NTNU – Trondheim
Norwegian University of
Science and Technology

Modelling of Photovoltaic Modules with Battery Energy Storage in Simulink/Matlab

With in-situ measurement comparisons

Håvard Breisnes Vika

Master of Energy and Environmental Engineering
Submission date: June 2014
Supervisor: Ole-Morten Midtgård, ELKRAFT
Co-supervisor: Iromi Ranaweera, ELKRAFT

Norwegian University of Science and Technology
Department of Electric Power Engineering

Sammendrag

Bruken av fornybare energiressurser er økende og de vil spille en sentral rolle i fremtidens energisystemer. Solenergi er variabel og umulig å kontrollere. Energisystemer med mye solcelleanlegg vil derfor trenge energilagring når andelen solenergi øker.

En solcellemodell med fem parametere har blitt implementert i Simulink/Matlab. Variablene i modellen bestemmes av et skript som tilnærmer verdiene slik at modellen passer til spesifikasjoner fra datablader. Solinnstråling og lufttemperatur er modellens input. De interessante verdiene ut fra modellen er først og fremst effekt, celletemperatur og spenning, men alle variablene kan måles hvis ønskelig. Virkningene av å endre modellparameterne er blitt demonstrert og en MPPT-algoritme basert på P&O-metoden er implementert.

En batterimodell basert på data fra utladningskurver fra datablader er blitt implementert slik at det er en grunnleggende battericelle som blir modifisert for å konstruere forskjellige batterimoduler. En effektutjevningsalgoritme som går ut på at ta gjennomsnittet av effekten en gitt tid tilbake er også implementert. Denne algoritmen gir en effektreferanse til batterisystemet.

Resultater fra solcellemodellen har blitt sammenlignet med målte verdier fra samme type modul. Det er brukt målinger fra to forskjellige dager, en med veldig variabel innstråling, og en mer stabil innstråling. Det er liten forskjell mellom simulert effekt og målt effekt. Forskjellen er stort sett nede i noen få prosent. Noen store og korte avvik er til stede. Disse kommer sannsynligvis av at MPPT-algoritmen ikke klarer å holde følge med de hurtige endringene i solinnstråling.

Tilnærmingen for celletemperatur avviker mye fra målte verdier. Ved simulering med målte celletemperaturer, bekreftes det at denne dårlige tilnærmingen er en av de største kildene til avvik.

Virkningen av effektutjevningen er demonstrert med flere forskjellige batteristørrelser. Det blir demonstrert at å endre på hvor langt tilbake gjennomsnittet strekker seg i effektutjevningsalgoritmen bør tilpasses batteristørrelsen.

Abstract

The use of renewable energy sources is increasing and will play an important role in the future power systems. The unpredictable and fluctuating nature of solar power leads to a need for energy storage as the prevalence increases.

A five parameter model of PV modules has been implemented in Simulink/Matlab. The parameters of the model are determined by an approximation method using data sheet values. Inputs to the model include light intensity and ambient temperature. The outputs are any measurements of interests as well as power, cell temperature and voltage. Effects of varying the model parameters are demonstrated. A maximum power point tracking algorithm is used to keep the voltage at the maximum power point at all times.

A battery model based on discharge curve fitting is implemented. The model is based on a fundamental battery cell which can be modified to construct many different module configurations. Power smoothing algorithms which average the input over a set time, are used to provide a power reference to the battery system.

The PV model power output is compared to in-situ measurements by giving the model inputs of measured irradiance profile and ambient temperature. Measurements of two different days, one with little variation in irradiance and one with a lot of variation, are used to shed light on different effects. The difference between the output of the simulation and the measured values is very small, in the range of a few percent, especially when there is little variation in irradiance. Large peaks in difference are probably caused by the maximum power point tracking not being able to follow rapidly changing conditions.

The approximation used in the modelling of cell temperature deviates significantly from the measured cell temperature. This is confirmed as one of the largest causes of deviance by running simulations with actual measured cell temperature.

Power smoothing efficiency is demonstrated with different battery module sizes. Changing the power smoothing algorithm parameters to suit the battery capacity is shown to be effective in providing as good smoothing as possible within battery constraints.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

1.1 Context and perspective

Emissions of CO_2 and other greenhouse gasses are leading to climate change[2]. The greenhouse effect of CO_2 is well understood and it is clear that the emissions must be reduced to avoid undesirable scenarios. To reverse the trend, there is a great need for accelerating the development and implementation of renewable energy technologies.

Use of renewable energy sources is rising at an increasing rate. The global PV capacity is escalating rapidly with an average annual growth for the last decade of 40%[3]. In figure 1.1 the development of total capacity up to year 2008 is shown. Predictions by IEA state that this trend will continue and that photovoltaics will provide 11% of the global electricity generation by 2050[3].

Output from PV systems can change rapidly because of cloud movements and other factors. If these systems are prevalent enough in a power system, problems with voltage and frequency control can arise[4]. On-site energy storage can be used to mitigate these issues.

Both photovoltaic energy and energy storage systems will play a major role in the future energy system. Energy storage technologies was the main theme in the specialization project[5]

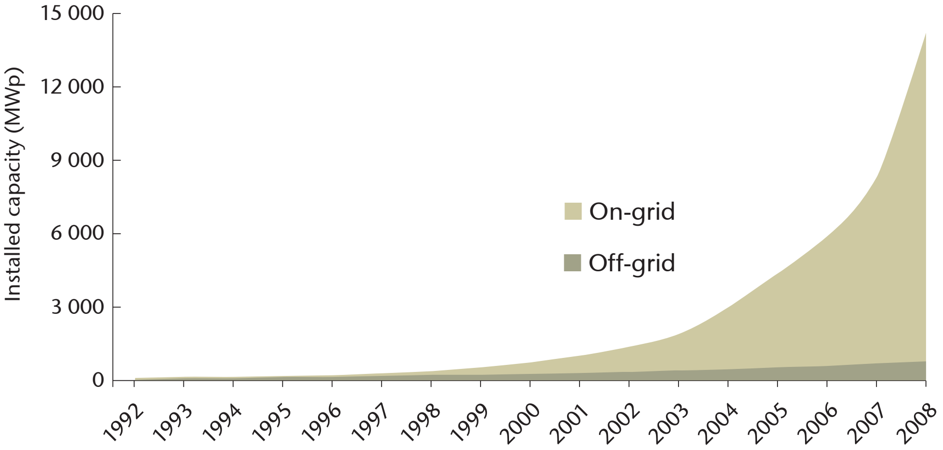


Figure 1.1: Global cumulative installed PV capacity from 1992 to 2008. Figure taken from [3]

1.2 Problem description

A simulation model of a general photovoltaic module is to be constructed in Simulink. The inputs to the model should be solar irradiance and ambient temperature. It must be able to operate at the maximum power point to compare the output to measured values.

A battery system as well as power smoothing algorithms should be implemented to evaluate the possibilities of power smoothing.

1.3 Aims and objectives

The main aim of this thesis is to construct an accurate model of a photovoltaic system with energy storage for power smoothing. The models should be as easily modifiable as possible. It is a goal to make the models so generally applicable so they can be used as tools.

A large portion of the work effort has been applied to make the models general and most of the results displayed in this paper has been created by some minor modification to the fundamental models.

1.4 Outline of thesis

Part I contains general introduction as well as a short introduction to some characteristics of batteries in chapter 2 and photovoltaic cells in chapter 3.

Chapter 2 contains a short introduction to battery energy storage systems. Some important features are presented and discussed.

The introduction to photovoltaic cells in chapter 3 contains a short introductions to the inner workings of the cell. An ideal PV is presented and some important parameters for solar cells are explained.

In part II the entire process of modelling both PV and battery modules are laid out.

The PV modelling method is presented in chapter 4. First, different variations of diode models are presented and discussed, then all the equations governing the behaviour of the five parameter model as well as the cell temperature are presented. A method of approximating the parameters from data sheet values is introduced and the conversion from a PV cell to a module is explored. Lastly, the effects of varying model parameters are presented as figures of I-V and P-V plots with curves resulting from different parameters.

Chapter 5 contains a presentation of different maximum power point tracking methods. The incremental conductance (IC) is presented. The last part of the chapter contains a presentation, implementation flow chart and discussion of the Perturb and Observe algorithm.

The method of modelling the battery modules is presented in chapter 6. In the first part of the chapter, some simple battery models are presented as well as the first implemented model together with a clarifications of why it was abandoned. The controlled voltage source model is explained at lengths and charging and discharging curves are displayed. Weaknesses of the model is discussed in the last part of the chapter.

Several power smoothing algorithms are presented, compared and discussed in chapter 7.

The Simulink implementation of all the models described in part II is presented in chapter 8. Some modifications has been made to make the simulation model work, these are discussed in section 8.5.1.

Part III contains results, discussion and conclusions. The results are discussed in the same sections they are presented, there is no separate discussion chapter.

In chapter 9 the results of comparing measured power values to simulated values are presented and discussed. The temperature simulation is examined. In the last section (9.4) contains a section where the temperature variable is removed from the picture to look at model accuracy without the temperature.

Chapter 10 contains results from power smoothing with different batteries and algorithms.

The final chapter (11) contains the conclusions.

Chapter 2

Battery energy storage

This chapter contains a short introduction to battery energy storage systems (BESS). In chapter 6 modelling of battery cells are reviewed in detail.

Various battery systems are currently some of the most applied energy storage systems in the world[6]. BESS can be manufactured to fit a wide range of applications from wristwatches to electric vehicles. It is one of the most adaptable energy storage technologies.

The quantity and diversity of battery energy storage systems are expanding in areas of electric vehicles, electric utility energy storage, portable electronics and storage system for energy management for renewable energy sources [6].

2.1 Battery energy storage technology

Some characteristics of BESS are presented in this section. Batteries modules are constructed by connecting cells in series and parallel. In the battery cells, chemical energy is converted to electrical energy when discharging and the opposite process takes place when charging. Each battery cell has some limitations on current and voltage due to the chemistry and structure of the cell. To construct a module with the desired capabilities, multiple cells must be connected in series and parallel.

2.1.1 Important features of batteries

Efficiency The efficiency of a battery is usually denoted by round-trip efficiency at nominal conditions and is often in the range of 70-80% [7], where the efficiency is dependent on discharge rate among other factors.

Life span The life span is often quantified by the number of charge/discharge cycles. Several factors affect the life span, amongst others, discharge rate, depth of discharge and temperature conditions.

State of charge (SOC) This refers to the relative capacity at the moment. A state of charge of 1 implies that the current battery capacity is equal to the rated capacity.

Depth of discharge (DOD) Depth of discharge is the reverse of the state of charge.

Self discharge Batteries in idle condition will discharge at some rate. Self discharge refers to the rate at which the battery discharges if not in use.

Operating temperature Temperature range for normal operation.

Specific energy density The amount of energy possible to store per kg, given in Watt hours per kg.

Specific power density The power capability in Watt per kg.

2.1.2 Typical discharge pattern

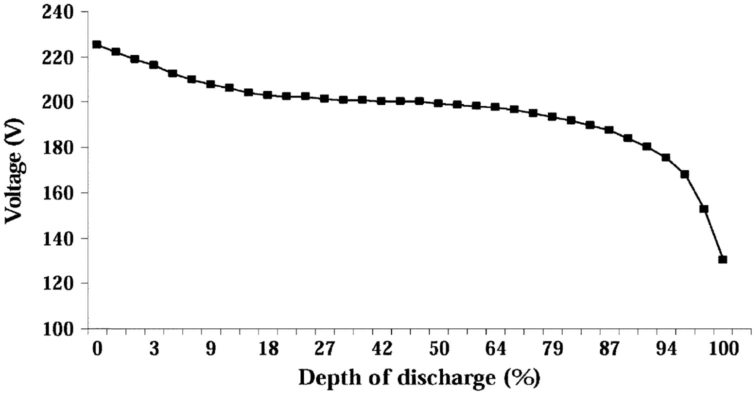


Figure 2.1: Typical discharge characteristics of a generic battery system plotted with voltage vs. depth of discharge. Figure taken from [8].

Figure 2.1 show a typical voltage to depth of discharge curve of a generic battery system. The same general shape can be observed by the battery model output in figure 6.8, but there it is plotted against SOC. There is a quite flat area from about 20% to 80% in the voltage curve. This enables very simple models of batteries to be reasonably accurate in large parts of the curve.

2.1.3 Battery types

There are many different battery types, some of them are plotted in figure 2.2 by their specific energy density and specific power density.

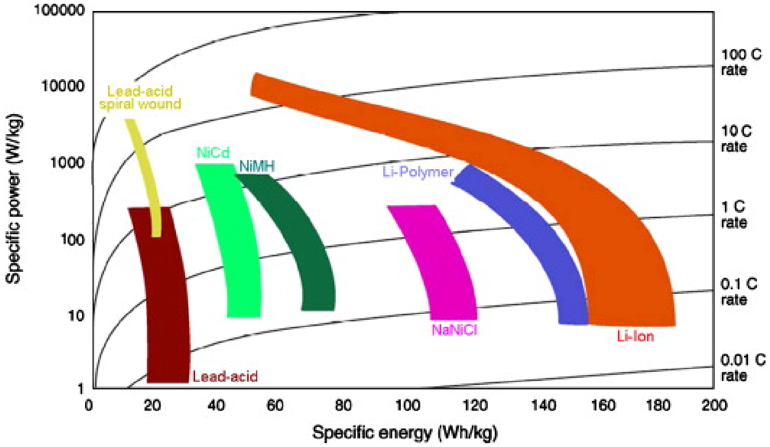


Figure 2.2: Distribution of different electrochemical batteries by their power and energy density[9, 10]. Figure taken from [10]

Lead acid This battery type is the oldest and most mature technology and has been used for many power system applications. The main advantage of lead-acid battery is the low cost compared to other types [10].

Li-Ion Lithium Ion batteries are very low-weight, energy dense, and power dense as shown in figure 2.2. These characteristics make them suitable for many portable applications. The production cost is high and the batteries lifetime degrade with deep discharges[7]. Data for from a lithium ion discharge curve is used in the battery simulation in this paper.

2.1.4 Battery capacity

The capacity of a battery is actually dependent on the rate at which it is discharged. This dependence is expressed by the Peukert equation (2.1).

$$C_p = I^k T \tag{2.1}$$

In equation 2.1 C_p is the Peukert capacity [Ah], I is the discharge current [A], T is the maximum discharge time and k is the Peukert coefficient. Manipulating the equation and introducing the rated discharge time and rated capacity produces equation 2.2:

$$It = C(\frac{C}{IH})^{k-1} \tag{2.2}$$

Where H is the rated discharge time in hours, C is the rated capacity at the discharge time of H, I is the actual current in [A], k is the Peukert coefficient and It is effective capacity [Ah] at the discharge rate of I.

It is clear from equation 2.2 that the higher the current, the lower the actual capacity will be.

The current amplitude dependence of the battery capacity is not included in the battery model in chapter 6, the only effect of apparent reduced capacity included is the one of reduced voltage because of the voltage drop over an internal resistance.

2.2 Discussion of battery energy storage

Battery energy storage will play a major role in the reliable and economic operation of smart electric grids with significant amounts of renewable power[7]. The technologies are mature, easily adaptable for different applications and are almost completely independent of topology and other local conditions except from temperature. These characteristics make BESS a suitable choice for many applications requiring energy storage.

Major downsides with batteries include limited cycle life, somewhat low round trip efficiency, self-discharge and increased losses with increased power.
Large scale battery storage systems designed for the electricity grid show some promise, and is being integrated into wind farms and PV plants[7, 11]. New battery types like vanadium redox flow batteries and Lithium-ion batteries will be extensively used instead of the conventional lead-acid types in the future[7, 12].

Chapter 3

Photovoltaic cells

3.1 Introduction

A photovoltaic cell directly converts sunlight into electricity. Cells can be connected together to form a module or an array. The direct output from a module or array may serve some small loads like DC-motors or lighting systems, but to provide power to fit more sophisticated demands, power electronic converters are needed. This also enables to tailor the load seen from the PV module to that which will make the module operate at the maximum power point (MPP). Photovoltaic cells connected together form a module or panel. A connection of one or more modules is called an array.

A photovoltaic cell is essentially a semiconductor diode where the p-n junction is exposed to light. The exposure leads to the breaking of electron bonds in the semiconductor. These charge carriers create an electric current when the cell is short circuited. The PV phenomenon may be described as absorption of solar radiation, the generation and transport of free carriers at the p-n junction, and the collection of these electric charges at the terminals of the PV device [13].

The review of the physics of PV cells is outside the scope of this paper. The following sections are related to characteristics of the behaviour of solar cells.

3.2 Ideal PV cell

The simplest model to represent a PV module is the ideal diode model with the equivalent circuit model shown in figure 3.1. This model consists of only a current source and a diode in parallel and it describes the behaviour of an ideal PV module.

$$I_{pv} = I_{ph} - I_0[\exp{(\frac{V_{pv}}{aN_cV_{th}})} - 1] \tag{3.1}$$

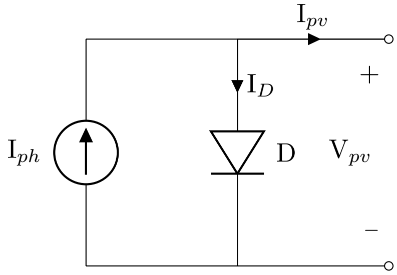


Figure 3.1: Ideal Diode model

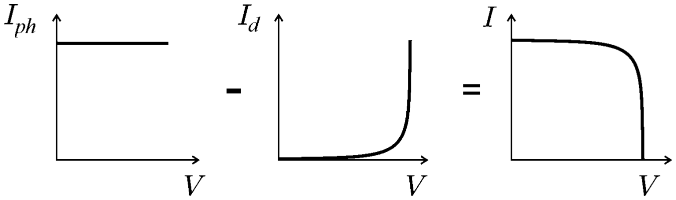


Figure 3.2: Characteristic I-V curve of the PV cell. The total current I is the photo current I_{ph} minus the diode current I_d. Figure taken from[13]

Equation 3.1 describes the current-voltage (I-V) characteristics. Figure 3.2 shows the three terms in equation 3.1, where the diode current I_d is equal to the second term on the right. This model is simple to implement and results in a low computational time, but the accuracy is poor [14]. More accurate and complex models are presented in the chapter about PV modelling (chapter 4).

3.3 PV cell parameters

The following are some parameters characterising a photovoltaic module.

Short circuit current, I_{sc} The short circuit current is the maximum current from a solar cell which occurs when the voltage is zero. It is for many solar cells very close to the photo current I_{ph} and it is therefore often characterized as the maximum current that can be drawn from the cell[13].

Open circuit voltage, V_{oc} The open circuit voltage is the maximum voltage from a solar cell that occurs when the current is zero. This voltage is dependent on cell temperature, it becomes smaller with higher temperature. This effect is shown in section 4.5.

Fill Factor, FF The Fill Factor is a measure of how good the voltage and current at maximum power point compares to the open circuit voltage and short circuit current. The calculation of fill factor is shown in equation 3.2.

$$FF = \frac{V_{mpp}I_{mpp}}{V_{oc}I_{sc}} \tag{3.2}$$

The fill factor refers to the degree at which the actual operating conditions lives up to the potential power of $I_{sc}V_{oc}$. It is impossible to have a fill factor of 1.

Efficiency Efficiency is defined as energy output divided by energy input. In the case of solar cells the efficiency equation can be presented in several ways as follows:

$$\eta = \frac{V_{mpp}I_{mpp}}{P_{in}} = \frac{V_{oc}I_{sc}FF}{P_{in}} \tag{3.3}$$

Here P_{in} is the irradiance from the sun on the area of the PV cell and η is the efficiency. Testing of efficiency is done at standard test conditions (STC), which is 1000 W/m^2 of sunlight intensity, 25°C temperature and 1.5 airmass.

Nominal operating cell temperature, NOCT The nominal operating cell temperature reflects that the operating conditions often are not STC. The NOCT refers to the cell temperature in a case of 800 W/m^2 of irradiance, 20°C ambient temperature and an average wind of 1 m/s. This temperature is given by manufacturers in solar cell specifications and it gives information on the temperature inside the cell. The NOCT is used in modelling the cell temperature in chapter 4

Ideality factor The ideality factor is a measure of how well the diode follows the ideal diode equation (equation 3.1). The effects of different ideality factors is shown in figure 4.10.

3.4 Resistive effects

3.4.1 Series resistance

The series resistance causes a slight voltage drop thereby reducing the fill factor. If the resistance is excessively high, even the short circuit current can be affected. It is generally a very small resistance. The series resistance is highlighted by a red colour in figure 3.3.

At open circuit voltage, the series resistance has no influence, but close to that voltage the effect is most clear. This can be seen in figure 4.11 where the effect of different values of R_s on the I-V curve is demonstrated.

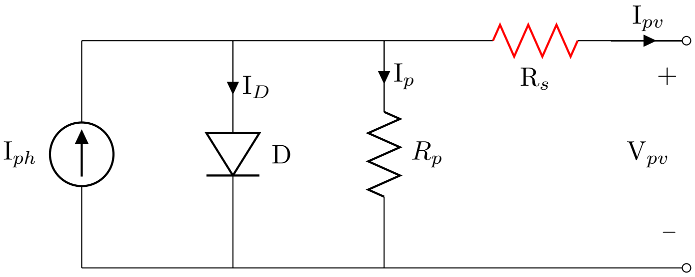


Figure 3.3: Single diode five parameter model with the series resistance element highlighted by red colour.

3.4.2 Shunt resistance

The parallel(or shunt) resistance is a cause of losses in the solar cells by providing an alternate current path. This equivalent resistance is usually quite high, although it can be low because of manufacturing defects[13]. The effect of the shunt resistance will be highest when the light levels are low because the voltage does not change that much with irradiance and when there is a small photo current, the current through the shunt resistance will be higher compared to the total.

The parallel/shunt resistance is highlighted in figure 3.4. The effect of varying

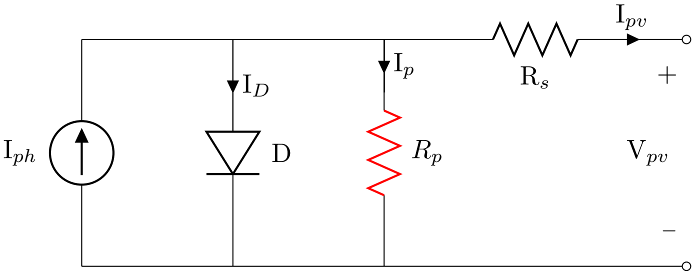


Figure 3.4: Single diode five parameter model with the shunt (also called parallel) resistance element highlighted by red color.

values of R_p is shown in figure 4.12.

Part II

Method

Chapter 4

Modelling of a photovoltaic module

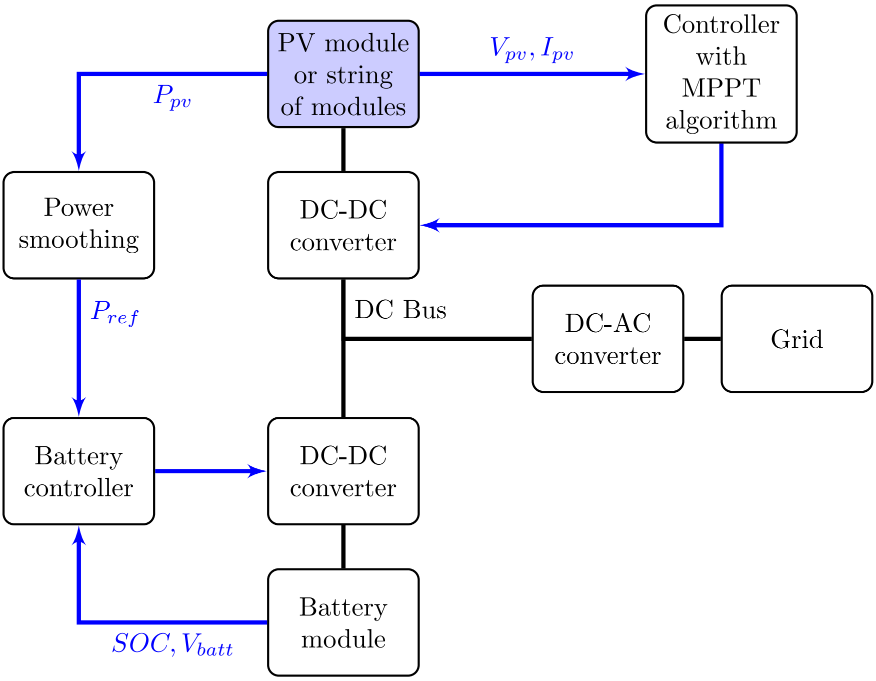


Figure 4.1: Overview of system used for simulation. The black lines indicate power flow and blue lines indicates signals. This chapter focuses mainly on the block ”PV module or string of modules” that is filled with blue colour. The DC-AC converter and grid is shown to illustrate that the system can be connected to the grid, but the converter is not discussed further in this paper.

Variable description	Name (Unit)
Maximum power	**Pmpp (W)**
Maximum power voltage	**Vmpp (V)**
Maximum power current	**Impp (A)**
Open circuit voltage	**Voc (V)**
Short circuit current	**Isc (A)**
Short circuit current temperature coefficient	**Ki (A/°C)**
Open circuit voltage temperature coefficient	**Kv (V/°C)**
Cell temperature proportionality factor	**Kt (°Cm^2/W)**
Number of cells in module	**Nc**
Photo electric current	**Iph (A)**
Series resistance	**Rs** (Ω)
Parallel resistance	**Rp** (Ω)
Diode Ideality factor	**a**
Electron charge	**Q (1.6x10^{-19}C)**
Boltzmann's constant	**k (1.38x10^{-23}J/K)**
Thermal voltage	**Vth (V)**
Diode reverse saturation current	**I$_0$ (A)**
Module current	**Ipv (A)**
Module voltage	**Vpv (V)**

Table 4.1: Variable name, descriptions and units of all the variables used in the model of photovoltaic cells in this report

4.1 Presentation of different diode models

The simplest model to represent a PV module is the ideal diode model presented in section 3.2.

4.1.1 Rs-model

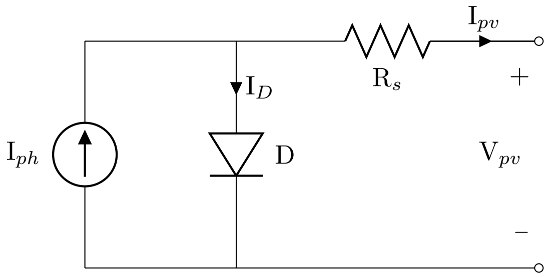


Figure 4.2: Rs-Model (Single diode four parameter model)

Figure 4.2 shows the equivalent circuit of the Rs-model. It has four unknown parameters and is known as the Single diode four-parameter model. Similar to the ideal diode model, this model is very simple and requires minimal computation power. The accuracy is better than the ideal diode model, but it still exhibits some serious deficiencies, particularly with temperature variations. This model is by far the most widely used in PV system simulation[14].

4.1.2 Rp-model

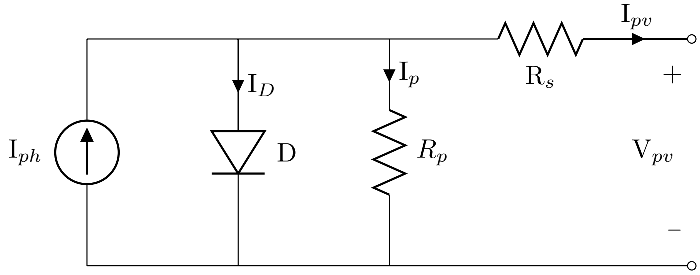


Figure 4.3: Rp-model (Single diode five parameter model)

The Rp-model, or single diode five parameter model, adds a shunt resistance (Rp) to the four parameter model and therefore considers the leakage current to the ground. Figure 4.3 shows the equivalent circuit and equation 4.1 shows the current-voltage (I-V) characteristics. This model is used for implementation in Simulink/MATLAB in this paper.

$$I_{pv} = I_{ph} - I_0[\exp{(\frac{V_{pv} + I_{pv}R_s}{aN_cV_{th}})} - 1] - \frac{V_{pv} + I_{pv}R_s}{R_p} \tag{4.1}$$

The Rp-model is insufficiently accurate when dealing with low irradiation levels[15]. This model is based on the assumption that the recombination loss in the depletion region is absent. In a real solar cell this loss is significant and the single diode model cannot be used to accurately model this effect[15].

4.1.3 Two diode model

The two diode model represents the PV cell more accurately than the single diode models. Two new variables are introduced (I_{02} and a_2) and this increases the complexity drastically. The single and two diode model show similar results at STC, but differ closer to Voc and with low irradiance[15]. Figure 4.4 show the equivalent circuit and equation 4.2 shows the current-voltage (I-V) characteristics.

$$I_{pv} = I_{ph} - I_{01}[\exp{(\frac{V_{pv} + I_{pv}R_s}{a_1V_{th1}})} - 1] - I_{02}[\exp{(\frac{V_{pv} + I_{pv}R_s}{a_2V_{th2}})} - 1] - \frac{V_{pv} + I_{pv}R_s}{R_p} \tag{4.2}$$

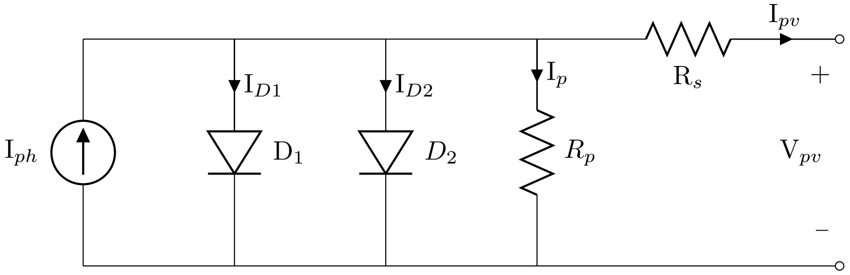


Figure 4.4: Two diode model

4.2 Five parameter model

In this section, the equations used to model a photovoltaic module are presented. In chapter 8 the implementation of these equations is shown.

