measurement and the prediction profile. A large value for α would lead to a larger emphasis on the measurement, and a small value would lead to a larger emphasis on the prediction profile. Seeing how the assumption for the prediction algorithms considers small changes in available energy for each time-slot, a larger emphasis will be placed on the measurement, while allowing for the prediction profile to still play a role in the estimation. For this reason, $\alpha = 0.7$ will be used for all algorithms.

The average error for day d can be calculated as

$$Error_d = \frac{1}{N} \sum_{n=1}^{N} abs \left(1 - \frac{E(n)}{\hat{E}(n)} \right) \tag{5.17}$$

where N is the amount of time-slots for one day, $E(n)$ is the measured value for time-slot n, and $\hat{E}(n)$ is the predicted value for time-slot n [12]. Calculating the Mean Absolute Percent Error (MAPE) for all D_{comp} days being compared can done by

$$MAPE = \frac{1}{D_{comp}} \sum_{d=1}^{D_{comp}} Error_d \tag{5.18}$$

When displaying the results of the algorithms, only the final four days of measurements will be plotted. The MAPE-values will be calculated over the final eight days for each algorithm.

Note that unless stated otherwise, the parameters used for D and K are $D = 5$, and $K = 2$. The effect of these two parameters will be examined for changes in accuracy and computation time.

5.2.1 Time-slot Duration

The chosen time-slot will be based on the how well the prediction algorithms perform on data obtained from office A. This is done in order to assess an appropriate time-slot duration for the prediction algorithm performance, which will be presented in more detail in Section 5.2.6.

Table 5.1: A table showing the changes in accuracy for increasing the time between time-slots. The values shown here are the MAPE-values, which represent the error percent. (lower is better)

time-slot duration	1 min	15 min	30 min	60 min
EWMA	82.351	75.04	44.072	41.77
WCMA	7.6434	31.741	31.603	45.445
Pro-energy	15.06	31.096	28.845	41.432
Min-Max	8.6849	30.399	29.991	42.44

Table 5.2: A table showing the changes in computation time for increasing the time between time-slots. The values here are shown in milliseconds. (lower is better)

time-slot duration	1 min	15 min	30 min	60 min
EWMA	0.9	0.6	0.6	0.6
WCMA	116.3	8.4	4.4	2.4
Pro-energy	3092.2	158.2	74.7	35
Min-Max	402.6	26.7	13.6	6.9

Tables 5.1 and 5.2 show how the accuracy and computation time change for varying durations for each time-slot respectively. Computation times were obtained using the system in Table 5.3, which can be found in Section 5.2.6. It can be seen that while the accuracy of most predictions are at their highest for one minute time-slots, it can also be observed that this time-slot duration leads to a high computation time. A time-slot duration of 30 minutes allows for a significant reduction in computation time, while still allowing for an accuracy that competes with a 15 minute time-slot. For this reason, the performance algorithms will be examined using 30 minute time-slots.

