

Adaptive PI Controlled PV Emulator

Armand Horak - 31625045

SCHOOL OF ELECTRICAL, ELECTRONIC, AND COMPUTER ENGINEERING



FACULTY OF ENGINEERING

Mini-dissertation submitted in partial fulfilment for the requirements of the degree Bachelor of Engineering in Electrical and Electronic Engineering

Supervisor: Dr. M.G. Botha

Date: November 2022

Abstract

Typical photovoltaic (PV) systems consists of PV modules and power electronic circuits. The power electronic circuits consists of a maximum power point tracking (MPPT) system used to extract the peak power from the PV modules as well as inverters and other DC-DC converters. This is done by altering the load seen by the PV modules to shift their operating points to the maximum power point. These algorithms are constantly evolving and testing them requires field experiments to be conducted. The experiments include physical PV modules that delivers power based on the environmental conditions present. This places limits on the testing process, since the environmental conditions cannot be altered easily. Also, if they wanted to change the characteristics of the PV module, a new PV module must be used. In this mini-dissertation a PV emulator is designed and developed to provide a flexible solution for the testing of MPPT devices, inverters, and DC-DC converters.

The PV emulator consists of a PV model, a power converter, and a control strategy. The PV model performs parameter extraction to model a PV module solely from its datasheet standard test conditions (STC) parameters. This was completed using the genetic algorithm, whereafter the I-V curves were constructed. The power converter is able to output 800 W , at a maximum of 20 A , allowing for the emulation of future PV modules as well. The control strategy comprised of an adaptive PI controller, able to control the output of the PV emulator under any loading condition. The control strategy proved to be successful, controlling the PV emulator during step-responses and dynamic responses. The settling time of the closed-loop system was always below 100 ms , with 0% overshoot, and the steady-state errors of the PV emulator proved to very low.

Key words: *PV Emulator, Power Electronics, Digital Control System, Buck Converter, Adaptive PI Controller, Fuzzy Logic, Takagi-Sugeno Fuzzy Model*

Declaration

I, Armand Mensauw Horak, declare the following:

1. This assessment submitted is my own work and no unlawful help was obtained during the execution of this work.
2. I understand what plagiarism is, and that plagiarism is wrong.
3. I have not shared my work with anyone and have not and will not allow anyone to copy my work with the intention of passing it off as his/her own work.
4. I acknowledge that I am aware of the content of the Faculty of Engineering Honour Code document and the subsequent sanctions proposed for academic dishonesty.

Signature : 

Date : 24/10/2022

Contents

List of Figures	i
List of Tables	iv
List of Acronyms	v
1 Introduction	1
1.1 Background	1
1.2 Problem Statement	4
1.3 Objectives	4
1.3.1 Primary Objective	4
1.3.2 Secondary Objectives	4
1.4 Scope Definition	4
1.5 Deliverables	5
1.6 Anticipated Benefits of Solution	5
1.7 Project Methodology	5
1.8 Conclusion	6
2 Literature Review	7
2.1 Recent Research	7
2.1.1 A Brief History	7
2.1.2 Recent Research	8
2.2 The PV Model	8
2.2.1 Electrical Circuit Model	8
2.2.2 Interpolation Model	11
2.2.3 Hardware Models	12
2.2.4 Parameters	13
2.2.5 PV Simulator	14
2.3 The Power Converter	15
2.3.1 Programmable Power Supply	15
2.3.2 DC/DC Converters	15
2.4 PV Emulator Loads	17
2.5 The Control Strategy	18
2.6 Conclusion	19
3 Conceptual Design	20
3.1 Functional Flow	20
3.2 Operational Flow	21
3.3 Architectural Analysis	22
3.4 Resource Allocation	24

3.5 Conclusion	24
4 Preliminary Design	25
4.1 Technology Trade-off Studies	25
4.1.1 Controller / Processor	25
4.1.2 Microcontroller	27
4.1.3 Buck Converter	29
4.1.4 Sensing Circuits	31
4.2 Updated Architecture	33
4.3 Conclusion	33
5 Detailed Design	35
5.1 PV Model	35
5.1.1 Optimisation Algorithm	37
5.2 Power Converter	41
5.2.1 Buck Converter	41
5.2.2 Sensing Circuit	50
5.3 Control Strategy	51
5.3.1 State-Space Average Model Derivation	51
5.3.2 On-time state-space model	52
5.3.3 Off-time state-space model	53
5.3.4 State-Space Average Model (SSAM)	54
5.3.5 Small-Signal Behaviour	54
5.3.6 Transfer Function Extraction	56
5.3.7 Open-loop comparison	57
5.3.8 Controller Specifications	59
5.3.9 Controller Design	60
5.4 Conclusion	65
6 Implementation and Evaluation	66
6.1 PV Model	66
6.1.1 Genetic Algorithm	66
6.1.2 I-V Curves	68
6.1.3 GUI	69
6.1.4 Serial Communication	71
6.2 Power Converter	71
6.2.1 Buck converter	72
6.2.2 Measurement Circuit	75
6.2.3 Open-loop response comparison	76
6.3 Control Strategy	77
6.3.1 Digital PI Controller Implementation	77
6.3.2 Digital Filter Design	78
6.3.3 Embedded Software Implementation	79
6.4 Integration	80

6.4.1	Step-response	82
6.4.2	Dynamic Response	85
6.5	Conclusion	85
7	Conclusions and Recommendations	87
7.1	Project Overview and Conclusion	87
7.1.1	Chapter 1	87
7.1.2	Chapter 2	87
7.1.3	Chapter 3	87
7.1.4	Chapter 4	87
7.1.5	Chapter 5	88
7.1.6	Chapter 6	88
7.2	Future Recommendations	89
7.2.1	Analog to Digital Converter	89
7.2.2	Supply Circuit	89
7.2.3	Partial Shading	89
7.3	Limitations	90
7.4	ECSA Exit Level Outcomes	90
Bibliography		92
Appendix A	System Specification Requirements Document	96
Appendix B	Controller Design Data	109
B.1	Tuned Controller Raw Data	109
Appendix C	Genetic Algorithm Sensitivity Analysis	110
Appendix D	Code Repository	112

List of Figures

1.1	PV module circuit diagram	2
1.2	Annual degradation rate of PV module obtained from [4]	2
1.3	I-V curve of a PV module/cell	3
1.4	V-model for systems engineering obtained from [5]	5
2.1	Double exponential ($2D2R$) model adapted from [11]	9
2.2	Single exponential ($1D2R$) model adapted from [11]	9
2.3	Power curve of PV module adapted from [14]	10
2.4	I-V curve of a PV module/cell	12
2.5	Hardware based PV model obtained from [15]	12
2.6	Partial shading losses obtained from [19]	14
2.7	Basic Buck Converter	16
2.8	Synchronous Buck Converter	16
2.9	Change in reference current due to change in output voltage. Obtained from [30]	19
3.1	Level 0 of functional flow	20
3.2	Level 1 of functional flow showing the decomposition of function 3.0	20
3.3	Level 1 of functional flow showing the decomposition of function 6.0	21
3.4	Level 0 of operational flow	21
3.5	Level 1 of operational flow	22
3.6	Level 0 of physical architecture	23
3.7	Level 1 of physical architecture	23
3.8	Resource allocation	24
4.1	Utility functions for evaluation criteria	26
4.2	Microprocessor MCDM	26
4.3	Microcontroller and PC pair MCDM	27
4.4	Utility functions for evaluation criteria	28
4.5	ESP32 MCDM	28
4.6	STM32 Blue pill MCDM	29
4.7	Utility functions for evaluation criteria	30
4.8	Asynchronous buck converter MCDM	30
4.9	Synchronous buck converter MCDM	31
4.10	Utility functions for evaluation criteria	32
4.11	Sensing breakout board MCDM	32
4.12	Isolation amplifier MCDM	33
4.13	Level 1 of physical architecture	33
5.1	Single-diode model	36
5.2	Genetic Algorithm flow diagram	37

5.3	Visualised boundary conditions for initial population	38
5.4	Buck-Converter with non-ideal components	41
5.5	Inductor current	43
5.6	Diode voltage	44
5.7	Diode current	44
5.8	Capacitor current	45
5.9	Buck converter schematic	47
5.10	Simulated buck converter schematic	47
5.11	Driver circuitry waveforms	48
5.12	Output voltage and inductor current waveforms	48
5.13	Inductor current at minimum load	49
5.14	Output ripple voltage	49
5.15	HCNR200 sensing circuit	50
5.16	Buck-Converter with non-ideal components	51
5.17	Buck-Converter Mode 1	52
5.18	Buck-Converter Mode 2	52
5.19	Open loop step response comparison for a 800 W load and an output voltage of 40 V	58
5.20	Open loop step response comparison for a 100 W load and an output voltage of 20 V	58
5.21	Open loop step response comparison for a 40 W load and an output voltage of 70 V	59
5.22	Open loop step response comparison for a 800 W load and an output voltage of 70 V	59
5.23	Proposed control system	60
5.24	Fuzzy model	64
5.25	Response results of 5000 adaptive PI controlled systems	65
6.1	Fast convergence	67
6.2	Medium convergence	67
6.3	Slow convergence	68
6.4	Graphical User Interface Implementation	70
6.5	GUI after data has been entered and ‘Plot Curves’ button pressed	70
6.6	GUI during emulation	71
6.7	Implemented Power Converter	72
6.8	Switching Waveforms at 900.18 W	72
6.9	Switching Waveforms at 934.65 W	73
6.10	Switching Waveforms at 32.786 W	73
6.11	Switching delay	74
6.12	Efficiency versus duty cycle	75
6.13	Voltage sensor characterisation	75
6.14	Current sensor characterisation	76
6.15	Open loop comparison	76
6.16	Open loop comparison	77
6.17	Open loop comparison	77
6.18	Digital PI controller block diagram	78
6.19	Embedded software flow diagram	80
6.20	Integration	80

6.21	Variable resistive load	81
6.22	Operating Points integration	81
6.23	Operating Points errors	82
6.24	Sampling time step response comparison	82
6.25	Step response of PV emulator with 11.6Ω load	83
6.26	Step response of PV emulator with 11.6Ω load and boundaries indicated	83
6.27	Step response of PV emulator with 5.0Ω load	83
6.28	Step response of PV emulator with 5.0Ω load and boundaries indicated	84
6.29	Step response of PV emulator with 3.2Ω load	84
6.30	Step response of PV emulator with 3.2Ω load and boundaries indicated	84
6.31	Dynamic response when load is decreased	85
6.32	Dynamic response when load is increased	85

List of Tables

2.1	Diode quality factors obtained from [9].	11
2.2	DC/DC load matching [28]	18
3.1	Description of interfaces	24
5.1	Required parameters for PV model	35
5.3	Buck converter specifications	41
5.4	Selected Components for buck converter	46
5.5	Transfer function variable descriptions	57
5.6	Converter operating ranges	60
5.7	Controller specifications	60
5.8	Operating Points	61
5.8	Operating Points Continued	61
5.9	Arranged array	62
5.10	Tuned PI controllers at different operating points	63
6.1	Genetic Algorithm Parameter Values	66
6.2	JA Solar 525 W (JAM72S30 525-550/MR) [4]	68
6.3	Trina Solar 430 W (Trina Tallmax 430-450W)	69
6.4	Canadian Solar 295 W (KuPower CS3K-295 300 305 310P) [4]	69
6.5	LONGi Solar 525 W (LR5-72HPH 525~545M) [4]	69
6.6	Efficiency test results	74
B.1	Tuned PI controllers at different operating points	109
C.1	Crossover Rate Sensitivity Analysis	110
C.2	Mutation Rate Sensitivity Analysis	110
C.3	Selection Fraction Sensitivity Analysis	110
C.4	Population Size Sensitivity Analysis	111

List of Acronyms

AC	Alternating Current
DC	Direct Current
PV	Photovoltaic
I-V	Current-Voltage
P-V	Power-Voltage
R-V	Resistance-Voltage
STC	Standard Test Conditions
NOCT	Normal Operating Cell Temperature
MPPT	Maximum Power Point Tracking
MPP	Maximum Power Point
PWM	Pulse-Width Modulation
KCL	Kirchhoff's current law
IGBT	Insulated-Gate Bipolar Transistor
PI	Proportional Integral
PID	Proportional Integral Derivative
MCDM	Multi-Criteria Decision Matrix
PVEM	Photovoltaic emulator

Chapter 1

Introduction

The purpose of this chapter is to define, constrain, and analyse the problem and its context.

1.1 Background

Solar energy is one of the most popular renewable energy generation methods. The rising popularity is driven by the increase in the global energy demand. In Europe, the continent aims to produce 60% of their energy solely from renewable sources by 2050. The global cumulative photovoltaic (PV) capacity has reached 500 GW at the end of 2018, and this value is increasing exponentially [1]. Furthermore, the global increase in solar generation has resulted in more stakeholders entering the market, producing improved technologies at lower costs. The reduction in cost fuels the drive for more installations and the cycle repeats. The costs have declined significantly in the past fifteen years, with the USD/kW generation cost reducing by 70% from 2009 to 2021 [2]. It is clear that the market is proven, stable, and growing. New technologies, along with new electrical standards will continue to rise as the market grows.

Solar energy is generated by means of a PV system. These PV systems generally consist of an array of PV modules, a maximum power point tracker (MPPT) charge controller, a battery bank (typically optional, with grid-tied systems being more common), and an inverter. Most inverters also have built in MPPT's. Each of these devices need to be conformance tested (type tested) to ensure that they meet certain specification requirements, technical standards, and regulations. Testing these devices with PV modules can be a troublesome process due to the dependence of the PV modules on the environment. The PV modules cannot produce a specific output at any time, but rather produces an output subject to environmental conditions. The environmental conditions affecting a PV module's output cannot be altered easily, which is also true for the characteristics of a PV module. The ability to do so is necessary to type test the devices in a PV system.

Before solutions to address this need are explored, it is necessary to investigate parameters affecting a PV module's output. The first parameter to consider is irradiation (solar radiation). Irradiation is defined as the radiant flux received by a surface per unit area. In PV systems, sunlight irradiance is typically measured in watts per square meter (W/m^2), with values ranging from 0 W/m^2 to just above 1100 W/m^2 . A value of 1000 W/m^2 is typically described as the full-sun value.

Secondly is partial shading, which occurs when a PV module is partially covered with shade. This causes a mismatch in the cells within a solar module and occurs when at least a single cell is partially shaded. This then causes the bypass diode to be forward biased and bypasses the array of cells within the module as seen in figure 1.1b. This is in contrast with the normal operation of the PV

module as seen in figure 1.1a. The forward voltage over the other unaffected arrays is reversely applied to the shaded array. This causes a reverse current to flow through the module and heat is generated in the affected array where the resistance is higher. The affected array now becomes a load on the module rather than being completely shut off. The partial shading effects reduces the power produced by the PV module not only by deactivating the array, but also due to the array loading the module. [3]

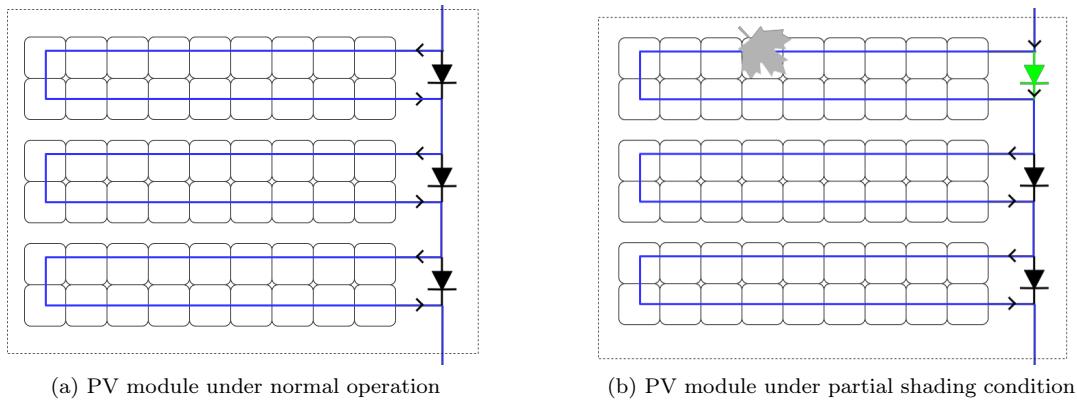


Figure 1.1: PV module circuit diagram

Another factor to consider is that PV panels age as time progresses, causing a linear drop in power output. Figure 1.2 depicts the linear reduction in power output obtained from a JA Solar datasheet. The value is provided as a percentage, typically in the range of 0.6% per annum.

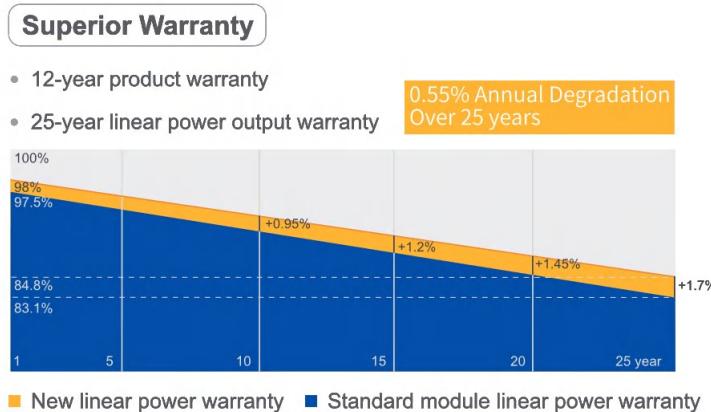


Figure 1.2: Annual degradation rate of PV module obtained from [4]

The temperature effects on the PV modules also have to be considered. PV modules operate more efficiently when they are cool and receive full sun as opposed to when they are warm and receive full sun. This is due to a high potential difference caused across the solar cell when the high energy photons hit the low energy electrons. When temperatures rise, the electrons have a higher energy and the difference between the photon energy and electron energy reduces. Temperature greatly affects V_{OC} , which lowers the maximum power point of the PV panel. Consequently, modelling the temperature is an important factor to consider when emulating/manipulating PV modules.

Lastly, the dynamic response should be considered. To emulate the PV module accurately, the dynamic response of the emulator has to be fast and accurate. This implies that a control loop will be present to ensure the dynamic response is sufficient.