4.2.1 Thermal voltage

The thermal voltage of a module with N$_c$ cells is given by equation 4.3[13].

$$V_{th,module} = \frac{kT}{q} = N_c V_{th,cell} \tag{4.3}$$

N$_c$ is the number of cells in a module, k is the Boltzmanns's constant ($1.38 * 10^{-23}$J/K) and q is the electron charge ($1.6 * 10^{-19}$C). In some papers [16, 17], this relation is not calculated separately, but used directly in the single diode model equations. The thermal voltage is used in the equations in this paper because it simplifies the modelling by separating out this relation into a subsystem. Because the thermal voltage calculation is done in a subsystem in the Simulink model, the equations in this paper are written with the same relations for consistency. The thermal voltage is expressed with the N_c factor in the other equations to underline how the equations can be modified to fit a single PV cell, a module or even a string of modules as shown in section 4.4

4.2.2 Diode reverse saturation current

The diode reverse saturation current can be calculated by substituting the open circuit conditions ($V = V_{oc}$ and $I = 0$) into equation 4.1 as shown in equation 4.4.

$$I_0 = \frac{I_{sc}}{\exp\left(\frac{V_{oc}}{aN_cV_{th}}\right) - 1} \tag{4.4}$$

Equation 4.4 does not depend on temperature (except from V_{th}) and it is therefore only accurate at the temperature at witch the input values is given. There are several ways to model the temperature dependence of I_0 and one of those is shown in equation 4.5[16]:

$$I_0(T) = I_0 \left(\frac{T}{T_{ref}} \right)^3 \exp \left([\frac{T}{T_{ref}} - 1] \frac{E_g}{aN_c V_{th}} \right) \tag{4.5}$$

Here I_0 is given by equation 4.4, E_g is the band gap energy of the semiconductor (1.12 eV for polycrystalline silicone at 25 °C [13]), V_{th} is given in equation 4.3 and T_{ref} is the reference temperature at which I_0 is calculated. The nominal or reference saturation current is usually calculated with values from data sheets. These values are given for at nominal temperature and the reference temperature in equation 4.5 is therefore often equal to the nominal temperature. This equation leads to some model errors around V_{oc} and therefore also at other sections of the I-V curve.

Several papers [13, 18, 15] suggests an improvement on equation 4.5 by substituting in equation 4.6.

$$I_0(T) = \frac{I_{sc} + K_i(T - T_{ref})}{\exp \left(\frac{V_{oc} + K_v(T - T_{ref})}{aN_c V_{th}} \right) - 1} \tag{4.6}$$

Here K_v is the open circuit voltage temperature coefficient and K_i is the short circuit current temperature coefficient. Equation 4.6 can be obtained by adding the temperature coefficients (K_i and K_v) and the temperature difference to equation 4.4. Both of the temperature coefficients usually appear in commercial data sheets. This equation simplifies the model and does not lead to model errors around the open circuit voltage like equation 4.5 does. This relation is used to calculate the saturation current in the Simulink model. The implemented subsystem containing this equation is shown in section 8.7.

4.2.3 Photoelectric current

The photoelectric current (I_{ph}) depends on the temperature and solar radiation as shown in equation 4.7.

$$I_{ph} = \left(I_{ph,ref} + K_t(T - T_{ref} \right) \frac{G}{G_{ref}} \tag{4.7}$$

Here $I_{ph,ref}$ is the photoelectric current at reference irradiance and temperature (usually 1000 W/m^2 and 25 °C), G is the incident irradiance (in W/m^2) and G_{ref} is the reference or nominal irradiance (usually 1000 W/m^2). $I_{ph,ref}$ can be assumed to be approximately equal to I_{sc}. This is a very common assumption in PV modelling [13]. The assumption gives a good approximation because the series resistance is usually very small and the parallel resistance is large.

4.2.4 Ideality factor

The ideality factor (a) can be arbitrarily chosen and still give decent accuracy in modelling[13]. It describes the degree of ideaity of the diode. The choice or otherwise determination of the value of the ideality factor used in modelling depends on the other parameters used in the model. As with the other parameters, a initial "guess" can be made and the best value approximated by iteration. There are several ways to determine approximate the ideality factor [19].

The approximation method discussed in section 4.3 does not include approximation of ideality factor. All the other parameters are determined by iteration based on the data sheet values and an assumed value of the ideality factor. Choosing different ideality factor values results in different approximated values for the other parameters. The approximation method determines the parameters so that the model fits the data sheet values even if the ideality factor chosen is not very accurate. Testing of model accuracy compared to measured power data with different ideality factors chosen revealed that the accuracy does not change much as long as the ideality factor is within certain limits. This is because the other parameters are determined in a way that compensates for an inaccurate guess. The least error when compared to measured data was encountered with an ideality factor of 1.

Like the other parameters in the model, the value really is not static. Figure 4.5 shows the variation in measured ideality factor during a day in different modules, but the difference was close to negligible.

The factor is usually between 1 and 1,5 [13, 19].

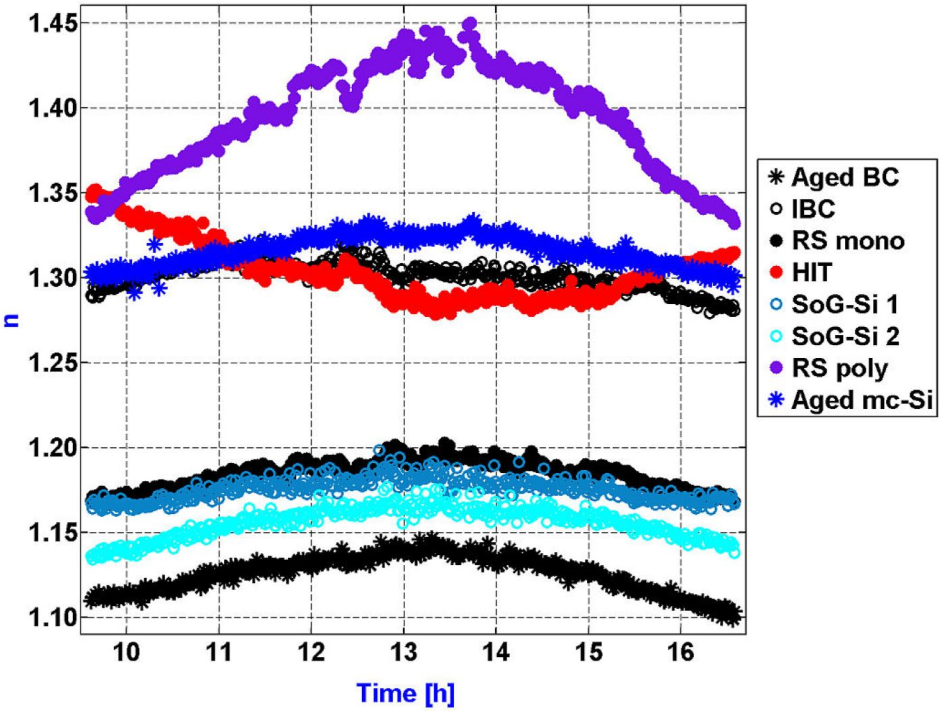


Figure 4.5: Ideality factor (n on y-axis here, a in this paper) variation on 29th of April, 2011. Figure taken from[20].

4.2.5 Cell operating temperature

The cell temperature will differ from the ambient temperature and can be assumed to be a linear function of the irradiance[21] as shown in equation 4.8

$$T_{cell} = T_{ambient} + K_t G \tag{4.8}$$

Here T_{cell} is the cell operating temperature, $T_{ambient}$ is the ambient temperature, K_t is the proportionality factor calculated in equation 4.9 and G is the incident irradiance.

$$K_t = \frac{NOCT(°C) - 20°C}{800W/m^2} \tag{4.9}$$

The NOCT (Nominal Operating Cell Temperature) is usually given in commercial module specifications.

This way of modelling the operating temperature is simple but has some disadvantages. At high irradiances, the simulated temperature will probably be higher than measured real world temperature because the increased convection of heat is not taken into account. Heat transfer is not a linear function, therefore this linear approximation has limitations to accuracy, particularly at high irradiance.

The cell temperature is used as input for the rest of the photovoltaic model.

4.3 Approximation model parameters from data sheet

In this section a method for approximating the five variables in the five parameter model (described in section 4.1.2) is outlined. A Matlab script with this algorithm is available in appendix A.1.

To approximate the variable for the five parameter model by this method, the specifications shown in table 4.2 are needed. In addition, an assumption of the diode ideality factor must be made if there is no real data available. Choosing an ideality factor value is discussed in section 4.2.4.

Specifications from data sheet	
Maximum power	**Pmpp (W)**
Maximum power voltage	**Vmpp (V)**
Maximum power current	**Impp (A)**
Open circuit voltage	**Voc (V)**
Short circuit current	**Isc (A)**
Short circuit current temperature coefficient	**Ki (A/°C)**
Open circuit voltage temperature coefficient	**Kv (V/°C)**
Number of cells in module	**Nc**

Table 4.2: Required input from data sheet

Reverse saturation current The first calculation that should be done, is to use equation 4.4 to calculate the nominal reverse saturation current of the diode (I_o). All the variables needed for this calculation are present in table 4.2.

Series resistance The series resistance can be calculated by the maximum power rating at STC. The derivative of maximum output power with respect to output voltage at the maximum power point is zero[22] as expressed in equation 4.10:

$$\left.\frac{dP}{dV}\right|_{mpp} = \left.\frac{d(IV)}{dV}\right|_{mpp} = I_{mpp} + V_{mpp}\left.(\frac{dI}{dV})\right|_{mpp} = 0 \tag{4.10}$$

Using equation 4.1, the differential of the module current with respect to module voltage can be calculated and substituted into equation 4.10[18]. The series resistance can then be calculated as shown in equation 4.11:

$$R_s = \frac{V_{mpp}}{I_{mpp}} - \frac{aN_cV_{th}R_p}{I_oR_p \exp{(\frac{V_{mpp}+I_{mpp}R_s}{aN_cV_{th}})} + aN_cV_{th}} \tag{4.11}$$

The parallel resistance (R_p) on the right side of the equation is still unknown. And the calculation of it requires a value for R_s.

Parallel resistance At the maximum power point, equation 4.4 results in equation 4.12:

$$\frac{P_{mpp}}{V_{mpp}} = I_{mpp} = I_{ph} - I_0[\exp{(\frac{V_{mpp} + I_{mpp}R_s}{aN_cV_{th}})} - 1] - \frac{V_{mpp} + I_{mpp}R_s}{R_p} \tag{4.12}$$

This can be rearranged into equation 4.13:

$$R_p = \frac{V_{mpp} + I_{mpp}R_s}{I_{ph} - I_{mpp} - I_o[\exp{(\frac{V_{mpp}+I_{mpp}R_s}{aN_cV_{th}})} - 1]} \tag{4.13}$$

There is two unknown parameters on the right side of this equation, the series resistance R_s and the photoelectric current I_{ph}.

Photoelectric current It is often assumed that the photoelectric current (I_{ph}) is equal to the short circuit current (I_{sc}). A better approximation can be made by using the series and parallel resistors to calculate the current as shown in equation 4.14:

$$I_{ph} \approx \frac{R_p + R_s}{R_p} I_{sc} \tag{4.14}$$

This calculation requires values for both of the resistances.

Iteration method Because all of these parameters are mutually dependent, iteration is a decent way to find values that fits well and gives a small error. A way to determine how "good" the values are is needed.

In [18] it is suggested to iterate the parameters and minimize the error values given in equation 4.18.

$$Err1 = \frac{V_{mpp}}{I_{mpp}} - \frac{aN_cV_{th}R_p}{I_oR_p \exp{(\frac{V_{mpp}+I_{mpp}R_s}{aN_cV_{th}})} + aN_cV_{th}} - R_s \tag{4.15}$$

$$Err2 = \frac{V_{mpp} + I_{mpp}R_s}{I_{ph} - I_{mpp} - I_o[\exp{(\frac{V_{mpp}+I_{mpp}R_s}{aN_cV_{th}})} - 1]} - R_p \tag{4.16}$$

$$Err3 = \frac{R_p + R_s}{R_p} I_{sc} - I_{ph} \tag{4.17}$$

$$Err = (Err1)^2 + (Err2)^2 + (Err3)^2 \tag{4.18}$$

The parameters can be iterated with small steps to give good accuracy. After the iteration, the values that gave the smallest error are the ones that fits the data sheet values the best according to this method.

A flowchart of the approximation algorithm is shown in figure 4.6 and the full implementation in Matlab is available in A.1.

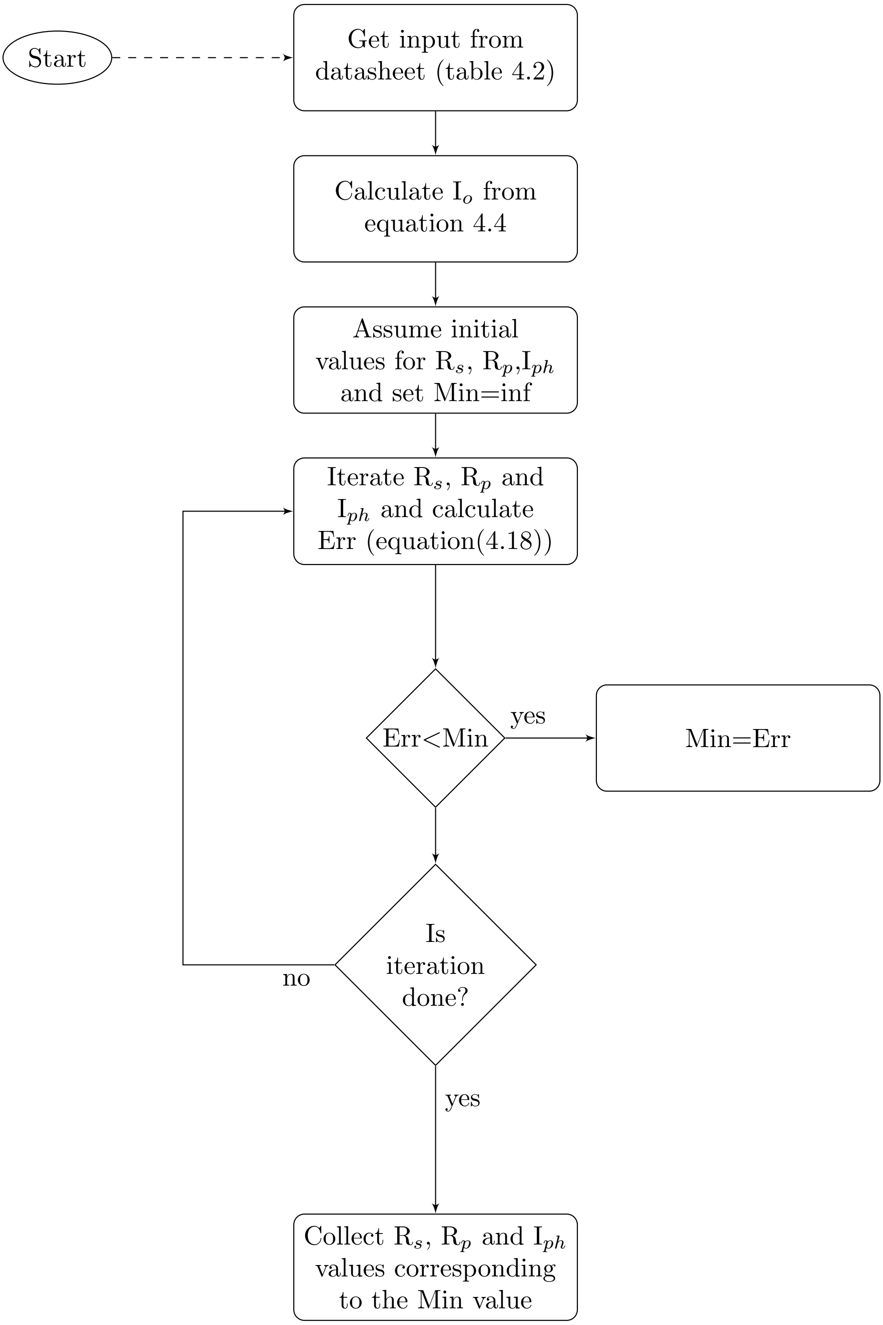

Start
Get input from datasheet (table 4.2)
Calculate I_o from equation 4.4
Assume initial values for R_s, R_p,I_{ph} and set Min=inf
Iterate R_s, R_p and I_{ph} and calculate Err (equation(4.18))
Err<Min
yes
Min=Err
Is iteration done?
no
yes
Collect R_s, R_p and I_{ph} values corresponding to the Min value

Figure 4.6: Flowchart of the parameter approximation algorithm

4.4 From cell to module

The modelling explained in section this chapter can be valid for just a photovoltaic cell, a string of cells, a module or string of modules. The variables in the model must be adjusted as will be outlined in this section to fit the system configuration that is to be modelled.

In this section, it is assumed that all the cells connected in series are identical. This is only an approximation since there are differences between individual cells in real systems. In addition to identical cells, it is assumed that the temperature and irradiance are identical for the entire block of series connected cells. There is a major downside of lumping together many cells and assuming uniformity in characteristics and input in that partial shading, dirt on the panel and other effects are not simulated with sufficient accuracy.

In real modules, bypass diodes are used to reduce the problem of uneven distribution of solar irradiance or other not uniformly distributed characteristics. A module can be constructed with several series connected blocks of cells with bypass diode. As an example, the modules in the REC Peak energy series, consists of 3 strings of 20 cells with bypass diodes for each string[23]. Modelling this system accurately requires three subsystems with 20 cell strings that can be lumped together and simulated with different inputs to model partial shading of the module.

Figure 4.7 shows two cells coupled in series. This equivalent circuit diagram can be simplified by lumping together the series resistances. The horizontal coupling between the diodes, parallel resistances and the current sources can be removed without changing the circuits behaviour because the cells are assumed to be identical and there will therefore flow zero current in this connection. These simplifications are shown in figure 4.8.

The circuit in figure 4.8 can be further simplified by replacing the currents sources with a single current source, lumping the parallel resistances together and lumping the diodes together as is shown in figure 4.9 where the number of series connected cells is N instead of 2. The voltage over the two diodes in figure 4.8 is equal to $(V_{pv} - 2R_sI)$ and the voltage over one diode will be $(V_{pv} - 2R_sI)/2$. This gives a diode current of a series connection of N_c cells as shown in equation 4.19:

$$I_D = I_0[\exp{(\frac{V_{pv} + N_cR_sI_{pv}}{aN_cV_{th}})} - 1] \tag{4.19}$$

With these simplifications of an equivalent circuit of a series connection of N_c cells, the new I-V relationship can be described by equation 4.20:

$$I_{pv} = I_{ph} - I_0[\exp{(\frac{V_{pv} + I_{pv}N_cR_s}{aN_cV_{th}})} - 1] - \frac{V_{pv} + I_{pv}N_cR_s}{N_cR_p} \tag{4.20}$$

When approximating the five parameters of a PV module, the values that are produced are dependent on the input. If the input are the ratings of a complete

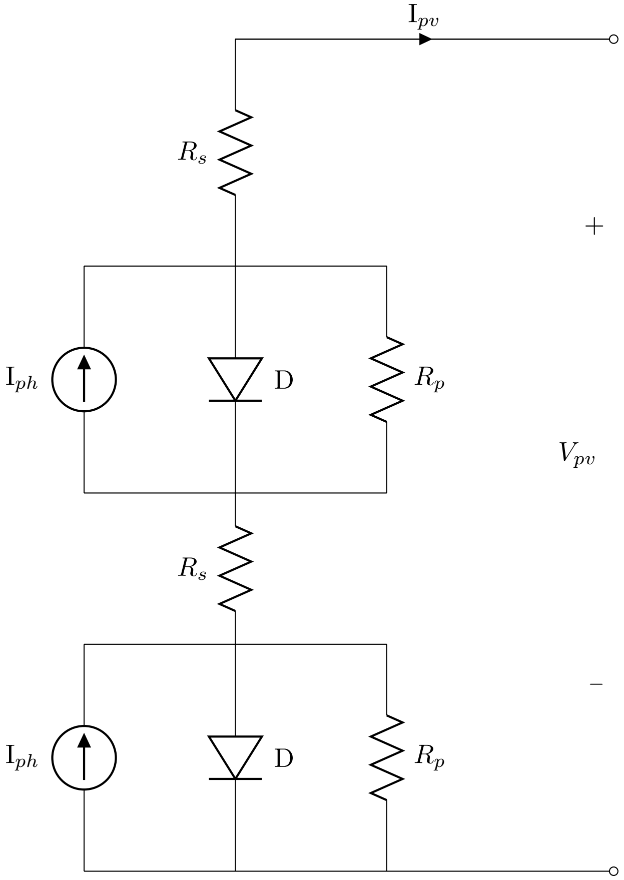


Figure 4.7: Circuit diagram of two PV cells coupled in series. I_{pv} and V_{pv} is the current and voltage output of the series coupling.

module, the resulting values from the approximation method described in section 4.3 will be the values corresponding to a complete model of a module. This entails that, as an example, the R_s-value obtained from the algorithm will in reality be equivalent to $(R_{s,cell} * N_{cells})$. The resulting values of the algorithm are written without the number of cells as a factor for simplicity and because the equations in this chapter are valid as long as all the variables are derived from the same rating. Parameters for each individual cell can easily extracted if the complete module parameters has been approximated by dividing the resulting values by N_c.

The only place in the equations where the factor N_c must always be included, is in the calculation of the diode current shown equation 4.19 and the five parameter model equation 4.1.

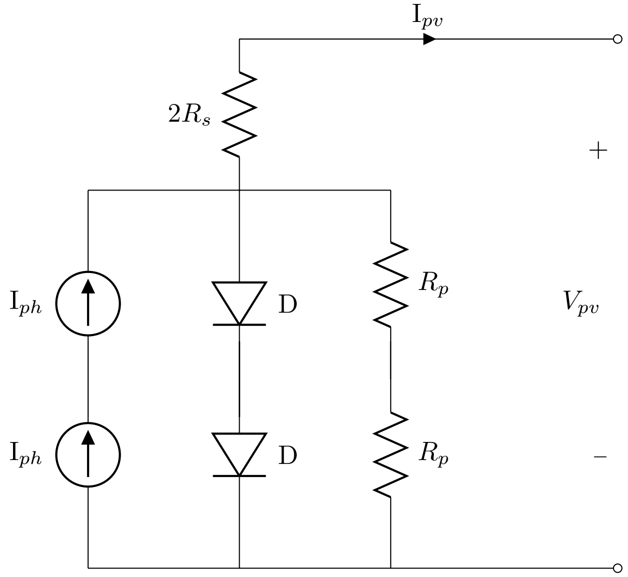


Figure 4.8: Circuit diagram of two PV cells coupled in series with some simplifications. I_{pv} and V_{pv} is the current and voltage output of the series coupling.

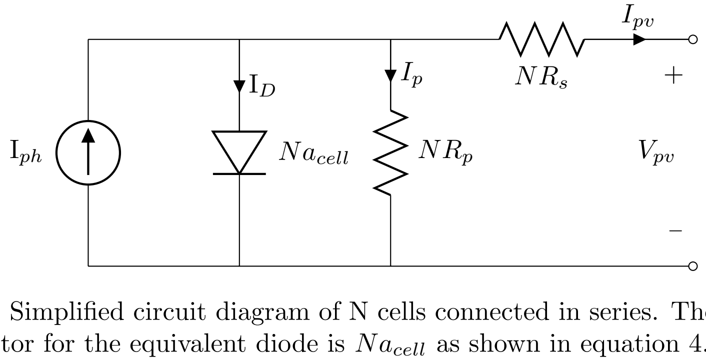


Figure 4.9: Simplified circuit diagram of N cells connected in series. The modified ideality factor for the equivalent diode is Na_{cell} as shown in equation 4.19.

4.5 Effects of model parameters and input

This section contains a lot of plots demonstrating what kind of influence tweaking of the model parameters has on the output of the model. In addition, the effect of different temperatures and irradiances has on the model output is shown. For every plot, all other variables and inputs are set to the values that are given in table 4.3.

Parameter values and test conditions		
Data sheet values		
Rated Power	Pmpp	85 W
Voltage at mpp	Vmpp	18 V
Current at mpp	Impp	4.72 A
Open circuit voltage	Voc	22.03 V
Short circuit current	Isc	5 A
Current coefficient	Ki	0.00325 A/°C
Voltage coefficient	Kv	-0.08 V/°C
Temperature coefficient	Kt	0.0338 °C/Wm^2
Number of cells	Ncells	36
Nominal operating cell temp	Tnoct	47 °C
Reference cell temp	Tref	320.15 °K
Reference irradiance	Ir0	1000 W/m^2
Results from script		
Parallel resistance	Rp	414 Ω
Series resistance	Rs	0.22 Ω
Reverse saturation current	Io	4.2263e-09 A
Photo current	Iph	5.0559 A
Ideality factor	a	1.14
Test conditions		
Irriadiance	Irradiance	1000 W/m^2
Ambient temperature	Tamb	273.15 + 25 °K

Table 4.3: Parameter values used for simulation in section 4.5. Data sheet values from [1]

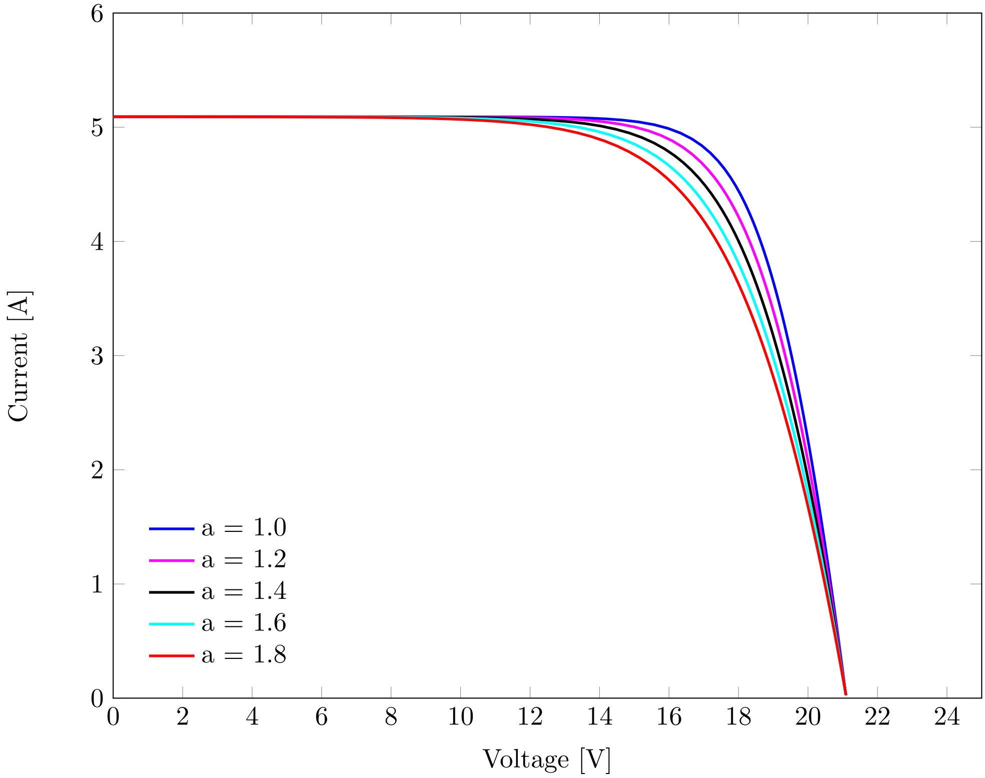


Figure 4.10: I-V curves with different ideality factors

4.5.1 Ideality factor (a)

Figure 4.10 demonstrates the effect the ideality factor have on the output. A higher factor leads to a lower maximum power point and a lower fill factor. The value of ideality factor used in this model is 1.14. This value was chosen based on the data in figure 4.5, where the approximate average value of the module of interest is about 1.14. Choosing an ideality factor for modelling a module can be done by iteration if real performance data is available. If this data is not available, the factor can be chosen based on the type of material used to make the cells as discussed in section 4.2.4. The approximation method of determining parameters described in section 4.3 causes the choice of ideality factor to not be too important. This is because all the other parameters are adjusted to fit the data sheet values, therefore a ”wrong” ideality factor will lead to other parameter compensating. This will probably cause the model to be inaccurate in other areas than around MPP at STC because some of the parameters may be maladjusted.

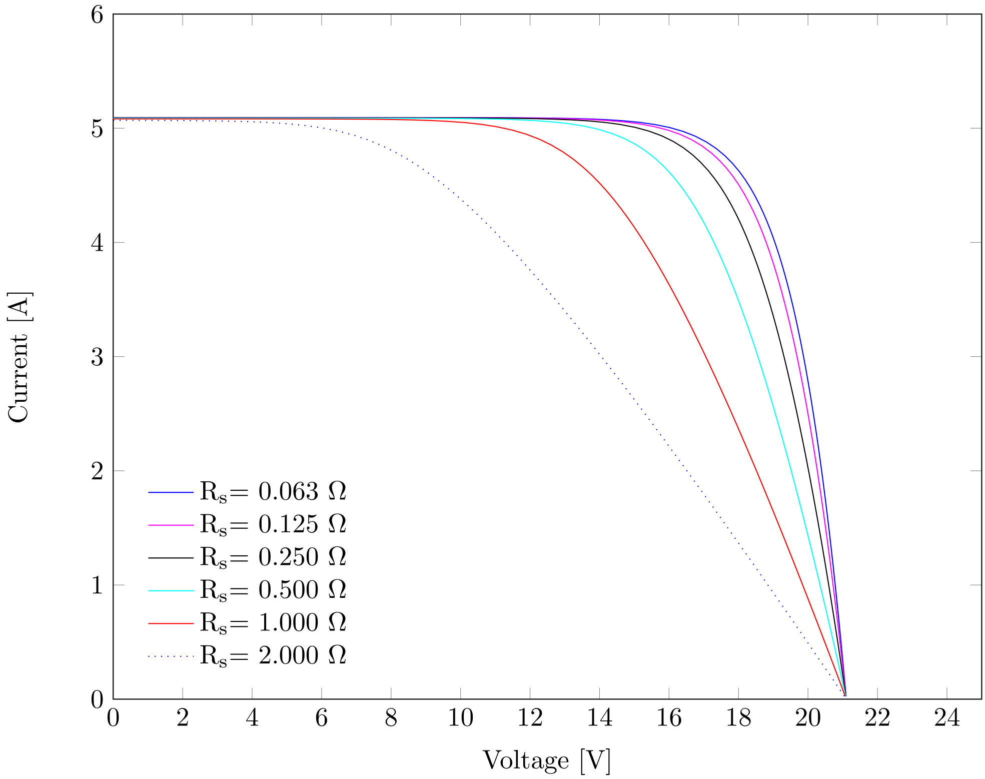


Figure 4.11: I-V curves with different Rs values [Ω]

4.5.2 Series resistance (R_S)

Figure 4.11 demonstrates the effect different values of the series resistance has on the output. Higher resistances reduce the maximum power and the fill factor. As a simplification, it can be said that the series resistance affects the I-V curve in the area where the current is no longer approximately constant. A higher resistance leads to lower voltage at the same current in this area of the graphs. The voltage at which the current starts to drop is lower at higher resistances, resulting in a lower current at maximum power point. In addition, the rate of decline in current is slower with higher resistances. Even larger resistances would even affect the short circuit current. The series resistance value resulting from the script A.1 is 0.22 Ω for reference. Ideally, the resistance should be as low as possible.