5.2.2 Exponentially Weighted Moving Average

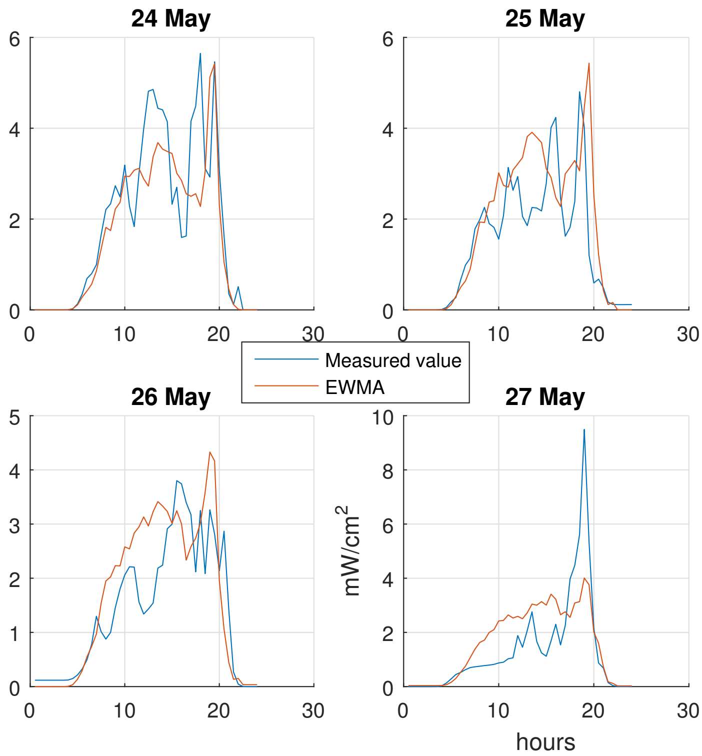

Figure 5.1: Four plots illustrating the performance of EWMA.

Figure 5.1 shows four plots, illustrating the performance of the EWMA prediction algorithm, giving a MAPE of 44.072%.

5.2.3 Weather Conditioned Moving Average

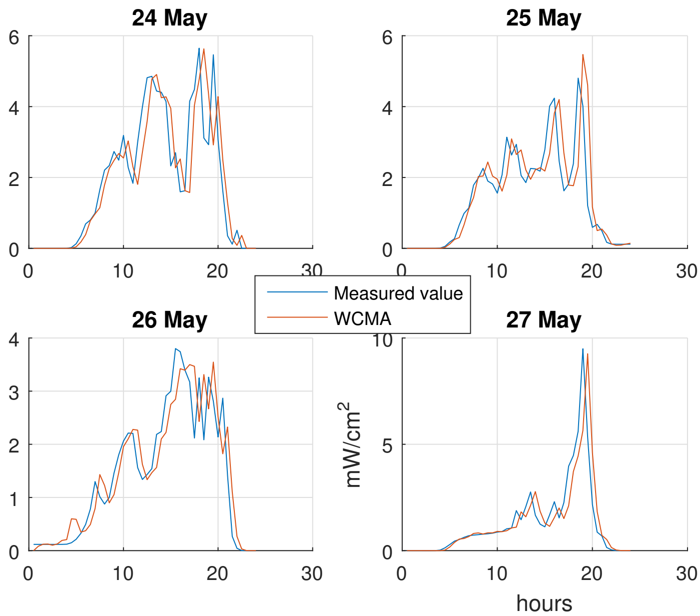

Figure 5.2: Four plots illustrating the performance of WCMA.

Figure 5.2 shows four plots, illustrating the performance of the WCMA prediction algorithm, giving a MAPE of 31.603%.

5.2.4 Pro-energy

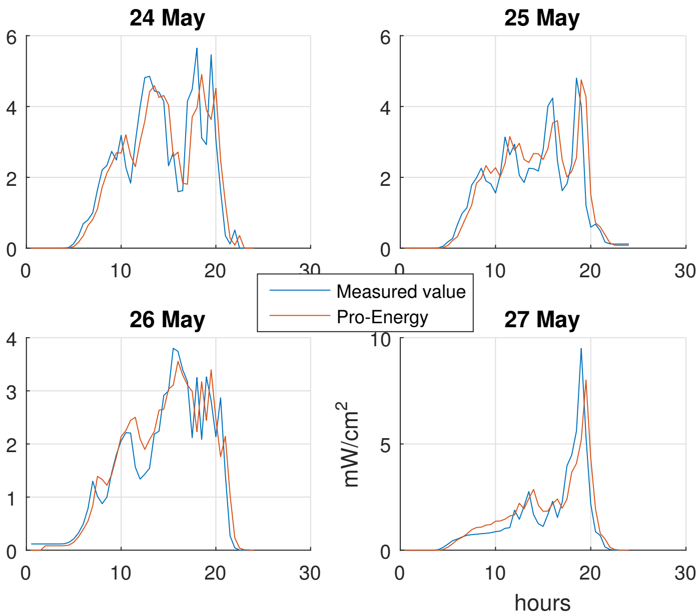

Figure 5.3: Four plots illustrating the performance of Pro-Energy.