All of the above factors affect the PV module's I-V curve in some manner. The I-V curve of a solar cell graphically depicts the current-voltage relationship of the module. The I-V curve of the cell is specific for a combination of ambient conditions and varies quite significantly. Figure 1.3 depicts a typical I-V curve of a PV module. The operating point on the curve depends on where the load line intersects with the I-V curve, indicated in figure 1.3. Variable voltage and current techniques are necessary for the PV emulator, since it must be able to operate at any point where the load line intersects the curve. It can also be seen on that when the voltage is zero the current is at the short-circuit current value, I_{sc} , and where the current is zero the voltage is at the open-circuit value, V_{oc} . Also, there exists a point on the curve where the voltage and current combination transfers maximum power to the load, given by V_{mp} and I_{mp} respectively. Ideally, one would always like to operate the PV module at that specific point.

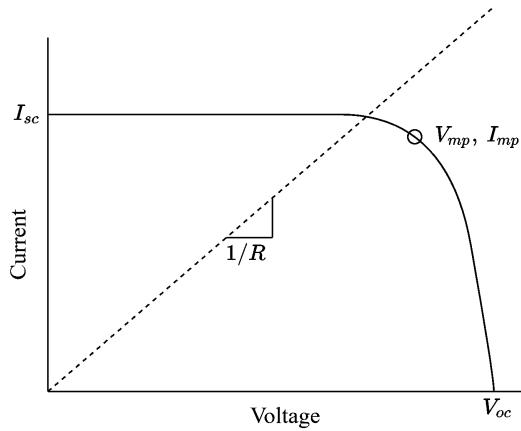


Figure 1.3: I-V curve of a PV module/cell

Various options are available to emulate or manipulate PV modules to produce outputs for the testing of the devices. The first option to consider is connecting the device to a PV module and manipulating the environment around the module to produce the desired outputs. This would effectively allow one to simulate the operating conditions of actual PV modules, but has the drawback that you would have to have the availability of a PV module. This can be extremely costly due to the inability to change and manipulate the PV array characteristics to the potential environmental conditions it can experience. Also, the PV modules could be subject to ageing in the testing environment causing a drift and reducing accuracy of tests in the long term. A typical setup for this option is to place a variable light source over a PV module to vary the irradiance incident on the module. To vary the temperature of the module would be a more challenging task, as sufficient cooling and heating systems are needed to uniformly vary the temperature of the entire panel. This would also be a time consuming process and rapidly varying the temperature of the panel could damage the structure. Partial shading can be controlled more accurately with this setup as realistic shading can be tested on the panel.

A more elegant option and the focus of this project is a PV emulator, which is much more flexible in terms of changing parameters and environmental conditions than the previous option. An emulator

is defined as a system that behaves like another system. Thus, a PV emulator behaves like real PV modules but without the presence of PV modules. The PV emulator recreates the characteristic curves of a PV module and outputs power by means of a power converter. To achieve this, the emulator needs parameters typically found in datasheets to model the desired PV module and to output the power at the operating point. The exact PV module electrical circuit equivalent contains two diodes in parallel connected to a parallel resistor and a series resistance. The values for these resistances are typically not given in datasheets, thus it is crucial to derive values for these parameters from other given values. The parameters typically present in PV module datasheets are the open-circuit voltage (V_{OC}), the short-circuit current (I_{SC}), the voltage and current at maximum power (V_{MP} , I_{MP}), and the temperature coefficients. Also, parameters at STC are present. From these parameters the circuit models can be derived and further analysis can be done to obtain the I-V curves and operating points. Alternatively, the values from the datasheet mentioned above can be used to directly construct the I-V curves of the models. The PV emulator is a very powerful tool for type testing PV system devices, since it can accurately recreate the output of a PV module under arbitrary ambient conditions. Existing commercially available solutions for PV emulation devices are extremely expensive and a cost effective and competitive solution that meets the commercial standards is needed.

1.2 Problem Statement

Conformance testing devices contained in PV systems is a troublesome process without a PV emulator. Commercially available PV emulators are expensive and not easily obtainable, while others compromise on their stability or speed of their dynamic response. The crucial parameters to be emulated are the irradiance and operating temperature.

1.3 Objectives

1.3.1 Primary Objective

- The primary objective of the project is to analyse, design, implement, and characterise a working, cost effective PV emulator that has a user friendly interface without compromising its accuracy and speed of the dynamic response. The main focus is on the control strategy used, since it differentiates a power supply from a PV emulator.

1.3.2 Secondary Objectives

- The system should be designed and implemented with a cost not exceeding R3000.
- The system must be able to output current and voltage levels high enough so that any PV module can be emulated, and that room for future improvement of PV modules are left.

1.4 Scope Definition

The scope of the project is to design and implement a cost-effective PV emulator that can emulate a PV module solely with the parameters obtained from the datasheet. The PV emulator must be able

to successfully and accurately emulate the terminals of a PV module while taking into account the irradiation and temperature. The PV emulator does not have to be self contained and may make use of external resources such as the user's PC for processing and displaying data. The PV emulator's control system must have a stable response without compromising on the speed of the dynamic response.

1.5 Deliverables

The deliverables for the project is as follows:

- The design of the PV emulator.
- The working implementation of the designed PV emulator.

1.6 Anticipated Benefits of Solution

The system will greatly aid in the development and testing speed of MPPT algorithms and circuitry. This is crucial in modern times as PV is an ever growing market, and with the global energy crisis at the moment, any development in the power sector is positive. The PV emulator will also allow more people to enter the MPPT development market as it gives the developers the ability to rigorously test their devices without expensive equipment.

1.7 Project Methodology

The methodology that will be followed throughout this document is known as the systems engineering methodology. Systems engineering serves as a guide for the engineering of complex systems, ensuring that the system is developed in the correct order with all steps accounted for. The systems engineering approach is often iterative, ensuring verification and validation throughout the phases. Several systems engineering models exists, but the most popular one, the v-model, will be utilised. A graphical representation of the v-model is depicted in figure 1.4.

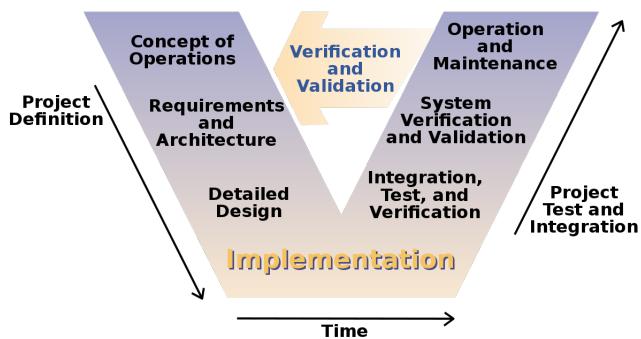


Figure 1.4: V-model for systems engineering obtained from [5]

The problem first defined, analysed, and constrained in detailed before any specialist engineering occurs. This ensures that an appropriate solution is developed that will address the specifications and needs of the client, before time is wasted on specialist engineering. After each phase of the methodology,

a verification and validation step occurs. Chapters 1 and 2 covers the concept of operations, appendix A and chapters 3 and 4 the requirements and architecture phase, while chapter 5 covers the detailed design phase. The implementation, integration, testing, and verification occurs in chapter 6.

1.8 Conclusion

In this chapter, the problem and its context was defined, constrained and analysed. A PV emulator must be designed which can successfully emulate a PV module when given typical datasheet parameters. The PV emulator must be able to take parameters such as irradiance and temperature into account. The anticipated benefits of the solution were also covered followed by the methodology followed throughout this project.

Chapter 2

Literature Review

The purpose of this chapter is to gain an understanding of existing research in the PV emulator field.

The PV emulator was briefly introduced in Chapter 1 and a more detailed definition is needed. The core of the term *PV emulator* can be described by the definition of the word *emulate*, where *emulate* is defined in the Oxford Dictionary [6] as follows:

- **Emulate something** (computing): (of a computer program, etc.) to work in the same way as another computer, etc. and perform the same tasks.

From this definition it can be said that if one system behaves like another it is an emulator, which is exactly what the PV emulator is designed to do. This then naturally leads to the derived definition of a PV emulator:

- A **PV emulator** is a non-linear power supply that mimics the characteristics of a PV module while also taking into account its ambient conditions.

Thus, a PV emulator can supply power to components connected to the PV module and it does so based on the characteristics of the PV module and ambient conditions determined by the user of the emulator. It emulates the terminals of a PV module and outputs the voltage and current at a specific operating point. The parameters affecting the determined operating points are discussed in this chapter.

2.1 Recent Research

This section covers a brief history of PV emulators and how far research in the field has progressed.

2.1.1 A Brief History

One of the earliest documented PV emulators came into existence in 1985 when the authors of [7] created a very primitive but working PV emulator. They called it a PV simulator rather than an emulator, since only PV simulators existed back then. A simulator is the same as an emulator except that the emulator outputs the power after calculating the operating point while a simulator only gives the operating point digitally. Thus, they were one of the first to physically output the voltage and currents obtained from the simulated results. The emulator contained a programmable power supply which received inputs from a CBM-4016 microcomputer and then fed its output to a power amplifier. A programmable multimeter measured the voltage over a shunt resistance which then determined the operating point of the simulator.

A PV emulator a decade down the line (1996) was developed by [8]. This emulator was also called a simulator, but its hardware is very similar to recent emulators described in [9]. It utilised a single phase AC supply, a rectifier, and a PWM controller buck converter containing an IGBT. The mathematical PV model still assumed that all parameters were available and no parameter extraction was done yet. Three methods were identified to construct the I-V curves for a PV model and all three were clearly very accurate for varying temperatures and irradiances. [8]

In 2002 the authors of [10] constructed a three-phase AC supplied PV emulator utilising a CPU board, a DC chopper, and a control loop. This emulator was identified as one of the first to completely run from CPU boards without the help of additional desktop computers. [10]

2.1.2 Recent Research

As the interest in the PV market grew over the years, increasingly more research has been done within the PV emulator field. PV emulators have progressed significantly in both the mathematical model used to obtain the operating point and I-V curves, as well as the hardware and control used to output the voltage and current of the operating point. Several topologies for PV emulators currently exist, each relying on different components. It is crucial to review these topologies to identify their advantages and shortcomings which will aid in the selection of a topology that meets the requirements of this project.

The amount of research done in the PV emulator field was enough for the authors of [9] to write an extensive review on the topic. Their idea was to reduce the time on the information gathering process for those investigating PV emulators. However, some topics are not covered by [9] as it only serves as a basis for the researcher. In [9], the authors suggests that a PV emulator can be broken into three main parts, namely: the PV model, control strategy, and power converter. Not every paper explicitly breaks the PV emulator into these components, but it was found that the three main components are present in each of them. A section for each of these components are present in this chapter.

2.2 The PV Model

The PV model generates the I-V curve of the PV emulator and finds the operating point for the emulator. The PV model itself can be divided into two subcategories, namely an electrical circuit model and a interpolation model [9].

2.2.1 Electrical Circuit Model

The electrical circuit model is an electrical circuit representation of the PV module. The equations for the PV model are derived from the electrical circuit, provided in figures 2.1 and 2.2, with Kirchhoff's current law (KCL). The characteristic I-V, P-V, and R-V curves can then be found from these equations given that all parameters present in the equations are known. The resistor values and diode constants are specific to the environmental conditions and material used for the PV module. The non-linear curve is dictated by the diodes present within the circuit. The exact electrical equivalent circuit of a PV module is displayed in figure 2.1. It can be seen that two diodes ($2D$), and two resistors ($2R$) are present in the model, yielding the $2D2R$ model.

The series resistance represents the power losses due to current circulation throughout the PV module [9]. A change in series resistance affects the short circuit current parameter. From a typical

I-V curve presented in [4] it is clear that the short circuit just about doubles as the irradiance doubles. Thus, the series resistance reduces to about half of its value as the irradiance doubles. The parallel resistance is present in the circuit due to the leakage current in the p-n junction of the PV module. A change in parallel resistance affects the open circuit voltage of the PV module. [9]. The current through diode D_1 , represents the ideal recombination current in the p- and n-side of the PV module, while the current through diode D_2 represents the non-ideal recombination current in the depletion region of the PV module. The series resistance of the PV module is represented by R_s , while the parallel resistance is represented by R_p . I_{ph} represents the photon-generated current. [11]

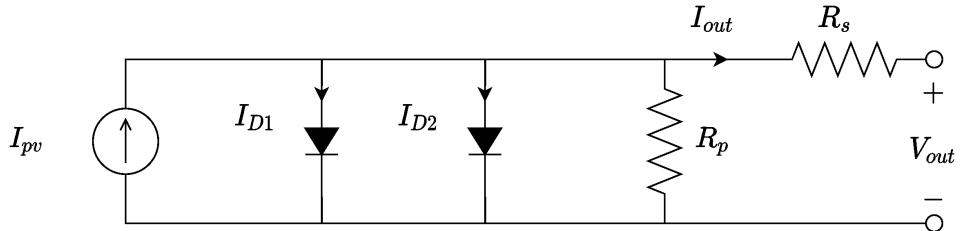


Figure 2.1: Double exponential (2D2R) model adapted from [11]

This model is often too complex to use due to the exponential equations modelling the diode characteristics. Two diodes are present in the 2D2R model, which yields a double exponential equation. A simpler and more frequently used model is the 1D2R model, which combines diodes D_1 and D_2 , with the assumption that the recombination of the PV cell can be described by a single diode [11]. This model is used more often due to its reduced complexity, but compromises on accuracy are made [9]. However, the dynamic response of the model is improved due to smaller calculation times [9]. It is extremely complex to model the 2D2R model with the parameter extraction method described in the following subsections and it often hinders one's ability to perform parameter extraction [12]. The accuracy of the 1D2R model is still extremely high and ultimately has an advantage over the 2D2R model due to its simplicity. Figure 2.2 depicts an electrical circuit representation of the 1D2R model.

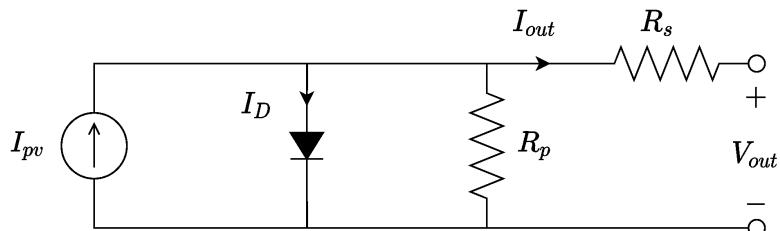


Figure 2.2: Single exponential (1D2R) model adapted from [11]

After deriving the equations for the 1D2R model it will become clear that not all the parameters can be calculated using the datasheet parameters. From the five parameters (I_{out} , I_{ph} , R_p , R_s , and a) only two have equations. The other three parameters are contained within the equations and they are specific to various temperature and irradiance conditions [13]. The problem at hand is an optimisation problem due to there being more variables than equations. Many algorithms exist for obtaining solutions to the problem, but all of them yield different results and have different computation times. Both the speed and accuracy of the solution is important due to the need for low computational times and the ability

to accurately recreate the I-V characteristics. The options for solving this problem are discussed below.

Parametric Model

The first solution is to avoid the optimisation problem and parameter extraction completely. This is possible when all of the parameters are known. The parametric model assumes that all the parameters for the PV module are available on the datasheet. This includes values like module constants, module surface area, energy gap of the device material, an array constant and series resistance. These values have to be present at different ambient conditions, which would make the datasheet verbose and difficult to interpret. Therefore, datasheets typically do not provide information like this directly and a need exists to extract these parameters from other values. Both [7] and [8] used this model.

Parameter Extraction

A very desirable quality for a PV emulator to have is the ability to model the PV module and emulate its terminals solely from the datasheet parameters. Parameters typically available on datasheets are: open circuit voltage (V_{oc}), short circuit current (I_{sc}), voltage at maximum power (V_{mp}), current at maximum power (I_{mp}), PV module material, module efficiency, module degradation rate, rated power, temperature coefficients, power tolerance, number of cells, and number of diodes. The values for V_{oc} , I_{sc} , V_{mp} , and I_{mp} are typically given at STC and NOCT.

The parameter extraction of the 1D2R model involves deriving the equations for the photo-generated current, I_{ph} , and the load current, I_o , analytically. Thereafter, the series and parallel resistance, along with the diode ideality factor a is calculated using an optimisation model [12]. The parameter extraction optimisation technique utilises the derived equations to solve the problem [12, 13]. After that an error function to be minimised is constructed in the maximum power point and then the objective functions are defined [12, 13]. Upon investigation of figure 2.3, it is clear that the slope at the maximum power point is zero on the P-V curve. This yields a reference value for the optimisation algorithm as it is the only known value at any given temperature and irradiance.

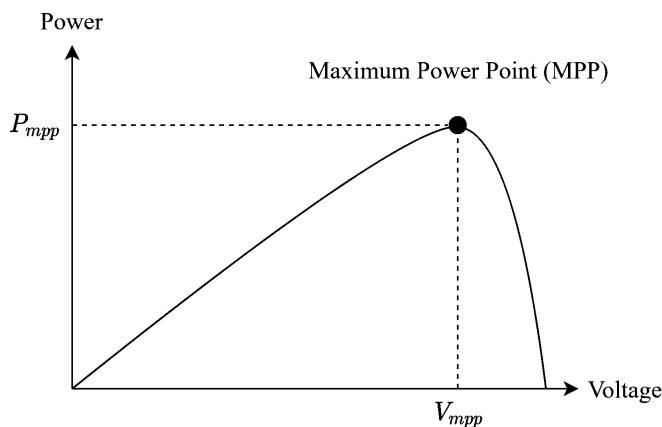


Figure 2.3: Power curve of PV module adapted from [14]

The minimisation of the error function yields a value for R_s , R_p , and a . The values for these parameters all differ depending on which optimisation algorithm is used. In [12], the authors reviewed

evolutionary optimisation algorithms as a way to improve the speed of the algorithms typically used. Evolutionary algorithms are a category of optimisation algorithms aimed at recreating certain optimisation behaviours present in organisms. The bacterial foraging algorithm specifically imitates the behaviour of bacteria moving to nutrient rich areas. These evolutionary algorithms are very accurate and fast as opposed to some of the more conventional mathematical models. An example of a typical mathematical model used is the conjugate gradient method as in [13]. Even though [13] yielded very rapid results, the evolutionary algorithms proved to be faster [9]. From the evolutionary algorithms the Bacterial Foraging Algorithm (BFA) proved to be much faster and accurate than both the Genetic Algorithm (GA) and the Artificial Immune System (AIS) [12]. The conjugate gradient method is much easier to implement, but the speed of the algorithm is an important aspect to consider. The Genetic Algorithm is the simplest and most popular of the evolutionary algorithms, and it is a proven optimisation algorithm.