4.5.3 Parallel resistance (R_p)

Figure 4.12 demonstrates the effect that different parallel resistances has on the model output. The effect is simply to lower the current I_{pv} with lower resistance values. This lowers the power at maximum power point and R_p should ideally be

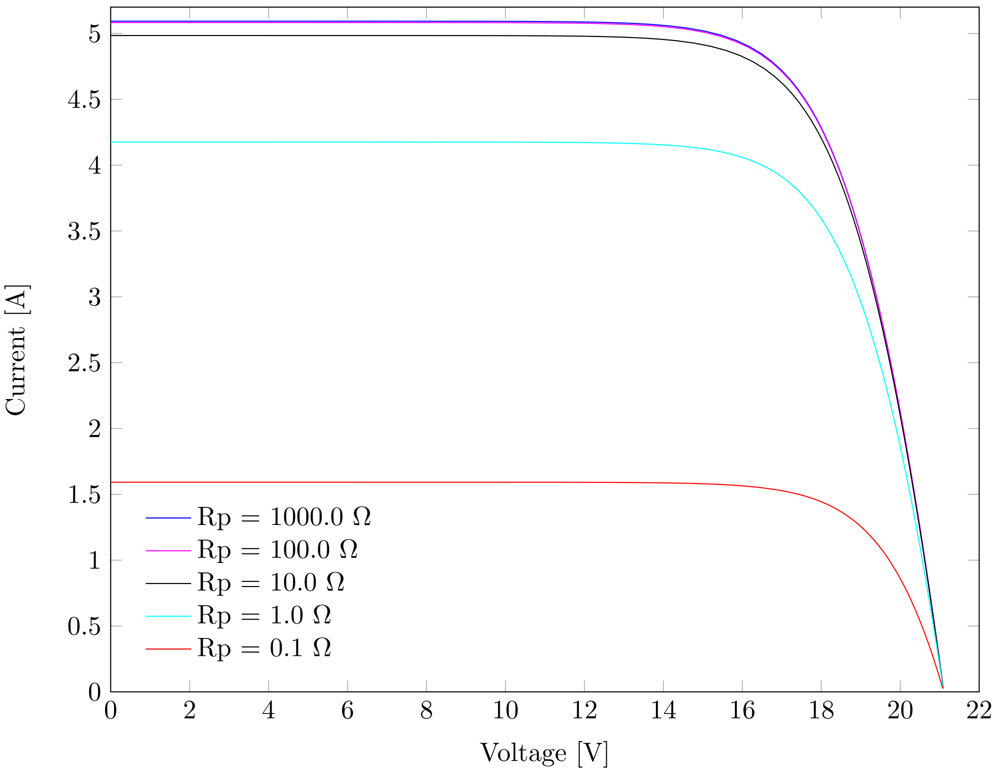


Figure 4.12: I-V curves with different Rp values [Ω]

infinite (open circuit). The value resulting from the approximation script A.1 is 414 Ω.

4.5.4 Ambient temperature

Figure 4.13 and 4.14 demonstrates the effect the ambient temperature has on the I-V and P-V curves. There are two effects that is clearly shown in figure 4.14. First, the open circuit voltage is higher with lower temperature. This is given as an open circuit voltage coefficient (K_v) specified in table 4.3. The other effect is the slight reduction of the short circuit current with lower temperature that is determined by the short circuit current coefficient (K_i).

The most important effect of the ambient temperature is that the power output is affected greatly. Power output decreases with increasing temperature and it does so to a significant degree. As can be seen in figure 4.13, the power is reduced by approximately 5 Watts by a 10 °C temperature increase. That is a quite large reduction in power output for a module rated at 85 Watts at STC.

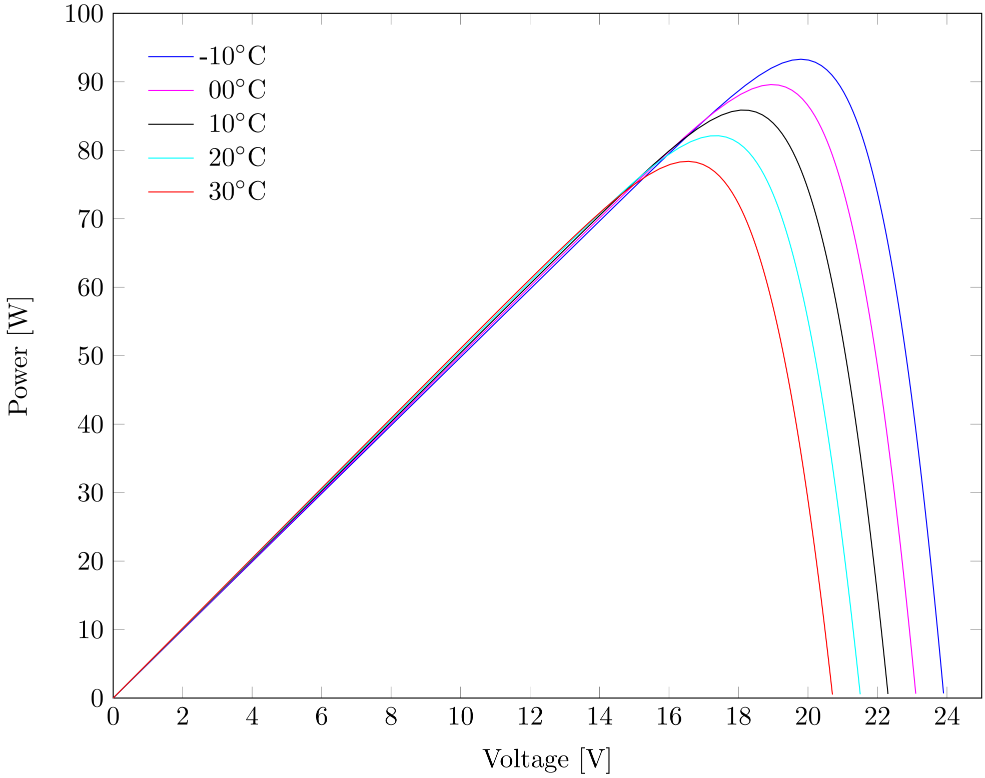


Figure 4.13: P-V curves with different ambient temperatures

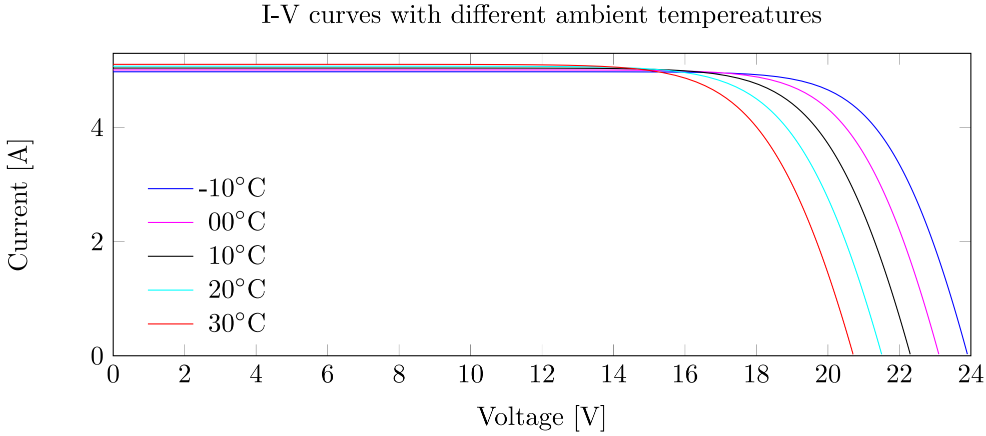


Figure 4.14: I-V curves with different ambient temperatures

4.5.5 Solar irradiance

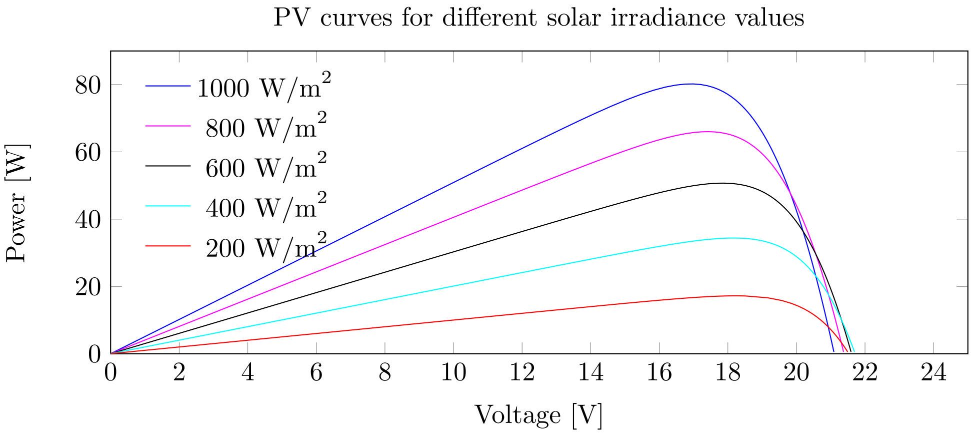


Figure 4.15: P-V curves with different solar irradiance values

Figure 4.15 and 4.16 shows the effect that different levels of solar irradiance have on the I-V and P-V curves. The main effect at work here is the reduction of photo current (I_{ph}) that is calculated by equation 4.7. The photo current varies linearly with irradiance level. This can be easily seen in figure 4.16.

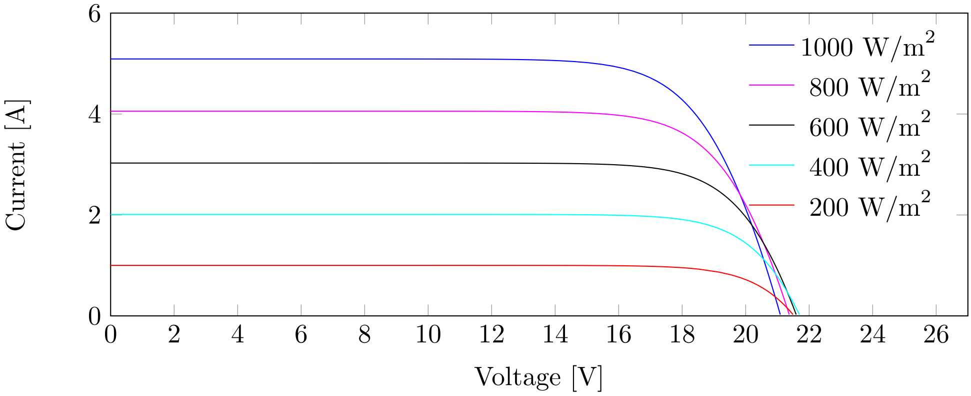


Figure 4.16: I-V curves with different solar irradiance values

There are also two more subtle effects at play here. One of them is the slight raising og the short circuit current in comparison to what is expected if the cell temperature is independent of irradiance. In this paper, the temperature coefficient relation (K_t) explained in equation 4.8 gives a cell temperature that is dependent on irradiance level.

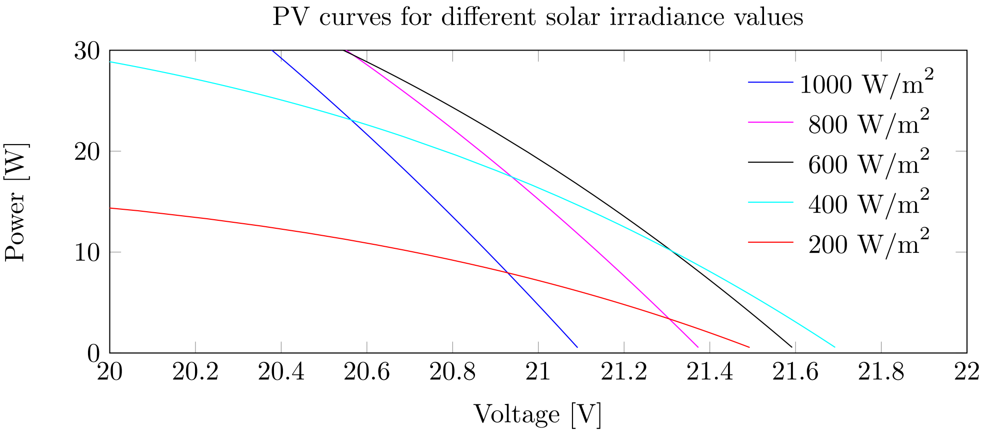


Figure 4.17: P-V curves with different solar irradiance values zoomed in on open circuit voltage

The other subtle effect is the reduction of open circuit voltage due to the higher cell temperature that higher irradiance leads to. It can be seen in figure 4.15 that the open circuit voltages of the curves with lower irradiance are higher than the voltages for the curves with higher irradiance.

The effect can be seen more distinctly in the zoomed image in figure 4.17. This model gives this effect because of the relation shown in equation 4.8. Higher cell temperature results in a lower open circuit voltage as shown in figure 4.14. The reduction of V_{oc} with higher irradiance levels is not commonly seen in PV literature. It is a direct consequence of modelling the cell temperature as a sum of ambient temperature and irradiance times a coefficient.

Chapter 5

Maximum power point tracking

5.1 Introduction

It is important to try to extract the maximum amount of power possible from photovoltaic arrays. The current-voltage and the power-voltage relationship is shown in the figures in section 4.5. At any time, there is a voltage level that will give maximum power. To always operate at this voltage level, DC-DC converters controlled by maximum power point tracking (MPPT) algorithms is inserted after the PV modules to ensure optimal operating conditions.

Methods OF maximum power point tracking can generally be divided into two categories: Direct and indirect methods[24].

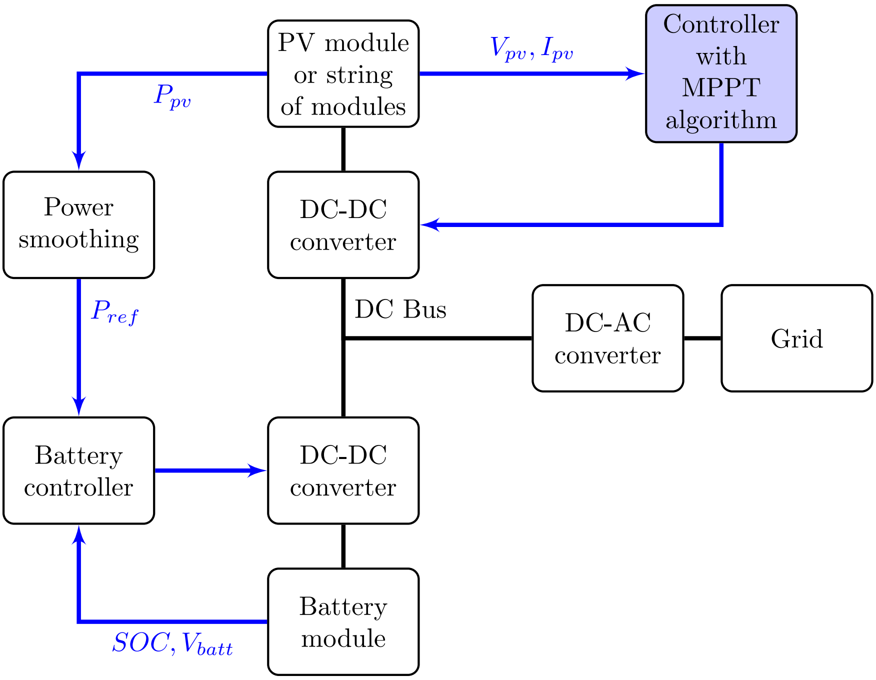


Figure 5.1: Overview of system used for simulation. The black lines indicate power flow and blue lines indicates signals. This chapter focuses mainly on the block "Controller with MPPT algorithm" that is filled with blue colour.

5.2 Direct and indirect methods

The MPPT methods can be categorized as direct or indirect. The indirect methods can also be classified as "quasi seeking" since there is no actual seeking, but only approximation by some other method.

Direct methods use measurements of the photovoltaic voltage (V_{pv}) and current (I_{pv}) and obtaining the operating point by changing the voltage based on their response to obtain the MPP[25].

5.2.1 Indirect methods of maximum power point tracking

Indirect methods entail estimating the MPP by measurements of the photovoltaic voltage (V_{pv}) and current (I_{pv}) and using some model of the system to give an approximation of the MPP. This model can be mathematical equations and numerical approximations applied by look-up tables. These equations or numerical approximations are dependent on the actual system of which it is to give an approximation of MPP. In other words, an indirect method of MPPT is only valid for the exact PV generator it is designed for. Aging and other changes in system characteristics will reduce the accuracy of the method. The fact that the methods needs to be modified for each system and that changes are not handled well are the major drawbacks of using indirect methods.

Varying irradiance and temperature must be accounted for by measurement and either input to mathematical equations or input to look-up tables. The look-up tables and mathematical equations can never perfectly model the real conditions. The look-up table method can never account precisely for all conditions as the data has to be measured and regression used between actual data.

There are several indirect methods available: Curve-fitting, look-up table, open-voltage PV generator, short circuit PV generator and the open circuit cell. These methods will not be reviewed further in this paper but are explained in detail in [25].

5.2.2 Direct methods of maximum power point tracking

Direct methods of acquiring the MPP do not require any prior knowledge of the system characteristics. The algorithms use measurements of voltage and/or current and takes into account the variation of these state variables. These methods can also be called ”true seeking” methods, since the correct operating point is obtained by ”seeking” and not by any approximations of system characteristics and behaviour. The ability to find the MPP is unaffected by variables like isolation, temperature or degradation levels[25, 26]. The major drawback of these methods are that they can be more complicated and that undesirable errors can affect the tracker accuracy.

The following methods/algorithms are included under the ”direct method” category: Perturb and observe (P&O), incremental conductance, differentiation, feedback voltage (current), auto oscillation, fuzzy logic and others[25]. The perturb and observe and the incremental conductance method are reviewed further in this paper.

5.3 Incremental conductance method

The incremental conductance (IC) method is based on equation 5.1 the derivative of the output power with respect to the output voltage. The maximum power point will always be on the highest point of a P-V curve as can be seen in figure 4.15. Tracking of the MPP is done by comparing the instantaneous conductance ($\frac{I_{pv}}{V_{pv}}$) to the incremental conductance ($\frac{\Delta I_{pv}}{\Delta V_{pv}}$)[24].

$$\frac{dP_{pv}}{dV_{pv}} = \frac{d(V_{pv}I_{pv})}{dV_{pv}} = I_{pv} + V_{pv}(\frac{dI_{pv}}{dV_{pv}}) = I_{pv} + V_{pv}(\frac{\Delta I_{pv}}{\Delta V_{pv}}) \qquad (5.1)$$

At the MPP, the derivative of the power with respect to voltage will be zero, giving the relation shown in equation 5.2 for detecting this point.

$$\frac{dP_{pv}}{dV_{pv}} = 0 \Longrightarrow \frac{\Delta I_{pv}}{\Delta V_{pv}} = -\frac{I_{pv}}{V_{pv}} \tag{5.2}$$

To the right of the MPP, on the side with too high voltage, the derivative of power with respect to voltage is negative, giving the relation shown in equation 5.3. In other words, if the term $\frac{dP_{pv}}{dV_{pv}}$ is less than zero, the voltage is too high and the current state is a position to the right of the MPP on a standard P-V curve. This can be detected by current and voltage and the change in these from the last cycle.

$$\frac{dP_{pv}}{dV_{pv}} < 0 \Longrightarrow \frac{\Delta I_{pv}}{\Delta V_{pv}} < -\frac{I_{pv}}{V_{pv}} \tag{5.3}$$

To the left of the MPP, the derivative of the power will be positive as shown in equation 5.4.

$$\frac{dP_{pv}}{dV_{pv}} > 0 \Longrightarrow \frac{\Delta I_{pv}}{\Delta V_{pv}} > -\frac{I_{pv}}{V_{pv}} \tag{5.4}$$

An MPPT algorithm can be implemented using equations 5.2, 5.4 and 5.3 to detect which way to increment the voltage by a step value. A large step value will result in fast tracking but also increased oscillation about the MPP.

The main advantage of the incremental conductance method is that it produces good results under rapidly changing conditions and that the level of oscillations around the MPP is low compared the P&O methods[24]. Unlike the P&O method, the IC method can determine in which direction the voltage has to be changed, even during rapid changes in conditions where the P&O algorithm will make "mistakes" as discussed in section 5.4. Another advantage is that by this method, it can be detected if the MPP is achieved (equation 5.2), thereby reducing the oscillation considerably under stable conditions.

One of the most significant drawbacks of the IC method is the cost of implementation because more measurements are required[27]. IC and P&O algorithms have no significant difference in efficiency[27].

Several attempt were made to implement this algorithm to compare with the P&O algorithm, but problems with algebraic loops and other instabilities was encountered. These issues might be caused by mistakes in the implementation or it might be an indication that this algorithm, to a higher degree than the other, contributes to instability in the model.

5.4 Perturb and observe method

The Perturb and observe method of maximum power point tracking is the most widely used algorithm due to the simple practical implementation[27]. The output

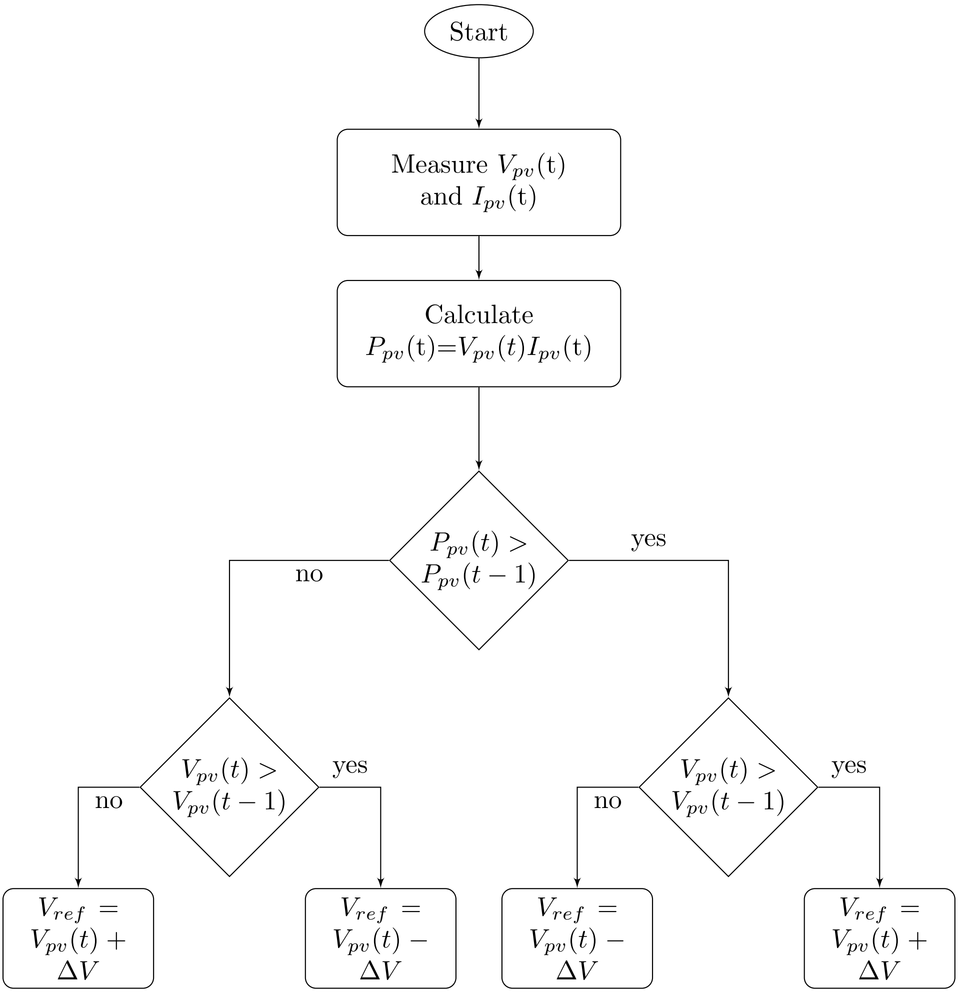

Start
Measure V_{pv}(t) and I_{pv}(t)
Calculate P_{pv}(t)=$V_{pv}(t)I_{pv}$(t)
$P_{pv}(t) > P_{pv}(t-1)$
no
yes
$V_{pv}(t) > V_{pv}(t-1)$
no
yes
$V_{pv}(t) > V_{pv}(t-1)$
no
yes
$V_{ref} = V_{pv}(t) + \Delta V$
$V_{ref} = V_{pv}(t) - \Delta V$
$V_{ref} = V_{pv}(t) - \Delta V$
$V_{ref} = V_{pv}(t) + \Delta V$

Figure 5.2: Flowchart of the perturb and observe maximum power point tracking algorithm. The Matlab implementation of this algorithm is available in appendix A.4

voltage (V_{pv}) is perturbed and th PV output power is then compared with power resulting from the previous perturbation. If the power is higher, then the voltage is perturbed in the same direction. If the power was lower, then the voltage is perturbed in the opposite direction. A flow diagram of the method is shown in figure 5.2.

The voltage is increased or decreased with a value (V_{step}). A high value increases tracking speed but it also increases the oscillation around MPP. There are several ways to improve this simple algorithm. Suggested methods of improving these problems include peak current control[28], optimizing of voltage step to converter and system characteristics[29] and variable voltage step methods[30, 31].

One of the major drawbacks of the P&O method is that it does not handle rapid variations in irradiations well. When a change in conditions happens, the algorithm will interpret the resulting difference in output power as a result of the last voltage perturbation. The next perturbation will therefore likely be in the wrong direction. If the change in conditions is such that this mistaken path og voltage change is reinforced, the algorithm can change the voltage to a value quite far from the MPP before "discovering" that it is the wrong point when the conditions stabilize enough for it to detect that.

The algorithm implemented in this paper is the basic P&O algorithm outlined in the flowchart in figure 5.2 without any implementation of varying voltage step or other improvements.

One of the worst conditions for an P&O algorithm to operate is when there is a ramping input condition. This kind of case is shown in figure 5.3. In this plot, there are shown the resulting voltage output with two different irradiance profiles. The blue line is with a input of an irradiance profile in a ramp function that increases with 10 W/m^2 per time unit. When the algorithm does a perturbation in one direction and then measures an increase in output power, the next perturbation will be in the same direction. Because the irradiance increases linearly, the output power will likely increase in most cases even if the last voltage change was in the wrong direction. The algorithm will therefore lead the voltage in the wrong direction until the power is actually reduced and the next voltage change will be in the opposite direction and it will continue until the power is reduced again. This process will repeat itself until the conditions stabilize sufficiently or the change slows down enough for the algorithm to jump out of this cycle earlier.

The red line in figure 5.3 is the output voltage from the same system with a different input irradiance. It starts at 200 W/m^2 and sits at this value until $t = 40$ when it jumps to 1000 W/m^2. Some variation can be seen in the stable condition. This is the algorithm constantly changing the voltage above and below the MPP. The variation seems to be about one voltage step above and below the MPP indicating maximum stability. In other words, if this variation is to be reduced, the voltage step has to be changed.

Notice from figure 5.3, that the V_{mpp} is lower at higher irradiance values, this is consistent with figure 4.15.

Figure 5.4 is the power output corresponding to the voltages in figure 5.3. The output is very smooth even with the relatively large variations in voltage. This is because the voltage variations are in an area of the P-V curve where small changes in voltage does not change the power by a lot. Some deviation can be seen in the

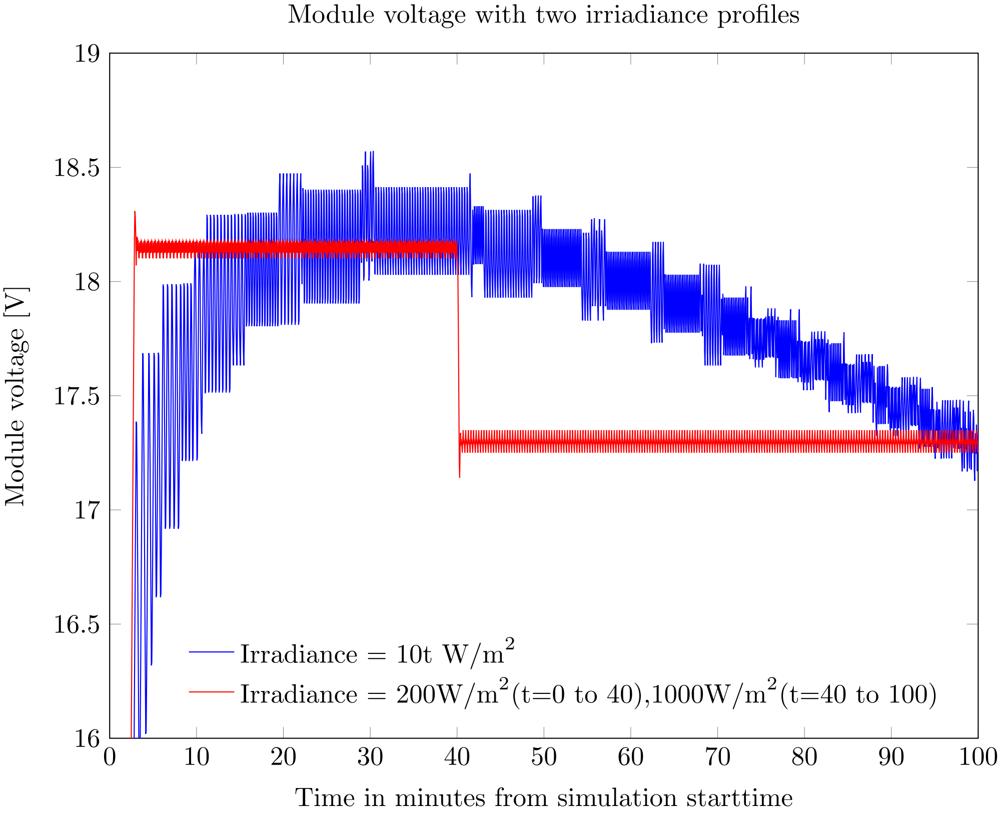


Figure 5.3: Voltage curves resulting from two different irradiance input profiles. The blue voltage curve is the voltage produced from an irradiance function that starts at zero and increases linearly by the equation $Irradiance = 10t$ as given in the figure legend. The red curve is the voltage resulting from first a stable irradiance value of 200W/m^2 and then 1000W/m^2 after t=40. The step value, $V_{step} = 0.1V$, is the same in both cases. It is clear the case with a ramp function, ($Irradiance = 10t$), is not well handled by the perturb and observe algorithm. Conditions and variable values used are the same as in table 4.3. The power output corresponding to these voltage curves is given in figure 5.4.