Figure 5.3 shows four plots, illustrating the performance of the Pro-Energy prediction algorithm, giving a MAPE of 28.845%. This MAPE-value is somewhat larger than that obtained from Cammarano et al. [13]. While the parameters are quite different than those used by the article, it is also likely that the character of changes could be slightly less predictable for this project, resulting in a higher MAPE-value. Furthermore, no updating for the stored profiles was implemented in order to minimize the required computations. Instead, only the past D days are used for profiling. However, it can be seen here that the performance of the Pro-Energy algorithm produces slightly less errors than the WCMA algorithm overall when only predicting the next time-slot. This is an indication that the algorithms do not contain any faults in implementation, but rather less effective for the used data-sets and parameters.

5.2.5 Proposed Approach: Weighted Min-Max Profile

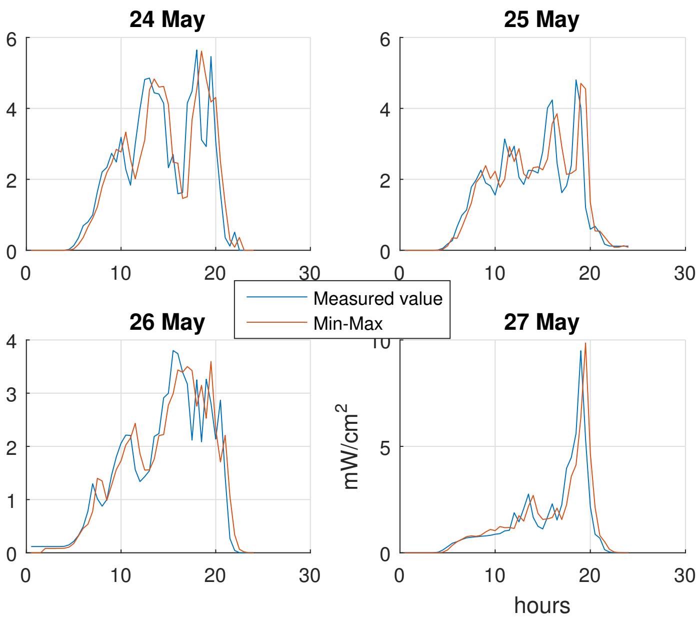

Figure 5.4: Four plots illustrating the performance of Min-Max.

Figure 5.4 shows four plots, illustrating the performance of the Min-Max prediction algorithm, giving a MAPE of 29.991%.

5.2.6 Comparison

In order to obtain an accurate estimate of the algorithms' performance, the data-set ¨*SetupC_merged*_2009_11_7_2010_9_13.*txt*" contributed by Gorlatova et al, of 261 days, consisting of indoor solar irradiance measurements, was used [25]. Furthermore, the computation times were obtained from a desktop computer using the specifications shown in Table 5.3

Table 5.3: A table showing the specifications of the computer used for obtaining the computation time of the algorithms.

Operating System	Windows 10 Education 64-bit
CPU Processor	Intel Core i7-6700K (not overclocked)
RAM Memory size	16 GB
RAM Memory frequency	2133 MHz
Motherboard	MSI Z170A Gaming M5
GPU	Gigabyte AMD Radeon HD 7870 2GB GDDR5
GPU frequency	Base frequency (not overclocked)

The results are displayed in the tables below.

5.2.6.1 Prediction of time-slot n+1

Table 5.4: A table of the MAPE-values for all four algorithms using the data-set contributed by Gorlatova et al.

Algorithm	MAPE (%)
EWMA	59.797
WCMA	16.139
Pro-Energy	20.98
Min-Max	15.872

According to the MAPE-values in Table 5.4, the EWMA gives the least accurate predictions, whereas Min-Max gives the most accurate ones. This is not the case for the measurements taken for office A in this project, as seen in Sections 5.2.2 through 5.2.5, since the most accurate algorithm in that case is Pro-Energy. While this may mean that Pro-Energy might be better suited for the examined area, it is also worth considering the fact that only 8 days were tested for the MAPE-values in office A, as opposed to over 250 days for the results of the data-set shown in Table 5.4.