A method of speeding up the optimisation problem is to add more constraints which reduces the complexity of the model. The diode ideality factor, a , can be estimated instead of being calculated. This is true since the ideality factor is related to the material used for the p-n region, which is available on the datasheet. The values typically range between one and two, with the ideal value being one. However, these values are never the ideal value and table 2.1 depicts the practical values for each material typically used. It must be noted that the selection of the a value beforehand might have drastic effects on the outcome of R_s and R_p when using the optimisation value. This is because the possibility exists that the typical value for the ideality factor might not be the optimal value to generate the I-V curves. Thus, deciding on a ideality factor beforehand would compromise the accuracy of the model.

Table 2.1: Diode quality factors obtained from [9].

Technology	Ideality Factor
Monocrystalline silicon	1.2
Polycrystalline silicon	1.3
Hybrid amorphous silicon	1.8
Cadmium telluride	1.5

2.2.2 Interpolation Model

Another method to obtain the I-V curves is the interpolation model. This method entails directly calculating the I-V curve from the datasheet parameters, as opposed to first deriving an electrical equivalent model. This is done by interpolating the I-V curves given on the datasheet, since typical datasheet parameters are points on the I-V curves. Figure 2.4 displays the points on the I-V curve and it is clear that the I-V curve can be broken into two main regions, namely, the constant current region to the left of the MPP and the constant voltage region to the right of the maximum power-point (MPP).

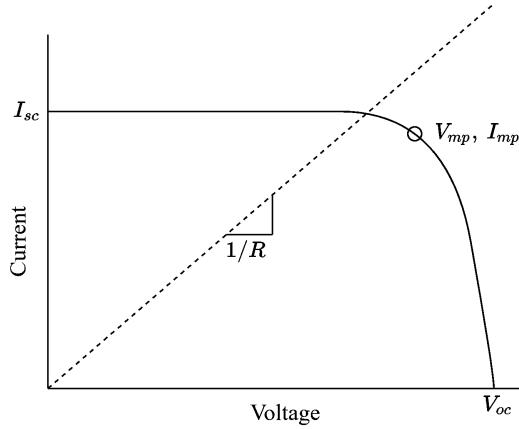


Figure 2.4: I-V curve of a PV module/cell

The advantage of this method is that the curves can be generated more accurately due to eliminating a step in the process when compared to the electrical circuit models. However, the drawback of this is that the data points are not available at a wide range of temperatures and irradiances and it is often difficult to find a relationship between the curves at different temperatures and irradiances. The electrical circuit model aids in this scaling step and therefore it is a more reliable method to model the I-V curves solely from the datasheet for a wider range of ambient conditions. [9, 8]

2.2.3 Hardware Models

In [15, 16] the authors created a hardware implemented PV model. This is different to the PV models previously discussed as they are mathematical models. The hardware based method in [15] was constructed using physical diodes connected in the same manner as PV cells in a PV module. A series of diodes were connected as shown in the figure below to form a ‘PV module’. The transistors are connected as current sources which are then connected to a selector switch for resistances to represent different irradiances which would produce different current.

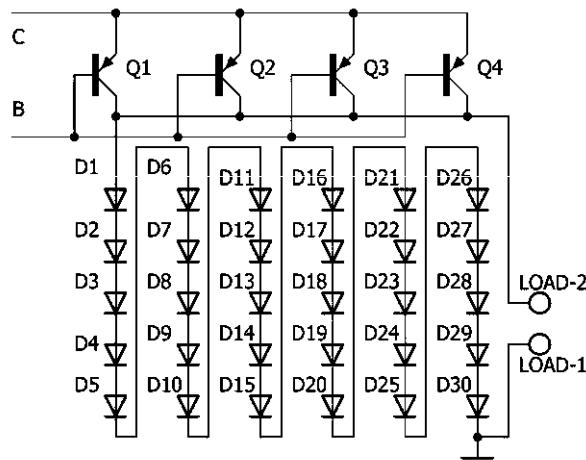


Figure 2.5: Hardware based PV model obtained from [15]

Although this method is simple and effective, the characteristics of the emulated PV module cannot be changed as they are subject to the responses of the diodes. The selector switch configuration also only allows for a set amount of irradiances which can be simulated. Another factor to consider is that the physical diodes become hot when conducting current and their characteristics alter the output under varying temperatures. The temperatures for the diodes used in these strings rose to 63.6°C presenting relatively high power losses [16]. After a comparison of this model with the electrical circuit model and interpolation model, the accuracy of this model was found to be very poor. [15, 16]

An Alternative Approach

In [17] the authors used actual PV modules with variable light sources. The setup contained a chamber containing variable light sources in which PV modules were placed. The chamber only allowed for varying irradiances. This method is restrictive due to the fact that changing the PV module characteristics would entail replacing the entire PV module in the chamber. A wide variety of PV modules are available and it could become an expensive process if a new PV module is required each time.

2.2.4 Parameters

Parameters typically included into the PV emulator are considered in this subsection.

Irradiance And Temperature

The irradiance and temperature used in the model are determined by the user and its effects on the module is contained within the mathematical formulation of the electrical circuit model [12]. Irradiance affects the output of the PV module more than any other parameter. Upon inspection of the JA Solar 525 – 550W module datasheet [4] it is clear that the short circuit current almost doubles upon doubling the irradiance incident on the module. A change in temperature varies the open circuit voltage slightly. When utilising the interpolation model, the irradiance and temperature are modelled with coefficients in order to scale parameters such as V_{oc} , I_{sc} , V_{mp} , and I_{mp} accordingly. The irradiance in the hardware model was changed in two different ways. The diode based hardware model utilised a selector switch to simulate different irradiances, while the chamber method utilised a variable light source.

PV Module Age

The PV module is subject to ageing effects, which can be categorised as optical and electrical effects. The optical effects relates to the reduction in the amount of solar irradiation (transmissivity) that passes through the glass of the PV module and ultimately onto the PV cell contained in the module. The electrical effects relates to an increase in series resistance over time and therefore an increase in losses as the module ages. Each of these effects are described by an equation, where equation 2.1 represents the drop in transmissivity and equation 2.2 the increase in series resistance. It can be seen that there is a linear drop in transmissivity and a linear increase in series resistance over time. [18]

$$\tau(T) = \tau_0(-\alpha_{opt}T + 100\%) \quad (2.1)$$

$$R_s(T) = R_{s0}(\alpha_{Rs}T + 100\%) \quad (2.2)$$

However, on datasheets these parameters are usually not given. For example, in [4] it can be seen that a single value is given for an annual decrease in power output due to a combination of the effects described in equations 2.1 and 2.2. Since the idea is to solely model the PV module from the datasheet, a better option, although less accurate, would be to model it using the annual degradation rate provided on datasheets. The annual degradation rate is given as a percentage power loss [4].

Partial Shading

The partial shading effects on a PV module result in complex P-V curves. Under partial shading condition the PV cells within the PV module are subject to different irradiance levels [19]. The curve becomes segmented and a more than one local maximum power point exists. Figure 2.6 depicts a typical P-V curve of a PV module experiencing partial shading compared to one experiencing no shading.

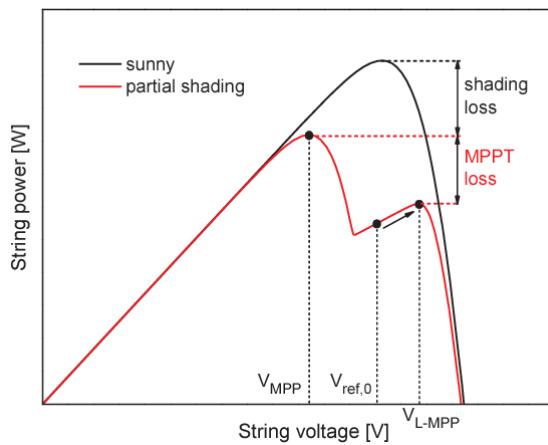


Figure 2.6: Partial shading losses obtained from [19]

Instead of working on module level, partial shading effects are calculated on cell level [19]. The proposed method in [19] is to evaluate the shading condition of each cell and then determine if the bypass diode of the string is conducting or non-conducting. Afterwards the shading effects can be calculated per cell with a set of equations. The authors of [19] considered pioneers in the PV modelling field and proposed the above mentioned method for calculating the partial shading effects [20]. The convenience of working on cell level is that the user of the PV emulator can select shaded cells via an interface, which can aid the user to simulate practical shading effects.

2.2.5 PV Simulator

Up to this point, only mathematical models of PV emulators were discussed. Upon the implementation of the mathematical models, with the addition of simulated loads, a PV simulator is constructed. The PV simulator is transformed into a PV emulator after the implementation of a power converter and control strategy discussed in the following sections. It is convenient to create a modular system in such a way that the PV emulator can function both as a simulator and an emulator. This would be useful since there are use cases in which the user would want to know the outputs of a PV module under specific conditions without needing the physical power output of the PV emulator.

2.3 The Power Converter

The power converter's main purpose is to output the voltage and current determined by the PV model. This section discusses the options available for a power converter and the advantages and disadvantages of each investigated option.

2.3.1 Programmable Power Supply

The programmable power supply (PPS) is another name for a lab bench power supply. These power supplies can typically connect to a computer and take input commands to output a specific voltage or current. These power supplies are often expensive, especially when compared to a self-constructed power converter of the same power output. An advantage of these power supplies are that off the shelf they can be connected in series or parallel to produce more power. This is a very desirable quality when considering a situation where a single PPS emulates a single PV module. Several PPSs can then be connected in series or parallel just like several PV modules are connected in series or parallel. The PPS includes the necessary electrical isolation to do this, which is not necessarily included in a self-constructed power converter. The authors of [21] utilised a PPS in the PV emulator application.

2.3.2 DC/DC Converters

The most commonly used power converter in PV emulator applications is a step-down converter. The step-up operation often isn't needed when a single PV module is emulated as the most common supply voltages are 120 V 60 Hz, and 230 V 50 Hz, both of which are higher than a single PV module's voltage when rectified. A single module's open circuit voltage at STC is typically around 50 V [4], but some high voltage modules exist with voltages in the order of 70 V. When a string of modules have to be emulated from a single emulator a step-up converter might be necessary. The other option is to change the single phase AC supply to a three phase AC supply for a higher voltage and power output and then use a step-down converter while emulating the terminals of an array of PV modules.

Linear Regulator

The linear regulator is a step-down type converter utilising operational amplifiers and zener diodes. This device can be controlled with a variable resistor to create a variable steady output voltage. The linear regulator typically comes in an IC package and constructing one would be similar to reinventing the wheel. The linear regulator cannot supply a large amount of current and some current boosting circuitry is needed in order to increase the power output of the regulator. Some applications of the linear regulator was found for the PV emulator, but often the power loss of the linear regulator is too high, and power output too low for the PV emulator applications. The more preferred DC-DC converter is the conventional buck converter. [9]

Buck Converter

The buck converter is a switching circuit used to step down the input voltage. These circuits operate at high efficiencies and large amounts of power can be transferred with them. These circuits are more preferred in the application of PV emulators as they can be controlled fairly easily. When designed correctly, this circuit is extremely robust. The challenge of these circuits are often the capacitor and

inductor values present in the circuit. The capacitance and inductance values in the design can typically not be obtained from the shelf directly and it is necessary to construct the inductor contained in this circuit. Regarding the capacitor, some iterations in the design have to occur to find the optimal value in terms of quality, ripple voltage, and cost. The buck converter can easily be controlled with a PWM signal, but an upper and lower bound exists on the practical duty cycle. Thus, it is foreseeable that the open circuit voltage and short circuit current might be an issue as it might be difficult to produce low enough voltage and high enough current and vice versa. The conventional buck converter circuit diagram is displayed in figure 2.7.

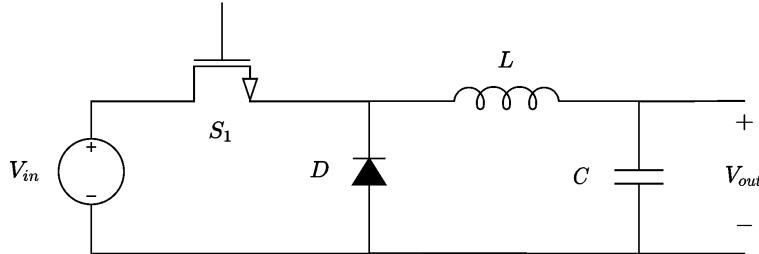


Figure 2.7: Basic Buck Converter

The buck converter can either be synchronous or asynchronous. The conventional buck converter displayed in figure 2.7 is an asynchronous buck converter. The synchronous buck converter is often chosen for its higher efficiency, but the design is a bit more complex [22]. The synchronous buck converter replaces the diode present in the asynchronous buck converter with a transistor to reduce the voltage drop at the diode from between 0.5–1V to less than 0.3V [22]. The synchronous buck converter shows efficiency increases of at least 10% when compared to the asynchronous buck converter. Figure 2.8 depicts a synchronous buck converter, where the diode in figure 2.7 is replaced with the MOSFET S_2 . The term synchronous is used since the PWM signals applied to MOSFETS S_1 and S_2 in figure 2.8 must be applied in a synchronous complementary manner.

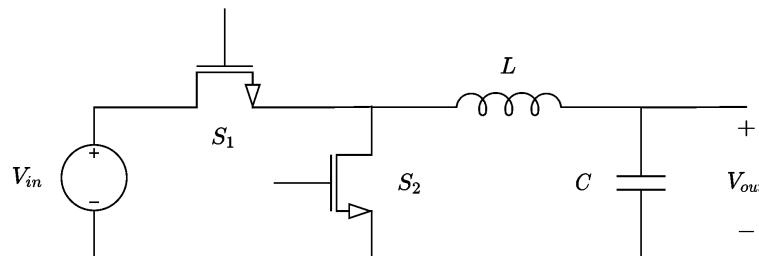


Figure 2.8: Synchronous Buck Converter

The synchronous buck converter plays an important role when considering extremely low voltages. This is due to the relatively large voltage drop across the diode in the asynchronous buck converter that would ensure an extremely low efficiency for the converter since the voltage across the diode would be comparable to the output voltage.

Several studies [10, 8, 23, 24, 25] made use of the asynchronous buck converter for the PV emulator, while only a few [26, 27] utilised the synchronous buck converter. The reason less people utilise the synchronous buck converter is due the difficult task of switching these device in a perfectly complementary

manner. For example, if S_1 and S_2 in figure 2.8 are on at the same time a short-circuit occurs. Thus, perfect complementary switching should be implemented.

Another issue regarding buck converters in general are the size of them. The inductors and capacitors can become extremely large when the power output of the converter is relatively high. An option to decrease the size of these converters are to increase the switching frequency which reduces the size needed for the storage elements. However, if the switching frequency becomes too high, the switching losses become excessive. Thus, a trade-off study in the design is necessary to determine the optimal switching frequency.

Switching the transistors

Switching the transistors in converter circuits typically require currents and voltages outside of the capacity of microcontrollers. Thus, gate driver circuitry is typically required. Also, it is necessary to electrically isolate the controllers from these higher voltages and currents to avoid damage to the microcontroller. Optocouplers are typically used for this application.

2.4 PV Emulator Loads

The PV emulator will in most cases be used to test MPPT devices and inverters. Most inverters contain MPPT devices and therefore the MPPT device is the most important connection on the PV module. The MPPT devices will try to alter the effective resistance seen by the PV module in order to force the PV module to its maximum power point. This could have significant effects on the behaviour of the PV emulator, as it should be able to respond to changes forced by the MPPT device. The effective resistance will change and thus, this is the only parameter which is necessary to consider. The operating point will have to move accordingly and stay on the I-V curve of the PV module selected. This is where the control strategy discussed in section 2.5 is crucial.

The MPPT converter uses a DC/DC converter to implement its load matching feature. Maximum power is transferred from the source to the load when the series resistance of the PV module is equal to the load resistance. The MPPT device is connected in a cascaded manner with the PV module and the load. It varies the load seen by the PV module in order to increase the power output of the PV module. The MPPT device varies the load resistance seen by the module by varying the duty cycle of the DC/DC converter it uses. Table 2.2 displays various DC/DC converters and how their input resistance varies with the duty cycle. It can be seen that the buck converter is the only converter that cannot range its input resistance from zero to infinity, thus it is not preferred in the application of MPPT devices. [28]

Table 2.2: DC/DC load matching [28]

Topology	V_{out}	R_{in}	$R_{in-range}$	I_{in}
Buck	DV_{in}	$\frac{1}{D^2}R_{load}$	$R_{load} \sim +\infty$	Discontinuous
Boost	$\frac{1}{1-D}V_{in}$	$(1-D)^2 R_{load}$	$0 \sim +\infty$	Continuous
Buck-Boost	$\frac{D}{1-D}V_{in}$	$\frac{(1-D)^2}{D^2}R_{load}$	$0 \sim +\infty$	Discontinuous
Ćuk	$\frac{-D}{1-D}V_{in}$	$\frac{(1-D)^2}{D^2}R_{load}$	$0 \sim +\infty$	Continuous
SEPIC	$\frac{D}{1-D}V_{in}$	$\frac{(1-D)^2}{D^2}R_{load}$	$0 \sim +\infty$	Continuous
Zeta	$\frac{D}{1-D}V_{in}$	$\frac{(1-D)^2}{D^2}R_{load}$	$0 \sim +\infty$	Discontinuous

It is clear that the core functionality of a MPPT device is simply a variable resistance. Thus, connecting an MPPT device to a PV emulator should be no different than just connecting a regular variable resistance and setting the resistance to the optimal resistance for maximum power transfer. For example, in [29] a MPPT device was used to test the constructed PV emulator and yielded the same results as the variable resistor, but only with a slightly longer settling time towards the MPP.

2.5 The Control Strategy

The control strategy is the heart of the PV emulator, since it ensures that the PV emulator is always operating on the I-V curve. On varying loads, the control strategy has to ensure that the PV emulator is always operating at the operating point. This can especially cause problems when the PV emulator is connected to MPPT devices as these devices will continuously vary the load seen by the PV emulator. The control system must then be able to keep the PV emulator on the I-V curve and converge to the operating point as quickly as possible.