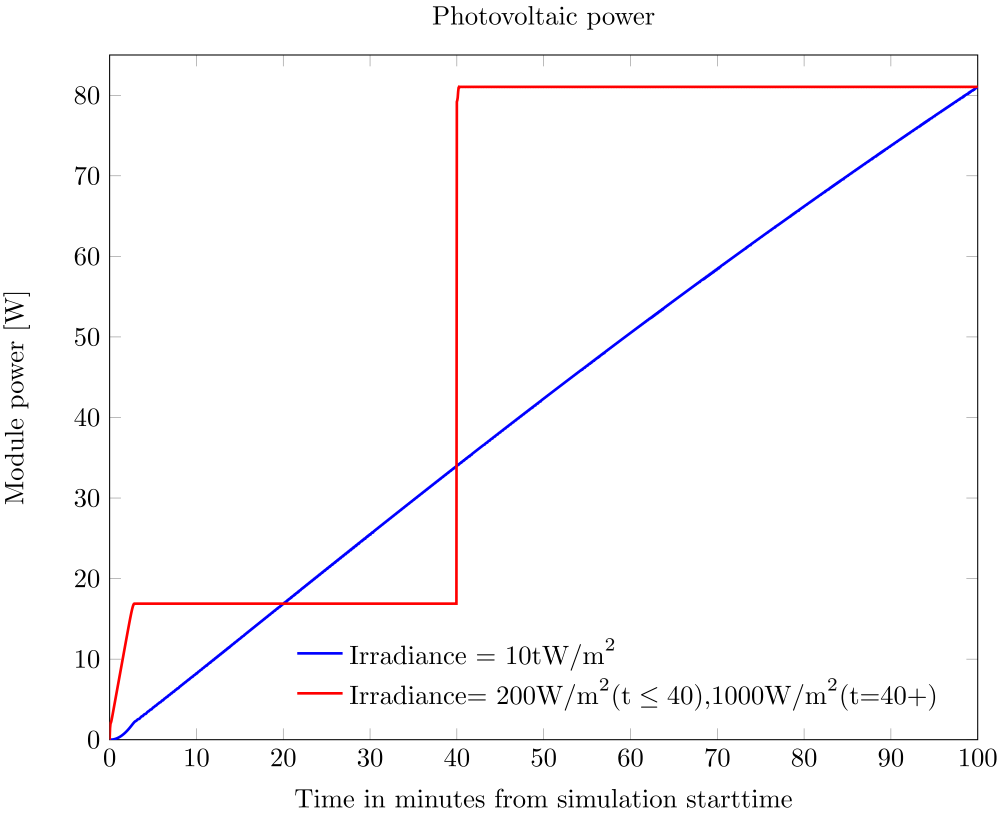


Figure 5.4: Module power output curves resulting from the voltage curves in figure 5.3. The two different irradiance profiles are denoted in the figure legends. The blue line is the product of a linear increase while the red line is the result from a step between two values. The highly variable voltage in the ramping case does not result in a very variable power curve since the variation in voltage is very close to the MPP and it causes little difference in output power.

start of both of the power curves. The red curve uses some time to reach the MPP power. This is because the starting voltage in the simulation is set to zero, and it takes some time to reach the appropriate voltage by the small step (V_{step}) for each iteration. The transition at $t = 40$ is a lot quicker because the change in voltage is much smaller.

5.5 Limitations of the MPPT implementation

In this paper, the MPPT operates on the same time scale as the other input data. The amount of perturbations per time unit is limited because of the amount of data to be handled. If there is 10 perturbations per data point of input irradiance, there will be tens of thousands of data points. Every data point of irradiance represents one minute in real time. A realistic perturbation frequency in real systems can be around 1-10 Hz[32]. 10 perturbations per data point would be equal to 1/6 Hz in perturbation frequency (one per 6 seconds) which is perhaps a too low frequency to represent how a real system would behave to great detail. Because this paper looks at phenomenons that are one the minute scale, rather than second scale, it is still a reasonable approximation as long as the voltage step value is sufficient. The transition in figure 5.4 is decently quick and this kind of response to change is speedy enough to closely followed measured data as can be seen in section 9.3.

Chapter 6

Battery modelling

6.1 Introduction

This chapter contains review of some ways to model battery modules and a more thorough explanation of the modelling method used in the simulation case.

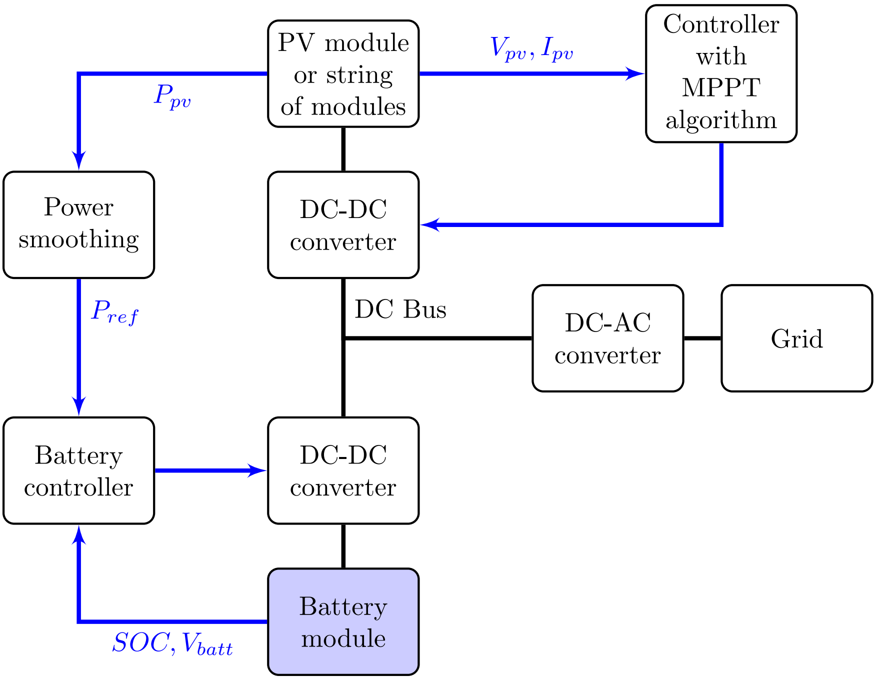


Figure 6.1: Overview of system used for simulation. The black lines indicate power flow and blue lines indicates signals. This chapter focuses mainly on the block "Battery module" that is filled with blue colour.

6.2 Simple linear battery model

One of the most commonly used battery models is shown in figure 6.2[33]. This model consists of only a DC voltage (V_0) and a constant resistance ($R_{internal}$). The main advantage with this model is its simplicity and easy extraction of parameters. Only two measurements are needed: One open circuit measurement at full charge to determine V_0 and another measurement with a connected load to calculate $R_{internal}$. The variables used are listed with units in table 6.1.

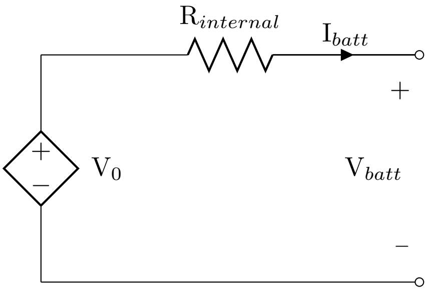


Figure 6.2: Circuit diagram of a simple linear battery model

This model can be suitable if the state of charge is not considered or the battery is assumed to have unlimited energy. The model does not take into account the effects of varying state of charge and is therefore not suitable if the available voltage at different charge levels is of interest. Even with an extremely simple model like this, the discharge curve can be modelled somewhat accurately in the linear area that can be seen in figure 2.1 from around 20 % to around 80% of discharge.

The voltage as a function of discharge level produced by this model is a straight horizontal line that goes up or down based on the amplitude of the discharging current.

6.3 State of charge dependent resistance model

A small modification of the model outlined in section 6.2 can be made by including a variable resistor that changes value based on the state of charge (SOC). The equivalent circuit shown in figure 6.3 is based on equation 6.1.

$$E = V_0 - (R_b + \frac{K}{SOC})I_{batt} \tag{6.1}$$

Equation 6.1 is an empirical mathematical model shown in[33]. The variables in the equation are explained in table 6.1.

The model is very simple and can be accurate over a large part of the SOC-range. The K/SOC-resistance adds a sharp reduction in voltage when the state of

Variable descriptions from sections 6.2 and 6.3	
$\mathbf{I}_{batt}$	Battery current[A]
E	Terminal Voltage[V]
$\mathbf{V}_0$	Open circuit voltage[V]
K	Polarization constant [Ω]
$\mathbf{V}_{batt}$	Battery terminal voltage[V]
$\mathbf{R}_b$	Terminal resistor, independent of SOC [Ω]
SOC	State of charge, number between 0 and 1[no unit]

Table 6.1: Explanation of variables in equation 6.1, figure 6.3 and figure 6.2

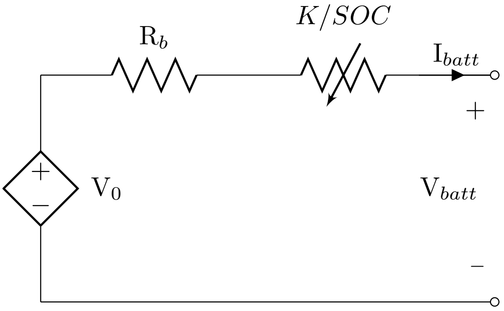


Figure 6.3: Circuit diagram based on equation 6.1

charge is low as can be seen in figure 6.4. The rise in voltage when the state of charge approaches 1 is absent and this model therefore is very inaccurate in that range.

This model was implemented in Simulink/Matlab. The parameters R$_b$ and K was chosen based on standard values from[33] and the voltage V$_0$ is chosen to be approximately equal to the nominal voltage of a lithium ion battery cell. The figure 6.4 is not representative of a real battery cell because there has been no effort to approximate the values to that of a real battery cell. It is included as an illustration of how a general charge/discharge curve from this model will be.

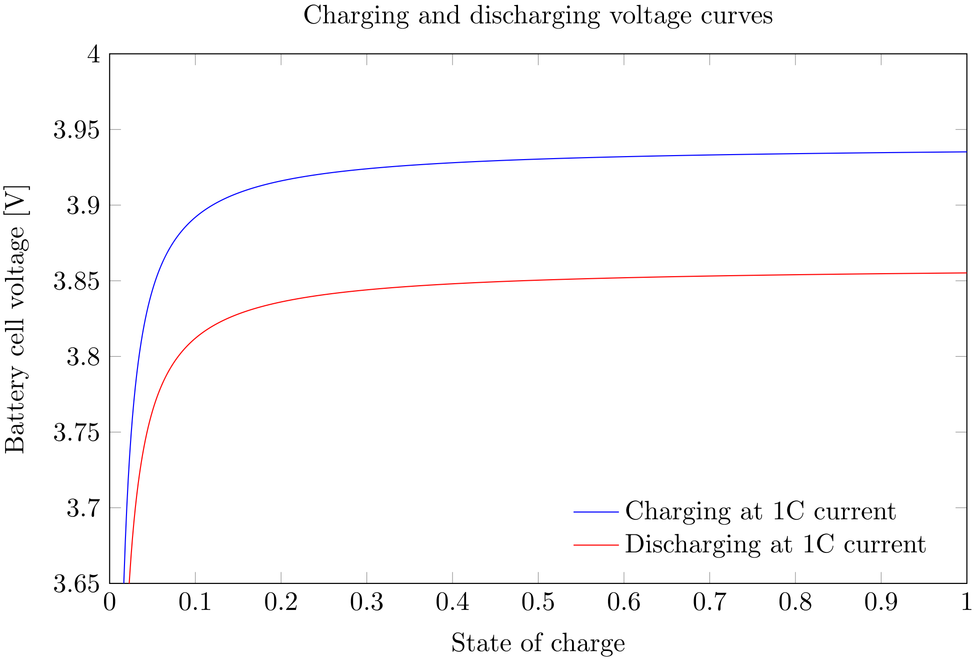


Figure 6.4: Charging and discharging voltage curve as a function of state of charge resulting from the implementation of the battery model discussed in section 6.3. The curve is not accurate at a high state of charge, but it can be decently accurate at low SOC and in the more flat area.

This model was abandoned because the voltage characteristics was not modelled accurately and the implementation suffered some problems with algebraic loops and instability. Other papers mention algebraic loop problems [34] and there might be a problem with the implementation.

6.4 Controlled voltage source model

Accurate modelling of a battery is a very complex process that requires knowledge of the electrochemical processes within. For crude energy simulation purposes, a model that accounts for the varying voltage that is available based on the SOC can be sufficient.

The model presented in this section uses only the SOC as a state variable. The voltage is modelled by a controlled voltage source that is dependent on the SOC as shown in equation 6.2. The explanation of the variables in the following equations and circuit diagrams is available in table 6.2.

$$E = E_0 - K\frac{Q_0}{Q_0 - Q} + A\exp -BQ \tag{6.2}$$

In equation 6.2, the variable Q represents the actual battery charge and the variable Q_0 is the rated battery capacity. The SOC is therefore the relation Q/Q_0. The actual battery charge is modelled by integrating the battery current and adding the start charge as shown in equation 6.3.

$$Q_{start} + \int_0^t -I_{batt}dt = Q \tag{6.3}$$

The starting charge (Q_{start}) is omitted in the equivalent circuit diagram in figure 6.5, but is a part of the calculation at all times. Notice that the sign of the current in the charge calculation is negative. The current is defined as positive in the direction of the arrow in figure 6.5. If the current is positive, the integral in equation 6.3 will result in negative charge, reducing the total charge of the battery. This is consistent with the battery delivering current and a reduction of state of charge.

The battery voltage according to this model is expressed by equation 6.4.

$$V_{batt} = E - I_{batt}R_{internal} \tag{6.4}$$

The charging voltage will be higher than a discharge voltage with the same current by $2I_{batt}R_{internal}$ because the discharge voltage is reduced by $I_{batt}R_{internal}$ since the current is positive, the opposite is true for a charging voltage.

Figure 6.5 shows the equivalent circuit of this controlled voltage source model of a battery cell described by equations

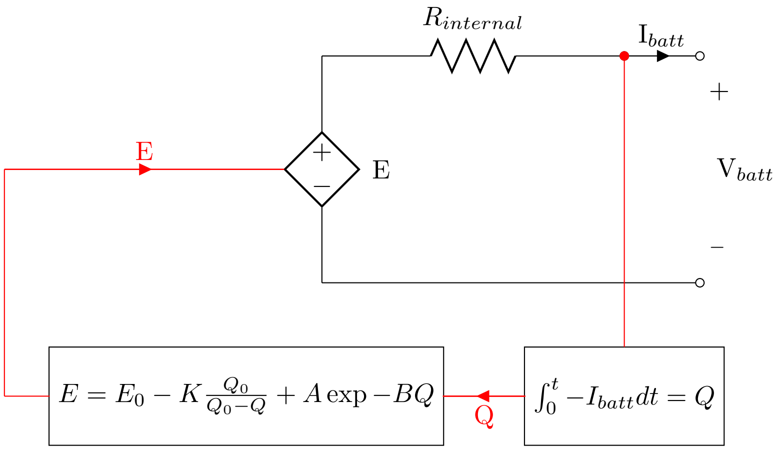

$R_{internal}$
I_{batt}
+
E
E
V_{batt}
−
$E = E_0 - K\frac{Q_0}{Q_0-Q} + A\exp -BQ$
Q
$\int_0^t -I_{batt}dt = Q$

Figure 6.5: Circuit diagram of the battery model with a non-linear controlled voltage source. Black lines indicate power flow and red lines indicate information flow.

Variable descriptions from figure 6.5

$\mathbf{I}_{batt}$	Battery current [A]
E	Voltage at no load [V]
K	Polarisation voltage [V]
Q	Actual battery charge [Ah]
$\mathbf{Q}_0$	Rated battery capacity [Ah]
$\mathbf{E}_0$	Battery constant voltage [V]
$\mathbf{V}_{batt}$	Battery terminal voltage [V]
$\mathbf{R}_{internal}$	Constant internal resistance [Ω]
A	Exponential zone amplitude [V]
B	Exponential zone time constant inverse [Ah^{-1}]

Table 6.2: Description of variables in battery model shown in figure 6.5 and equations 6.2 and 6.4.

6.4.1 Model assumptions

There are several approximations and assumptions that are done by modelling a battery in the way described in figure 6.5.

Constant internal resistance The internal resistance ($R_{internal}$) is assumed to be constant and independent on current amplitude and SOC that affects the real internal resistance in a battery.

Valid model parameters The parameters extracted from the manufacturer discharge curve are assumed to be valid for other conditions. The charging characteristics are assumed to be the same as the discharging characteristics.

Battery capacity The battery capacity is assumed constant. There is no variation of capacity with respect to current amplitude. In other words, the model does not consider the Peukert effect (discussed in section 2.1.4).

Temperature No temperature dependence is included in the model.

Self-Discharge There is no consideration of the self discharge of the battery.

6.4.2 Model limitations

Voltage The maximum battery voltage is not limited by the model and therefore has to be limited outside the battery module. The minimum no-load voltage (E) is 0 V.

Battery capacity The state of charge is not limited by the model itself and it can exceed 100 %. This has to be managed by the battery controller.

Battery capacity The Peukert effect (discussed in section 2.1.4) is not considered.

6.4.3 Approximation of parameters

The parameters can be extracted from manufacturers discharge curves. This process is described in detail in [34]. This section contains a short description of how these variables can be approximated from discharge curve, but the values used in this paper is taken from [34] shown in table 6.3. The parameters of the Lithium-ion battery cell is used for simulation.

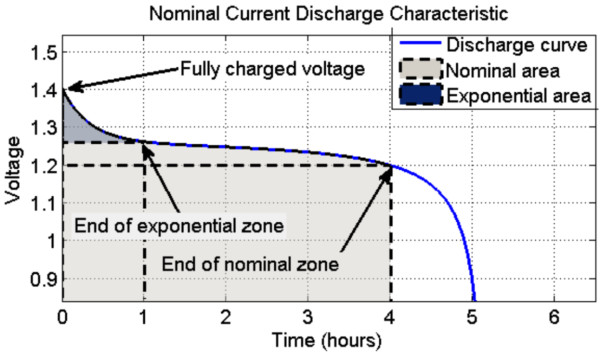


Figure 6.6: Typical discharge curve with arrows showing points of interest for approximation of battery model parameters. Figure taken from [34]

The parameter A can be approximated by equation 6.5:

$$A = E_{full} - E_{exp} \tag{6.5}$$

In equation 6.5 the voltage E_{full} corresponds to the fully charged voltage in figure 6.6 and the voltage E_{exp} corresponds to the voltage at the end of the exponential zone.

Parameter B can be approximated by the relation in equation 6.6

$$B = \frac{3}{Q_{exp}} \tag{6.6}$$

Here, Q_{exp} is the battery charge at the end of the exponential zone.

Battery type **Parameters**	**Lead-Acid**	**Nickel-Cadmium**	**Lithium-Ion**	**Nickel-Metal-Hydrid**
Nominal voltage [V]	12	1.2	3.6	1.2
Nominal capacity [Ah]	1.2	1.3	1	6.5
E_0 [V]	12.6463	1.2505	3.7348	1.2848
R [Ω]	0.25	0.023	0.09	0.0046
K [V]	0.33	0.00852	0.00876	0.01875
A [V]	0.66	0.144	0.468	0.144
B [Ah^{-1}]	2884.61	5.7692	3.5294	2.3077

Table 6.3: Parameters for different battery cell types extracted from manufacturer's discharge curve at 1C current[34]

The polarization voltage can be calculated by using values at another point on the discharge curve, the end of nominal zone voltage (E_{nom}) and charge (Q_{nom}):

$$K = \frac{(E_{full} + A(\exp - BQ_{nom}))(Q_{rated} - Q_{nom})}{Q_{nom}} \tag{6.7}$$

The internal resistance ($R_{internal}$) is important to model the voltage drop caused by current variation. Commercial data sheets usually provides a value, but these values does not necessarily result in good curve fitting when using this model[34] and it is suggested to use the nominal current, charge and efficiency to give a value resulting in better curve fitting.

The battery constant voltage (E_0), can be calculated using the fully charged voltage as shown in equation 6.8:

$$E_0 = E_{full} + K + R_{internal} I_{batt} - A \tag{6.8}$$

Here the current (I_{batt}) is the current that is used to give the discharge voltage curve.

6.4.4 Discussion of approximation method

The paper [34] demonstrates that this model results in discharge curves that track those of the manufacturer closely. The model and manufacturer curves track better at the discharge rate at which the parameters was extracted.

The general shape of the discharge curve and the voltage variation with state of charge is modelled at a level of accuracy that probably is sufficient to the application in this paper.

6.4.5 Battery charge and discharge curves

This section contains charging and discharging curves of the battery model implementation in Simulink. In figure 6.7, it can be seen that the charging voltage is consistently higher than the discharging voltage. This is because of the voltage drop over the internal resistance $R_{internal}I_{batt}$. The current used to produce these plots, is of 1C amplitude. That is equal to the value of total rated charge capacity of the battery in amps. In this case the total capacity is 1 Ah and the 1C current is therefore 1 A.

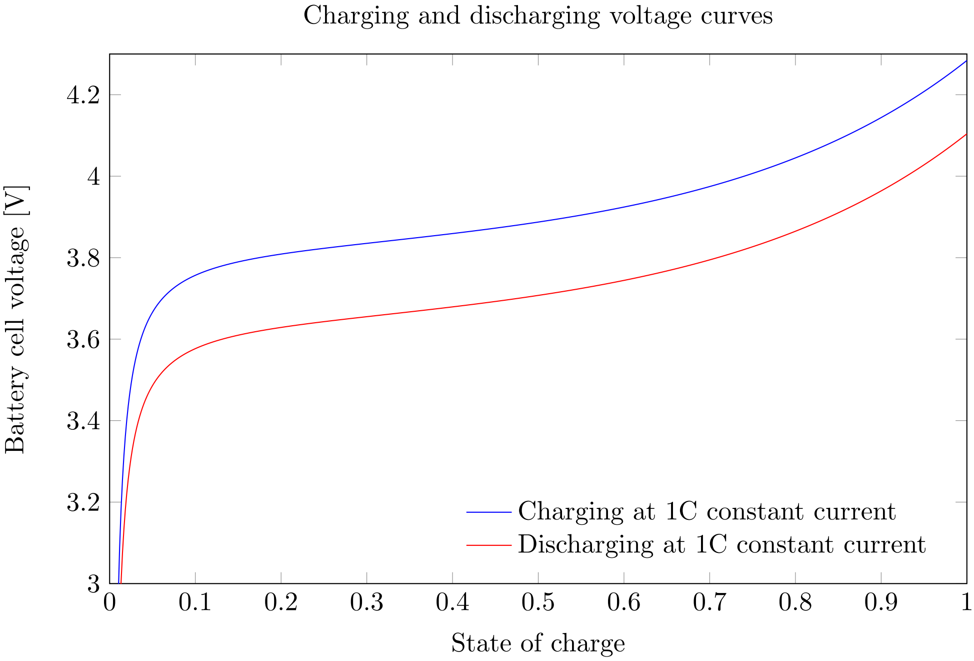


Figure 6.7: Charging and discharging voltage curve as a function of state of charge

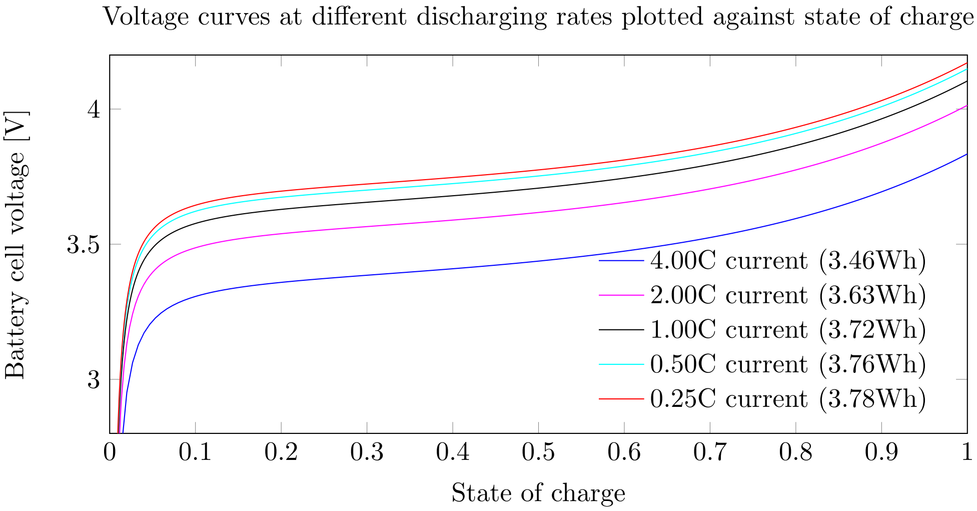


Figure 6.8: Discharge voltage curves with different constant currents labelled by C-values plotted against state of charge. Total energy delivered from the battery during the discharge is indicated in parentheses in the figure legend.

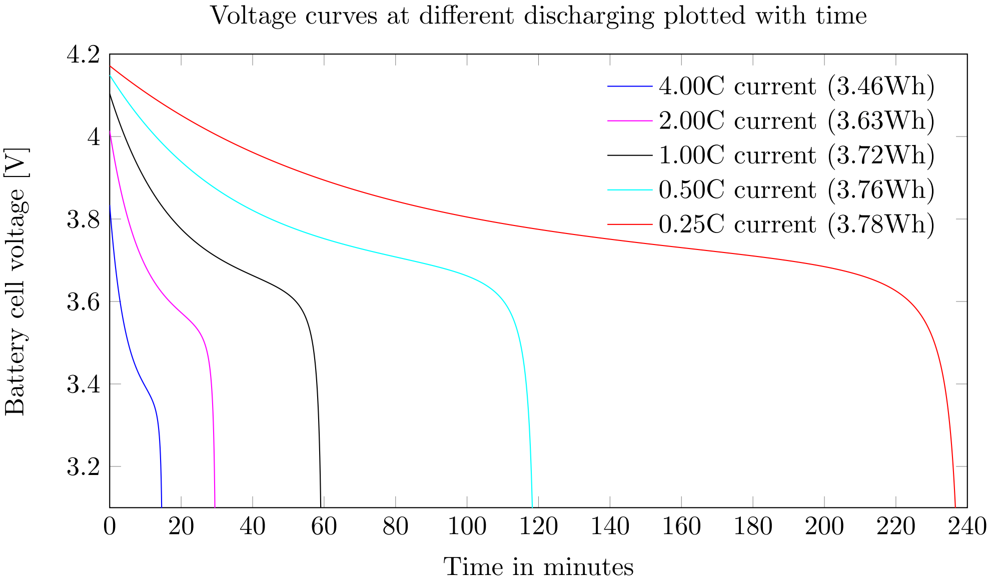


Figure 6.9: Discharge voltage curves with different constant currents labelled by C-values plotted against time. Total energy delivered from the battery during the discharge is indicated in parentheses in the figure legend.

6.5 From cell to module

In this paper, various configurations of batteries are needed for different simulation cases. The battery model implementation needs to be modifiable by scripts so that effects of different capacities, power rating and other limitations can be examined.

Parameters of one battery cell are available and this fundamental unit of one battery cell can be used to construct many different battery modules. The way these modifications are done in this paper, is that at all times the fundamental model of the battery cell is run with the same basic parameters. This way, there are always the same fundamental characteristics in the model and no model parameters has to be changed. The modifications are done outside of the cell. Figure 6.10 shows how the controlled voltage source model can be modified in a case of N_s cells connected in series and N_p strings in parallel. The voltage source is still controlled by the state of charge level, but this equation is not shown in the figure for simplicity.

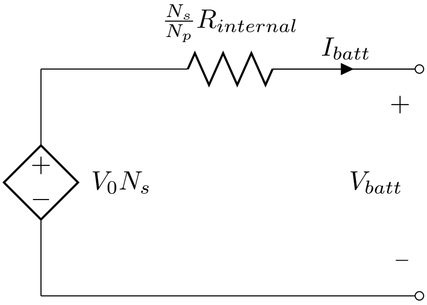


Figure 6.10: Circuit diagram of a battery module with N_s battery cell for each string and N_p strings in parallel. The voltage source is the controlled voltage source that is discussed in section 6.4

In figure 6.10 there is one modification that must be done that is not displayed. With multiple strings or cells connected in parallel, there will be a higher ampere-hour rating and a higher maximum charging current. Both will change by a factor of N_p. This must therefore be implemented both in the battery module model and the controller.

6.6 Battery module control

There are several constraints to consider when controlling a battery module. The manufacturer usually specifies maximum charging voltage, cut-off discharge voltage, maximum charging current and maximum discharge current.

The controller for the battery module implemented in this paper uses constraints from [35] shown in table 6.4. The maximum and minimum SOC are arbitrary values chosen as limits. The factor N_s denotes number of cells in series.

Battery constraints	
Maximum discharge current	2C
Maximum charging current	1C
Maximum charging voltage	4.2VN_s
Discharge termination voltage	3.0VN_s
Max SOC	90%
Min SOC	10%

Table 6.4: Battery controller constraints

The battery controller receives a power reference signal from the power smoothing algorithm block and checks if the battery is able to match the reference within the constraints discussed here. If it is, the voltage is set to a level that will give the reference power. If not, then the output of the battery is matched as closely as possible within the constraints. The controller then calculates the duty ratio of the converter which then supplies the voltage to the battery model. The converter is the same as in figure 8.9b except form the RC-filter.

The full implementation of the battery controller is available in appendix A.5.

6.6.1 Weakness and suggestions for improvement

There are several weaknesses with the way the battery controller is implemented. It only takes into account if the reference is within the constraints and there is no consideration to limit the output to ensure that the battery does not reach the limits of state of charge. Some simple modifications in control are suggested in [4] where the irradiance and the distance from the desired SOC level is taken into consideration. Improving the battery model control is an area where future work should be done.

Chapter 7

Power smoothing algorithms

The battery model is to be used for smoothing of photovoltaic power output. An algorithm needs to produce a reference power for the battery module to smooth the PV output. Rapid fluctuations in power need to be smoothed and the general trend needs to be tracked.