Table 5.5: A table of the computation times for all four algorithms using the data-set contributed by Gorlatova et al.

Algorithm	computation time (ms)
EWMA	3.2
WCMA	176.5
Pro-Energy	27559
Min-Max	723.1

As can be seen in Table 5.5, it is clear that EWMA requires the least computation time in order to complete, and Pro-Energy requires the most. For small devices, it is convenient to minimize all unnecessary energy consumption, while at the same time maximizing efficiency. This means that it would be preferable to use the least computation time possible while obtaining the most accurate predictions. The WCMA algorithm would be the preferable case here, since its computation time is much lower than Min-Max, while offering MAPE-values that do not have a large deviation from Min-Max. However, this algorithm cannot perform any predictions for time-slots after the next one. This means that if predictions of time-slots with longer prediction horizons are required, the most preferable algorithm among the four mentioned ones would be the Min-Max algorithm from a computational point of view. The Min-Max algorithm takes about several times less to complete when compared to Pro-Energy, while still providing competetive, if not improved MAPE-values.

Table 5.6: A table of the MAPE for Pro-Energy and Min-Max using the data-set contributed by Gorlatova et al.

Algorithm	MAPE
Pro-Energy	20.238
Min-Max	19.405

Table 5.6 shows the MAPE-values for Pro-Energy and Min-Max when the parameter D was reduced from 5 till 2. Here, it can be seen that the Min-Max algorithm performs better than Pro-Energy when fewer days are available for profiling.

Table 5.7: A table of the MAPE-values in office A for WCMA, Pro-Energy and Min-Max with a D-parameter value of 2.

Algorithm	MAPE
WCMA	32.588
Pro-Energy	29.596
Min-Max	30.581

Table 5.7 shows that for office A, the Min-Max algorithm displays slightly lower accuracy than that of Pro-Energy, and improvements over WCMA.

Table 5.8: A table of the MAPE-values in office B for WCMA, Pro-Energy and Min-Max with a D-parameter value of 2.

D-parameter	5	2
WCMA	38.244	45.653
Pro-Energy	47.478	51.62
Min-Max	41.411	48.815

Table 5.9: A table of the MAPE-values in office C for WCMA, Pro-Energy and Min-Max with a D-parameter value of 2.

D-parameter	5	2
WCMA	40.918	44.043
Pro-Energy	47.605	48.249
Min-Max	43.802	45.026

Tables 5.8 and 5.9 show that for both offices B and C, Min-Max displays higher accuracy than that of Pro-Energy, but lower accuracy than WCMA.

So far, the Min-Max algorithm has provided MAPE-values in between Pro-Energy and WCMA when performing predictions for the next time-slot. This suggests that, for an area with higher variations in irradiance, predictions performed by WCMA are preferable, whereas for an area with lower variations in irradiance, Min-Max shows signs of improvements over both WCMA and Pro-Energy.

Table 5.10: A table of the computation times in milliseconds for the entire procedure, comparing WCMA, Pro-Energy and Min-Max using the data-set contributed by Gorlatova et al, with varying parameter D.

D-parameter	5	2
WCMA	175.8	172
Pro-Energy	27559	26754
Min-Max	723.1	616.2

Table 5.10 quantifies the decrease in the computation time for a smaller value of the parameter D. This indicates that it is desirable to reduce this parameter, since doing so would reduce power consumption.

Table 5.11: A table of the computation times in milliseconds for the entire procedure, comparing WCMA, Pro-Energy and Min-Max using the data-set contributed by Gorlatova et al, with varying parameter K.

K-parameter	2	4	6
WCMA	172	193.96	212.2
Pro-Energy	26780	26741	26595
Min-Max	610	603.3	614

Table 5.11 shows how the computation time increases with increasing values of K. Here, the parameter value of D remains $D = 2$. These results indicate minor changes in computation time when changing the values of K while varying between K = 2, 4 or 6.