The conventional control methods of PV emulators employ the PI controller with the current value at the operating point as a reference. The variables fed from the output of the PV emulator to the controller are the output voltage and current. The PI controller operates at a critically damped condition to ensure stability. This configuration creates an unstable control method, especially near the MPP. In the constant current region a small step in reference current is needed for a large change in output voltage. Thus, this region is stable, but in the constant voltage region a small step in output voltage creates a large step in reference current. This causes an unstable condition which causes an oscillating reference point when entering the constant voltage region. When a reference voltage method is used, the opposite in each region occurs. Figure 2.9 displays this phenomenon. This can especially be a problem when an MPPT device constantly varies the load seen by the PV emulator. [30]

A method to solve this is to ensure that the controller operates at an over-damped condition, but this creates a very slow dynamic response. A better method proposed in [30] is an adaptive PI controller specific to a level of irradiance and temperature. The adaptive controller also calculates the critical resistance, which is the resistance intersecting the MPP on a specific curve in order to create two PI

controllers, one for the constant current region and one for the constant voltage region. This produces an accurate and fast dynamic response which is stable. [30]

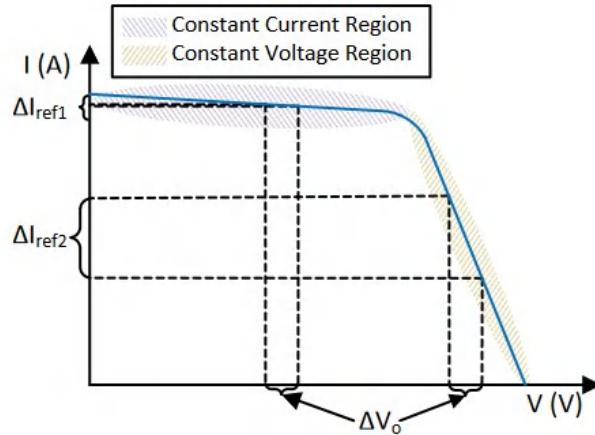


Figure 2.9: Change in reference current due to change in output voltage. Obtained from [30]

Most PV emulators employ the conventional control method as only variable resistances are connected to them and not MPPT devices. Surely, an MPPT device would expose the instability of the PV emulator utilising this simple control method. It is therefore crucial to ensure that a robust control method is utilised when the PV emulator is implemented.

2.6 Conclusion

In this chapter a literature review was conducted in order to develop a familiarisation of current literature amongst the reader and to provide the reader with a better understanding of the aspects of a PV emulator. Several papers, from more than three decades ago up to some of the most recent emulators, were reviewed to consider all perspectives and to form an unbiased view of existing solutions for the PV emulator. Much success has already been achieved in the field of PV emulators, yielding emulators with high accuracy, fast dynamic responses and high power, but most emulators lack robust control methods.

Chapter 3

Conceptual Design

This chapter covers the conceptual design of the PV emulator. The conceptual design sets out to provide a clear path for the detailed design process.

The conceptual design phase of a system breaks the system down in such a way that the exact working and operation of the system can be visualised and understood. This is a key aspect of the engineering design process as it allows one to follow a methodical approach to the design and implementation of a system. This phase of the design also ensures that the final design performs all the necessary functions required to create a PV emulator.

3.1 Functional Flow

The functional flow of the system is covered first, which defines the functions and how they piece together. The functional flow is decomposed until all of the functions are either solely performed by the PV emulator (PVEM) or until they are detailed enough to understand what is required of the PV emulator. Figure 3.1 displays level 0 of the functional flow. The letter *F* as a prefix to the numbers of the blocks represents the word *function*. Furthermore, it is necessary to decompose functions 3.0 and 6.0. Figures 3.2 and 3.3 depict the decomposition of functions 3.0 and 6.0, respectively.

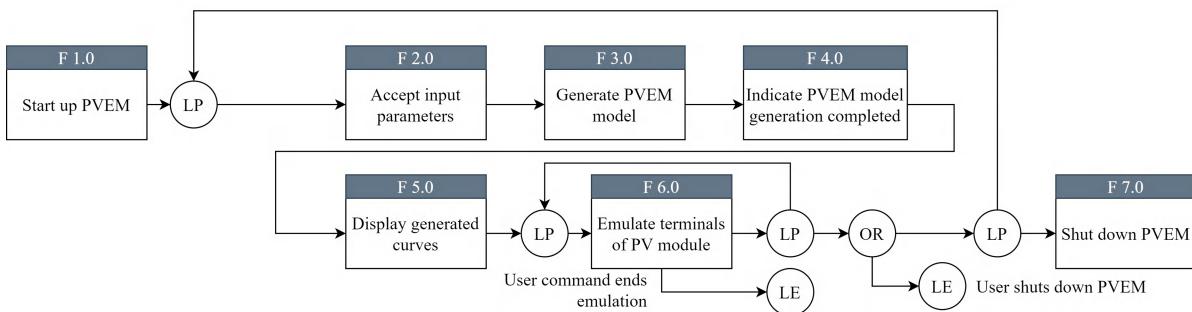


Figure 3.1: Level 0 of functional flow

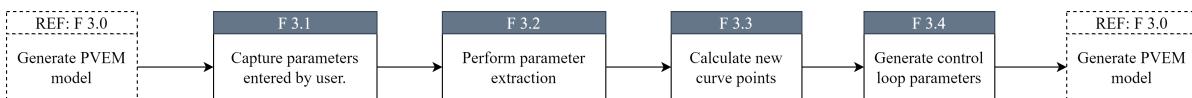


Figure 3.2: Level 1 of functional flow showing the decomposition of function 3.0

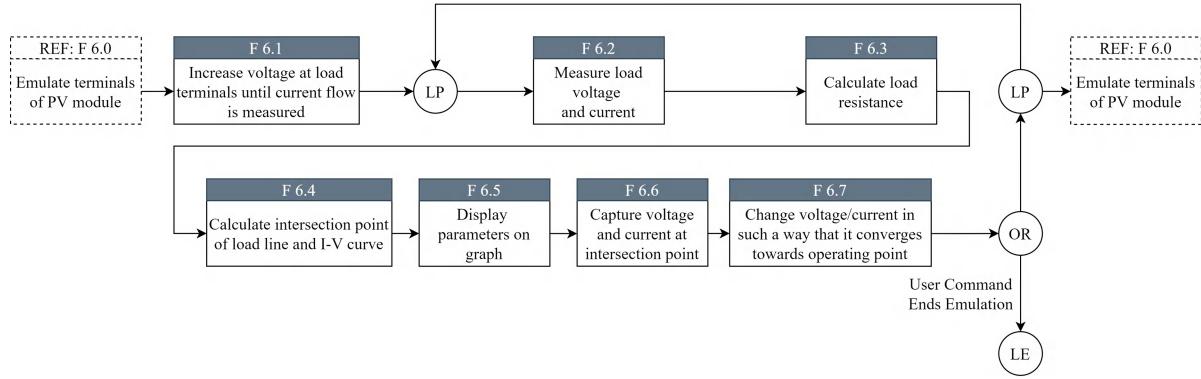


Figure 3.3: Level 1 of functional flow showing the decomposition of function 6.0

3.2 Operational Flow

The operational flow is next, which determines how the user will interact with the PV emulator. Level 0 of the operational flow is depicted in figure 3.4, which only contains three functions. Function 3.0 can be expanded to understand the operation of the PV emulator better.

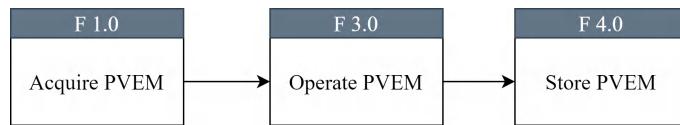


Figure 3.4: Level 0 of operational flow

The expanded version of function 3.0 is shown in figure 3.5.

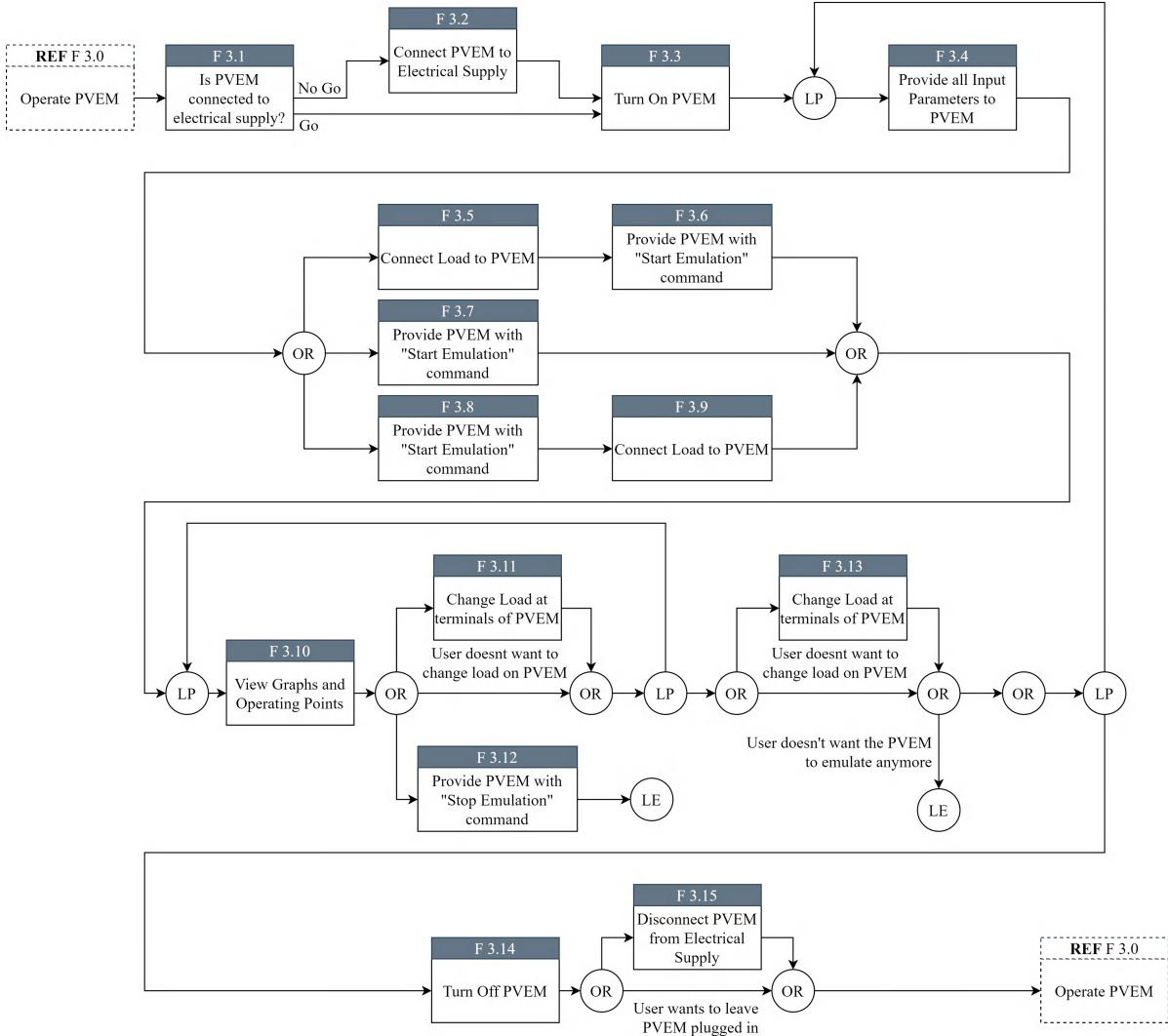


Figure 3.5: Level 1 of operational flow

3.3 Architectural Analysis

The architectural analysis visualises the physical architecture of the system. Level 0 of the physical architecture can be seen in figure 3.6. The system boundary is also visualised in this figure, showing which interfaces and functional units will be designed and implemented. The PVEM can be seen in functional unit (F/U) 3.0. The PVEM has to be decomposed to get a full-view of the components needed to construct the PVEM.

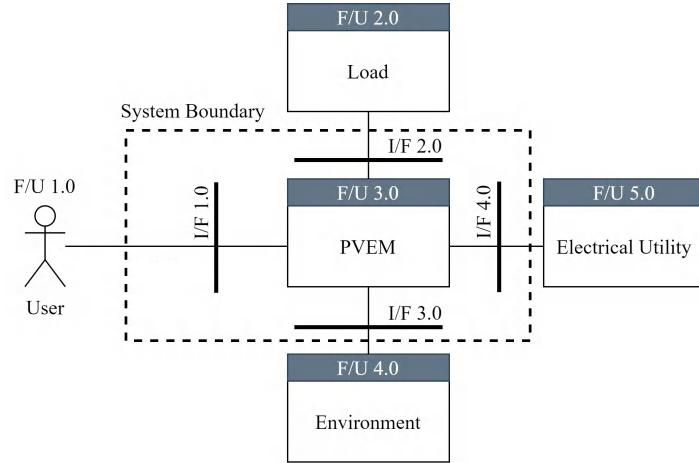


Figure 3.6: Level 0 of physical architecture

The decomposed physical architecture of the PVEM is displayed in figure 3.7.

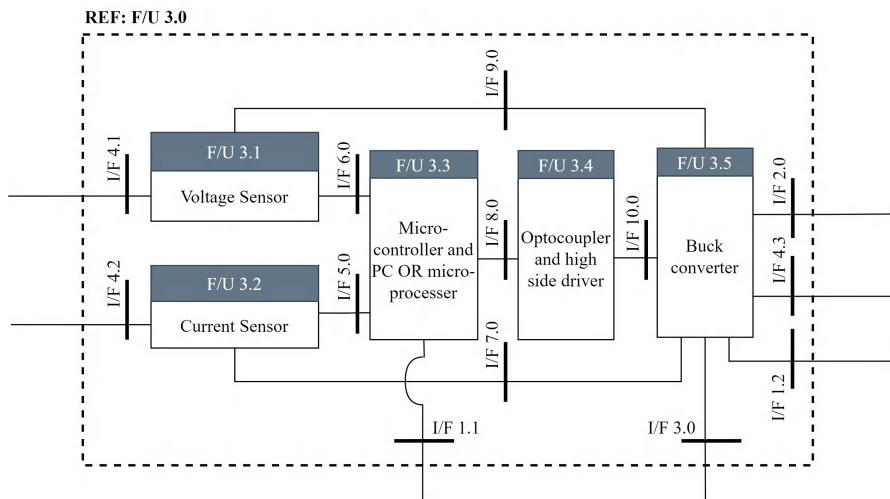


Figure 3.7: Level 1 of physical architecture

The technologies that will be utilised for each of the functional units are covered in the following chapter by means of trade-off studies. Each of the interfaces in figure 3.7 are described briefly in table 3.1.

Table 3.1: Description of interfaces

Interface	Description
I/F 1.1	User interface contained on PC or microprocessor.
I/F 1.2	Load terminals that user can connect loads to.
I/F 2.0	Load terminals that load is connected to.
I/F 3.0	The buck converter interfaces to the environment by heat dissipation
I/F 4.1	Power supply to voltage sensor
I/F 4.2	Power supply to current sensor
I/F 4.3	Power supply to buck converter
I/F 5.0 - I/F 10.0	Electrical wiring

3.4 Resource Allocation

The resource allocation is depicted in figure 3.8. The purpose of the resource allocation is to ensure that all architectural components perform functions, which is indeed the case.

Functions (Functional Flow)	Resources									
	F/U 1.0	F/U 2.0	F/U 3.1	F/U 3.2	F/U 3.3	F/U 3.4	F/U 3.5	F/U 4.0	F/U 5.0	
F 1.0	■			■	■	■	■	■	■	
F 2.0					■	■		■	■	
F 3.1					■			■	■	
F 3.2					■			■	■	
F 3.3					■			■	■	
F 3.4					■			■	■	
F 4.0					■			■	■	
F 5.0					■			■	■	
F 6.1		■			■	■	■	■	■	
F 6.2	■		■	■	■		■	■	■	
F 6.3	■				■		■	■	■	
F 6.4					■			■	■	
F 6.5	■				■			■	■	
F 6.6					■			■	■	
F 6.7		■			■	■	■	■	■	
F 7.0	■		■	■	■	■	■	■	■	

Figure 3.8: Resource allocation

3.5 Conclusion

In this chapter, the conceptual design of the PVEM was completed in order to formulate the working of the system along with the physical architecture needed to execute the functions performed by the system.

Chapter 4

Preliminary Design

This chapter covers the technology trade-off studies for the architecture defined in the previous chapter. The preliminary design shapes the high-level solution into a more specific one that is used as a framework for the detailed design.

4.1 Technology Trade-off Studies

Technology trade-off studies are used to determine the best fit technologies for the architectural components defined in the previous chapter.

4.1.1 Controller / Processor

The microcontroller and PC pair or microprocessor is responsible for the computation of the PV model, the control of the output voltage, as well as the interaction with the user. Using a microprocessor allows the PV emulator to be self contained, not relying on the processing power of a PC, but less computational power is available. However, the speed of the control loop can be much faster. On the other hand, using a microcontroller allows for a simpler design. The user will require not extra peripherals to be connected to the PV emulator since the user's PC's peripherals will be used. This increases the user friendliness of the solution.

The evaluation criteria for the trade-off studies are as follows:

Evaluation Criteria

- **Risk** (Less is better) - Risk evaluates the familiarity of the technology and the application thereof in this specific situation.
- **Cost** (Less is better)
- **Architecture** (More is better) - The architecture relates to the speed of arithmetic operations, which is crucial for this problem due to the mathematics involved.
- **Processing Speed** (More is better)
- **Peripherals required** (Yes or no)

Each of the evaluation criteria has its own function used to score the solutions, depicted in figures 4.1a to 4.1d. A graph is not displayed for the 'Peripherals required' since it is only a binary decision.