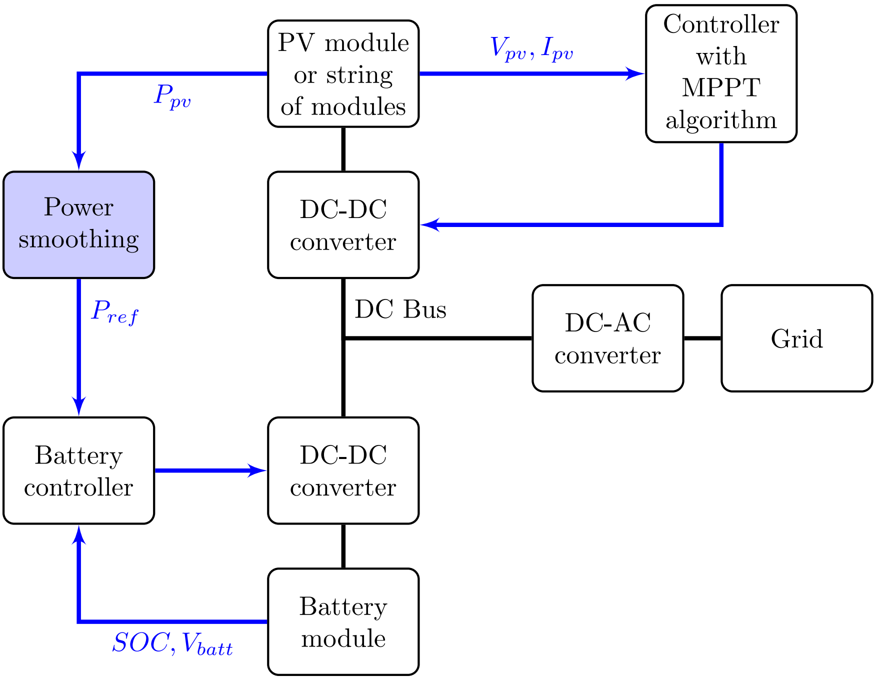


Figure 7.1: Overview of system used for simulation. The black lines indicate power flow and blue lines indicates signals. This chapter focuses mainly on the block "Power smoothing" that is filled with blue colour.

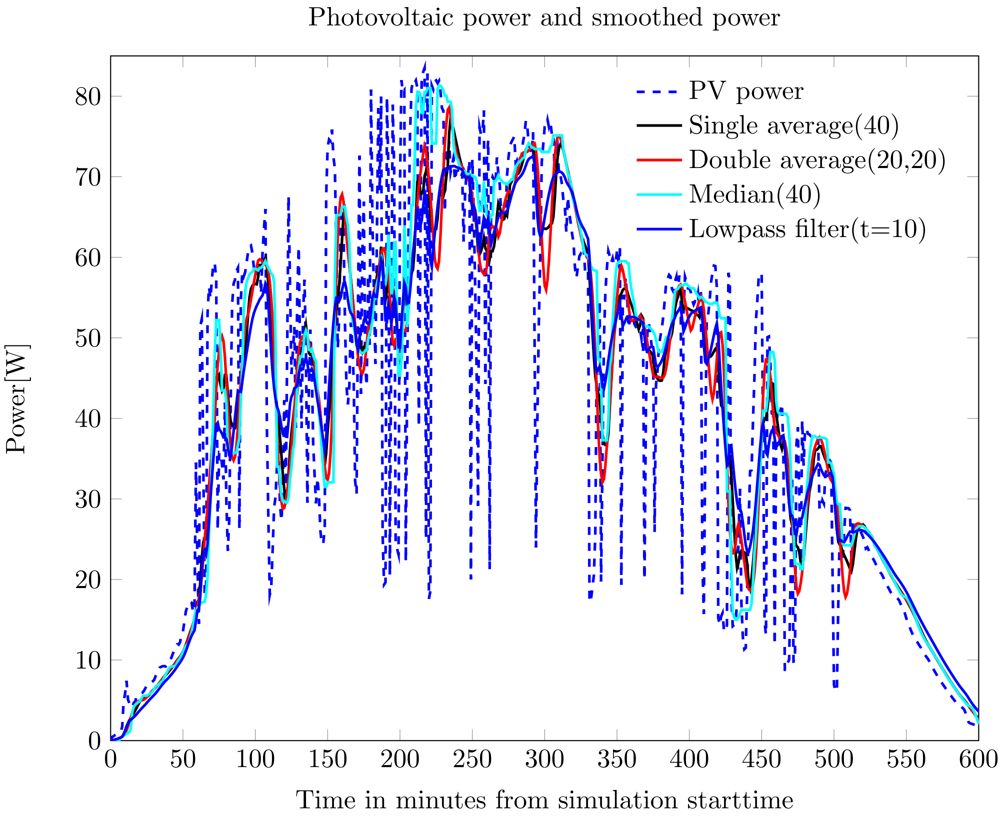


Figure 7.2: Plot of photovoltaic power on a variable day and the result from smoothing algorithms by single and double averaging. The averaging times are given in parentheses. Double averaging gives more smoothing but more delay. The PV power data is the same as the data presented in chapter 9

There are several different ways to implement a smoothing algorithm. The resulting smoothness can be manipulating by adjusting the windows size that the algorithms uses. All the algorithms use some part of the PV power for averaging or other operations. How large a part of the power output is used is given by window size. In figures 7.2, 7.3 and 7.4 the window size is given in minutes. The algorithms "remembers" the values as far back as given by window size and calculates the output based on these values.

Larger window sizes leads to smoother output but also a larger delay. Choosing the appropriate window sizes for the algorithms is therefore a problem of choosing between very smooth output with some delay and bad tracking or an output with good tracking but a lot of fluctuation.

Bad tracking and delay compared to the PV power output can be a problem because it can lead to some unnecessary charging or discharging of the battery module and larger window sizes therefore raises the required energy capacity of the battery module.

7.1 Moving average algorithms

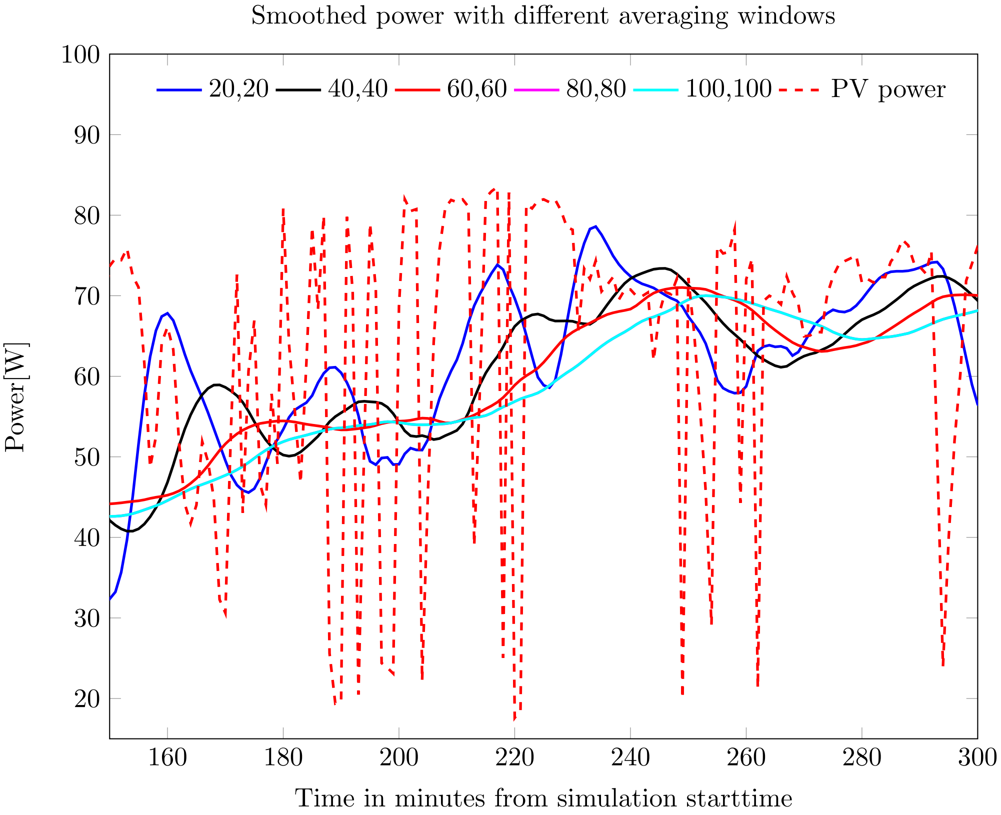


Figure 7.3: PV power and reference power resulting for a double averaging algorithm with minutes of memory given in parentheses. The averaging window is given on this format (window size 1 in minutes, window size 2 in minutes) in the legend. Longer averaging windows gives smoother output but also more delay. This is graph from parts of the same output PV data set as in figure 7.2.

7.1.1 Single moving average

The moving average algorithm is very simple. Window size and the power are the only two inputs. The average value of the power as far back as the window size is calculated and the resulting value is the output from the algorithm. This algorithm is commonly used to calculate the power reference in smoothing applications[36, 37, 4]. This kind of algorithm has been implemented in real PV power smoothing battery systems with an averaging window of up to 60 minutes[4]. The implementation of this algorithm is in appendix A.3.

7.1.2 Double moving average

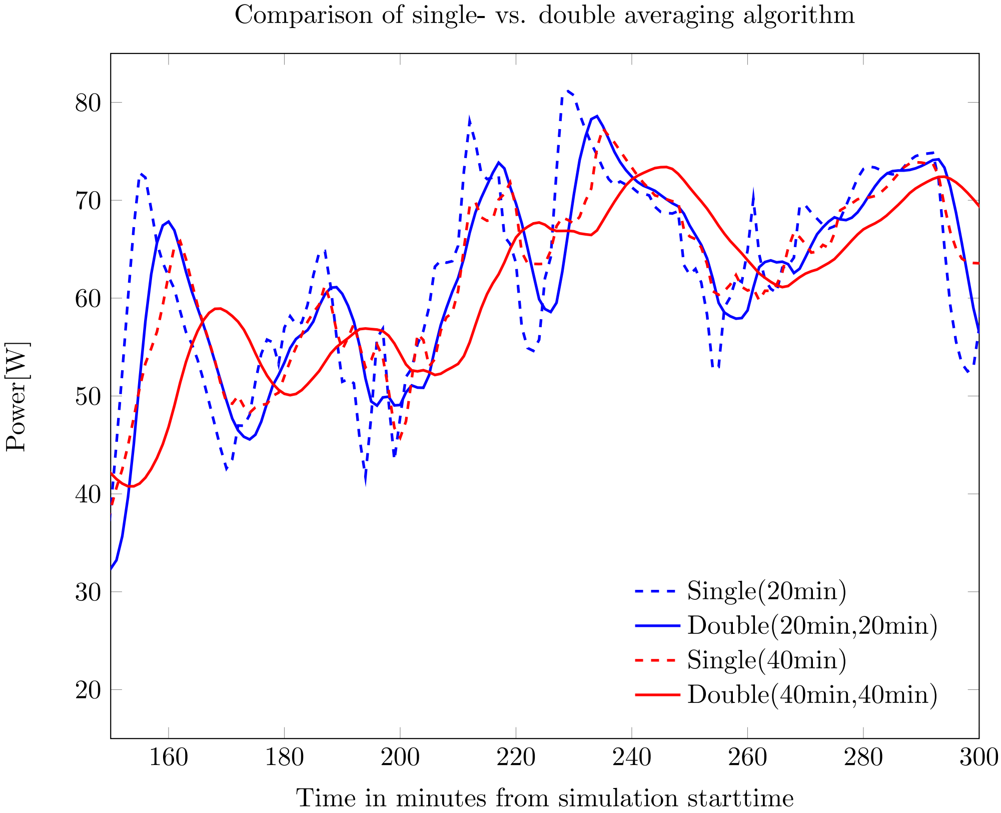


Figure 7.4: Graph showing difference between single and double averaging algorithms. The notation of the legend is averaging windows in minutes. This is graph from parts of the same output PV data set as in figure 7.2.

The moving average algorithm can be used twice in a row by supply the resulting output of the first into the second. By doing this, the output can be made smoother while maintaining better tracking and shorter delay than that of an equally smooth output from single averaging. The resulting output of single and double averaging outputs are shown in figure 7.4.

As can be seen in figure 7.4, the double averaging algorithm results in significantly smoother output while not increasing the delay as much as required by a single averaging algorithm of the same resulting smoothness. To achieve the same degree of smoothness, the single averaging implementation needs a window that is about twice as large.

7.2 Moving median

Calculating the median value in a power vector of length given by window size is another way to smooth the output. The median has an advantage over the averaging algorithm in that rapid and transient fluctuations is underplayed. If the output power dataset inside the window contains a lot of outlying data points, these are more efficiently handled by a moving median. The moving median produces less smoothness for the same window size as the outer smoothing method as can be seen in figure 7.2.

7.3 Low pass filter

A very simple way to smooth the power, at least in a simulation environment, is to just apply a low pass filter to the PV output. The low pass filter output reference power shown together with the other outputs in figure 7.2 has a time constant of 10 minutes. This value has been chosen to give a result similar to the resulting output from the other methods. The output is about as good at that of the averaging algorithms. This method is used in the literature in simulation cases[38]. In a real system, a low pass filter with a time constant (RC-value) of 10 minutes or more may not be viable. Using a filter also excludes any modification of the averaging on the fly.

7.4 Discussion of algorithms

Of the smoothing methods presented, the double moving average algorithm and the low pass filter seems to provide the best smoothing without too much delay. The low pass filter method has a major drawback in that it is a static implementation. The smoothing of the filter cannot be changed depending on what is needed at any time.

It is plausible that a battery pack controlled with a reference power from a double moving average algorithm can be used for other purposes if the PV power is already decently smooth because of stable weather. An averaging algorithm can use a rapidly changing window size to provide the required smoothing at any time within its limits. A passive filter will not have this capability. The double averaging smoothing method is for this reason used for the simulation in the rest of this paper.

Chapter 8

Simulink implementations

8.1 Introduction

This chapter contains an exploration of the modelling process and structure.

During the entire modelling process, there has been a focus on easy adjustment. Every parameter in all constructed models is imported from state variables from Matlab. Before running a Simulink simulation case, a Matlab script that defines the parameters for the system has to be run to set the state variables to the appropriate values. This way of implementing the systems has some major advantages.

Producing output with different model parameters is as simple as defining the desired variable and run the model inside a for-loop. Most of the figures in this paper is produced by a script that defines the variables and runs simulation for different cases. An example of such a script is available in appendix A.2. The resulting data from Simulink is automatically imported to Matlab as state variables. A script can be written to make plots and save the plotted data to appropriate file formats for plot construction in LaTeX.

The implementation is very flexible because of this ease of modification. In fact, it is so flexible that a script could prompt a user for the input data sheet values for the solar module he or she wants to simulate together with other optional variables like number of PV modules in a string and the script can define the variables necessary and run a simulation with available irradiance data. This is possible because the parameter approximation script (appendix A.1) automatically returns approximated values as Matlab state variables that can be used for a Simulink model.

To produce all the figures and results in this paper, several configurations of PV module models, other systems and battery models must be constructed. Some of the blocks implemented in Simulink are not modified for different simulation cases and include: Battery module, PV module, MPPT controller, Power smoothing and DC-DC converter. These systems are copied into different simulation cases and are only modified by the parameter values defined in the scripts that run the simulations.

8.2 Implementation of PV model

This section contains printed figures of the Simulink/Matlab implementation of the photovoltaic module. The systems are explained in the figure captions.

Large blocks with inputs with variable names are subsystems in the model. A map of the structure of subsystems is shown in figure 8.1.

Rectangular boxes with variable names are inputs from Matlab. Scripts running the simulations can define all of these variables. This enables the construction of the figures in section 4.5 by running several simulations where one of the variables are changed for each simulation run.

The PV system is a model of a string of one or more modules. It can even be parts of a module if it interesting to examine effects of partial shading. The parameters of one fundamental unit are used to model a string of these units. The unit can be module, a cell or a string within a module. A parameter called "Num-Modules" is used to factor the voltage up to the desired number of units in a series connection. To model several strings in parallel, the system showed in figure 8.2 can be copied and connected together or a multiplication of the output current could be made. Modelling partial shading and other similar non-uniform conditions can be done by creating a larger system consisting of the PV model presented here.

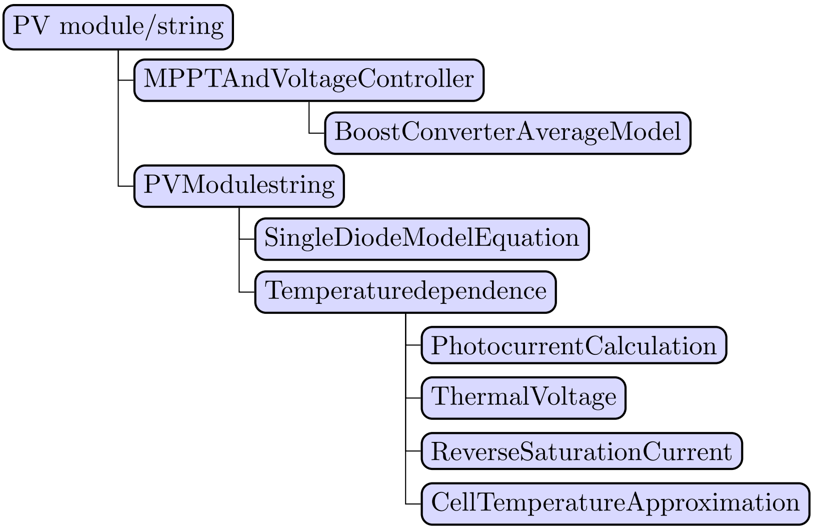


Figure 8.1: Photovoltaic module model structure. Blocks below and to the right are subsystems of the parent blocks that they are connected to.

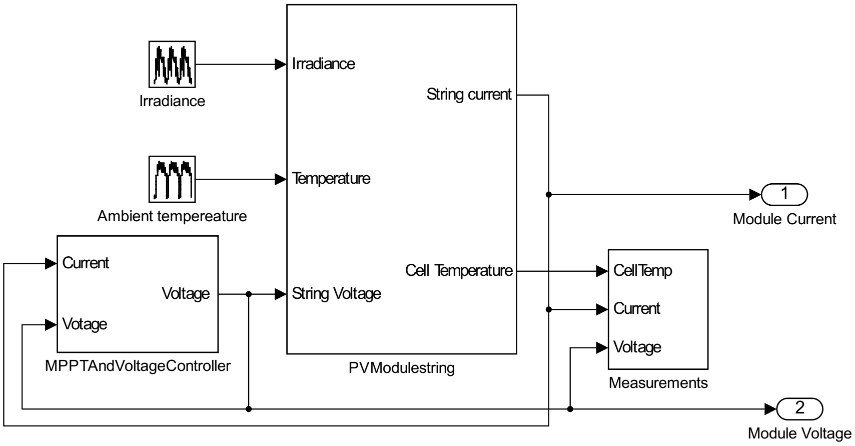


Figure 8.2: This is the top PV module/array model that contains all the other subsystems. Here are the inputs of irradiance and ambient temperature and outputs of module voltage and current. The MPPT algorithm and converter are under the "MPPTAndVoltageController" subsystem and all the equations modelling the behaviour of a PV module are under the "PVModulestring" subsystem

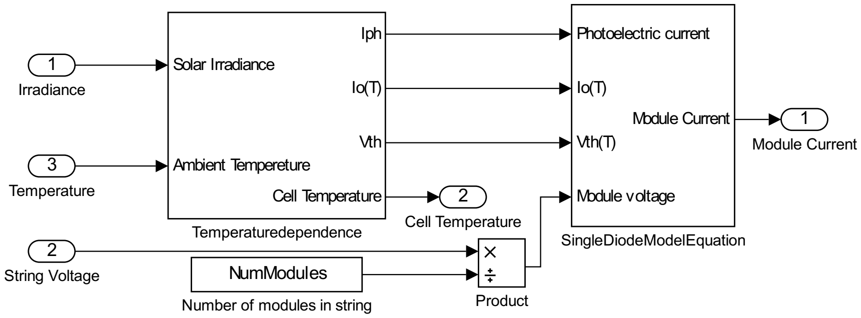


Figure 8.3: This is the subsystem named "PVModulestring" from figure 8.2. The model is here divided into a temperature dependent part in subsystem "Temperaturedepedence", and a part that has no temperature dependence in subsystem "SingleDiodeModelEquation". The parameter "NumModules" is used to set the voltage at the value that each module in a string will have.

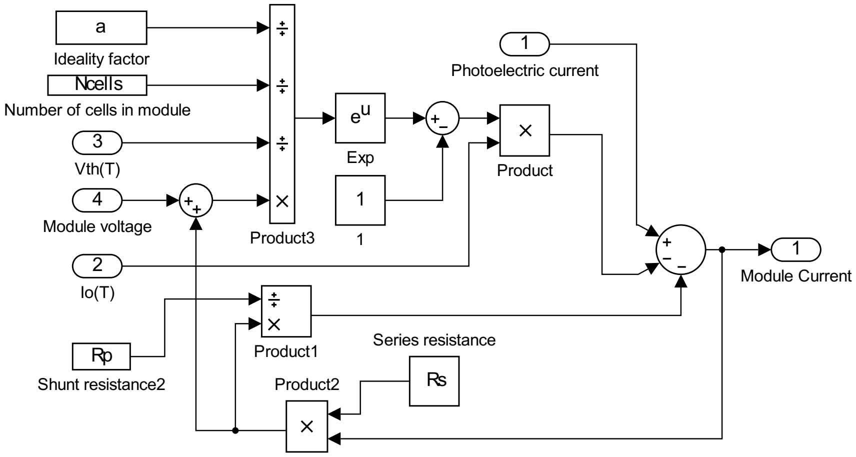


Figure 8.4: This is the Simulink implementation of the single diode equation discussed in section 4.1.2. This is the subsystem named "SingleDiodeModelEquation" in figure 8.3

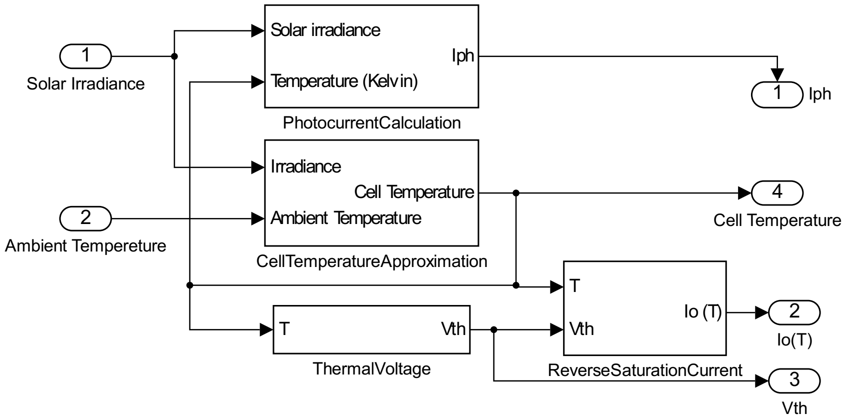
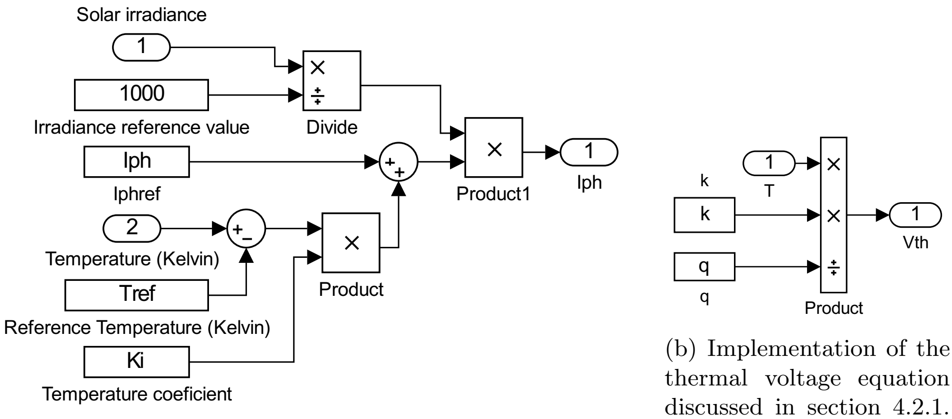


Figure 8.5: This is the subsystem "Temperaturedependence" in figure 8.3. All the blocks here are dependent on the temperature and the resulting calculations are fed into the single diode equation in figure 8.4. The block "CellTemoeratureApproximation" (figure 8.8) calculates the cell temperature, which is then fed into the other blocks.



(a) This the subsystem labelled "PhotocurrentCalculation" in figure 8.5 and it is the implementation of the photo current equation discussed in section 4.2.3.

(b) Implementation of the thermal voltage equation discussed in section 4.2.1. This is the subsystem labelled "ThermalVoltage" in figure 8.5.

Figure 8.6: Two of the subsystems in figure 8.5

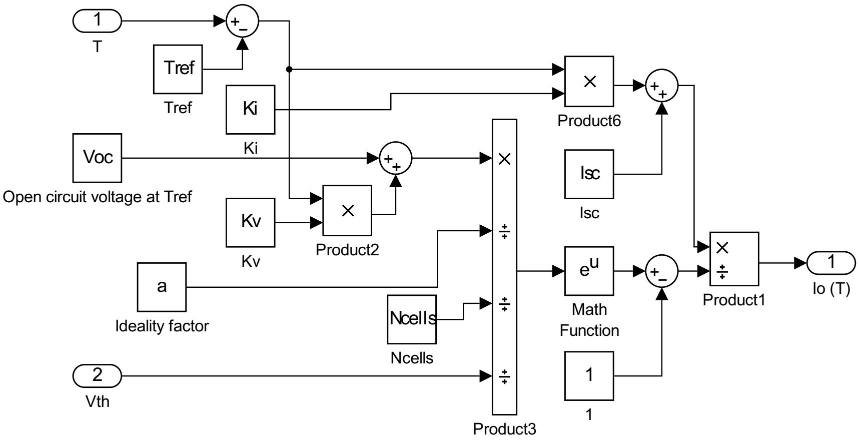


Figure 8.7: Implementation of the reverse saturation current equation (equation 4.6) discussed in section 4.2.2. It is the subsystem named "ReverseSaturationCurrent" in figure 8.5

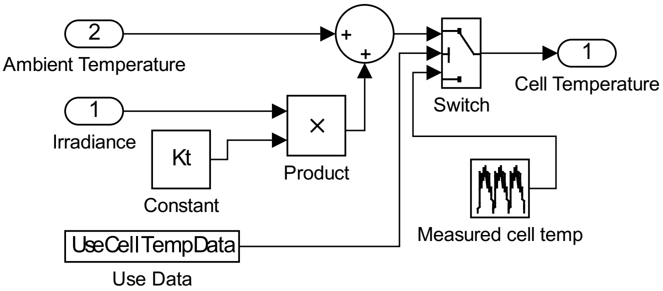
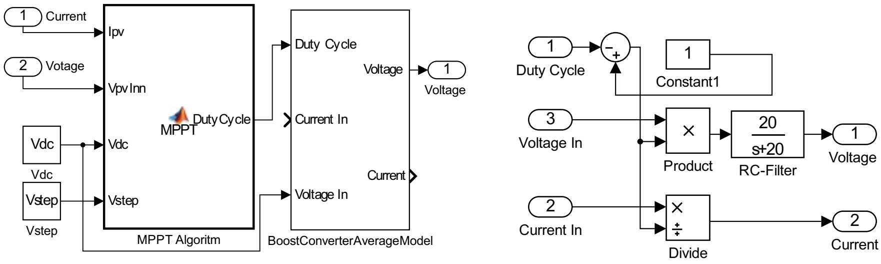


Figure 8.8: This is the subsystem "CellTemperatureApproximation" in figure 8.5. Here an approximation of cell temperature by irradiance,ambient temperature and a temperature coefficient (K_t) is done as discussed in section 4.2.5. Notice the switch and the input of "Measured cell temp", this is an option to switch between actual measured values and the approximation. This switching is done by setting the value "UseCellTempData" in the script running the simulations and this is exploited in section 9.4.



(a) Implementation of the MPPT algorithm discussed in chapter 5. The full Matlab code inside the block ”MPPT Algorithm” is available in appendix A.4. This is the subsystem labelled ”MPPTAndVoltageController” in figure 8.2.

(b) This the subsystem labelled ”BoostConverterAverageModel” in figure 8.9a and it is an average model of an ideal boost converter. The RC-filter has been added for reasons discussed in section 8.5.1.

Figure 8.9: The two subsystems under the system labelled ”MPPTAndVoltage-Controller” in figure 8.2.

8.3 Implementation of battery model

This section contains figures of the Simulink/Matlab implementation of the battery model discussed in chapter 6. The explanations of the model parts are included in the figure captions.

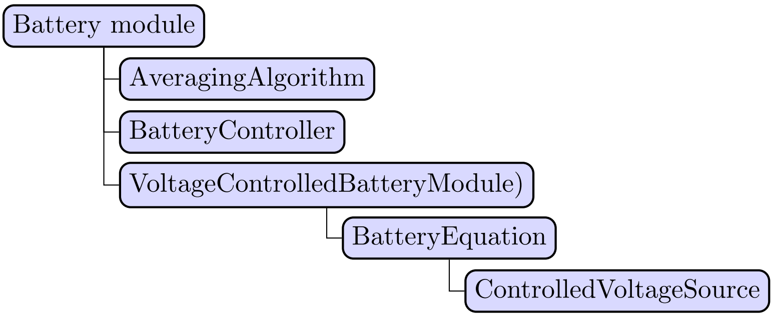


Figure 8.10: Overview of the battery model structure. Blocks below and to the right are subsystems of the block they are connected to.

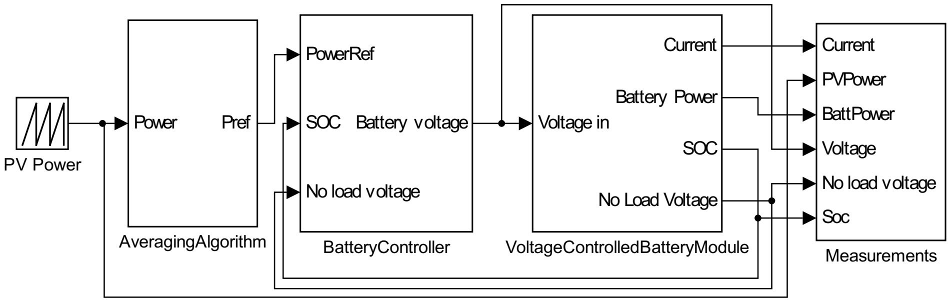


Figure 8.11: This is the implementation of the battery model described in chapter 6 and the power smoothing algorithms described in chapter 7. In the model displayed here, the input (”PV Power”) is a previously saved data set from the PV model. This is a way to reduce the runtime of the model.