Table 5.12: A table of the MAPE of Pro-Energy and Min-Max with varying values for the K-parameter, using the data-set contributed by Gorlatova et al.

K-parameter	2	4	6
WCMA	20.493	20.571	20.72
Pro-Energy	20.238	20.389	20.506
Min-Max	19.405	20.16	38.949

Negligible changes were observed in accuracy upon increasing the K-parameter for WCMA and Pro-Energy. For Min-Max on the other hand, an increased value for K resulted in a deterioration in accuracy. This suggests that a lower value for the parameter K is desirable in terms of both accuracy and energy consumption.

5.2.6.2 Prediction of time-slot n+i

All four algorithms so far have displayed the ability to predict the next time-slot. However, only Pro-Energy and Min-Max can provide the ability to perform predictions when longer prediction horizons are required.

Table 5.13: A table presenting the MAPE-values for an increased prediction horizon with parameter $D = 2$, using the data-set contributed by Gorlatova et al.

Time-slot	Pro-Energy	Min-Max
n + 2	97.718	36.369
n + 3	99.328	42.132
n + 4	100.76	52.696

Table 5.13 shows the MAPE-values for an increased prediction horizon with up to three time-slots after the next one. Here, it can be seen that the performance of the Min-Max algorithm shows an improvement over the Pro-Energy algorithm. This is to be expected, since Pro-Energy is designed for using a profile consisting of multiple days in order to obtain an accurate estimate.

Table 5.14: A table presenting the MAPE-values for an increased prediction horizon with parameter $D = 5$, using the data-set contributed by Gorlatova et al.

Future time-slot	Pro-Energy	Min-Max
n + 2	56.974	25.283
n + 3	57.51	31.931
n + 4	57.941	37.565

Table 5.13 shows the MAPE-values for an increased prediction horizon with up to three time-slots after the next one and an increased value for the parameter D. Here it can be seen that the performance of the Min-Max algorithm still shows improvements over Pro-Energy. It can also be observed that improvements arise when increasing the amount of days available for building a profile. Pro-Energy's design allows for small deterioration in accuracy when increasing the prediction horizon. As can be seen in the tables above, it can be observed that Min-Max shows a quicker increase in MAPE-values when compared to Pro-Energy.

So far, the Min-Max algorithm has shown promise by delivering lower error rates when compared to the Pro-Energy algorithm. However, the data-set contributed by Gorlatova et al has somewhat smaller variations than that of the measured values in office A.

Table 5.15: A table presenting the MAPE-values for an increased prediction horizon with parameter $D = 5$, using the data from office A over the course of eight days.

Future time-slot	Pro-Energy	Min-Max
n + 2	43.31	43.18
n + 3	43.882	52.24
n + 4	44.236	62.876

Table 5.15 the MAPE-values for an increased prediction horizon for office A. Clearly, the Min-Max algorithm does not perform as well for this office as it did for the area used from the data-set contributed by Gorlatova et al.

Table 5.16: A table presenting the MAPE-values for an increased prediction horizon with parameter $D = 2$, using the data from office A over the course of eight days.

Future time-slot	Pro-Energy	Min-Max
n + 2	64.921	48.189
n + 3	66.394	62.035
n + 4	68.486	66.391

Table 5.16 the MAPE-values for an increased prediction horizon for office A, with a reduced value for the D-parameter. It can be seen that the performance of the Min-Max is reduced here due to the reduction of the D-Parameter. Despite this reduction in performance, it can be observed that Min-Max produces an improved accuracy over Pro-Energy when using a lower value for the parameter D. Note however that although Min-Max provides lower MAPE-values than Pro-Energy when $D = 2$, Pro-Energy still offers higher accuracy when $D = 5$. This indicates that, for longer prediction horizons, Pro-Energy offers a more stable deterioration than Min-Max and has the ability to provide more accurate predictions if the D-parameter is not constrained.