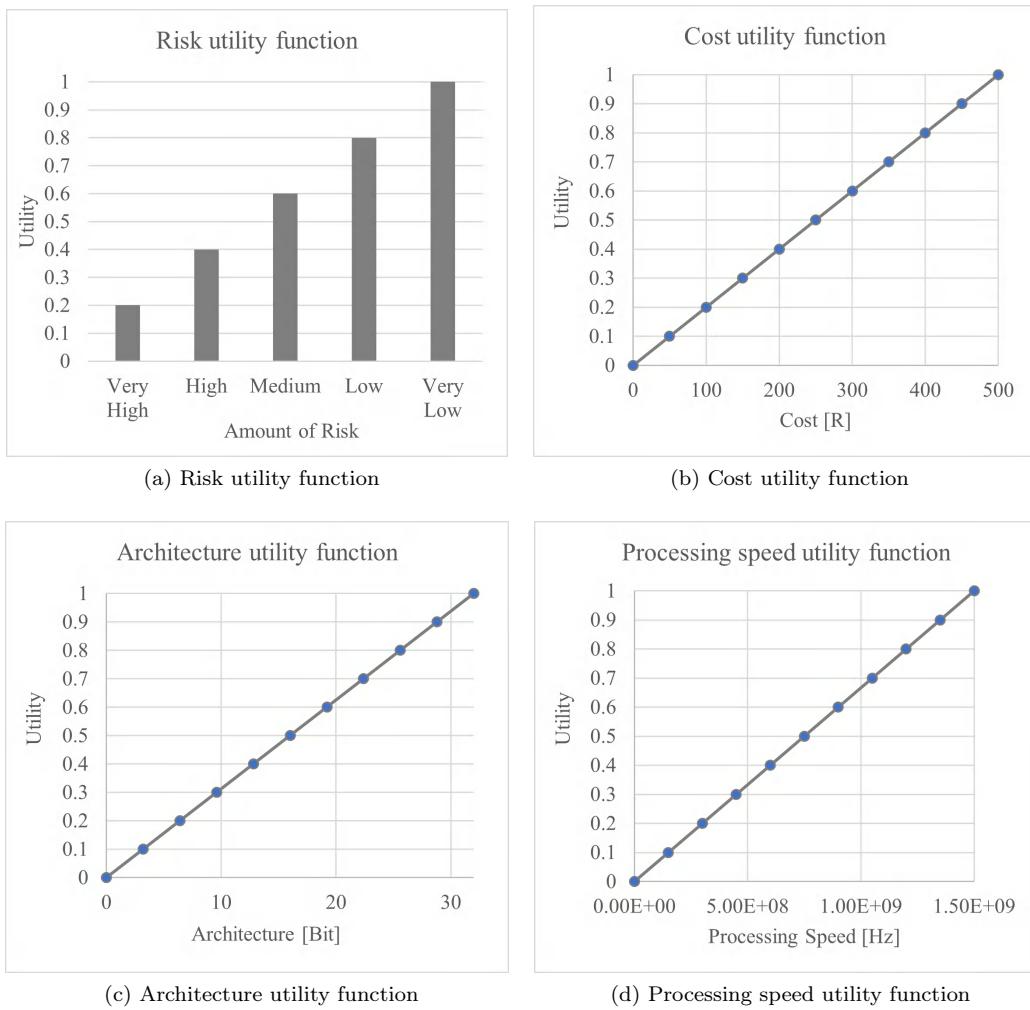


Figure 4.1: Utility functions for evaluation criteria

Multi-Criteria Decision Matrices (MCDM)

Figures 4.2 and 4.3 displays the scoring of each of the technologies.

0.53	Microprocessor				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.200	Low	0.8	0.16
Cost	Linear	0.200	About R400	0.2	0.04
Architecture	Histogram	0.200	32-Bit	1	0.20
Processing Speed	Linear	0.200	1GHz	0.67	0.13
Peripherals Required	Linear	0.200	All	0.2	0.04
		1.000			

Figure 4.2: Microprocessor MCDM

0.55	Microcontroller				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.200	Very Low	1	0.20
Cost	Linear	0.200	About R150	0.7	0.14
Architecture	Histogram	0.200	32-Bit	1	0.20
Processing Speed	Linear	0.200	80MHz	0.05333	0.01
Peripherals Required	Linear	0.200	PC	0.8	0.16
		1.000			

Figure 4.3: Microcontroller and PC pair MCDM

Conclusion

The microcontroller and PC pair yielded a lower cost, number of peripherals required, and risk when compared to the microprocessor. The processing speed of the microprocessor was much higher, and the architecture of both options were the same. The selected solution is the microcontroller and PC pair having a score of 0.55 as opposed to the 0.53 of the microprocessor.

4.1.2 Microcontroller

The microcontroller was selected in the previous subsection, and the specific microcontroller is now selected. The microcontroller must maintain a balance between speed and cost, since the cost-effectiveness is crucial. Only 32-bit controllers will be evaluated as they provide a decent amount of speed for their cost. The controllers compared are the ESP32 and the STM32 Blue pill controllers. The evaluation criteria of the controllers are shown below.

Evaluation Criteria

- **Risk** (Less is better)
- **Cost** (Less is better)
- **Familiarity** (More is better) - Familiarity implies how easily and quickly the solution will be implemented based on my familiarity with the controller.
- **Clock Speed** (More is better)

The evaluation criteria graphs are depicted in figures 4.4a to 4.4d.

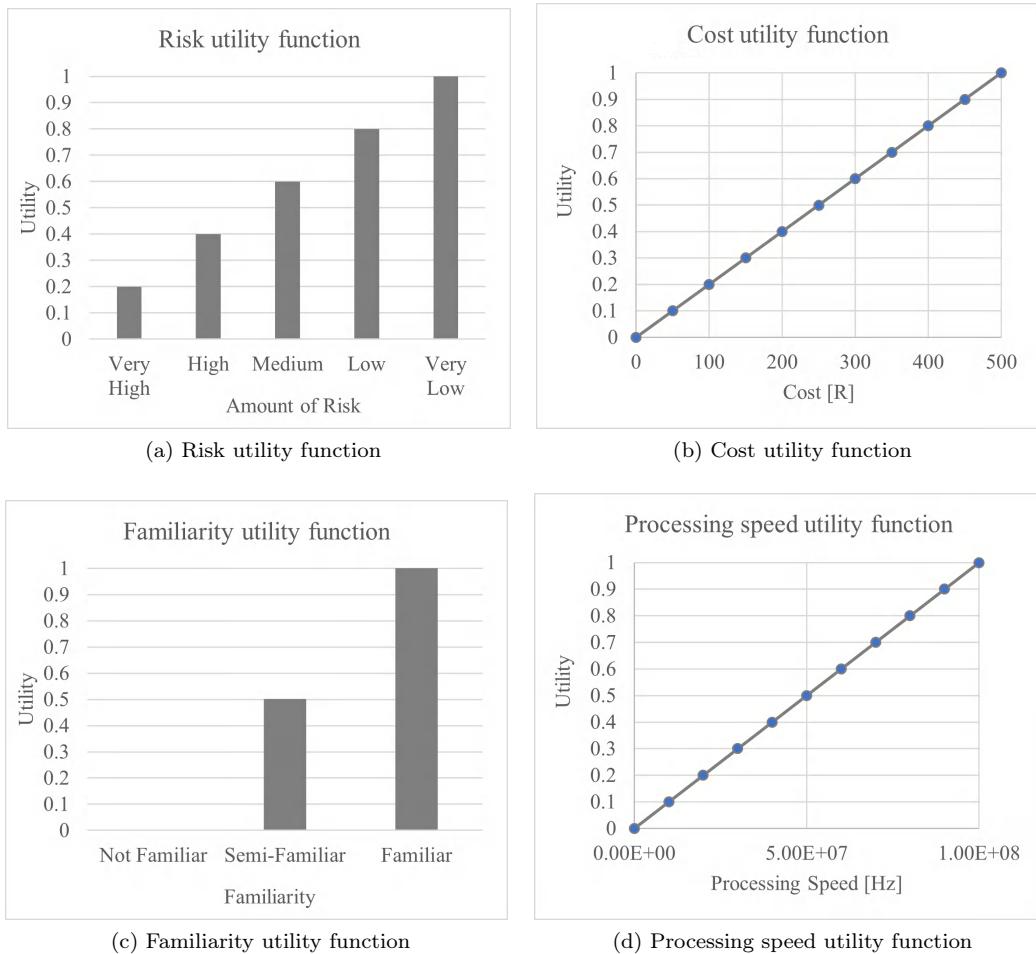


Figure 4.4: Utility functions for evaluation criteria

Multi-Criteria Decision Matrices (MCDM)

Figures 4.5 and 4.6 depicts the scoring of each of the options.

0.86	ESP32				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.250	Very Low	1	0.25
Cost	Linear	0.250	R189.00	0.622	0.16
Familiarity	Histogram	0.250	Familiar	1	0.25
Clock Speed	Linear	0.250	80 MHz	0.8	0.20
		1.000			

Figure 4.5: ESP32 MCDM

0.63	STM32 Blue Pill				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.250	Very Low	1	0.25
Cost	Linear	0.250	R350.00	0.3	0.08
Familiarity	Histogram	0.250	Semi-Familiar	0.5	0.13
Clock Speed	Linear	0.250	72 Mhz	0.72	0.18
		1.000			

Figure 4.6: STM32 Blue pill MCDM

Conclusion

The selected microcontroller is the ESP32, which yielded a much higher score than the STM32 Blue pill. The ESP32 scored better than the STM32 in every criteria except for risk, which was a tie. Thus, the ESP32 is cheaper, faster, and easier to work with than the STM32.

4.1.3 Buck Converter

The buck converter is the circuit typically used for the PV emulator. However, as discussed in chapter 2, the buck converter can either be synchronous or asynchronous. This is an important decision as it will effect the design parameters of the circuit. These two topologies will be compared in the trade-off study to determine the best alternative. The evaluation criteria is listed below.

Evaluation Criteria

- **Risk** (Less is better) - Risk implies evaluates the familiarity of the technology and the application thereof in this specific situation.
- **Complexity** (Less is better) - Complexity in this case refers to the switching complexity of the circuit.
- **Efficiency** (More is better)
- **Forward Voltage** (Less is better) - The forward voltage of the secondary switch relates to the lower output voltage limit that can be achieved with the circuit.

The functions utilised in the scoring of the alternatives are displayed in figures 4.7a to 4.7d.

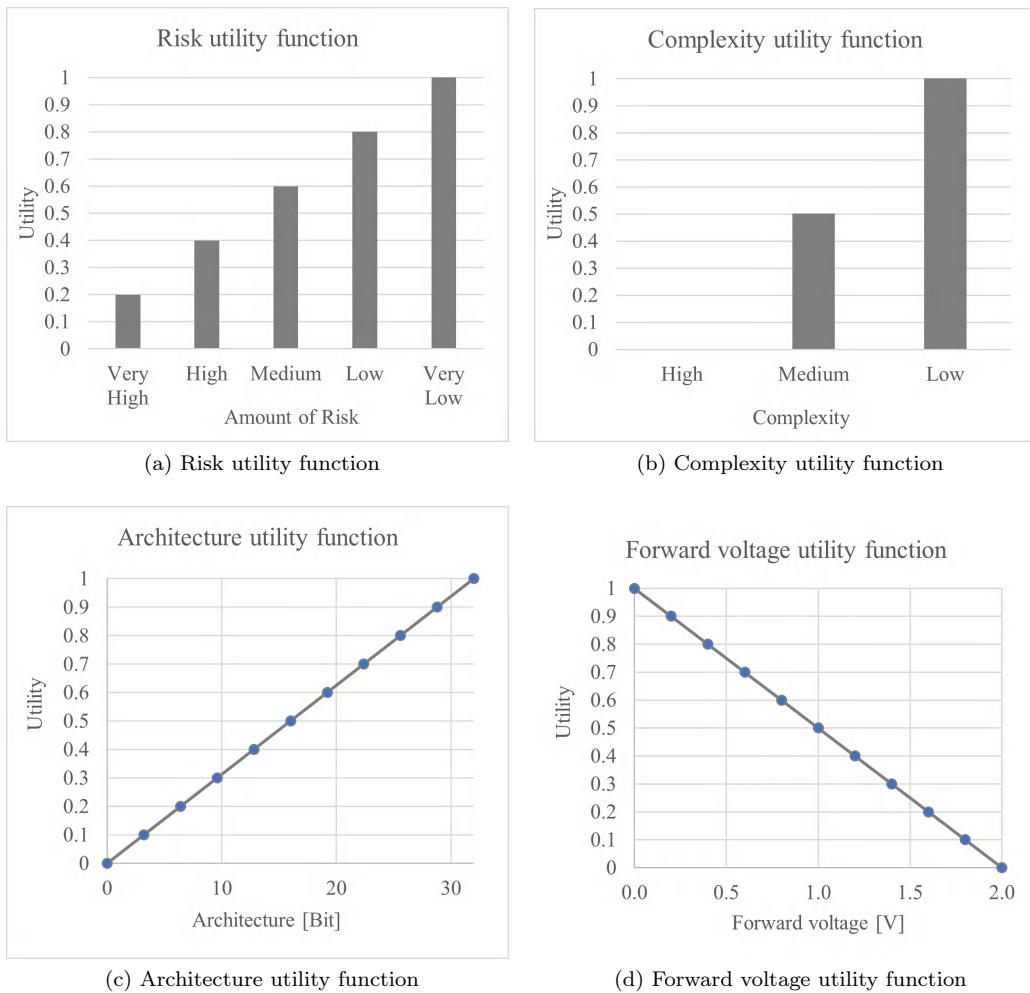


Figure 4.7: Utility functions for evaluation criteria

Multi-Criteria Decision Matrices (MCDM)

Figure 4.8 and 4.9 displays the scoring of each of the options.

0.71	Asynchronous Buck				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.250	Very Low	1	0.25
Complexity	Histogram	0.250	Very Low	1	0.25
Efficiency	Linear	0.250	In order of 85%	0.85	0.21
Forward Voltage	Linear	0.250	In order of 2V	0	0.00
		1.000			

Figure 4.8: Asynchronous buck converter MCDM

0.69	Synchronous Buck				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.250	Low	0.8	0.20
Complexity	Histogram	0.250	Medium	0.5	0.13
Efficiency	Linear	0.250	In order of 95%	0.95	0.24
Forward Voltage	Linear	0.250	In order of 1V	0.5	0.13
		1.000			

Figure 4.9: Synchronous buck converter MCDM

Conclusion

The selected solution is the asynchronous buck converter having a score of 0.71 as opposed to the 0.69 of the synchronous buck converter.

4.1.4 Sensing Circuits

The sensing circuits are responsible for the measurement of voltages and currents within the circuit. Two option are available; the first is to construct your own sensing circuit by means of an isolation amplifier, and the second to buy a breakout board. Each of them has their own advantages and disadvantages, thus a trade-off study is performed to reach an unbiased decision. The criteria used in the trade-off study is shown below, along with a brief description of the criteria that is not self-explanatory.

Evaluation Criteria

- **Risk** (Less is better) - Risk implies evaluates the familiarity of the technology and the application thereof in this specific situation.
- **Cost** (Less is better)
- **Tunable?** (More is better) - Can the output range of the sensor be varied?
- **Non-linearity** (Less is better)

Each of the evaluation criteria has a specific function used for scoring, depicted in figures 4.10a to 4.10d.

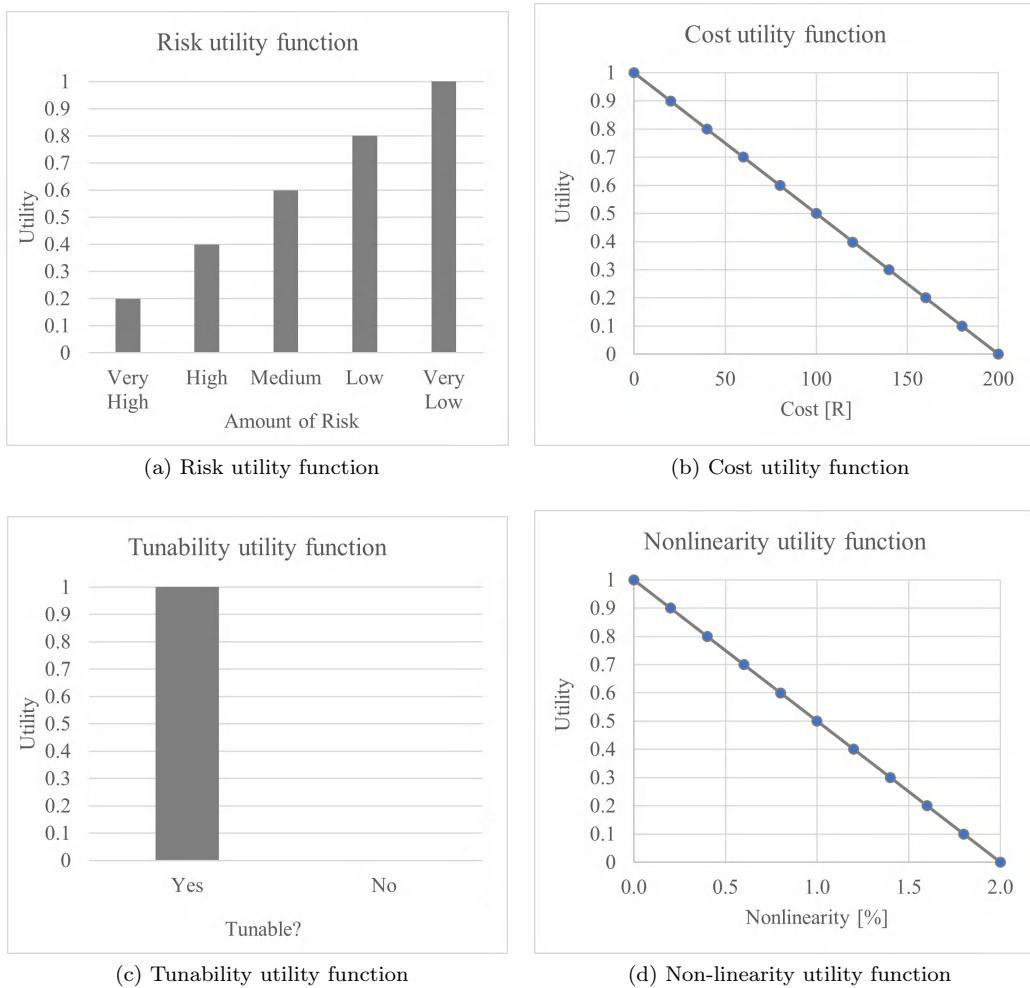


Figure 4.10: Utility functions for evaluation criteria

Multi-Criteria Decision Matrices (MCDM)

Figures 4.11 and 4.12 displays the scoring of each of the options.

0.36	Sensing Breakout Board				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.250	Very Low	1	0.25
Cost	Linear	0.250	R160.00	0.195	0.05
Tunable?	Yes or No	0.250	No	0	0.00
Nonlinearity	Linear	0.250	Approximately 1.5%	0.25	0.06
		1.000			

Figure 4.11: Sensing breakout board MCDM

0.84	Isolation Amplifier Sensor				
	Function Type	Weight	Score	Utility	Utility Score
Risk	Histogram	0.250	Low	0.8	0.20
Cost	Linear	0.250	R90.00	0.55	0.14
Tunable?	Yes or No	0.250	Yes	1	0.25
Nonlinearity	Linear	0.250	Approximately 0.01%	0.995	0.25
		1.000			

Figure 4.12: Isolation amplifier MCDM

Conclusion

The isolation amplifier proved to have a much lower cost than the breakout board, a significantly lower non-linearity, it is tunable, and it has slightly higher risk. The isolation amplifier sensor is ultimately better having a score of 0.84 as opposed to the 0.36 of the breakout board.