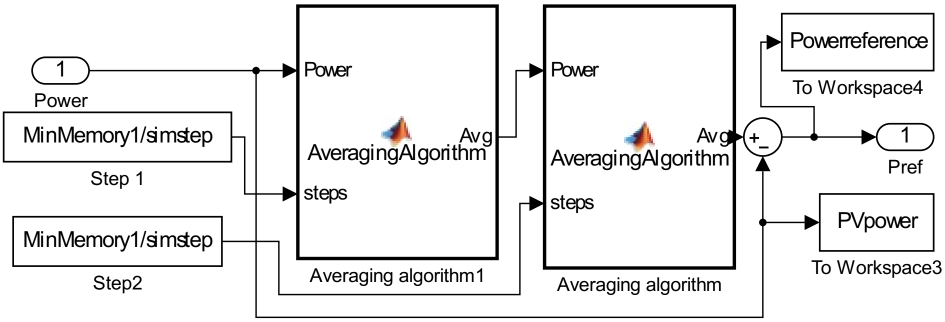


Figure 8.12: This is the subsystem under the block ”AveragingAlgorithm” in figure 8.11. There are two blocks containing the same algorithm (available in appendix A.3) where the output of the first one is the input of the next. This is how the double averaging algorithm is implemented. The inputs are Power and steps, where steps gives the length of the averaging window. The effect of different averaging windows are discussed in section 7.1.2

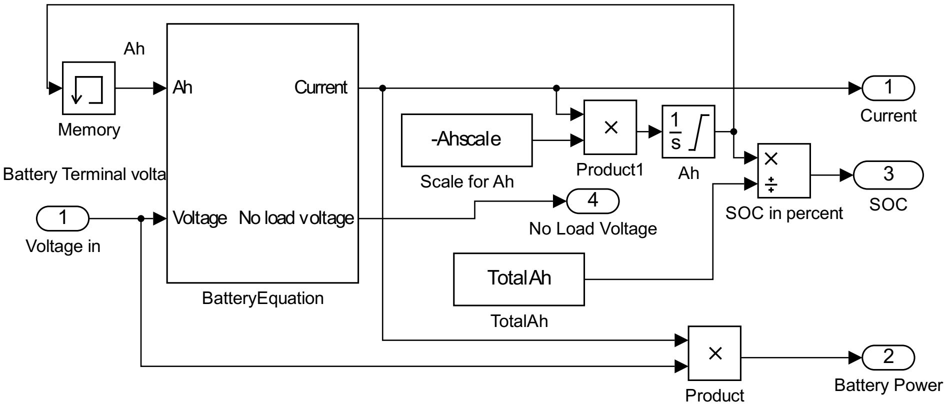


Figure 8.13: This is an implementation of the charge level and power of the battery and it is the subsystem ”VoltageControlledBatteryModule” in figure 8.11. The current is integrated by the block ”Ah” with a scale of ”-Ahscale”. The scale takes into account what time unit is used in the simulation to ensure that the unit is correct for the integration. The negative sign in ”-Ahscale” is there because the current is defined as positive out from the battery. To drain the capacity when the battery delivers power, the sign in the integration must be negative. The ”Memory” block introduces a unit delay and is there to help the simulation by removing the algebraic loop that causes problems with convergence there otherwise. This delay does not change the dynamics to a significant degree because the value ”Ah” changes very slowly and only affects the controlled voltage source very slightly with small differences in charge level.

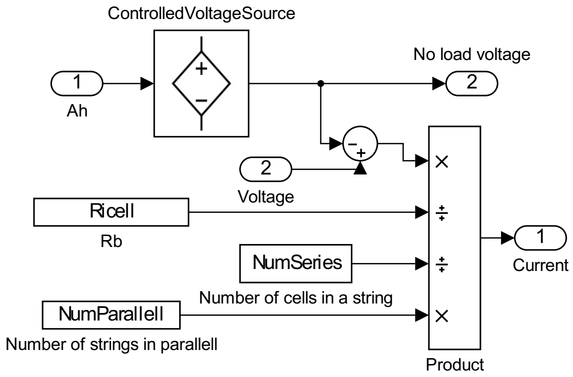


Figure 8.14: This is the subsystem "BatteryEquation" in figure 8.13. The input "NumSeries" and "NumParallell" enable automatic resizing of the battery module. "NumSeries" denote the number of battery cells in series an "NumParallell" denote the number of strings in parallel. These values can be changed by a script that runs the simulation to fit the scenario to be examined. This resizing of the battery module is implemented as discussed in section 6.5. With N_s cells connected in series the current will be equal to that of one cell, but the voltage will be N_s times higher. The equivalent resistance must therefore be N_s as large as that of one cell. A similar argument can be made for a parallel connection

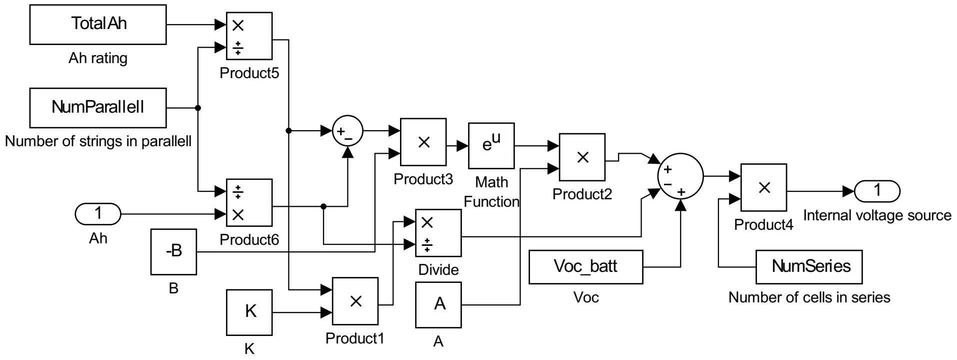


Figure 8.15: This is the implementation of the controlled voltage source equation (6.2) discussed in section 6.4. The only difference in the implementation is the modification bu the blocks "NumParallell" and "NumSeries". Both the parameter "ToltalAh" and the state variable "Ah" are divided by "NumParallell" and is therefore converted to the value that one battery cell in the configuration would have. The output voltage is increased by a factor of "NumSeries" to model a series connection. The other parameters are set to values corresponding to one battery cell, while "TotalAh","NumParallell" and "NumSeries" modify the model to different configurations based on a single cell.

8.4 Case specific models

Several of the situations examined in this paper requires a special configuration of the basic model blocks described in this chapter. For example, the figures in section 4.5 are produced by removing the maximum power point tracking block and controlling the input voltage with a variable resistor as shown in figure 8.16.

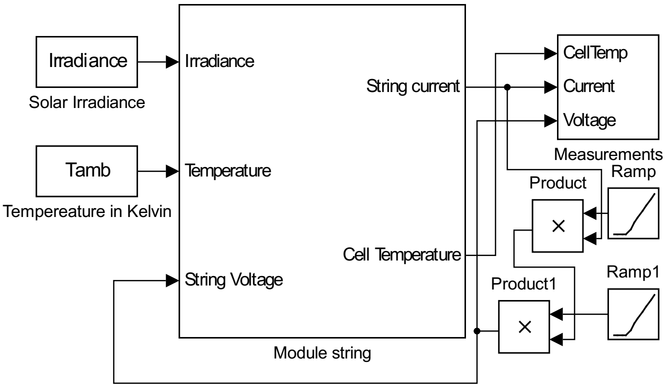


Figure 8.16: Model used for producing all the figures in section 4.5. The MPPT algorithm and converter is removed and replaced by two ramping functions.

The ramp functions in figure 8.16 that are multiplied by the output current acts as linearly increasing variable resistors. The voltage input will be $V_{pv} = I_{pv} * Ramp * Ramp1$ There are two of them to produce good resolution in plots both in the rapidly changing area, and in the area where the voltage is closing in on V_{oc}. The second ramp function only activates a long time into the simulation to provide a much higher resistance to make the voltage close to V_{oc} at the end of the simulations.

The model shown in figure 8.16 is an example of a case specific Simulink model. Several other models similar to this one has be implemented to produce the outputs in this paper.

8.5 Discussion of model challenges and weaknesses

There are some major and systemic difficulties with the models implemented in this paper. Some simplifications have been done to make the models work properly.

8.5.1 Algebraic loop problems

Problem explanation

Both the PV model and the battery model are based on equations in which the state variables are dependent on themselves. When running a simulation, it will often be stopped because the solver in Simulink does not converge. This problem arises because the variables are dependent on themselves, and it is especially common at the start or other cases where some variable changes very rapidly. One of the battery models that has been implemented, suffered a lot from this problem. The model shown in section 8.3 is not as prone to these problems [34] because the outputs are not as dependent on themselves as in the model discussed in section 6.3.

The single diode equation implementation is very prone to the algebraic loop problems. The current is to a high degree dependent on itself. The PV model often does not converge when one or more of the inputs has a large step change in values.

The battery model is prone to this problem at the start of the simulation.

Modifications to tackle problems

A simple modification can be made to the battery model. A memory block that provides a unit delay decouples the algebraic loop. This block can be seen in figure 8.13. The starting value of this block can be set to the same as the starting conditions of the integrator. This measure has proven to be effective in eradicating convergence problems in the battery model. Introducing a unit time delay in the absolute charge value ("Ah") should not affect the results to any significant degree. This variable changes value very slowly and only influences the voltage infinitesimally by the change that happens in a couple of time units.

The modification done to the PV model is more drastic. A RC-filter is introduced after the MPPT algorithm as shown in figure 8.9b. This is a low pass filter that functions as a small smoothing of the output of the algorithm. This removes any step changes in output, but also affects the voltage amplitude and delay to some degree. The intervention of a memory block that is done in the battery model could not be done here because the whole algorithm is built to shift the voltage by some value and measuring the resulting change in power the following iteration. A unit delay would alter the dynamics completely. The filter retards the voltage to some degree while providing some smoothing.

The effects of the filter can be seen in figure 8.17. It dampens the voltage a bit and delays the output. Lower frequency signals are not affected to the same degree as higher frequency signals. This is apparent in the small difference in the left side of figure 8.17 compared to in the higher frequency signals on the right side.

In effect the RC-filter lowers the voltage step value by a small value and lower frequency trends are not modified as much as higher frequencies. It probably affects the efficiency of the P&O algorithm because of the time delay, but this can

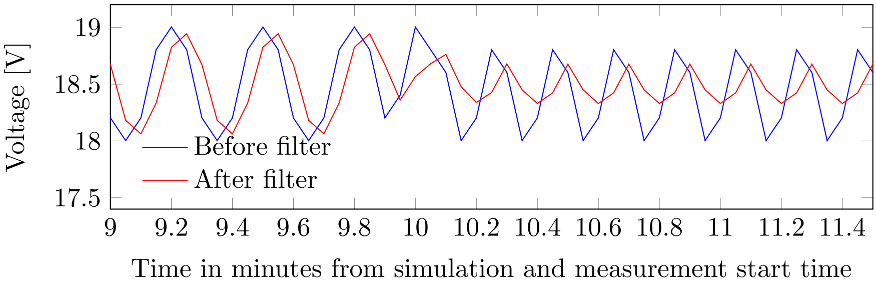


Figure 8.17: The output voltage from the ideal converter before and after RC-filter

not be shown with the current model implementation because it does not converge at all times without the filter.

In a real system, a RC-filter is present because of the power converter. This filter will have a much lower time constant than the one implemented here. The input data is on the minute scale. A practically fast RC-Filter would have almost no effect because the frequencies that is filtered out is much higher than almost all the signals in this model. If the simulation would be on millisecond to second time scale, there would be a physical equivalent RC-filter present.

Introducing a filter is an approximation, and it probably is the greatest weakness of the model.

Part III

Results and discussion

Chapter 9

PV Model accuracy

9.1 Input data

The results in this chapter are simulations using real measured data from the 29th and 4th September 2013. Measured data values are provided by the University of Agder. The figure 9.1 shows the measured irradiance on 29th September 2013 and is used as an example of a variable day. When "variable day" is mentioned in other part of this paper, this is the data that is referenced. The output power measurements from the same day can be seen in figure 9.9 but then together with simulation output resulting from the irradiance data shown in figure 9.1. The power output measurements is from a BP Solar 585F module and the data sheet values used in the simulation is from the same module[1].

PV cell and ambient temperature data is also used in the simulations. The measured cell temperature is displayed in conjunction with the temperature estimation data from the simulation data in figures 9.6 and 9.5. The ambient temperature data is displayed in figure 9.3 and 9.4.

All the input has one data point per minute. The data shown in figure 9.1, and other data from the same day, contains 602 measurements and the data from 4th September consists 734 values.

The simulations are done with 10 to 20 steps for each data point to ensure that the MPPT algorithm is able to reach MPP for each point.

Using battery systems to smooth the output of one module might not be realistic, but this is done in chapter 10 because there is no other PV data available. The data is not scaled so that all the power output curves are using the same input for consistency.

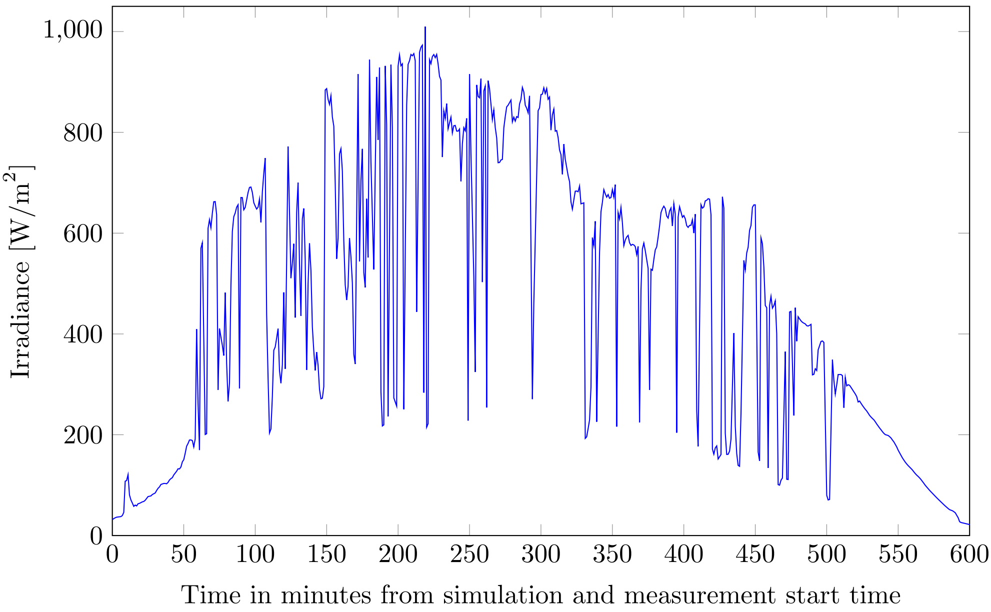


Figure 9.1: Irradiance profile from 29th September 2013. This irradiance profile together with the corresponding power output data is used as an example of a variable irradiance and output power.

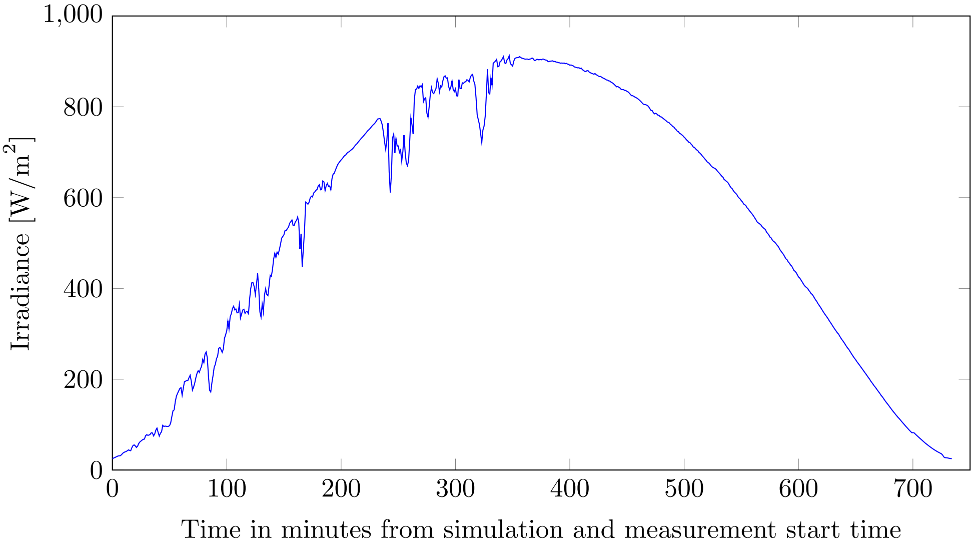


Figure 9.2: Irradiance profile from 4th September 2013. This profile together with the power output data on the same day is used as an example of a smooth day.

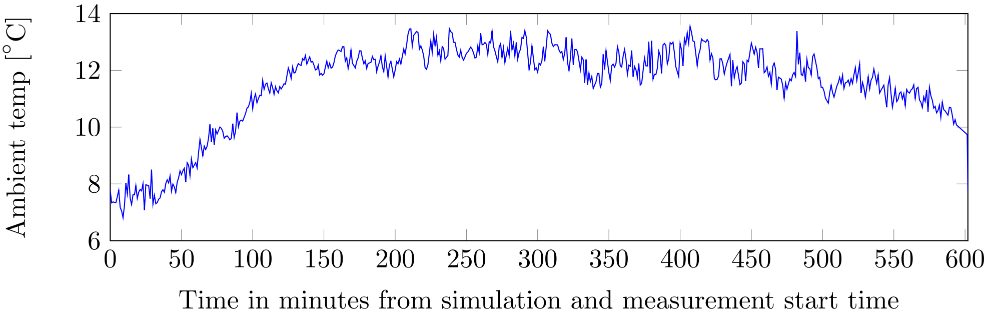


Figure 9.3: Ambient temperature data from 29th September 2013

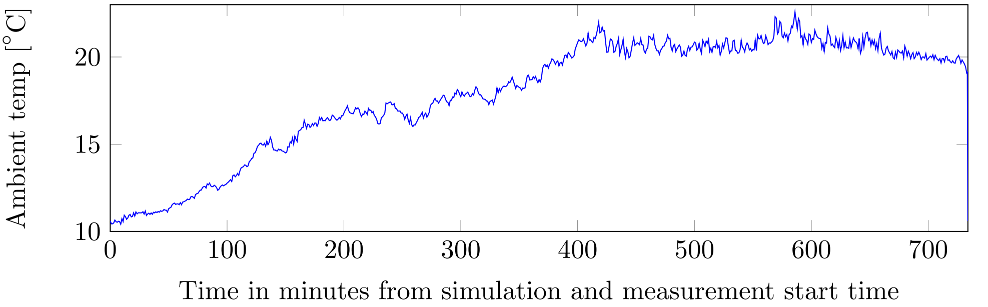


Figure 9.4: Ambient temperature data from 4th September 2013

9.2 Temperature modelling accuracy

The temperature modelling method used in this paper is explained in section 4.2.5. It is a simple way of estimating the cell temperature by adding the irradiance value with a temperature constant (K_t). The simulated temperature is therefore a sum of the ambient temperatures (shown in figures 9.3 and 9.4) and K_t times the irradiance values in figure 9.1 and 9.2.

In figure 9.5, there is significant deviation between the temperature resulting from the simulation and the measured values. When the irradiance is high, the approximation method overestimates the cell temperature consistently by around 5 to 10°C. The approximation is much closer to the measured values in the first half of the day at the same irradiance levels as later in the data set. This is most clearly apparent in figure 9.5 and it might be because of heat retention or perhaps just that it was more windy in the later part of the day. Wind will increase the heat exchange from the panel to the environment, thereby increasing the cooling of the module. In addition to the possible influence from weather, the irradiance can vary even more than is apparent in the data shown here. The data points are separated by one minute each and it is therefore possible that additional fluctuation

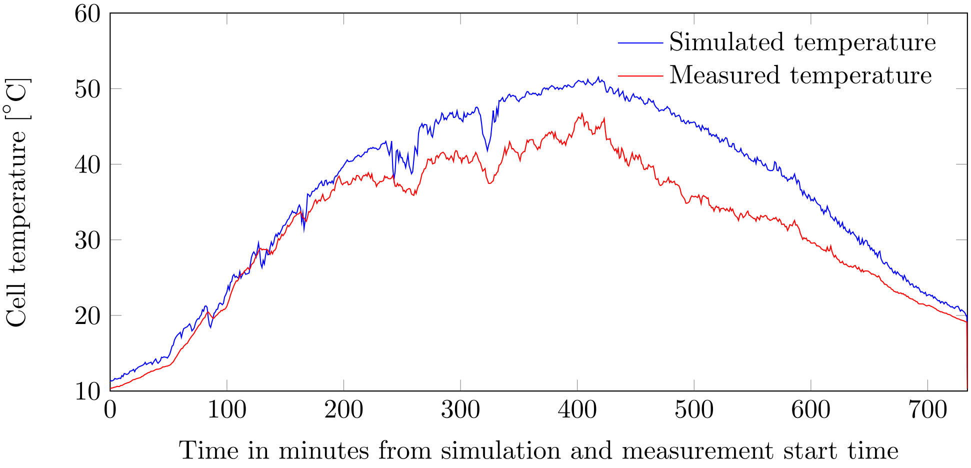


Figure 9.5: Measured cell temperature from 4th September 2013 and the cell temperature resulting from the simulation

is happening without it being apparent in the dataset

The considerable deviation apparent in figure 9.5 can have multiple explanations. The approximation lacks any appreciation of the non-linear nature of heat convection. A linear approximation may be decently accurate in a certain range, but since heat transfer is much more efficient when the temperature is higher, the approximation probably becomes inaccurate at temperatures different from the one that the approximation is based upon.

In figure 9.6, one of the major weaknesses of the approximation is exposed, namely the lack of any kind of heat retention modelling. Notice the rapid changes in temperature caused by the quickly changing input irradiance (figure 9.1). The simulated temperature changes by up to 25°C in one minute (at t∼220) while the actual measured value changes by about 3-5°C in the same time interval. To model the temperature behaviour during instantaneous changes in conditions properly, heat retention must be considered. This is a promising area of improvement and future work on cell temperature modelling.

There is a larger stationary disparity between approximated and measured temperature on the variable day (figure 9.6) than on the smooth day (figure 9.5). This can possibly be caused by more wind or it may be the heat retention making the module slow to escalate the temperature when there is frequent ”breaks” in irradiance.

The temperature in the simulation tracks the real measured temperature crudely,

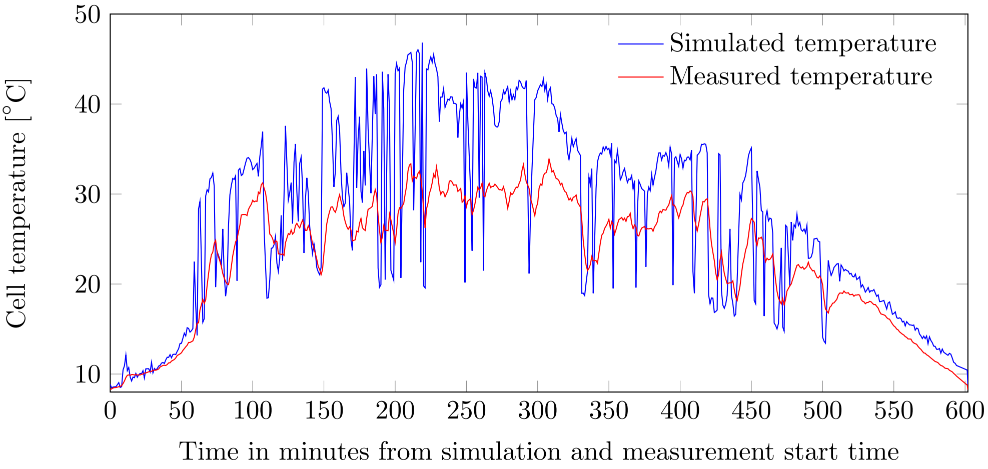


Figure 9.6: Measured cell temperature from 29th September 2013 and the cell temperature resulting from the simulation

but better than if only ambient temperature would be used. When the conditions are irregular, the temperature approximation method implemented here is unreliable.

9.3 Power modelling accuracy

9.3.1 Smooth day

Figure 9.7 shows a comparison of output from the simulation and the measured power. Even with the imperfect cell temperature estimations discussed in section 9.2, the simulated output fits the measured power accurately. The average absolute difference is around 1W. That is an average deviation of about 1,3 % of the highest output value that day.

With some close inspection, it is apparent in figure 9.7 that the disparity between the two curves is higher in the later half of the day. The same is true of the temperature curves shown in figure 9.5.

The larger discrepancy towards the end of the day can be more plainly seen in figure 9.8 which is a plot of the difference between the measured data and simulation output.

In section 9.4 it is confirmed that it is the error in temperature that causes the larger error towards the end of the day. This can be clearly seen from figure 9.14 because the power difference is close to symmetrical from early day to late day once the temperature modelling is taken out of consideration.

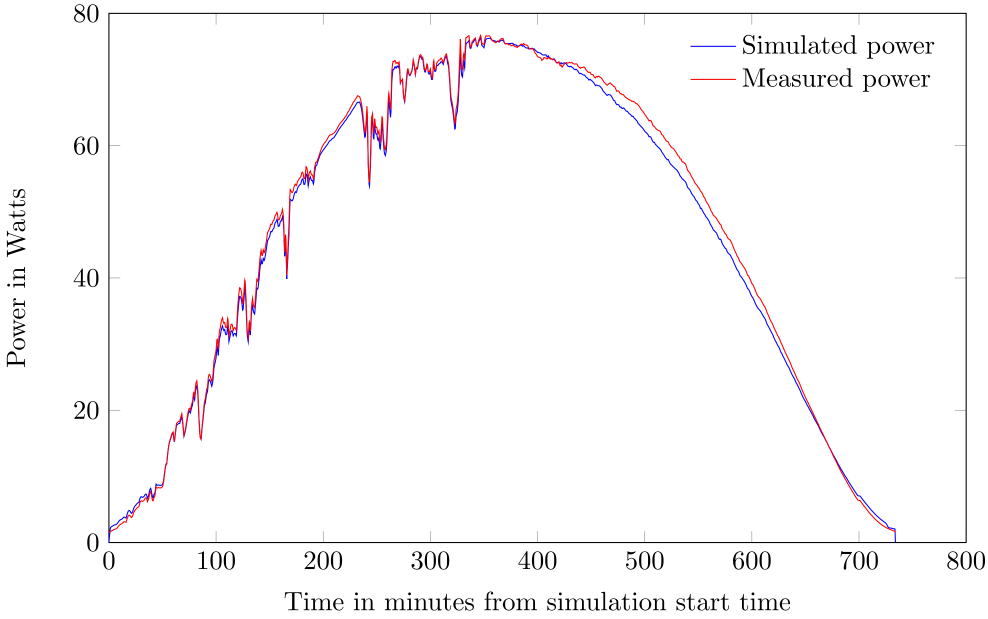


Figure 9.7: Simulated vs. measured power on a smooth day

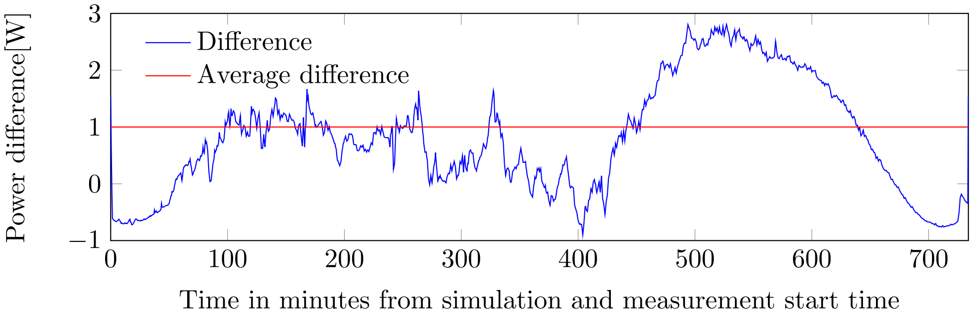


Figure 9.8: Difference between Simulated and measured power on a smooth day. This difference is 1W and is calculated as an absolute value of ($P_{measured} - P_{sim}$. This is the same data as in figure 9.7. The average difference line shown as the red line, while the true average value is 0.80W

9.3.2 Variable day

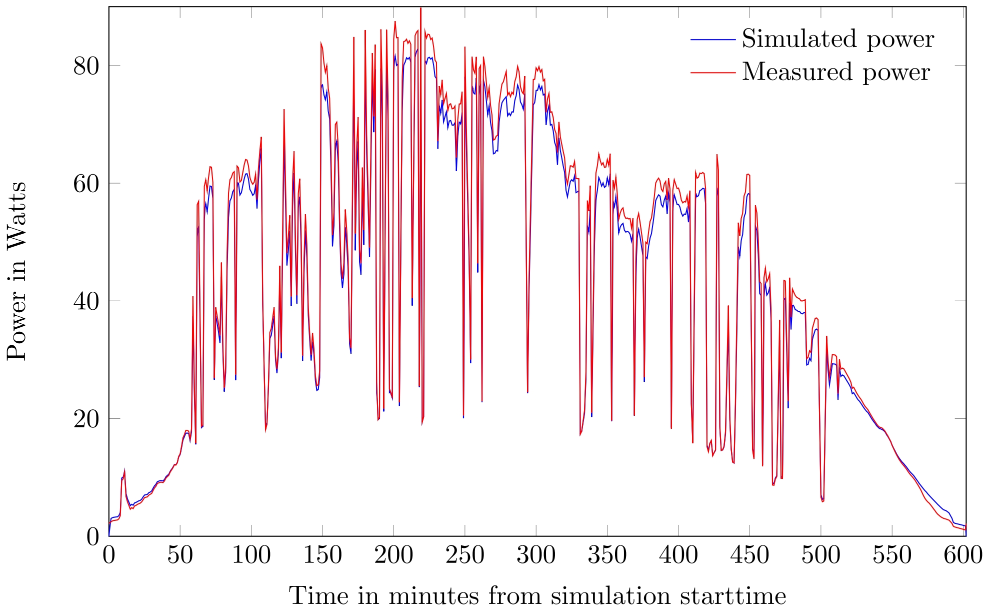


Figure 9.9: Simulated vs. measured power on a variable day. The measured data is from 29th September 2013.

The measured power output and the simulated output is shown in figure 9.9. It is not as easy to see as in the case with a smoother power, but there definitely is more deviation from the measured values in the case of the variable day than in the other. This is more obvious in figure 9.10 that shows the difference between the measured power data and the results from the simulation.

There is a trend of underestimation of power in the simulation that is apparent in figure 9.10. In comparison figure 9.8, there are some very high peaks of difference. This is probably because the maximum power point tracking does not react swiftly enough. Notice that all the peaks apparent in figure 9.10 is in the direction of the measured power being greater than the simulated indicating that the simulation is delayed compared to the input. This is to be expected because of the constraints put on the simulation resolution by simulating an entire day's worth of data in one run. These plots are produced with a simulation resolution of about 20 steps per data point and a display resolution of 2 points per data point from the measured values. As discussed in section 5.5, a realistic amount of perturbations per second for an P&O controlled maximum power point tracker is at least one per second and often 10 or more per second. Some deviation because of insufficient amount of perturbations is to be expected when the conditions are fluctuating a lot like in this case.