The Min-Max algorithm is designed in such a manner, that it uses weights for the highest and lowest recorded value for each time-slot, in the past D days in order to determine its prediction. This means that this algorithm relies heavily on the assumption that future time-slots will have the same distance from the minimum and maximum profile built over the past D days. In other words, the Min-Max algorithm only constructs a profile based on two profiles (minimum and maximum values for each time-slot), which means that large deviations have a great impact. On the other hand, the Pro-Energy algorithm bases its predictions by weighting all D stored profiles individually. This means that the predictions made from Pro-Energy rely heavily on the character of how those days change, instead of assuming an interval to base its prediction on. Here, all deviations for the D stored days are weighted individually. This indicates that an improved prediction accuracy can be achieved over the Min-Max algorithm when variations in weather conditions are high, since all variations of each time-slot are weighted individually.

From the obtained results, it can be suggested that WCMA shows signs of outperforming Min-Max when the variations in available solar irradiance is high, whereas Min-Max shows signs of outperforming WCMA for more stable changes. When comparing the computational cost of WCMA and Min-Max, it can be seen that WCMA offers a clear reduction in computation time, making it preferable for short-term prediction. However, WCMA is unable to perform predictions for longer prediction horizons, while Pro-Energy and Min-Max can. For larger horizons, Min-Max offers improvements in predictions over Pro-Energy when the amount of available D days for profiling is low, while at the same time offering a significantly lower computational cost. However, Min-Max is unable to outperform Pro-Energy if the parameter D is not constrained.

In conclusion, The Min-Max algorithm offers a balance between the benefits of WCMA and Pro-Energy. Min-Max provides the ability to perform predictions for longer prediction horizons, which WCMA cannot do. Furthermore, Min-Max shows signs of improved accuracy when the available D days are limited, while still maintaining a lower computational cost than Pro-Energy.

6
Conclusion

This chapter will summarise the conclusions of the previous chapters, and state in short the reasons behind these conclusions.

6.1 Hardware

It was concluded that the use of an Arduino Uno proved to be beneficial over a smaller micro-controller, such as the ATtiny85, due to the popularity of Arduino devices. The Arduino Uno comes complete with an oscillator crystal, USB-socket and various purchasable modules available at local hobby stores. On-board components, such as a voltage-regulator allowed for avoiding issues with jitter. Furthermore, the open-source libraries, along with the development platform, offered a convenient method of programming all of the necessary hardware.

It was also concluded that the TSL230BRD sensor was the most appropriate choice for sampling the available light energy. This sensor was only available for purchase using a SOIC-socket, which is why it was soldered onto a socket adapter in order to use it with a DIP-socket breadboard. Conducting measurements with the sensors without using a capacitor showed undesirable qualities when examining the logged data. Furthermore, shadow from the wiring around the measurement sensors resulted in a negative impact on the quality of the measurements. It was concluded that the effects of these factors could not be ignored, and precautions to prevent undesirable behaviour was taken in the form of improving the wiring and using capacitors for all sensors, as proposed by the specification.

6.2 Characterisation

Characterising the available light energy was done by constructing a profile. Each day contained a total of 48 time-slots, where each time-slot represents the mean value of the past half hour. By calculating the mean value for all days in each time-slot, a profile was obtained that gave a reasonable estimate for the character of sunlight. It was concluded that, while the mean gave an illustration as to how the character of available light energy typically changed during a day, values representing how the light varied for each time-slot were also necessary. Calculating the standard deviation for each time-slot gave insight as to how much the value of each time-slot typically varied throughout the day.

Depending on the angle that the measurements were taken, it was suggested that an increase in available energy could be achieved by prioritising non-peak hours over peak hours. It was estimated that more energy could be harvested overall if the duration of direct sunshine was low.