4.2 Updated Architecture

The architecture created in the previous chapter can now be updated. The resulting architecture of the system can now be seen in figure 4.13. Its clear that another functional unit was added to the diagram. An additional interface was also added, namely interface 11.0, which is the serial communication between the PC and the ESP32.

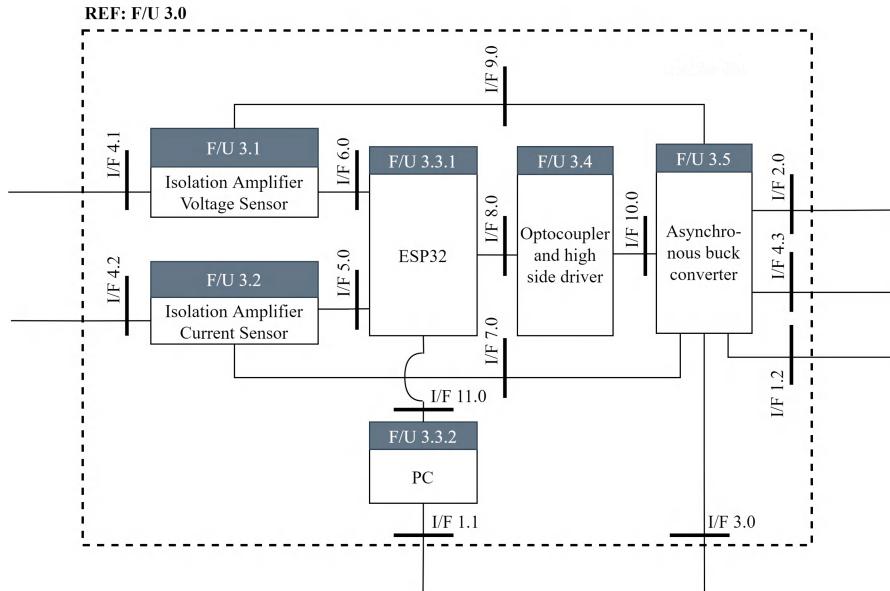


Figure 4.13: Level 1 of physical architecture

4.3 Conclusion

Several trade-off studies were completed in this section in order to refine the architecture for the detailed design process. Thereafter the updated architecture was defined and an extra interface was created,

namely, the serial communication between the microcontroller and the user's PC. The following chapter covers the detailed design of all the sub-systems pertaining to the PV emulator.

Chapter 5

Detailed Design

This chapter covers the detailed design of all subsystems pertaining to the PVEM. The chapter is structured as follows: First, the PV model is designed, which is responsible for the mathematical modelling of a PV module, whereafter the power converter, responsible for delivering power to the load, is designed. This is followed by the design of the control strategy, responsible for ensuring that the PV emulator tracks the reference point swiftly and accurately.

5.1 PV Model

The PV model equations will be derived in this section in order to mathematically model a PV module. To create the PV model for a specific PV module, several parameters are required as input from the user. All of the required parameters are obtainable from the datasheet of a typical PV module. The parameters are summarised in table 5.1 along with their descriptions.

Table 5.1: Required parameters for PV model

Symbol	Description
V_{ocn}	Open-circuit voltage at STC
I_{scn}	Short-circuit current at STC
V_{mpn}	Maximum power point voltage at STC
I_{mpn}	Maximum power point current at STC
k_v	Open-circuit voltage temperature coefficient
k_i	Short-circuit current temperature coefficient
N_s	Number of cells in series
N_p	Number of cell arrays
T	Cell temperature
G	Irradiance

From the parameters in table 5.1 the open-circuit and maximum power voltage, along with the short-circuit and maximum power current can be calculated at the user defined T and G . This is done using (5.1) through (5.4), where G_n is the irradiation at STC [9].

$$V_{oc} = V_{ocn} + V_t \ln \left(\frac{G}{G_n} \right) + k_v \Delta T \quad (5.1)$$

$$V_{mp} = V_{mpn} + V_t \ln \left(\frac{G}{G_n} \right) + k_v \Delta T \quad (5.2)$$

$$I_{sc} = (I_{scn} + I_{scn} k_i \Delta T) \left(\frac{G}{G_n} \right) \quad (5.3)$$

$$I_{mp} = (I_{mpn} + I_{mpn} k_i \Delta T) \left(\frac{G}{G_n} \right) \quad (5.4)$$

ΔT is given by (5.5).

$$\Delta T = (T - T_n) \quad (5.5)$$

The single diode model circuit is repeated in figure 5.1 for convenience.

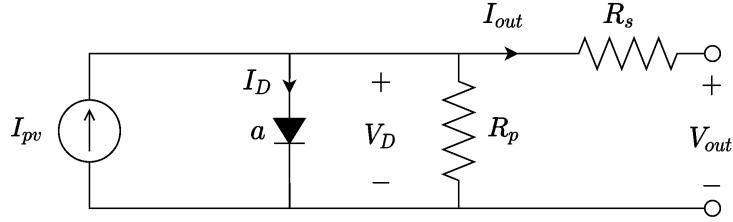


Figure 5.1: Single-diode model

Using KCL the output current of the model can be written as

$$I_{out} = I_{pv} - I_D - \frac{V_D}{R_p}, \quad (5.6)$$

where V_D represents the diode voltage. The diode current, I_D , is given by

$$I_D = I_o \left(e^{V_D/(aV_t)} - 1 \right), \quad (5.7)$$

where a is the ideality factor of the diode [9]. The thermal voltage, V_t , is given by

$$V_t = N_s \times \frac{kT_k}{q}. \quad (5.8)$$

N_s is the number of cells connected in series per array within the PV module, k is the Boltzmann constant, q is the electron charge, and T_k is the cell temperature in Kelvin. Finally, the output voltage can be written from figure 5.1 as

$$V_{out} = V_D - I_{out} R_s. \quad (5.9)$$

Five parameters, I_{pv} , I_o , R_s , R_p , and a must be calculated to solve the circuit model, where I_{pv} can be calculated as follows:

$$I_{pv} = (I_{scn} + k_i \Delta T) \left(\frac{G}{G_n} \right). \quad (5.10)$$

The reverse saturation current of the diode, I_o , can be computed using equation 5.11.

$$I_o = \frac{I_{pv}}{\exp[(V_{oc} + k_v \Delta T) / (aV_t)] - 1} \quad (5.11)$$

After I_{pv} and I_o have been calculated, R_s , R_p , and a remain unknown. An optimisation model is required since there are too many unknowns within the problem.

5.1.1 Optimisation Algorithm

To solve for the unknowns, a parameter extraction approach is followed. The genetic algorithm was chosen as the optimisation algorithm since it is a proven and tested algorithm, yielding acceptable accuracy with sufficient speed and simplicity. The generic approach for the genetic algorithm is depicted in figure 5.2 and is adapted from the approach followed in [31].

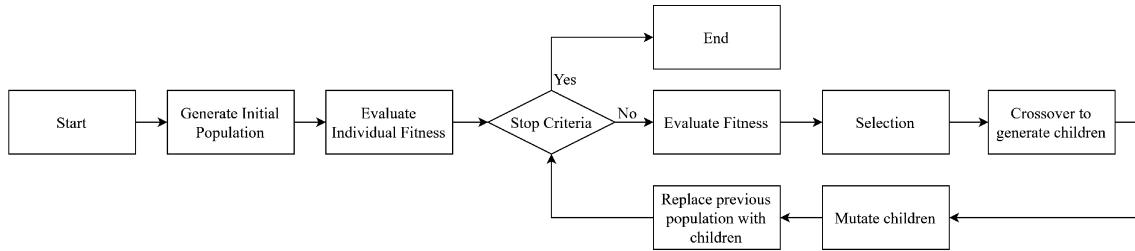


Figure 5.2: Genetic Algorithm flow diagram

Each of the steps of the Genetic Algorithm are explained in detail next, but the basic concept is as follows: Firstly, random solutions for the parameters are generated, whereafter the solutions are evaluated by means of a fitness function. The best solutions are then selected to be adapted slightly for the next generation. This is done by means of the evaluation, crossover and mutation steps. The solution then improves from generation to generation, since the best solutions are kept and modified, while the poor solutions are discarded.

Generate Initial Population

The genetic algorithm generates an initial generation of random solutions. This involves generating random values for R_p , R_s , and a within the solution boundaries. The initial population will be generated with the boundaries given by equations (5.12) to (5.14). Equation (5.12) yields the boundary for the PV module's series resistance, R_s , while (5.13) yields the boundaries for the parallel resistance, R_p . It is also well-known that the ideality factor of the diode typically ranges between 1 and 2, hence the boundary in 5.14.

$$0 \leq R_s \leq \frac{V_{oc} - V_{mp}}{I_{mp}} \quad (5.12)$$

$$\frac{V_{mp}}{(I_{sc} - I_{mp})} \leq R_p \leq \infty \quad (5.13)$$

$$1 \leq a \leq 2 \quad (5.14)$$

Figure 5.3 visualises the selected boundaries for an arbitrary PV module. The search areas of the genetic algorithm can now be seen clearly, indicated by the arrows. As from the equations above, the line from V_{mp} to V_{oc} is dictated by R_s , while the line between I_{sc} and I_{mp} is dictated by R_p .

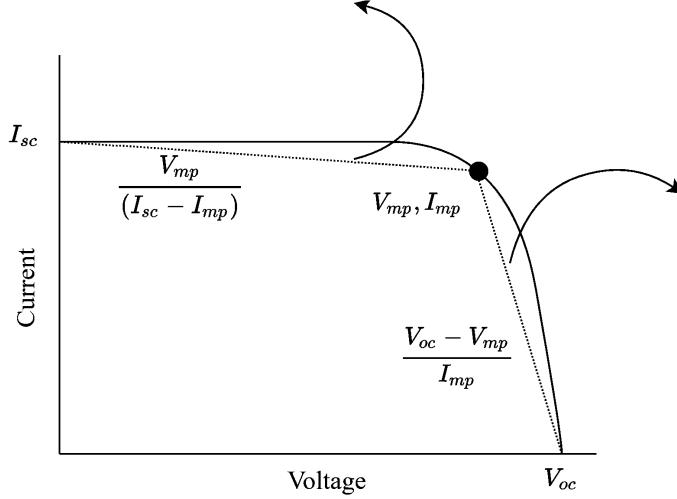


Figure 5.3: Visualised boundary conditions for initial population

Evaluate Individual Fitness

The fitness function's sole purpose is to assign a score to a solution, which can then be used to determine how good a solution is. Two fitness functions, f and g , are adapted from [13] and utilised in the genetic algorithm of the PV emulator. Equation (5.15) is rewritten from (5.10) and the maximum power point is substituted into the equation yielding the first fitness function, f .

$$f(R_s, R_p, a) = I_{ph} - I_o \exp\left(\frac{V_{mp} + I_{mp}R_s}{aV_t}\right) - \left(\frac{V_{mp} + I_{mp}R_s}{R_p}\right) - I_{mp} = 0 \quad (5.15)$$

The zero slope at the maximum power point, which is the second fitness function (g), can be written as

$$g(R_s, R_p, a) = \frac{dP}{dV} \Big|_{I_{mp}, V_{mp}} = I_{mp} - V_{mp} \frac{\Gamma/R_p}{1 + \left(\frac{R_s}{R_p}\right)\Gamma} = 0, \quad (5.16)$$

where Γ is given by

$$\Gamma = \left(\frac{I_{sc}R_p - V_{oc} + I_{sc}R_s}{aV_t} \right) \exp\left(\frac{V_{mp} + I_{mp}R_s - V_{oc}}{aV_t}\right) + 1. \quad (5.17)$$

The error function can then be constructed as the sum of the squares of (5.15) and (5.16) to yield (5.18). The sum of the squares yields drastically larger errors for the smallest of differences, allowing larger rewards for better performing solutions.

$$E(R_s, R_p, a) = f^2(R_s, R_p, a) + g^2(R_s, R_p, a) \quad (5.18)$$

The problem ultimately turns into a minimisation problem, since the slope at the maximum power point must be minimised, which easily be solved by means of the genetic algorithm. Its rather natural to evaluate fitness on the basis of larger is better, thus the fitness value calculated in the genetic algorithm is the inverse of the error obtained by (5.18).

Stop Criteria

The stop criteria of the algorithm is typically the tolerance of the desired solution or the maximum allowable number of generations. The tolerance is used as a stop criteria when a solution is found having an error less than the maximum allowable error.

Selection

The selection process selects parent solutions for the crossover and mutation steps of the algorithm. The most popular and preferred selection method is the tournament selection method. This method involves randomly selecting a solution and evaluation its score against a portion of the population. The best solution out of the portion of the population wins the tournament and is selected for the next steps of the algorithm by replacing all the solutions that it won against. The selection process ensures that the better solutions are selected for the next generations, creating a better solution after each iteration.

Crossover

The crossover step in the algorithm ensures that the solutions evolve. This imitates the reproduction process in nature, combining aspects of two solutions to create a new solution. Given two arbitrary binary values of 1011 and 1110, crossover is performed as follows. Both must be of the same length, whereafter a random point in the binary value is selected. Lets say this value is 3. This implies that the first binary value, 1011, will be split at its 3rd position, yielding 101 and 1. The same implies for the second binary value leaving 111 and 0. The two solutions are now combined. The 101 of 1011 goes to the 0 of 1110, and visa versa. This yields the crossover solution of 1010 and 1011. The crossover rate is the chance that two parents gets a child. Thus, some of the old solutions remain, allowing for some form of backtracking, but new solutions also exists, with evolution occurring. The crossover step is mostly performed with binary variables, thus the decimal values must be converted to binary, whereafter crossover and mutation is performed, and then converted back to decimal. The crossover rate will be selected using a sensitivity analysis, discussed in the implementation chapter.

Mutation

After the crossover step, mutation can now occur. Mutation is the way in which a new variation enters the population. As mentioned above, the mutation step also occurs in the binary state. Continuing the previous example, the crossover solution was 1010 and 1011. The mutation step has a mutation rate, which is the chance that mutation will occur. If the solution is selected for mutation, a bit is randomly altered. For example, if the first bit of the first solution, 1010, is now selected for mutation, the solution will change the variable to 0010, yielding a mutated solution. This step is responsible for drastically

changing the solution, allowing for the genetic algorithm to explore all directions. The mutation rate will be selected using a sensitivity analysis, discussed in the implementation chapter.

Replacement

The replacement step just involves overwriting the previous population with the new solution created by the selection, crossover, and mutation steps. After the replacement step, the process is repeated.

Parameter Selection

The input parameters to the genetic algorithm is the population size, maximum generations, crossover rate, mutation rate, and selection sample. All of these parameters are specific to the solution, thus no guidelines exists on selecting these parameters. During the implementation phase of the Genetic algorithm, a sensitivity analysis will be performed to determine the parameters. The main criteria of the solution is accuracy and speed. The accuracy is controlled by the tolerance, thus the solution will always be within its accuracy specification. Thus, the speed of the GA will be optimised during the selection of these parameters. The sensitivity analysis involves keeping all variables fixed and varying a single parameter while evaluating its effects on the speed of the algorithm.

After R_s , R_p , and a is obtained from the genetic algorithm, using (5.6), (5.7), and (5.9) we can write (5.19) to solve for the output current.

$$I_{out} = I_{pv} - I_o \left(e^{(V_{out} + I_{out}R_s)/(aV_t)} - 1 \right) - \frac{V_{out} + I_{out}R_s}{R_p} \quad (5.19)$$

It should be noted that (5.19) is an implicit equation, which complicates the solution for the equation. One of two methods can be used to solve this problem, but it should be kept in mind that both speed and accuracy is crucial. The first is to use a root finding algorithm to firstly solve for discrete points on the I-V curve, whereafter a high-order polynomial (an order of ten or higher) can be fitted in order to find a continuous solution for the I-V curve. However, this method is cumbersome and a much more elegant solution to the problem exists. The reason for the implicit nature of (5.19) is due to the diode equation that contains the output current in the exponential function. To solve this problem, the Lambert W function, named after its creator, Johann Heinrich Lambert was used. The Lambert W function is a multivalued function that aids in obtaining explicit solutions for equations containing implicit exponential functions. From [32], the explicit solution to (5.19) is given by

$$I_{out} = \left[\frac{R_p (I_{sc} + I_o) - V_{out}}{R_s + R_p} \right] - \left(\frac{aV_t}{R_s} \right) W \left[\left(\frac{R_s R_p I_o}{aV_t (R_s + R_p)} \right) * \exp \left[\frac{R_p (R_s I_{sc} + R_s I_o + V_{out})}{aV_t (R_s + R_p)} \right] \right], \quad (5.20)$$

where W is the shorthand for the Lambert W function. Now that a continuous function exists for the I-V curve, the intersection point between the I-V curve and the load line can easily be found. This is simply done by using the bisection root finding algorithm.

5.2 Power Converter

The power converter is a single-stage converter operating with an input voltage of 85 V, and a maximum output voltage of 70 V. The 85 V input voltage was selected to provide a 20% above the maximum desired output voltage, which implies that the maximum output voltage will be easily achievable in reality. A buck converter was selected as the power converter and its design follows.

5.2.1 Buck Converter

The buck converter's main function is to vary the output voltage in the range of 0 to 70 V for loads up to 800 W. The circuit must be designed for a minimum load in order to operate in continuous conduction mode (CCM), thus a minimum load of 5% is chosen. Typical minimum loads are in the range of 10%, but the problem at hand is not typical. Also, the input voltage to the circuit is a fixed value of 85 V, and the switching frequency is selected as 25 kHz, which is just outside of the audible frequency spectrum. Lastly, the duty cycle of the buck converter will be allowed to vary from 5% to 95%. The upper- and lower limits of the duty cycle exists, since in practice a point is reached where the switch can no longer switch on or off due to stray components present in the circuit. A summary of the buck converter design specifications is provided in table 5.3.

Table 5.3: Buck converter specifications

Parameter	Minimum	Fixed	Maximum
Input Voltage, V_{in}	-	85 V	-
Duty Cycle, k	0.05	-	0.95
Output Power, P_o	40 W	-	800 W
Switching Frequency, f_s	-	25 kHz	-

An asynchronous buck converter will be designed as determined in chapter 4. The target efficiency of the converter is 75%, as set out in appendix A, which is easily achievable with the asynchronous topology. The circuit diagram of an asynchronous buck converter, along with its non-ideal components, is depicted in figure 5.4.