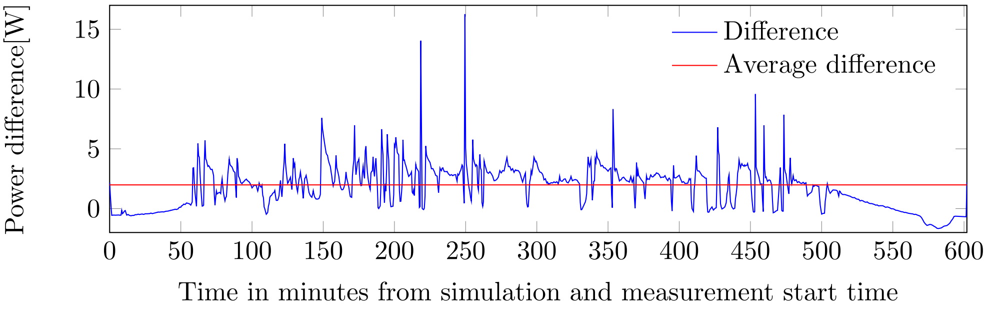


Figure 9.10: Difference between Simulated and measured power on a variable day. The average difference is 1.99W and is calculated as absolute value of $(P_{measured} - P_{sim})$. This is the same data as in figure 9.9. The red line displayed in the figure is the average of the absolute value of the difference while the true average value is 1.76W

There are some deviations that last longer and holds the difference at a close to constant values for some time. These cases are consistent with the with the influence the temperature curve in figure 9.6 should have on the deviation from the measured value. Figure 9.17 confirms this suspicion by removing these "blocks" of deviation. Comparison between figure 9.10 and 9.17 reveals that the approximated temperature deviation shown in figure 9.6 results in a consistently higher discrepancy between simulation output and measured power.

9.4 Simulation with measured cell temperature

The results in this section is produced by running a simulation where the actual measured cell temperature is used instead of the approximation and the output is then compared to measured results and those produce with the "normal" cell temperature approximation. This variation of cell temperature input is enabled by a switch that is implemented in the subsystem calculating the cell temperature (showed in figure 8.8).

In many of the figures in this section, three curves are plotted together. In the legend they are denoted as "Simulated power", "Measured power" and "Simulated Tmeasured". The curves denoted by "Simulated power" is the same data as in the preceding sections, meaning that the data is produced by a simulation with ambient temperature and irradiance as input data. "Measured power" contains the same measured power output values as in previous sections. The new plotted data, "Simulated Tmeasured", is produced by running a simulation with inputs of irradiance and cell temperature, thereby excluding the cell temperature approximation subsystem to investigate the modelling efficiency independent of temperature approximation accuracy.

9.4.1 Smooth day

Figure 9.11 shows the three power output curves from the smooth day data. Different portions of the curves displayed on a smaller time scale are provided in figure 9.12 and 9.13 for clarity.

The temperature does not make a big difference in the average deviation on the smooth day, it is only lowered by 0.1W. In the part with highest power in figure 9.11, the simulated with measured temperature actually deviates considerably more than the regular simulation. This is more clearly displayed in figure 9.12. The approximated temperature in this area is fairly overestimated and probably results in an underestimation of the simulated power. Considering that the simulated power with this inaccurate temperature approximation actually tracks the measured value better in some parts of the day, there must be some other influence that the model does not address adequately.

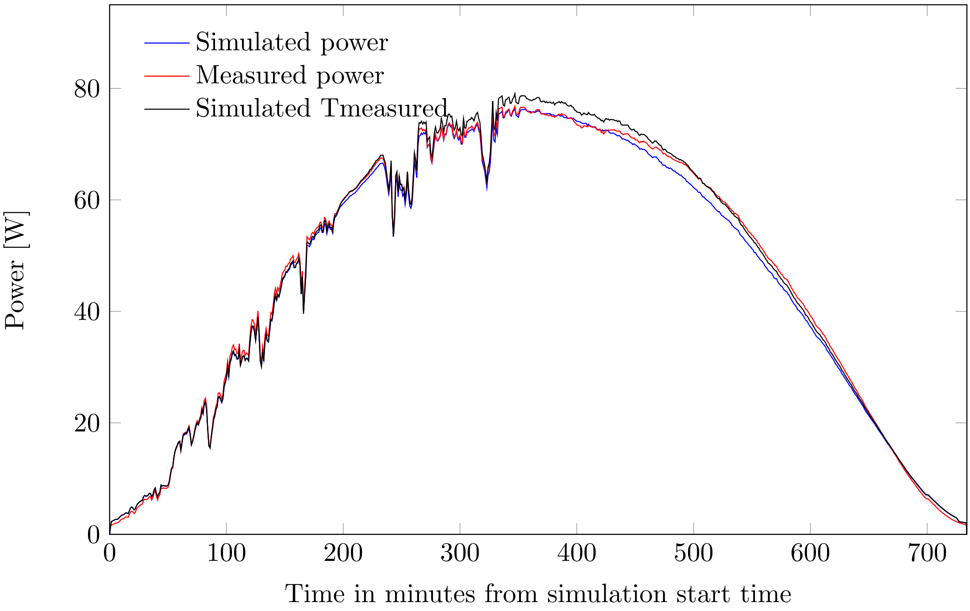


Figure 9.11: Measured output power vs. simulated power vs. simulated power where the cell temperature is the measured cell temperature. The input irradiance is from a smooth day.

Figure 9.13 is a closer look at the first part of the curves in figure 9.11. Both of the measured and simulated curve is consistently higher than the black curve. This is the opposite deviance compared to the one showed in figure 9.12.

The difference between measured power output and the output from the simula-

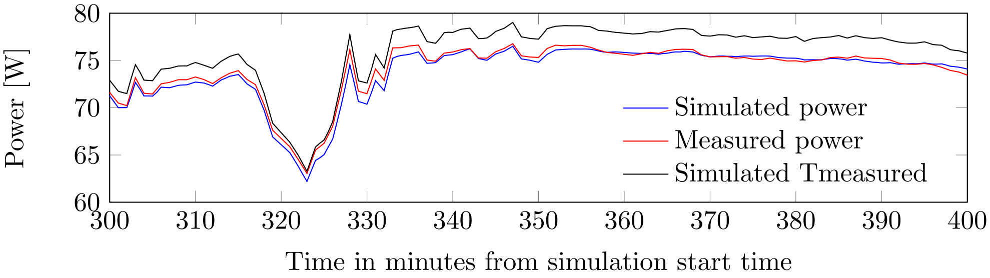


Figure 9.12: Measured output power vs. simulated power vs. simulated power where the cell temperature is the measured cell temperature. This is the same data as in figure 9.14.

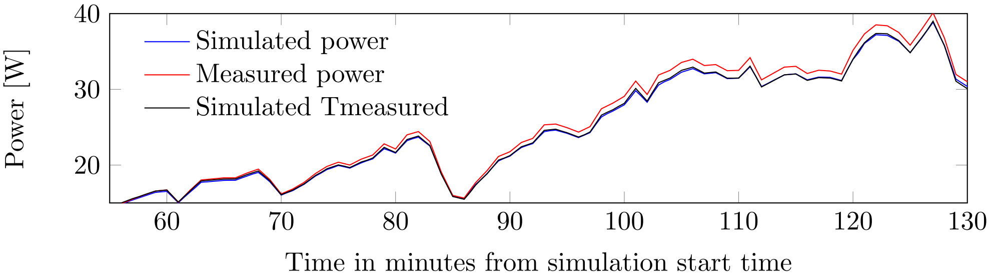


Figure 9.13: Measured output power vs. simulated power vs. simulated power where the cell temperature is the measured cell temperature. This is the same data as in figure 9.14.

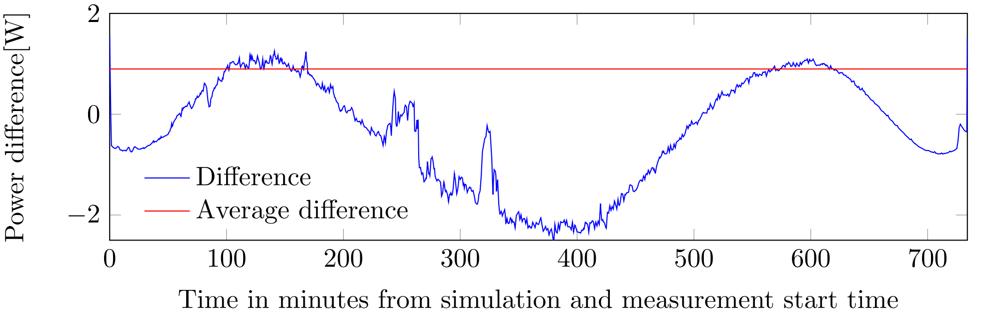


Figure 9.14: Difference between measured power and simulated power with measured cell temperature. The average value is the absolute value of the difference at any time. The actual average is -0.33 W while the absolute average is 0.90 W.

tion using the measured cell temperature is shown in figure 9.14. A clear pattern emerges from this plot compared to the one with the approximated temperature in figure 9.8. The deviation is close to symmetrical from early to late day if the more transient variations are ignored. This clear pattern of difference indicates a systemic inadequacy. In other words, there is something that this way of modelling a PV cell does not take into account, at least not enough to remove the tendency apparent in figure 9.14. This trend can also be seen in figure 9.17, but it is less obvious partly because of the other variation or it may not be present to the same degree as in the smooth case because the pattern may be related to the variation of power, irradiance or temperature.

One of the probable causes of the deviation independent of temperature is that all the parameters in the model are static. None of the parameters are truly static as they are just approximations of the behaviour of a PV module. An example of measured ideality factor variation is shown in figure 4.5.

Another possible explanation is that the modelling of the effect the temperature has on the current and voltage may not be very accurate, but this seems less likely to be the major contributor because then there should be a different pattern in deviance.

9.4.2 Variable day

Figure 9.15 shows three different power output curves on the variable day. The black curve clearly tracks the measured value better than the blue. This trend is easier to see in figure 9.16 which is a closer and more limited view of the same data.

The lower discrepancy between measured data and simulated data with measured cell temperature reveals that the approximation of cell temperature is a major contributor to deviation in figures 9.7 and 9.9.

The difference between measured, and simulated with measured temperature, power output is shown in figure 9.17 and 9.18. The large peaks in deviance is about the same as in figure 9.10, but the average discrepancy is much lower. It is in fact as low as in the smooth case in figure 9.7 with around 1W on average. On the basis of this data, it can be concluded that the large temperature deviancy apparent in figure 9.6 causes an additional departure from measured values of roughly 0.8W on average.

The high amplitude short duration peaks are not changed noticeably by using measured cell temperature. This is to be expected, as the presumed cause of these discrepancies is the inability of the power point tracking to react quickly enough.

As discussed in section 9.4.1, there appears to be a pattern in the simulation accuracy which is recognizable even in the very variable results shown in figure 9.18

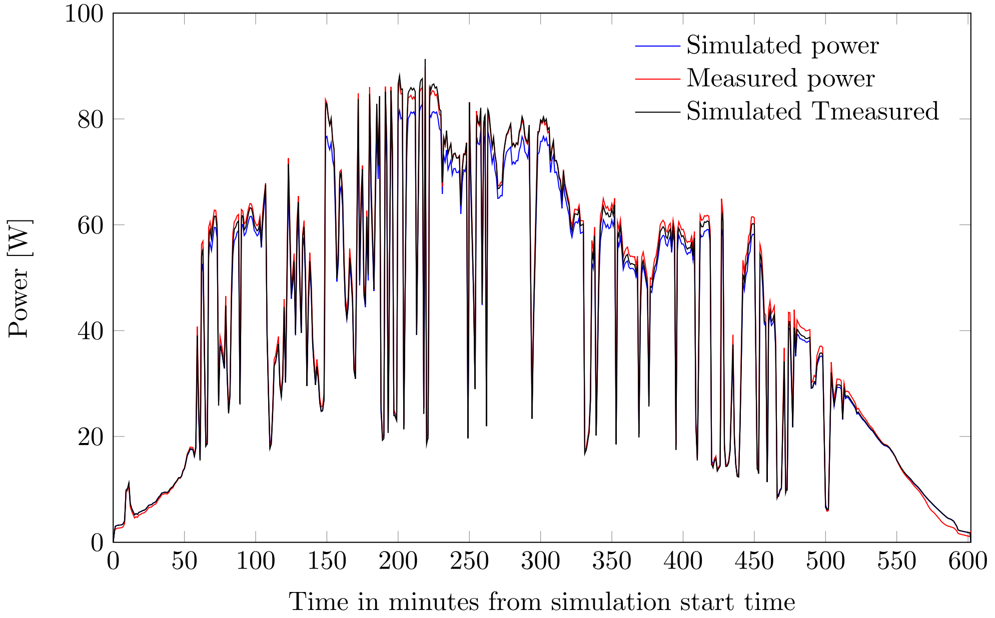


Figure 9.15: Measured output power vs. simulated power vs. simulated power where the cell temperature is the measured cell temperature. The input irradiance is from a variable day.

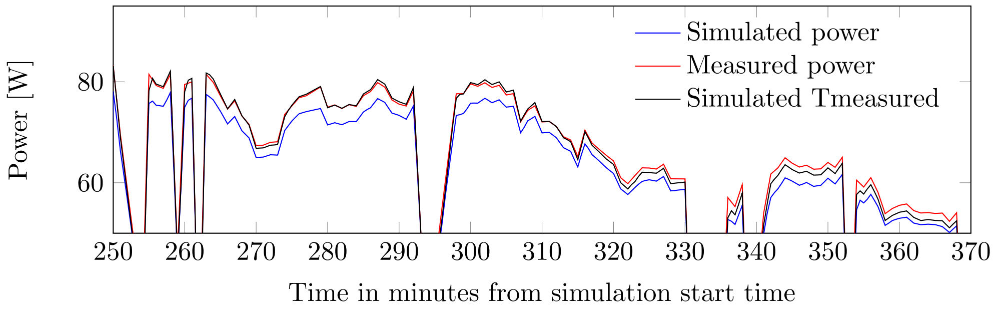


Figure 9.16: Measured output power vs. simulated power vs. simulated power where the cell temperature is the measured cell temperature. The input irradiance is from a variable day.

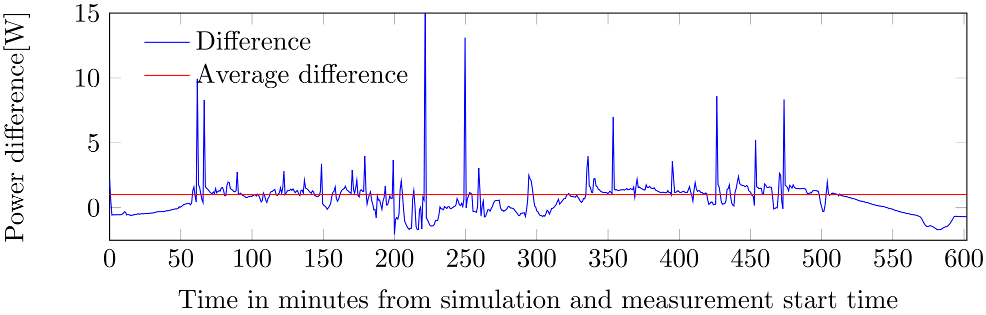


Figure 9.17: Difference between measured power and simulated power with measured cell temperature. The average value is the absolute value of the difference at any time. The actual average is 0.65 W while the absolute average value is 1.00 W (red line in figure)

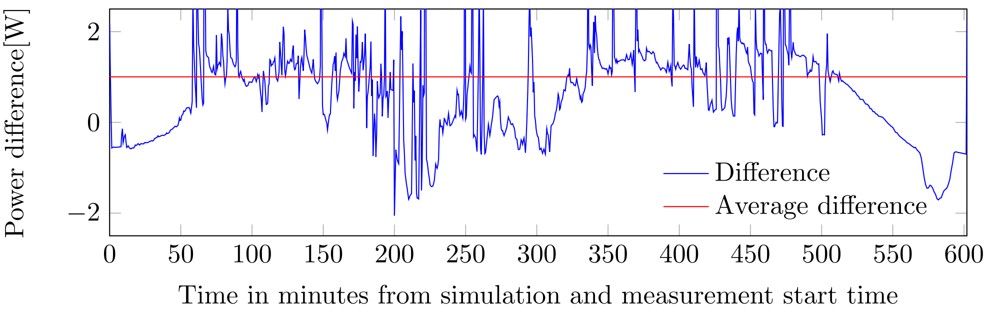


Figure 9.18: Difference between measured power and simulated power with measured cell temperature. This is the same data as in figure 9.17

Chapter 10

Power smoothing results

In this chapter, the resulting output curves from three different battery configurations that provides power smoothing is presented.

The battery systems are denoted by Wh and Ah in the figure legends. The Wh-rating is calculated by multiplying the total ampere hour capacity with the nominal voltage. Three different module configurations are used to illustrate how various battery modules can perform. The maximum power capabilities of the batteries are considered in the model simulation. These limits on the batteries are discussed in section 6.6 where the battery controller and constraints are discussed. Within this framework, the absolute maximum power of the battery modules here are 25.2W, 67.2W and 100.8W respectively. These values are the multiplication of maximum discharge current and maximum battery voltage and they are therefore the absolute highest power outputs that the batteries can produce.

The battery module configurations are constructed automatically by the script in appendix A.6 that takes in constraint of the battery cell, desired power output and energy capacity and configures a module built up by the fundamental battery cells. In the Simulink model it is only necessary to change two variables to make a different battery configuration. These variables are changed by a script wherein a for-loop is running the simulation model to produce the plotted curves in this chapter.

All the power smoothing figures up until section 10.3 are produced with averaging windows of 40 minutes with the double averaging algorithm.

10.1 Smooth day

Figure 10.1 shows the sum of battery power and PV Power with three different battery configurations. All the modules start at the same charge relative to their own capacity at 50% state of charge. The power reference provided by the double averaging algorithm is so close to the power of the largest battery module that they are indistinguishable on the time scale of these figures.

The biggest battery follows the reference close to perfectly, while the medium battery fills up to much during the mid-day to absorb additional energy and the smallest battery quickly becomes ”fully charged” in the start of the day and fully discharged in the later part of the day. ”Fully charged” or discharged is here used for describing an SOC of 90 and 10% respectively.

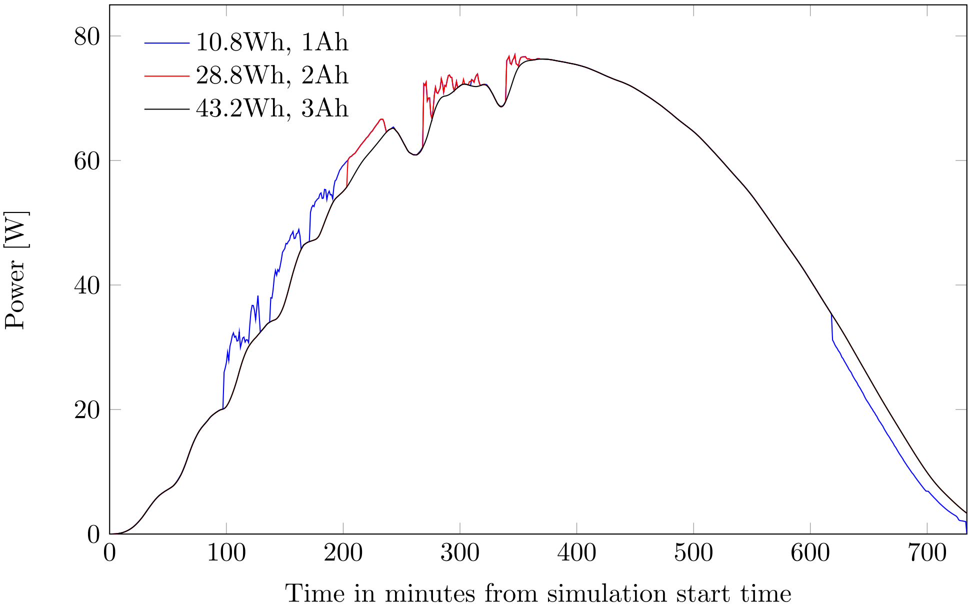


Figure 10.1: Smoothed power resulting from three different battery modules. The largest module is the only one that is able to follow the reference. The resulting output isn’t very variable even though the smaller batteries are not able to provide good smoothing. The PV power is not displayed because it would mask some of the variations in the curves. In figure 10.3, the PV power is displayed together with the curve from the largest battery.

The state of charge curves corresponding to the power curves in figure 10.1 can be seen in figure 10.2. It is clear that smaller batteries reaches the upper charge limit and therefore cannot absorb additional energy early in the day. This problem could be improved by starting at a lower charge value or by ensuring that the power reference tracks the PV power better by reducing the averaging window. Making the reference closer to the PV power will also increase the variation in power, but it can improve the output if it leads to the batteries to not reach any limits.

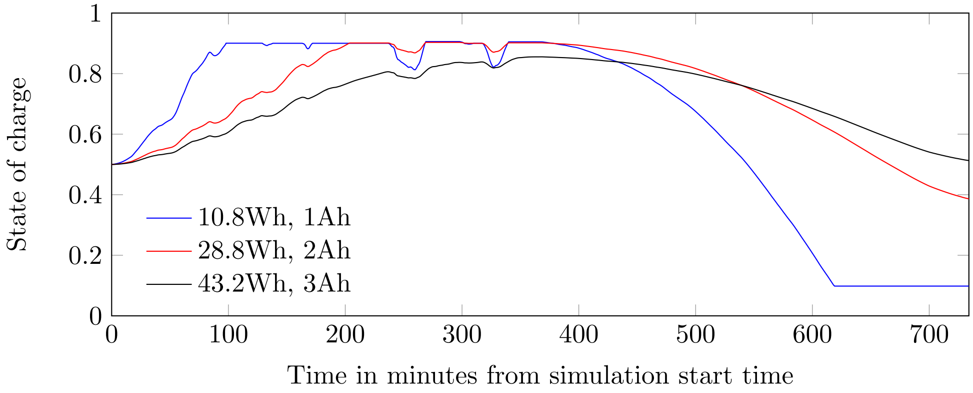


Figure 10.2: State of charge curves for the three battery modules providing smoothing in figure 10.1. Because of the delay in power reference with respect to PV power, the state of charge rises the first half the day and decreases on the second half.

The main reason for the pattern of SOC apparent in figure 10.2, namely rising SOC in the first half of the day and dropping charge in the later half, is that the power reference lags the PV power. This leads to the reference being lower than the PV power when the power is rising and higher when it declining. Reducing the averaging window reduces the discrepancy between the two, but it also reduces the smoothing. The averaging window should be different for different battery modules. An example of how this can utilize smaller batteries is shown in figure 10.7 and 10.8.

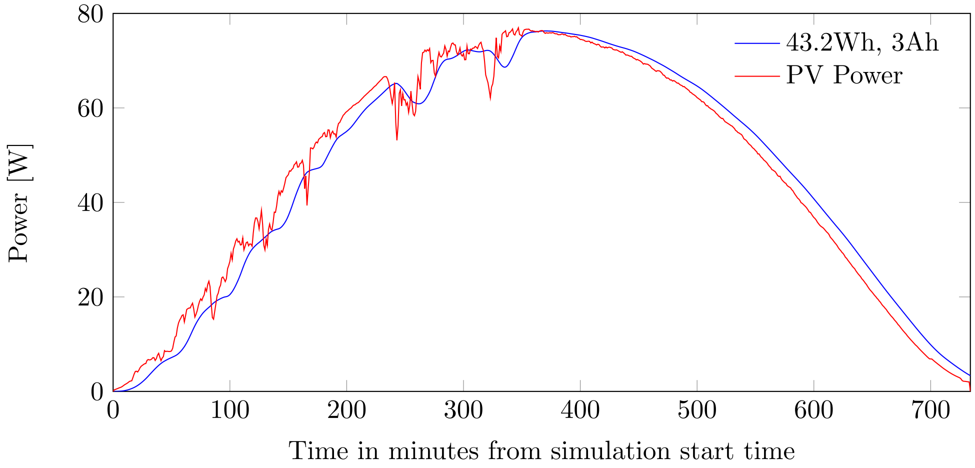


Figure 10.3: Smoothed power resulting from the largest battery module compared to the PV power. There is a clear smoothing effect as well as a visible delay in output.

10.2 Variable day

Smoothing the output power on the variable day (data from 29th September 2013) is a much bigger challenge than on the smooth day. Perhaps the most important change between the two scenarios is the power requirement. On the smooth day, the highest peak in the battery power reference is about 12 W. On the variable day however, the highest peak is around 55 W. This high power requirement means that the smaller battery modules will probably not be able to follow the reference because of power limitations. The smallest battery has a maximum power capability of 25W. This is apparent in figure 10.4 when considering that is only during the time around 50-100 minutes that the battery is limited by the controller because of high SOC.

The figure 10.6 shows the SOC variations during the day in the three batteries. In this case, the smallest battery does not consistently hover at 90% SOC like on the smooth day (figure 10.2). The variable power causes the battery to discharge more often instead of the steadier SOC in the smoother case. It is also apparent that the medium sized battery hits the upper limit of SOC a couple of times and the effect of this can be seen in the figure 10.4 where the red curve suddenly jumps to the actual PV power value at several occasions. Notice that the blue curve in

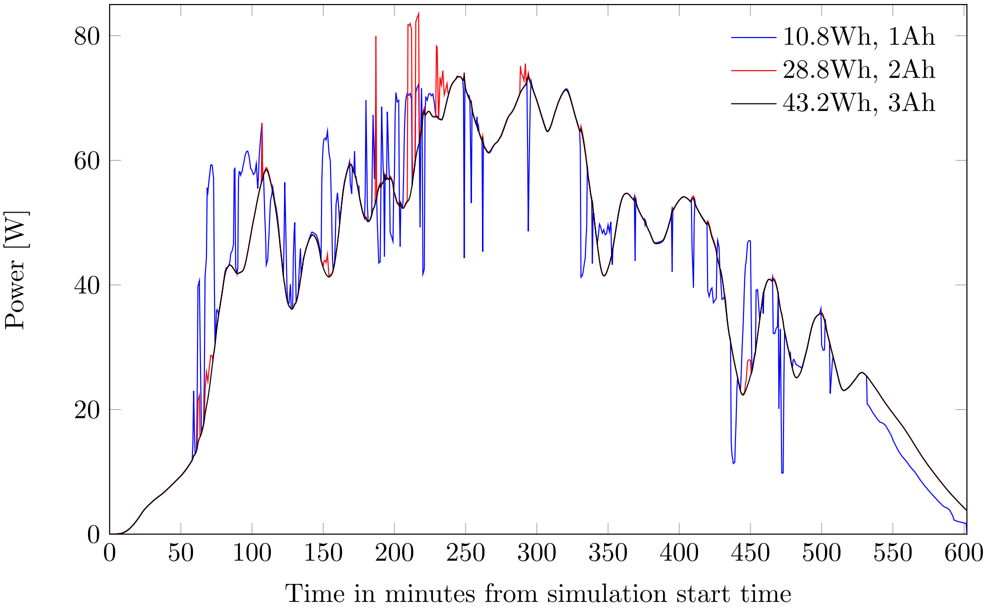


Figure 10.4: Resulting power output from the combination of PV power and battery power with three different battery module sizes. The PV power is not displayed because it would mask parts of the other curves. The measured power is shown together with smoothed power resulting from the largest battery in figure 10.5.

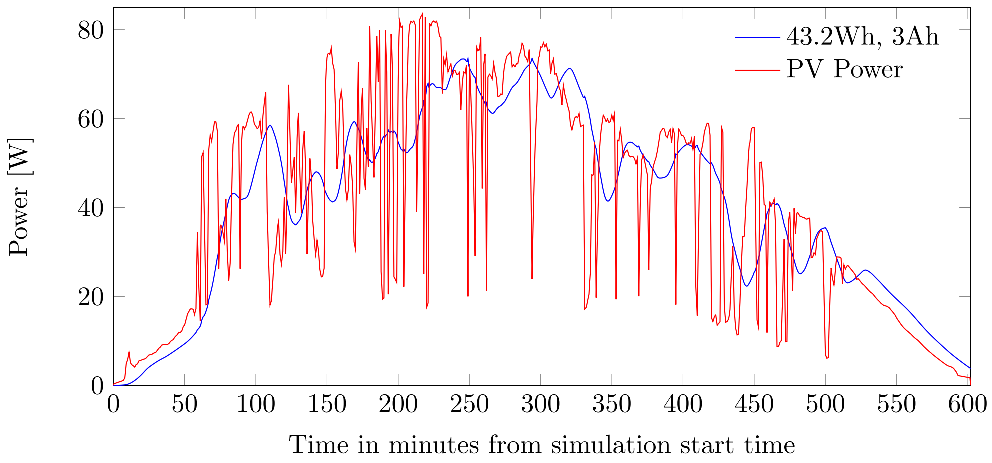


Figure 10.5: Resulting power output from the largest battery module compared to the PV power.

the same instances is actually closer to the reference because the smallest battery is able to provide smoothing since it is not near the limits of SOC at the same time.

The largest battery is able to follow the reference all day and narrowly avoids reaching the state of charge limits. In figure 10.5 shows that the battery even out the output significantly and that the power smoothing algorithm provides a decent reference as long as the battery is able to deliver what is required. To provide a better reference to the smaller batteries, the averaging windows can be lowered. This is discussed in section 10.3.

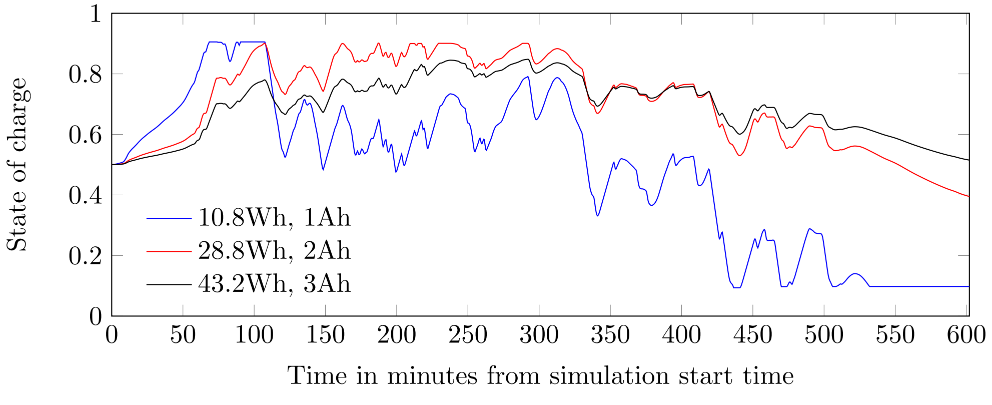


Figure 10.6: State of charge curves for the same case as in figure 10.4. The smallest module changes SOC quickly and hits the limit of 90% SOC quickly.