6.3 Machine-to-Machine Communications and Energy Consumption

A model of a low-power device using Machine-to-Machine (M2M) communications, with varying active periods was used. The energy consumption of this model was compared to the estimate of harvested energy, in order to asses whether the energy requirements for indefinite use were fulfilled. It was concluded that the estimated available energy provided sufficient energy for indefinite use of a low-power wireless sensor, depending on the size of the solar panel and the frequency of active periods chosen for the M2M model. However, results showed that some down-time was still present, despite the estimate suggesting fulfilled requirements. This issue was a consequence of a missing initial charge of the battery. It was concluded that an initial charge was mandatory in order to avoid initial down-time. In this case, it was assumed that the low-power sensor began its task during night time, where measurements showed no available energy in the environment.

6.4 Prediction

The four algorithms tested were Exponentially Weighted Moving Average (EWMA), Weather Conditioned Moving Average (WCMA), Pro-Energy and a weighted Min-Max profile algorithm. Min-Max was a proposed algorithm which was inspired by Pro-Energy, and was designed to require less memory and computational resources. All four algorithms provided the ability to perform a prediction for the next time-slot, but only Pro-Energy and Min-Max allowed for predictions of larger prediction horizons.

WCMA, Pro-Energy and Min-Max construct a profile using the past D days for each time-slot. A prediction value was obtained by summing two weighted terms, where one term was the measurement of the current time-slot, and the second was the profile value of the next time-slot. A weight, α for WCMA and γ for Pro-Energy, adjusted how much emphasis was placed on each term. The γ weight is the same as α when predicting the next time-slot, and different when predicting time-slots after the next one.

It was determined that the Min-Max algorithm offered a balance between WCMA and Pro-Energy, by providing the ability to perform predictions over longer prediction horizons, while at the same time reducing the computational cost. Furthermore, Min-Max showed signs of improved performance when the amount of available days were limited. It was concluded that the method used for determining the profile weights used in Min-Max, made the algorithm sensitive for small changes of the profile for each time-slot. While this approach proved beneficial for predictions in an environment where the variations in weather conditions were not too high, results for long-term predictions indicated otherwise. It was established that this algorithm was not preferable for long-term predictions, with increased variations in weather conditions, for the very reason that made it more suitable for short-term predictions. According to the design of Min-Max, the weights used for the next time-slot are also used for the entire prediction horizon. Using this approach, without adapting the weights for larger horizons, resulted in a rapid deterioration of accuracy. However, when the amount of D days available were constrained, this algorithm showed signs of outperforming Pro-Energy. This meant that although it may be preferable to use Pro-Energy for long-term predictions in highly varying weather conditions, it was suggested that Min-Max would still be preferable in the case of a constrained amount of D days available for profiling.

6.5 Future Work

In this work, measurements were conducted in a select few offices, with restricted conditions. The constructed sunlight profiles used could only give a representation for the month of May. Performing longer term measurements, that cover an entire year would provide a more accurate estimate for the available light energy, allowing for improved estimations for energy requirements of low-power sensors. Furthermore, the proposed prediction algorithm, Min-Max, could not provide improved prediction values for long prediction horizons with highly varying weather conditions, when using an unconstrained D days for profiling. Implementing a method for determining unique weights for each time-slot used in the prediction horizon would allow for an improvement in prediction accuracy. While this improvement would be beneficial, it would also be valuable to compare the results of the improved Min-Max weighting method with other prediction algorithms commonly used for predicting solar irradiance. Finally, this thesis has provided insights and estimations as to what can be done. Realising these estimations by constructing solar powered, low-power sensors, using M2M communication would allow for a conformation of the presented possibilities.

Bibliography

[1] T. Tirronen, A. Larmo, J. Sachs, B. Lindoff, and N. Wiberg, “Machine-to-machine communication with long-term evolution with reduced device energy consumption,” *Trans Emerging Tel Tech*, vol. 24, pp. 413–426, 2013.

[2] A. Wood, “The internet of things is revolutionising our lives, but standards are a must | media network | the guardian,” May 2016. [Online]. Available: http://www.theguardian.com/media-network/2015/mar/31/the-internet-of-things-is-revolutionising-our-lives-but-standards-are-a-must

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[4] Intel, “The internet of things is on the road to autonomous driving,” May 2016. [Online]. Available: http://www.intel.com/content/www/us/en/internet-of-things/infographics/iot-autonomous-driving-infographic.html

[5] W. K. G. Seah, Z. A. Eu, and H.-P. Tan, “Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges,” *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009*, pp. 1–5, 2009.