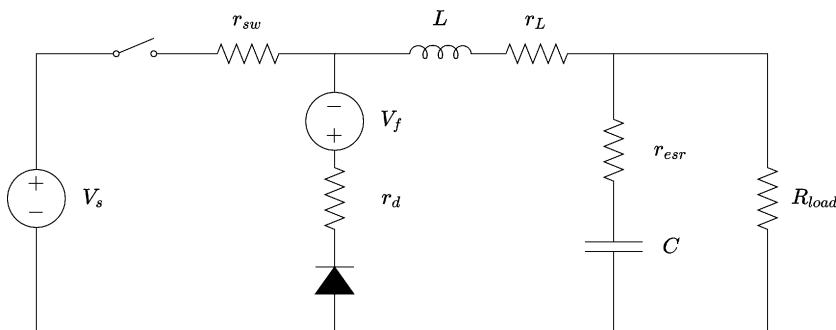


Figure 5.4: Buck-Converter with non-ideal components

The output voltage of a buck converter is given by

$$V_o = kV_{in}, \quad (5.21)$$

where k denotes the duty cycle, and V_{in} the input voltage. Using (5.21), the output voltage range is $4.25 \sim 80.75$ V, considering the duty cycle range. The converter will be designed for continuous conduction mode (CCM), implying that current will always flow in the inductor. The inductor value at which this condition is met is called the critical inductance, L_{crit} , and its equation is given by (5.22).

$$L_{crit} = \frac{V_o(1-k)}{\Delta i_L f_s} \quad (5.22)$$

The ripple current, Δi_L , at the critical inductance is given by

$$\Delta i_L = \frac{2P_{o,min}}{V_o}, \quad (5.23)$$

yielding $\Delta i_L = 18.82$ A at an output voltage of 4.25 V, and $\Delta i_L = 0.991$ A at an output voltage of 80.75 V, with $P_{o,min}$ being 40 W. Using 5.22, the inductor values for the upper and lower limit of the output voltages are determined as

$$L_{crit} = 8.58 \mu H; L_{crit} = 163.01 \mu H.$$

The largest inductor value must be selected, yielding an inductor of $L = 163.01 \mu H$ or larger. Next, the value for the output capacitance must be calculated. The capacitor value is given by

$$C_{out} = \frac{\Delta i_L}{8f_s \Delta v_L}, \quad (5.24)$$

where Δv_L is the maximum allowed voltage ripple. For a maximum voltage ripple of $\pm 0.5\%$, the voltage ripple is calculated as

$$\Delta v_L = 2 \times 0.005 \times V_{o,min} = 2 \times 0.005 \times 4.25 = 0.0425 \text{ V},$$

which yields the output capacitance as $C_{out} = 116.59 \mu F$.

Component Ratings

Next, it is important to determine the required ratings of the components to be selected. Each of the components will be rated with a safety factor of 50% included. The first component that will be selected is the inductor. The maximum allowable load current in the PVEM was chosen as 20 A, therefore a 200 uH, 20 A inductor is selected. The inductor current in a buck converter is depicted in figure 5.5.

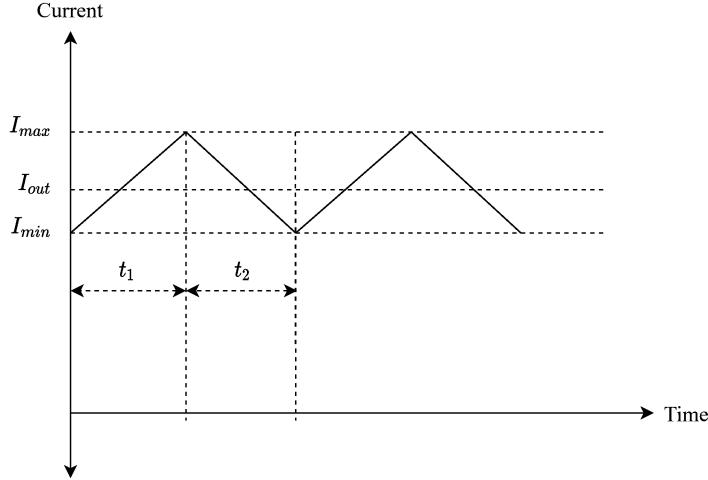


Figure 5.5: Inductor current

It can be seen that three currents are denoted; I_{max} , I_{out} , and I_{min} , where the output current, I_{out} , is the current that the inductor is rated for. The ripple current at an output current value of 20 A, and an output power of 800 W is given by

$$\Delta i_L = \frac{V_o(1-k)}{L f_s} = \frac{\frac{800}{20} \left(1 - \frac{800/20}{85}\right)}{200\mu(25k)} = 4.2353 \text{ A.}$$

The largest values of I_{max} and I_{min} can now be calculated as shown below.

$$I_{max} = I_{out} + \frac{\Delta i_L}{2} = 20 + \frac{4.2353}{2} = 22.118 \text{ A}$$

$$I_{min} = I_{out} - \frac{\Delta i_L}{2} = 20 - \frac{4.2353}{2} = 17.88235 \text{ A}$$

Figure 5.6 depicts the diode voltage in the buck converter. It can be seen that the absolute maximum voltage across the diode is V_{in} , which implies that the voltage rating of the diode should be at least 127.5 V. From figure 5.7, it can be seen that the maximum diode current is I_{max} , implying that the diode current should be rated for 33.177 A.

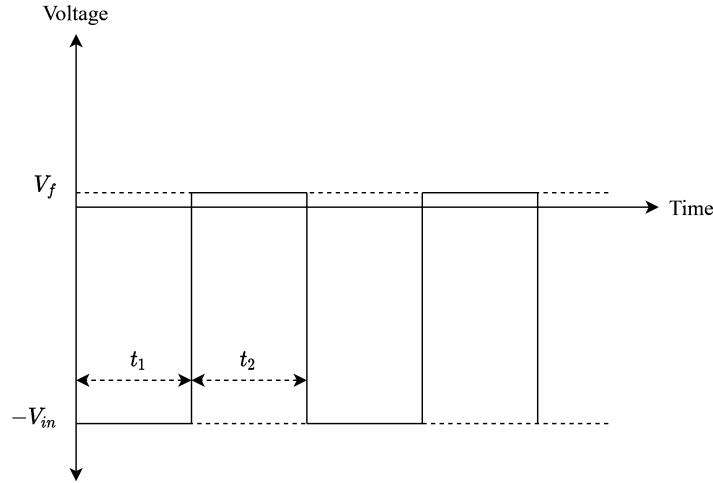


Figure 5.6: Diode voltage

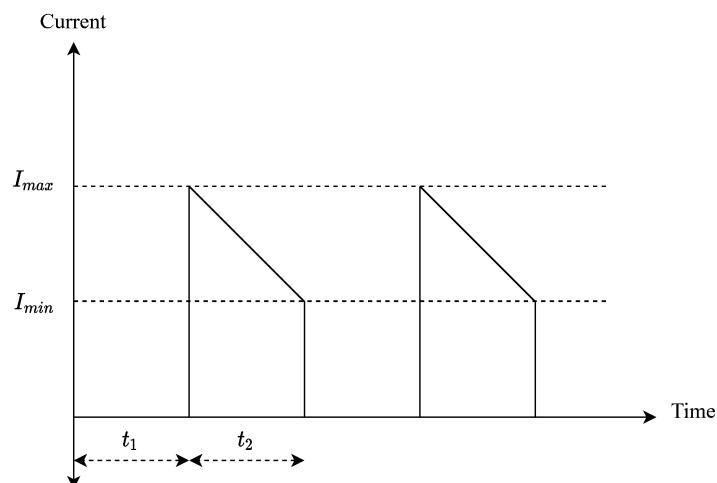


Figure 5.7: Diode current

The output capacitor current can be seen in figure 5.8. The values of $I_{Co,max}$ and $I_{Co,min}$ are calculated as follows.

$$I_{Co,max} = I_{max} - I_{out} = 22.118 - 20 = 2.118 \text{ A}$$

$$I_{Co,min} = I_{min} - I_{out} = 17.88235 - 20 = -2.118 \text{ A}$$

The preceding calculations imply that the capacitor must be able to withstand a ripple current of 2.118 A. Typically more than one capacitor is connected in parallel since a single capacitor cannot

withstand the ripple current. The capacitor voltage must be rated with respect to the maximum output voltage, implying a required voltage rating of at least $121.125 V_{dc}$.

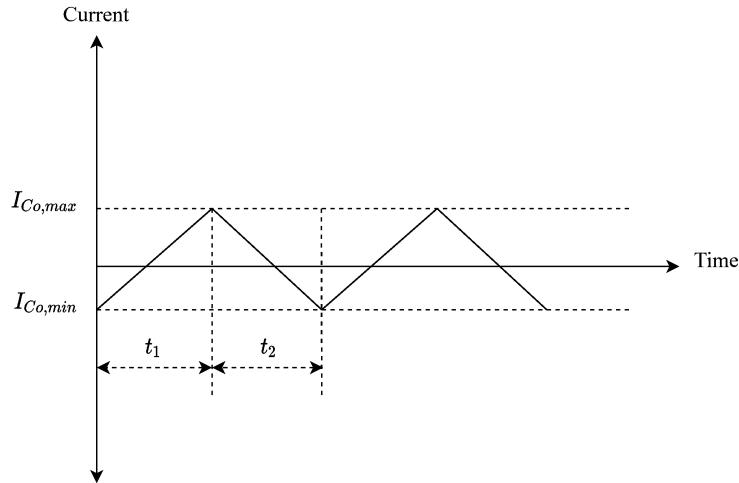


Figure 5.8: Capacitor current

Next, the ratings of the switch can be obtained. The selected switch type is an IGBT, due to the high power capabilities of an IGBT and the relatively low switching frequency. The IGBT will be on the high-side, implying that the maximum voltage is across the switch will be the sum of the input voltage and diode forward voltage. If the diode forward voltage is estimated as $1.7 V$, the IGBT must be rated at a voltage of $123.675 V$. The peak current through the IGBT is I_{max} , yielding a required rating of $33.177 A$.

Lastly, the input capacitor must be selected. Equation (5.25) allows for the calculation of the required input capacitor.

$$C_{IN} \geq \frac{k(1-k) \times I_o}{\Delta V_{IN_PP} \times f_{sw}} \quad (5.25)$$

The value for the peak to peak input ripple voltage is chosen as 0.5% of $85 V$, while the duty cycle is chosen as 0.47 , yielding an output current of $20 A$ at $800 W$. This gives the worst case required capacitor required as

$$C_{IN} \geq \frac{0.47(1 - 0.47) \times 20}{(0.425)(25k)} = 468.9 \mu F.$$

Selected Components

The components were selected to meet the required ratings at the lowest cost. Thus, some of the ratings are higher than required, but the cost of the component was lower in these cases. Also, apart from cost, the availability of some of the semiconductors are very low, which also greatly influenced the selected components. The final selected components for the buck converter are summarised in table 5.4.

Table 5.4: Selected Components for buck converter

Component	Code	Rating	Manufacturer
IGBT	GT50JR21	600 V, 50 A	Toshiba
Diode	FFH50US60S	600 V, 50 A	Onsemi
Inductor	NAC-20-0201	200 uH, 20 A	Tamura
Output Capacitor (Two in parallel)	250BXC68MEFC16X20	68 uF, 250 V _{dc} , 1.3 A	Rubycon

It should be noted that the inductor was measured at a frequency of 20 kHz, while the circuits switching frequency is 25 kHz. The inductance will be slightly smaller at 25 kHz, but the critical inductance was calculated as 163 uH, which implies that the 200 uH inductor will suffice.

Driver Circuitry

To drive the high-side switch of the buck converter, a high-side gate driver will be utilised. Specifically, the IRS2117 IC from Infineon is used as it is a 600 V, single input high-side driver with bootstrap operation. Also, the output of the driver is in phase with the output, simplifying the control. The IRS2117 requires a bootstrap diode and capacitor to function properly, with equation (5.26) giving the required bootstrap capacitor size. Equation (5.26) is obtained from the Infineon, along with the typical value for ΔV_{BS} , which is 1 V. The value of the total gate-charge, Q_{GTOT} , must be calculated from the selected IGBT's datasheet. Equation (5.27) is used to calculate the value of Q_{GTOT} .

$$C_{boot} \geq \frac{Q_{GTOT}}{\Delta V_{BS}} \quad (5.26)$$

$$Q_{GTOT} \approx 2 \times \left(Q_G + Q_{LS} + (I_{QBS}) \times D_{max} \times \frac{1}{f_s} \right) \quad (5.27)$$

Q_G represents the required turn-on gate charge of the IGBT, Q_{LS} is the level shift charge required per cycle which is typically 5 nC, and I_{QBS} is the maximum floating section quiescent current. Obtaining these values from the datasheet of the GT50JR21 yields a required bootstrap capacitor value of

$$C_{boot} \geq 316.24 \text{ nF.}$$

The bootstrap capacitor must be a capacitor with a very low ESR (to reduce losses), and must be placed as close as possible to the IRS2117 to reduce the voltage spikes which can trigger the undervoltage lockout threshold of the high side driver section. A 1 uF polyester film capacitor will be utilised. The voltage of the IRS2117 will be 12 V, which is consequently the gate-drive voltage.

To feed the input of the high-side gate driver, a microcontroller connected to an optocoupler will be used. Since the required logic '1' voltage of the IRS2117 is rather high at 9.5 V, a high current-transfer ratio (CTR) optocoupler is required. A very commonly used optocoupler for these type of applications is the 61N36 from Vishay. The optocoupler will be configured in a non-inverting manner, implying that the voltage fed to the gate-driver is the same as the PWM signal of the microcontroller. The gate resistor is selected as 22 Ω from the datasheet of the IRS2117. Also, a reverse diode is placed across the gate resistor to ensure that the internal capacitances of the IGBT discharge swiftly, due to lower resistance during turn-off.

Circuit Schematic

The schematic of the final designed circuit is displayed in figure 5.9.

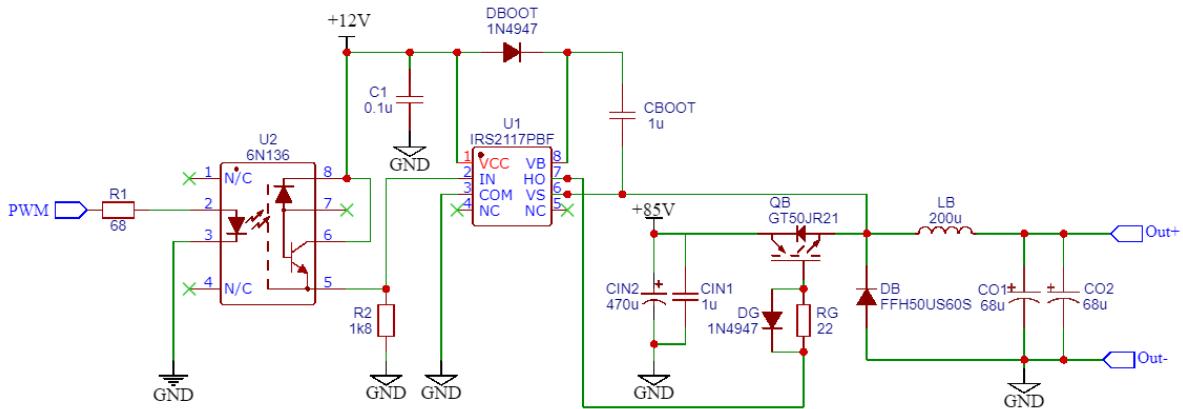


Figure 5.9: Buck converter schematic

Simulation

The schematic of the simulated buck converter is shown in figure 5.10. All non-ideal components are included within the simulation, and it can be seen that the 6N136 optocoupler and IRS2117 gate-driver is also included. The IGBT used in the simulation is a 50 A, 650 V IGBT, since this was the closest model to the selected GT50JR21 IGBT that could be obtained.

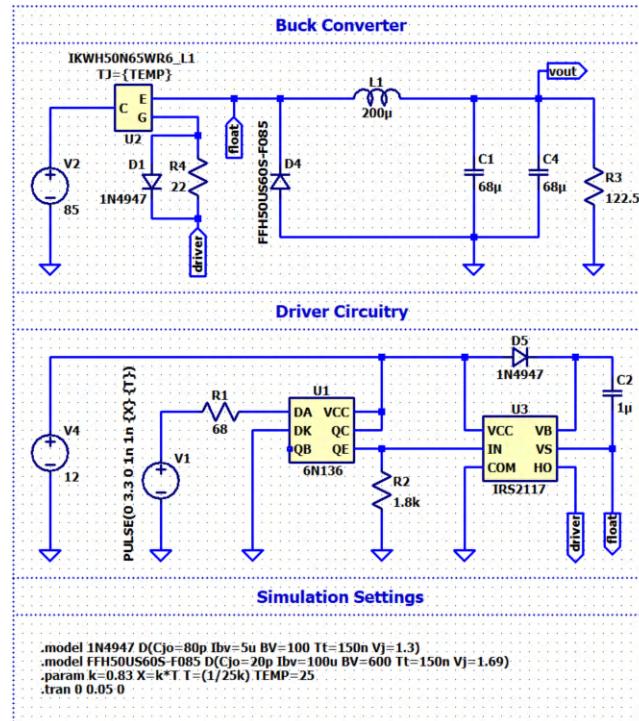


Figure 5.10: Simulated buck converter schematic

Figure 5.11 depicts the driver circuitry waveforms. It can be seen that the optocoupler input switches and the signal is transferred to the output of the optocoupler as a 12 V PWM signal. Thereafter, the output of the optocoupler serves as the input to the gate-driver which in turn produces a gate voltage signal with high-current capabilities.

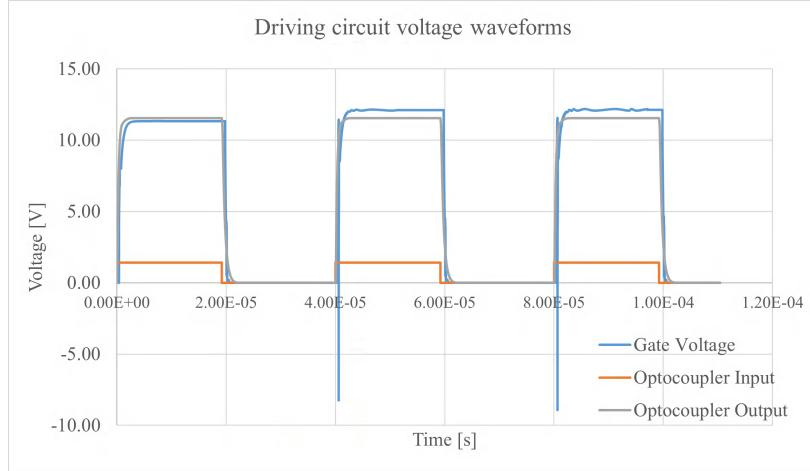


Figure 5.11: Driver circuitry waveforms

The circuit performs as expected since the switching signal successfully switches the IGBT to produce the output voltage and inductor current given by figures 5.12a and 5.12b, respectively.