10.3 Power smoothing with different window sizes

In figure 10.7 there is shown smoothed output with different battery modules with different reference power. The smallest battery module operates with a reference power produced by 10 minutes averaging window by the double averaging algorithm. This improves the utilization of the battery greatly compared to the case shown in figure 10.1.

It is evident that the adjustment done to the reference power utilizes the smaller batteries better and the resulting output is smoother because there is no area of the curve where the batteries are unable to provide smoothing. Figure 10.8 shows how the state of charge varies for each battery in the case shown in figure 10.7. The SOC curves are well aligned, indicating that the relative capacity demand on the batteries is the same. It can be concluded that varying averaging windows can be used to modify power and capacity requirements while reducing smoothing moderately.

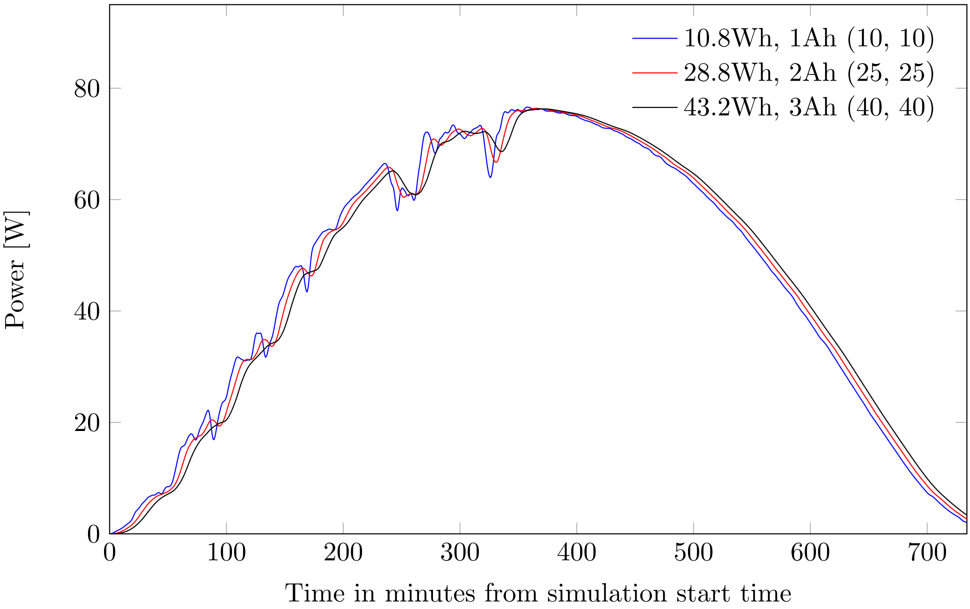


Figure 10.7: Curves of the sum of battery output power and PV power. The legend notation is: "Wh-rating", "Ah-rating", ("Window 1 in minutes", "Window 2 in minutes") where "Window" is referring to the window size of the averaging algorithm. The PV power output is not shown here because it would mask some of the curves and the curve resulting from the largest battery can be seen compared against the measured power in figure 10.3.

If the blue curve of power output in figure 10.7 is sufficiently smooth, that av-

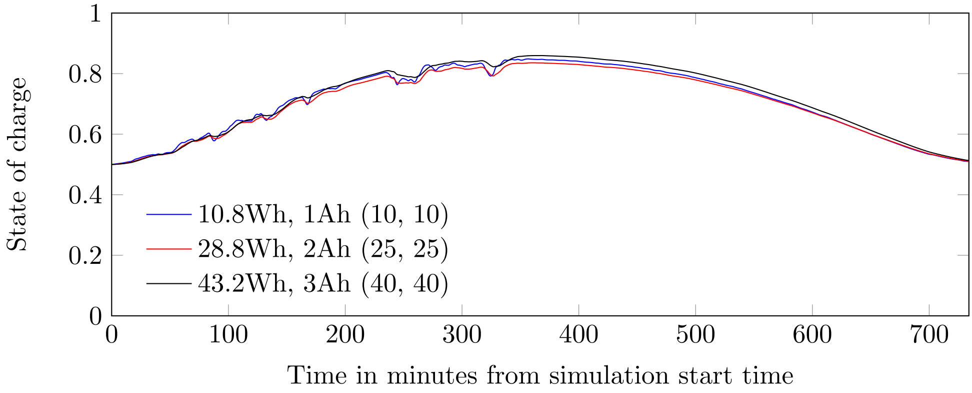


Figure 10.8: Curves of SOC for the three battery modules. The legend notation is: "Wh-rating", "Ah-rating", ("Window 1 in minutes", "Window 2 in minutes") where "Window" is referring to the window size of the averaging algorithm.

eraging window can be used irrespective of a larger battery capacity. This will free up some of the battery capacity of larger batteries for other purposes.

Chapter 11

Conclusions

A Simulink implementation based on the five parameter diode PV model has been constructed. The effects of varying model parameter values and the effect of different solar light intensity and ambient temperature has been demonstrated.

All the parameters in the PV model can be estimated automatically by the parameter approximation method outlined. This method is implemented in such a way that it enables easy simulation of different solar modules just by supplying data sheet parameter values to the script. The PV model can be automatically modified to simulate configurations ranging from a single PV cell to a string of modules.

The model implementation suffers from some instability because of algebraic loops. An RC-filter has been added to deter this problem. This is an approximation that probably affects the output to some degree.

Maximum power point tracking using the perturb and observe algorithm has been implemented. Some weaknesses with this tracking method has been exposed, and these include wrong reactions to changing inputs, slow voltage reaction, oscillation around MPP and a limited amount of perturbations per data point in these simulations. The maximum power point tracking implementations successfully keeps the voltage close to the MPP most of the time.

Modelling of battery cells is done by curve fitting from manufacturer's discharge curves. The implemented model utilizes a state of charge dependent controlled voltage source and an internal resistance. As with the PV model, it can be automatically modified to different configurations from a single cell to a module with several strings in parallel. The major weaknesses are that there is no modelling of current dependent capacity and no temperature dependence.

A double averaging algorithm has been implemented to create the power reference to the battery system. This method of power smoothing enables using different window sizes for averaging so that various battery sizes and smoothing requirements can be met.

The power output from the PV model has been compared to real measurements to evaluate the accuracy. Approximation of the cell temperature in the simulation is based on the light intensity and the ambient cell temperature. It is clear that the temperature approximation is not very accurate compared to the actual measured data and the difference is highest on the day with variable conditions. Even with this poor temperature modelling, the simulation matches the measured output to within a few percent's accuracy most of the day.

Using simulations with the measured cell temperature values, it is demonstrated that the crude approximation of cell temperature is one of the major contributors to deviance from measured values. These simulations are very accurate with a discrepancy of as low as 1W on average even for the very variable day case. A clear pattern of deviance when the cell temperature is no longer a factor is demonstrated. This indicates a systemic inaccuracy in the model. The most probable explanation is that it arises as an effect of having static parameters instead of them varying with conditions. The deviance is low enough to justify using static parameters, at least as long as other factors like cell temperature contribute as much to inaccuracy.

The efficiency of the power smoothing algorithms is demonstrated in simulations with different battery sizes. Smaller batteries are not able to follow the power reference for as long as larger batteries. The power output is significantly smoother in cases where the batteries are sufficiently large. It is demonstrated that changing the averaging windows of the algorithms reduces the requirements of the batteries while decreasing the smoothing somewhat. The main weakness of the method of power smoothing implemented is that the output is delayed compared to the PV power. This increases the needed capacity of the batteries.

The constructed model of photovoltaic modules is a flexible tool for simulation of different PV module types. It has been demonstrated to be very accurate compared to measured data when the only input other than the data sheet values is a good approximation of the ideality factor. The deviance in output power is not affected much by the choice of ideality factor. As long as there is sufficient data available to approximate the parameters, this model should give a decently accurate output power for many PV module types.

References

[1] BP Solar. Bp 585f data sheet,http://www.posharp.com/Businesses/daacacf6-2fa0-4fe2-b6f5-30211a36989a/Panel/BP585F.pdf.

[2] Intergovernmental Panel On Climate Change. Climate change 2007: The physical science basis. *Agenda*, 6(07):333, 2007.

[3] The international energy agency (iea). Technology roadmap: Solar photovoltaic energy,http://www.iea.org/publications/freepublications/publication/pv_roadmap.pdf.

[4] T.D. Hund, S. Gonzalez, and K. Barrett. Grid-tied pv system energy smoothing. In *Photovoltaic Specialists Conference (PVSC), 2010 35th IEEE*, pages 002762–002766, June 2010.

[5] Håvard Breisnes Vika. Literature study of energy storage technologies - with a simulation case of smes and battery energy storage,tet 4520 - specialization project, ntnu, december 2013.

[6] A. Joseph and M. Shahidehpour. Battery storage systems in electric power systems. pages 8 pp.–, 2006.

[7] K.C. Divya and Jacob Østergaard. Battery energy storage technology for power systems—an overview. *Electric Power Systems Research*, 79(4):511 – 520, 2009.

[8] J.R. Pillai and B. Bak-Jensen. Integration of vehicle-to-grid in the western danish power system. *Sustainable Energy, IEEE Transactions on*, 2(1):12–19, 2011.

[9] H. Ibrahim, A. Ilinca, and J. Perron. Energy storage systems—characteristics and comparisons. *Renewable and Sustainable Energy Reviews*, 12(5):1221 – 1250, 2008.

[10] Peter Van den Bossche, Frédéric Vergels, Joeri Van Mierlo, Julien Matheys, and Wout Van Autenboer. Subat: An assessment of sustainable battery technology. *Journal of Power Sources*, 162(2):913 – 919, 2006. Special issue including selected papers from the International Power Sources Symposium 2005 together with regular papers.

[11] C.A. Hill, M.C. Such, Dongmei Chen, J. Gonzalez, and W.M. Grady. Battery energy storage for enabling integration of distributed solar power generation. *Smart Grid, IEEE Transactions on*, 3(2):850–857, June 2012.

[12] Guishi Wang, M. Ciobotaru, and V.G. Agelidis. Integration of vanadium redox battery with pv systems: Modeling and operational characteristics. In *Industrial Electronics (ISIE), 2012 IEEE International Symposium on*, pages 1598–1603, May 2012.

[13] M.G. Villalva, J.R. Gazoli, and E.R. Filho. Comprehensive approach to modeling and simulation of photovoltaic arrays. *Power Electronics, IEEE Transactions on*, 24(5):1198–1208, May 2009.

[14] T.T. Yetayew and T.R. Jyothsna. Improved single-diode modeling approach for photovoltaic modules using data sheet. In *India Conference (INDICON), 2013 Annual IEEE*, pages 1–6, Dec 2013.

[15] N.M. Abd Alrahim Shannan, N.Z. Yahaya, and B. Singh. Single-diode model and two-diode model of pv modules: A comparison. In *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on*, pages 210–214, Nov 2013.

[16] Tarak Salmi, Mounir Bouzguenda, Adel Gastli, and Ahmed Masmoudi. Matlab/simulink based modeling of photovoltaic cell. *International Journal of Renewable Energy Research (IJRER)*, 2(2):213–218, 2012.

[17] N Pandiarajan and Ranganath Muthu. Mathematical modeling of photovoltaic module with simulink. In *IEEE Ist International Conference on Electrical energy Systems*, pages 257–263, 2011.

[18] V. Sangsawang and S. Chaitusaney. Modeling of photovoltaic module from commercial specification in datasheet. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference on*, pages 1–4, May 2012.

[19] Georgi Hristov Yordanov. *Characterization and Analysis of Photovoltaic Modules and the Solar Resource Based on In-Situ Measurements in Southern Norway.* PhD thesis, Norwegian University of Science and Technology, Department of Electrical Power Engineering, 2012.

[20] G.H. Yordanov, O.-M. Midtgard, and T.O. Saetre. Pv modules with variable ideality factors. In *Photovoltaic Specialists Conference (PVSC), 2012 38th IEEE*, pages 002362–002367, June 2012.

[21] Ole-Morten Midtgard, Tor Oskar Sætre, Georgi Yordanov, Anne Gerd Imenes, and Chee Lim Nge. A qualitative examination of performance and energy yield of photovoltaic modules in southern norway. *Renewable Energy*, 35(6):1266 – 1274, 2010.

[22] Chih-Hao Chang, Jia-Jun Zhu, and Huan-Liang Tsai. Model-based performance diagnosis for pv systems. In *SICE Annual Conference 2010, Proceedings of*, pages 2139–2145, Aug 2010.

[23] Rec peak energy series data sheet,`http://www.recgroup.com/en/Downloads/Product-downloads/?feed=Product%2bdatasheets`.

[24] Mei Shan Ngan and Chee Wei Tan. A study of maximum power point tracking algorithms for stand-alone photovoltaic systems. In *Applied Power Electronics Colloquium (IAPEC), 2011 IEEE*, pages 22–27, April 2011.

[25] V. Salas, E. Olías, A. Barrado, and A. Lázaro. Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems. *Solar Energy Materials and Solar Cells*, 90(11):1555 – 1578, 2006.

[26] T. Esram and P.L. Chapman. Comparison of photovoltaic array maximum power point tracking techniques. *Energy Conversion, IEEE Transactions on*, 22(2):439–449, June 2007.

[27] Nevzat Onat. Recent developments in maximum power point tracking technologies for photovoltaic systems. *International Journal of Photoenergy*, 2010, 2010.

[28] N.S. D'Souza, L. A C Lopes, and X. Liu. An intelligent maximum power point tracker using peak current control. In *Power Electronics Specialists Conference, 2005. PESC '05. IEEE 36th*, pages 172–, June 2005.

[29] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli. Optimization of perturb and observe maximum power point tracking method. *Power Electronics, IEEE Transactions on*, 20(4):963–973, July 2005.

[30] A. Al-Diab and C. Sourkounis. Variable step size p&o mppt algorithm for pv systems. In *Optimization of Electrical and Electronic Equipment (OPTIM), 2010 12th International Conference on*, pages 1097–1102, May 2010.

[31] Fangrui Liu, Shanxu Duan, Fei Liu, Bangyin Liu, and Yong Kang. A variable step size inc mppt method for pv systems. *Industrial Electronics, IEEE Transactions on*, 55(7):2622–2628, July 2008.

[32] Mohammed A Elgendy, Bashar Zahawi, and David J Atkinson. Assessment of perturb and observe mppt algorithm implementation techniques for pv pumping applications. *Sustainable Energy, IEEE Transactions on*, 3(1):21–33, 2012.

[33] H.L. Chan. A new battery model for use with battery energy storage systems and electric vehicles power systems. 1:470–475 vol.1, 2000.

[34] O. Tremblay, L-A Dessaint, and A.-I. Dekkiche. A generic battery model for the dynamic simulation of hybrid electric vehicles. In *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*, pages 284–289, Sept 2007.

[35] Panasonic. Lithium ion batteries handbook `http://industrial.panasonic.com/www-data/pdf/ACI4000/ACI4000PE5.pdf`.

[36] Feng Cheng, S. Willard, J. Hawkins, B. Arellano, O. Lavrova, and A. Mammoli. Applying battery energy storage to enhance the benefits of photovoltaics. In *Energytech, 2012 IEEE*, pages 1–5, May 2012.

[37] O. Lavrova, F. Cheng, S. Abdollahy, H. Barsun, A. Mammoli, D. Dreisigmayer, S. Willard, B. Arellano, and C. Van Zeyl. Analysis of battery storage utilization for load shifting and peak smoothing on a distribution feeder in new mexico. In *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, pages 1–6, Jan 2012.

[38] Manoj Datta, Tomonobu Senjyu, Atsushi Yona, and Toshihisa Funabashi. Photovoltaic output power fluctuations smoothing by selecting optimal capacity of battery for a photovoltaic-diesel hybrid system. *Electric Power Components and Systems*, 39(7):621–644, 2011.

Appendix A

Matlab Scripts

A.1 Script for approximating five parameters from data sheet

```
function [Rp,Rs,Io,Iph]=ApproximateModelParameters(Voc,Vmpp,Impp,Isc,
    Ncells,a)

%% Constants
k=1.3806e-23;     % Boltzman constant (J/oK)
q=1.602e-19;      % Elementary charge on an electron (oC)
Tref=273.15+25;
Eg=1.12*q;

%% Calculations
Vth = k*Tref/q;         % Thermal voltage

Io=Isc/(exp(Voc/(a*Ncells*Vth))-1);

%% Initialization
%Start conditions and increment value adjusted from previous run for best
%result
Rs_start=0.15;
Rs_step=0.001;
Rp_start=10;
Rp_step=1;
Iph_start=Isc;
Iph_step=0.001;
Min=inf;

%% Iteration and minimization of error
for i=1:1000
    for j=1:1000
        for k=1:100
               Rs=Rs_start+Rs_step*i;
               Rp=Rp_start+Rp_step*j;
               Iph=Iph_start+Iph_step*k;
               Error1=(Vmpp/Impp)-((a*Ncells*Vth*Rp)/((Io*Rp*exp((Vmpp+(
                   Impp*Rs))/(a*Ncells*Vth)))+(a*Ncells*Vth)))-Rs;
               Error2=(Vmpp+(Impp*Rs))/(Iph-Impp-Io*(exp((Vmpp+(Impp*Rs))
                   /(a*Ncells*Vth))-1))-Rp;
               Error3=(((Rp+Rs)/Rp)*Isc)-Iph;
               TotalError=Error1^2+Error2^2+Error3^2;
               if TotalError<Min
                   E1=Error1;
                   E2=Error2;
                   E3=Error3;
                   Min=TotalError;
                   Rsbest=Rs;
                   Rpbest=Rp;
                   Iphbest=Iph;
                   I=i;
                   J=j;
                   K=k;
               end
        end
    end
end

%% Output Values
Rp=Rpbest;
Rs=Rsbest;
Iph=Iphbest;
```

A.2 Script for producing plots with different variable values

```
%% This script produces plots demonstrating the effects of some of
%the variables in the model

% Ideality factor
clear
Variables
for i=1:5
    a=0.8+(0.2*i);
    simOut=sim('VariablesEffects');
    current(i,:)=PVcurrent2.signals.values;
    voltage(i,:)=PVVoltage2.signals.values;
    Power(i,:)=PVPower.signals.values;
end

f1 = figure(1);
plot(voltage(1,:),current(1,:),'b-',voltage(2,:),current(2,:),'m-',voltage
    (3,:),current(3,:),'k-',voltage(4,:),current(4,:),'c-',voltage(5,:),
    current(5,:),'r-','LineWidth',1);
xlabel('Voltage [V]')
ylabel('Current [A]')
legend('a = 1.0','a = 1.2','a = 1.4','a = 1.6','a = 1.8','Location','
    SouthWest')
legend BOXOFF
saveas(f1, 'IdealityfactorDemonstration.png');
```

```
matlab2tikz('IdealityfactorDemonstration.tex');% Converts the plot to '
    IdealityfactorDemonstration.tex'
matlab2tikz('IdealityfactorDemonstration.tikz', 'height', '\figureheight',
     'width', '\figurewidth');

% Rs - Series resistance
clear
Variables
for i=1:6
    Rs=2^(-5+i)
    simOut=sim('VariablesEffects');
    current(i,:)=PVcurrent2.signals.values;
    voltage(i,:)=PVVoltage2.signals.values;
    Power(i,:)=PVPower.signals.values;
end

f2 = figure(2);
grid on
plot(voltage(1,:),current(1,:),'b-',voltage(2,:),current(2,:),'m-',voltage
    (3,:),current(3,:),'k-',voltage(4,:),current(4,:),'c-',voltage(5,:),
    current(5,:),'r-',voltage(6,:),current(6,:),'b:');
xlabel('Voltage [V]')
ylabel('Current [A]')
legend('R_s= 0.063 \Omega','R_s= 0.125 \Omega','R_s= 0.250 \Omega','R_s=
    0.500 \Omega','R_s= 1.000 \Omega','R_s= 2.000 \Omega','Location','
    SouthWest')
legend BOXOFF
saveas(f2, 'RsDemonstration.png');
matlab2tikz('RsDemonstration.tex');
matlab2tikz('RsDemonstration.tikz', 'height', '\figureheight', 'width', '\
    figurewidth');
Rp - Shunt resistance
clear
Variables
for i=1:5
    Rp=10^(4-i)
    simOut=sim('VariablesEffects');
    current(i,:)=PVcurrent2.signals.values;
    voltage(i,:)=PVVoltage2.signals.values;
    Power(i,:)=PVPower.signals.values;
    legend_labels{i,:}=[sprintf('Rp = %# 6.1f',Rp) ' \Omega' ]
end
f3 = figure(3);
grid on
plot(voltage(1,:),current(1,:),'b-',voltage(2,:),current(2,:),'m-',voltage
    (3,:),current(3,:),'k-',voltage(4,:),current(4,:),'c-',voltage(5,:),
    current(5,:),'r-');
xlabel('Voltage [V]')
ylabel('Current [A]')
axis([0 22 0 5.2])
legend(legend_labels,'Location','SouthWest')%'R_p = 1.0 k\Omega','R_p =
    0.1 k\Omega','R_p = 10  \Omega','R_p = 1.0  \Omega','R_p = 0.1  \
    Omega',
legend BOXOFF
saveas(f3, 'RpDemonstration.png');
matlab2tikz('RpDemonstration.tex');
```

```
matlab2tikz('RpDemonstration.tikz', 'height', '\figureheight', 'width', '\
    figurewidth');
Ambient temperature

% Rp - Shunt resistance
clear
Variables
for i=1:5
    Tamb=273-20+(i*10);
    simOut=sim('VariablesEffects');
    current(i,:)=PVcurrent2.signals.values;
    voltage(i,:)=PVVoltage2.signals.values;
    Power(i,:)=PVPower.signals.values;
end

f4 = figure(4);
grid on
plot(voltage(1,:),current(1,:),'b-',voltage(2,:),current(2,:),'m-',voltage
    (3,:),current(3,:),'k-',voltage(4,:),current(4,:),'c-',voltage(5,:),
    current(5,:),'r-');
xlabel('Voltage [V]')
ylabel('Current [A]')
axis([0 24 0 5.3])
title('I-V curves with different ambient tempereatures')
legend('-10^\circC',' 00^\circC',' 10^\circC',' 20^\circC',' 30^\circC','
    Location','SouthWest')
legend BOXOFF
saveas(f4, 'TambIVDemonstration.png');
matlab2tikz('TambIVDemonstration.tex');
matlab2tikz('TambIVDemonstration.tikz', 'height', '\figureheight', 'width'
    , '\figurewidth');

f5 = figure(5);
grid on
grid minor
plot(voltage(1,:),Power(1,:),'b-',voltage(2,:),Power(2,:),'m-',voltage
    (3,:),Power(3,:),'k-',voltage(4,:),Power(4,:),'c-',voltage(5,:),Power
    (5,:),'r-');
xlabel('Voltage [V]')
ylabel('Power [W]')
legend('-10^\circC',' 00^\circC',' 10^\circC',' 20^\circC',' 30^\circC','
    Location','NorthWest')
legend BOXOFF
saveas(f5, 'TambPVDemonstration.png');
matlab2tikz('TambPVDemonstration.tex');
matlab2tikz('TambPVDemonstration.tikz', 'height', '\figureheight', 'width'
    , '\figurewidth');

Irradiance
clear
Variables
for i=1:5
    Irradiance=1200-(200*i)
    simOut=sim('VariablesEffects');
    current(i,:)=PVcurrent2.signals.values;
    voltage(i,:)=PVVoltage2.signals.values;
    Power(i,:)=PVPower.signals.values;
```

```
end

f6 = figure(6);
grid on
plot(voltage(1,:),Power(1,:),'b—',voltage(2,:),Power(2,:),'m—',voltage
    (3,:),Power(3,:),'k—',voltage(4,:),Power(4,:),'c—',voltage(5,:),Power
    (5,:),'r—');
xlabel('Voltage [V]')
ylabel('Power [W]')
legend('1000 W/m^2',' 800 W/m^2',' 600 W/m^2',' 400 W/m^2',' 200 W/m^2','
    Location','NorthWest')
title('PV curves for different solar irradiance values')
legend BOXOFF
saveas(f6, 'IrrPVDemonstration.png');
matlab2tikz('IrrPVDemonstration.tex');
matlab2tikz('IrrPVDemonstration.tikz', 'height', '\figureheight', 'width',
     '\figurewidth');

f7 = figure(7);
grid on
plot(voltage(1,:),current(1,:),'b—',voltage(2,:),current(2,:),'m—',voltage
    (3,:),current(3,:),'k—',voltage(4,:),current(4,:),'c—',voltage(5,:),
    current(5,:),'r—');
grid off
xlabel('Voltage [V]')
ylabel('Current [A]')
axis([0 27 0 6])
legend('1000 W/m^2',' 800 W/m^2',' 600 W/m^2',' 400 W/m^2',' 200 W/m^2','
    Location','NorthEast')
legend BOXOFF
saveas(f7, 'IrrIVDemonstration.png');
matlab2tikz('IrrIVDemonstration.tex');
matlab2tikz('IrrIVDemonstration.tikz', 'height', '\figureheight', 'width',
     '\figurewidth');

f8 = figure(8);
grid on
plot(voltage(1,:),Power(1,:),'b—',voltage(2,:),Power(2,:),'m—',voltage
    (3,:),Power(3,:),'k—',voltage(4,:),Power(4,:),'c—',voltage(5,:),Power
    (5,:),'r—');
xlabel('Voltage [V]')
ylabel('Power [W]')
axis([20 22 0 30])
legend('1000 W/m^2',' 800 W/m^2',' 600 W/m^2',' 400 W/m^2',' 200 W/m^2','
    Location','NorthEast')
title('PV curves for different solar irradiance values')
legend BOXOFF
saveas(f8, 'IrrPVDemonstrationZoom.png');
matlab2tikz('IrrPVDemonstrationZoom.tex');
matlab2tikz('IrrPVDemonstrationZoom.tikz', 'height', '\figureheight', '
    width', '\figurewidth');

function that prints the simulink model to .eps files.
print —dpdf —r300 —sVariablesEffects VariablesEffects.pdf
printalls('VariablesEffects','Y:\Masterprosjekt\Latexfiler\VariablesEffec\
    Printedmodels')
```

A.3 Averaging algorithm for power smoothing

```
function Avg =AveragingAlgorithm(Power, steps)
%steps=50;
persistent PowerVector;
if isempty(PowerVector)
    PowerVector=zeros(1,1000); %The length of the vector must initially be
                              %set at a larger value than step. An error
                              %occurs if not.
end

temp=PowerVector;
PowerVector(steps)=Power; %Add the last value to the powervector
i=steps;
while i>2
    PowerVector(i-1)=temp(i); %Move all datapoints one step farther back
    i=i-1;
end
PowerVector(1)=temp(2);
Avg=sum(PowerVector)/steps; % Calculate average value for output
end
```

A.4 Perturb and observe maximum power point tracking (MPPT) algorithm

```
function DutyCycle=MPPT(Ipv,VpvInn,Vdc,Vstep)
%#codegen
%% Algoritm for setting the voltage at Vmpp

persistent Pold Change Dold %Defines two persistent variables so that they
                            %can be stored between call to this function
if isempty(Pold)            %Testing if the variable Pold has been defined
    Pold=0;
end
if isempty(Change)         %Testing if the variable Change has been
    defined
    Change=1;
end
if isempty(Dold)          %Testing if the variable Dold has been defined
    Dold=0.7;
end
%% Setting voltage step size and change variable
DutyStep=Vstep/Vdc;
%% Calculate Power as a result of the previous iteration
P=VpvInn*Ipv;
%% Increase or decrease Voltage based on the conditions
if P>0
    if abs(P-Pold)>0
        if (P>Pold)             %Test if new power is higher than old
            if Change > 0
                Change=1;
            else
```

```
                Change=-1;
            end
        else
            if Change > 0
                Change=-1;
            else
                Change=1;
            end
        end
        DutyCycle=Dold+(Change*DutyStep);
    else
    DutyCycle=Dold;
    end
else
    DutyCycle=0.9;              %In case the algoritm makes the power
end                             % negative, the voltage is set at a value
Pold=P;
Dold=DutyCycle;
end
```

A.5 Battery controller

```
function [Voltage1,D]= BatteryController(Voltage,Powerref,
    MaxChargingCurrent,MaxVoltage,soc,Numseries,NumParallell,Ricell,
    MinVoltage,MaxDischargeCurrent,Vdc)
UpperSoc=0.9;
LowerSoc=0.1;
if(soc<UpperSoc)
    cancharge=1;
else
    cancharge=0;
end
if(soc>LowerSoc)
    candischarge=1;
else
    candischarge=0;
end
CurrentRef=Powerref/Voltage;

if(CurrentRef)<-MaxChargingCurrent
    Current1=sign(CurrentRef)*MaxChargingCurrent;
elseif CurrentRef>MaxDischargeCurrent
    Current1=MaxDischargeCurrent ;
else
    Current1=CurrentRef;
end
if(Current1<0)
    if cancharge
    else
        Current1=0;
    end
else
    if candischarge
    else
        Current1=0;
    end
end
Voltage1=Voltage+Current1*Ricell*Numseries/NumParallell;
if abs(Voltage1)>MaxVoltage
    Voltage1=MaxVoltage;
elseif abs(Voltage1)<MinVoltage
    Voltage1=MinVoltage;
else
end
%Convertion to duty cycle
D=1-(Voltage1/Vdc);

end
```

A.6 Battery module construction

```
function [Num_Series,Num_Parallell,BatteryModulePower,MaxChargingCurrent,
    TotalAh,V_max,MinVoltage,MaxDischargeCurrent,Whrating]=
    batteryModuleSize(BatteryPower,Max_Charging_Voltage,
    Max_Charging_Current,Max_Series,Ahcell,Max_discharge_Current,
    Min_Voltage,V_nom)
Max_Charging_Power_Cell=Max_Charging_Voltage*Max_Charging_Current;
Req_Cells=BatteryPower/Max_Charging_Power_Cell;
if(ceil(Req_Cells)>Max_Series)
    Num_Parallell=ceil(Req_Cells/Max_Series);
    Num_Series=Max_Series;
else
    Num_Series=ceil(Req_Cells);
    Num_Parallell=1;
end
TotalAh=Ahcell*Num_Parallell;
V_max=Max_Charging_Voltage*Num_Series;
MaxChargingCurrent=Max_Charging_Current*Num_Parallell;
BatteryModulePower=Num_Parallell*Num_Series*Max_Charging_Power_Cell;
MinVoltage=Min_Voltage*Num_Series;
MaxDischargeCurrent=Max_discharge_Current*Num_Parallell;
Whrating=V_nom*Num_Series*TotalAh;
end
```