[6] W. S. Wang, T. O’Donnell, L. Ribetto, B. O’Flynn, M. Hayes, and C. O’Mathuna, “Energy harvesting embedded wireless sensor system for building environment applications,” *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009*, pp. 36–41, 2009.

[7] P. I. U. Ltd., “Solar measuring device pce-spm1 | pce instruments,” May 2016. [Online]. Available: https://www.pce-instruments.com/english/measuring-instruments/test-meters/solar-measuring-device-pce-instruments-solar-measuring-device-pce-spm-1-det_62041.htm

[8] A. Ltd, “Single channel solar irradiance logger for industrial and commercial appliances,” May 2016. [Online]. Available: http://www.electrocorder.com/products/details/c/renewable-energy-loggers/p/single-channel-solar-irradiance-sr1r

[9] Onset, “Solar radiation data loggers & sensors | onset data loggers,” May 2016. [Online]. Available: http://www.onsetcomp.com/products/sensors/solar-radiation?taxonomy_vocabulary_8_tid_1=547&sort_bef_combine=created+ASC&items_per_page=16

[10] ——, “Hobo pendant temperature/light data logger 8k,” May 2016. [Online]. Available: http://www.onsetcomp.com/products/data-loggers/ua-002-08

[11] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Transactions on Embedded Computing Systems (TECS) - Special Section LCTES’05*, vol. 6, no. 4, p. 19, 2007.

[12] J. R. Piorno, C. Bergonzini, D. Atienza, and T. S. Rosing, “Prediction and Management in Energy Harvested Wireless Sensor Nodes,” *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009.*, no. 6 - 10, pp. 6–10, 2009.

[13] A. Cammarano, C. Petrioli, and D. Spenza, “Pro-Energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks,” *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pp. 75–83, 2012.

[14] Instructables, “Technology arduino featured instructables,” May 2016. [Online]. Available: http://www.instructables.com/tag/type-id/category-technology/channel-arduino/

[15] H. low tech group, “Programming an attiny w/ arduino 1.6 (or 1.0),” April 2016. [Online]. Available: http://highlowtech.org/?p=1695

[16] Arduino, “Arduino - home,” May 2016. [Online]. Available: https://www.arduino.cc/

[17] K. . Company, “Micro-sd-kortläsare för arduino - moduler | kjell.com,” May 2016. [Online]. Available: https://www.kjell.com/se/sortiment/el-verktyg/elektronik/arduino/moduler/micro-sd-kortlasare-for-arduino-p87983

[18] AMS, *TSL235R Light-to-Frequency Converter*, AMS.

[19] ——, *TSL230RD, TSL230ARD, TSL230BRD Programmable Light-to-Frequency Converters*, AMS.

[20] M. Gorlatova, A. Wallwater, and G. Zussman, “Networking Low-Power Energy Harvesting Devices: Measurements and Algorithms,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1853–1865, 2012.

[21] Atmel, *ATmega48A/PA/88A/PA/168A/PA/328/P*, Atmel.

[22] M. Areskoug, *Miljöfysik*. Studentlitteratur AB, Lund: Studentlitteratur, 2012.

[23] Seeed, “Buy 0.5w solar panel 55x70 [313070004] | seeedstudio,” May 2016. [Online]. Available: http://www.seeedstudio.com/depot/05w-solar-panel-55x70-p-632.html?cPath=155

[24] IXYS, *IXOLAR High Efficiency Solar Cells*, IXYS.

[25] M. Gorlatova, M. Zapas, E. Xu, M. Bahlke, I. Kymissis, and G. Zussman, “Crawdad dataset colombia/enhants,” May 2016. [Online]. Available: https://crawdad.cs.dartmouth.edu/columbia/enhants/20110407/energy/