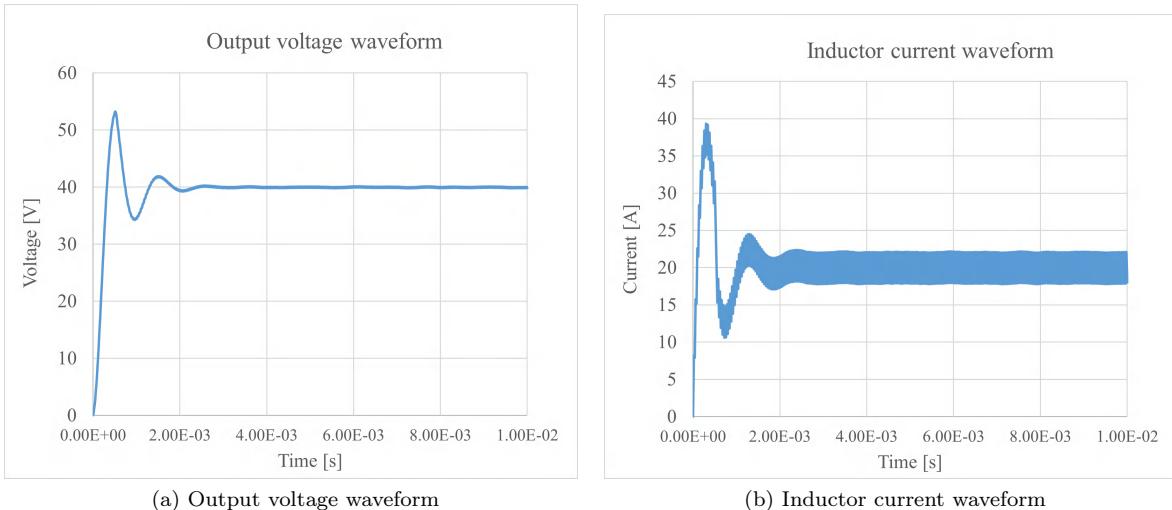


Figure 5.12: Output voltage and inductor current waveforms

Figure 5.13 depicts the inductor current at minimum load. This is the condition for which the critical inductance was designed in order to ensure that a current will always flow within the inductor. Its clear that the buck converter operates in CCM.

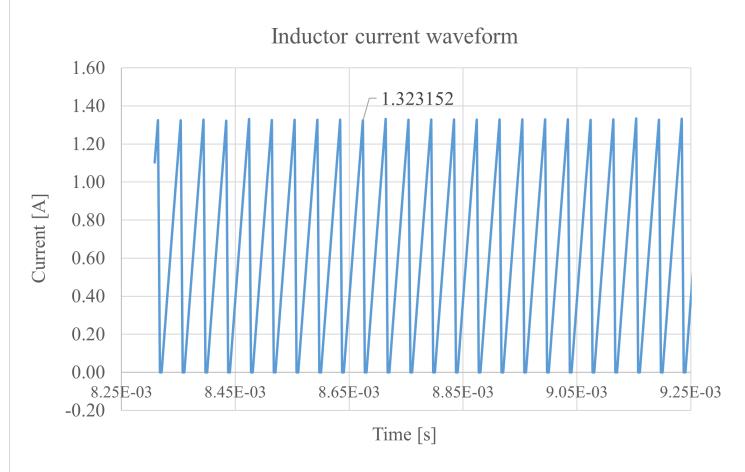


Figure 5.13: Inductor current at minimum load

Also from figure 5.13, the peak value of the current is 1.323 A, yielding a ripple current of 0.662 A, which is less than the estimated value of 0.991 A. This is expected due to the larger inductance added, enabling the circuit to perform better. Figure 5.14 displays the output voltage, where the ripple can be calculated as $\pm 0.4753\%$, which is within the designed value of $\pm 0.5\%$.

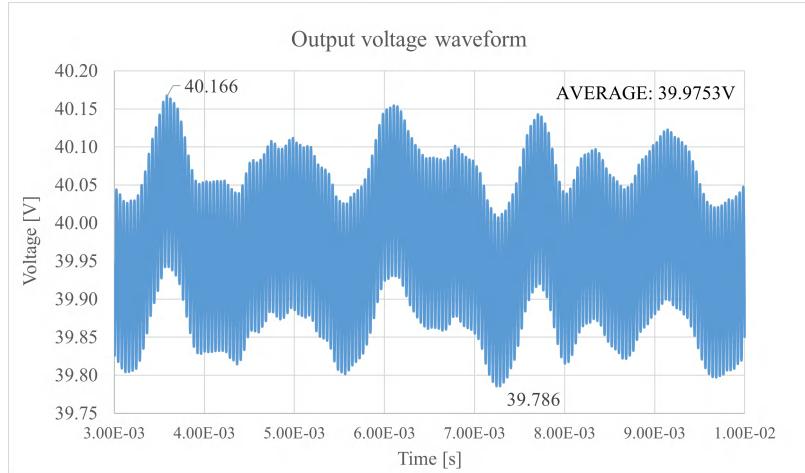


Figure 5.14: Output ripple voltage

The expected efficiency of the buck converter was also determined by means of simulation. As mentioned previously, the simulation included all non-ideal components, such as the inductor DC resistance, the capacitor ESR, the diode and switch resistances, and the driver IC losses. The efficiency at maximum load current is

$$\eta = \frac{P_{out}}{P_{in}} = \frac{790.96}{842.31} \times 100 = 93.90\%,$$

while at minimum load current the efficiency of the converter is

$$\eta = \frac{P_{out}}{P_{in}} = \frac{47.832}{60.033} \times 100 = 79.67\%.$$

It can be seen that both values are above the desired 75% value, implying a satisfactory design.

5.2.2 Sensing Circuit

The sensing circuitry will comprise of two voltage sensors. The current sensing will be inferred by means of a sensing resistor and amplifier network. All of the feedback circuits must pass through an isolation barrier to ensure that the microcontroller does not damage.

Three main methods exists for the isolated sensing of voltages. The first is known as an isolated amplifier which is an isolated operational amplifier. It was found that the isolated amplifier IC's are quite hard to obtain. The second is by means of a voltage to frequency converter in combination with an optocoupler. This method converts the measured voltage to a PWM signal with a frequency specific to the measured voltage, whereafter it gets passed through an optocoupler for a microcontroller to translated the measured frequency to the measured voltage. This method requires a lot of conversions and becomes quite cumbersome to implement. The third and simplest solution, and the one which will be employed in all of the PVEM sensing circuits, is a linear optocoupler. An optocoupler is known for its non-linearities and the linear optocoupler compensates for these non-linearities. By using a linear optocoupler, a linear relationship between the measured voltage and the voltage on the opposite side of the isolation barrier can be created. This allows for painless isolated measurement of voltages.

The IC that will be used is the HCNR200 manufactured by Broadcom. Its non-linearity is approximately 0.01%, allowing for very accurate measurements. The linear optocoupler is now used in conjunction with operational amplifiers to construct an isolated amplifier. A typical voltage measurement circuit can be seen in figure 5.15. The resistors R1 and R2 should be selected in such a way that the output voltage range is suitable for the application. The output voltage of the circuit is given by

$$V_{out} = \frac{R_2}{R_1} V_{in}. \quad (5.28)$$

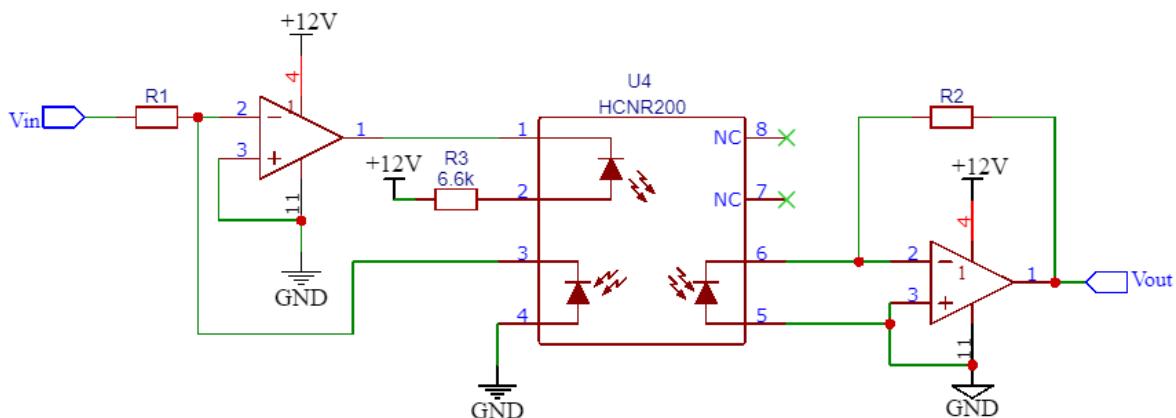


Figure 5.15: HCNR200 sensing circuit

The parameters that will be measured are the buck converter output voltage and current. Therefore, two HCNR200 circuits will be required to measure each of the signals specified above. Each of the measurement circuits will have the configuration of figure 5.15, with appropriate values of R1 and R2 selected for the specific measurement circuit.

5.3 Control Strategy

To design the controller for the buck converter, the transfer function of the buck converter must first be derived. The derivation of the transfer function is not a straight forward task, due to the switching nature of the buck converter, and the process is displayed below.

5.3.1 State-Space Average Model Derivation

The circuit diagram of the buck-converter containing non-ideal components is shown in figure 5.16. The buck converter has two operating modes; when the switch is closed, and when the switch is open.

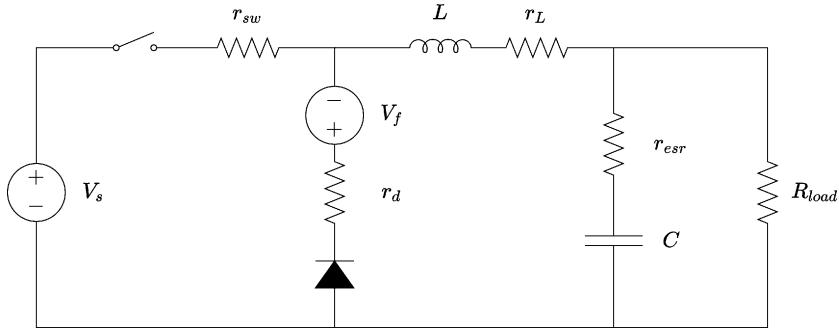


Figure 5.16: Buck-Converter with non-ideal components

The equations for the inductor current and capacitor voltages are derived below for the closed switch mode, depicted in figure 5.17. The inductor voltage is given by

$$v_L(t) = L \frac{di_L(t)}{dt} = V_s(t) - i_L(t)[r_{sw} + r_L] - V_o(t), \quad (5.29)$$

while the output voltage of the buck converter is given (5.30).

$$V_o(t) = v_C(t) + r_{esr} \left[i_L(t) - \frac{V_o(t)}{R_{load}} \right] \quad (5.30)$$

The output voltage can then be rewritten to form (5.31).

$$V_o(t) = v_C(t) \frac{R_{load}}{R_{load} + r_{esr}} + i_L(t) \frac{R_{load}r_{esr}}{R_{load} + r_{esr}} \quad (5.31)$$

By substituting (5.31) into (5.29) the inductor voltage is rewritten to form

$$v_L(t) = L \frac{di_L(t)}{dt} = V_s(t) - i_L(t) \left[r_{sw} + r_L + \frac{r_{esr}R_{load}}{R_{load} + r_{esr}} \right] - v_C(t) \frac{R_{load}}{R_{load} + r_{esr}}. \quad (5.32)$$

Lastly, the capacitor current is given by (5.33).

$$i_C(t) = C \frac{dv_C(t)}{dt} = i_L(t) - \frac{V_o(t)}{R_{load}} = i_L(t) \frac{R_{load}}{R_{load} + r_{esr}} - v_C(t) \frac{1}{R_{load} + r_{esr}} \quad (5.33)$$

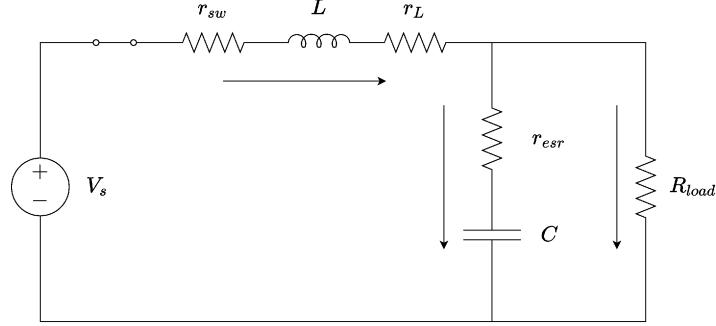


Figure 5.17: Buck-Converter Mode 1

During the second mode, the switch of the buck converter is open as shown in figure 5.18. The inductor voltage for this mode is given by

$$v_L(t) = L \frac{di_L(t)}{dt} = -i_L(t) \left(r_d + r_L + \frac{r_{esr} R_{load}}{R_{load} + r_{esr}} \right) - V_f - v_C(t) \frac{R_{load}}{R_{load} + r_{esr}}. \quad (5.34)$$

The capacitor current can be written as shown by (5.35), while the output voltage is given by (5.36).

$$i_C(t) = C \frac{dv_C(t)}{dt} = i_L(t) \frac{R_{load}}{R_{load} + r_{esr}} - v_C(t) \frac{1}{R_{load} + r_{esr}} \quad (5.35)$$

$$V_o(t) = v_C(t) \frac{R_{load}}{R_{load} + r_{esr}} + i_L(t) \frac{r_{esr} R_{load}}{R_{load} + r_{esr}} \quad (5.36)$$

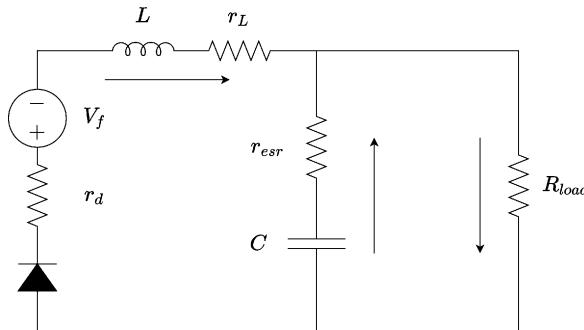


Figure 5.18: Buck-Converter Mode 2

The state-space model of the buck-converter is derived in the following subsections. To derive the state-space model, the state-space averaging (SSA) method will be used, as from [33] and [34]. The input for the system is chosen as the input voltage V_S , while the outputs are the inductor current, i_L , and the capacitor voltage, v_C .

5.3.2 On-time state-space model

Equations (5.29) through (5.33) can be used to determine the state-space model of the buck-converter during the on-time (switch-closed) as

$$\dot{x}(t) = \mathbf{A}_{on}x(t) + \mathbf{B}_{on}V_s(t) + \mathbf{C}_{on}V_f(t), \quad (5.37)$$

where

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_C(t) \end{bmatrix}. \quad (5.38)$$

The matrices \mathbf{A}_{on} , \mathbf{B}_{on} , and \mathbf{C}_{on} are given by (5.39), (5.40) and (5.41), respectively.

$$\mathbf{A}_{on} = \begin{bmatrix} -\frac{1}{L} \left(r_{sw} + r_L + \frac{r_{esr} R_{load}}{R_{load} + r_{esr}} \right) & -\frac{1}{L} \left(\frac{R_{load}}{R_{load} + r_{esr}} \right) \\ \frac{1}{C} \left(\frac{R_{load}}{R_{load} + r_{esr}} \right) & -\frac{1}{C} \left(\frac{1}{R_{load} + r_{esr}} \right) \end{bmatrix} \quad (5.39)$$

$$\mathbf{B}_{on} = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \quad (5.40)$$

$$\mathbf{C}_{on} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.41)$$

The output voltage of the on-time model is given by

$$V_o(t) = \mathbf{D}_{on} \mathbf{x}(t), \quad (5.42)$$

where

$$\mathbf{D}_{on} = \begin{bmatrix} \left(\frac{R_{load} r_{esr}}{R_{load} + r_{esr}} \right) \\ \left(\frac{R_{load}}{R_{load} + r_{esr}} \right) \end{bmatrix}. \quad (5.43)$$

5.3.3 Off-time state-space model

Equations (5.34) through (5.36) can be used to determine the state-space model of the buck-converter during the off-time (switch-open) as

$$\dot{\mathbf{x}}(t) = \mathbf{A}_{off} \mathbf{x}(t) + \mathbf{B}_{off} V_s(t) \mathbf{C}_{off} V_F, \quad (5.44)$$

where

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_C(t) \end{bmatrix} \quad (5.45)$$

The matrices \mathbf{A}_{on} , \mathbf{B}_{on} , and \mathbf{C}_{on} are given by (5.46), (5.47) and (5.48), respectively.

$$\mathbf{A}_{off} = \begin{bmatrix} -\frac{1}{L} \left(r_d + r_L + \frac{r_{esr} R_{load}}{R_{load} + r_{esr}} \right) & -\frac{1}{L} \left(\frac{R_{load}}{R_{load} + r_{esr}} \right) \\ \frac{1}{C} \left(\frac{R_{load}}{R_{load} + r_{esr}} \right) & -\frac{1}{C} \left(\frac{1}{R_{load} + r_{esr}} \right) \end{bmatrix} \quad (5.46)$$

$$\mathbf{B}_{off} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.47)$$

$$\mathbf{C}_{off} = \begin{bmatrix} -\frac{1}{L} \\ 0 \end{bmatrix} \quad (5.48)$$

The output voltage of the on-time model is given by

$$V_o(t) = \mathbf{D}_{off}\mathbf{x}(t), \quad (5.49)$$

where

$$\mathbf{D}_{off} = \begin{bmatrix} \left(\frac{R_{load}r_{esr}}{R_{load} + r_{esr}} \right) \\ \left(\frac{R_{load}}{R_{load} + r_{esr}} \right) \end{bmatrix}. \quad (5.50)$$

5.3.4 State-Space Average Model (SSAM)

The averaged state-space model can be obtained by multiplying the individual state-space models by the portion they are present in the switching cycle. Therefore the on-time state-space model can be multiplied by the duty cycle, k , and the off-time model can be multiplied by $k' = 1 - k$. The sum of the on-time and off-time state-space models then yield the SSAM as the set of equations (5.51) and (5.52).

$$\dot{\mathbf{x}}(t) = [k\mathbf{A}_{on} + k'\mathbf{A}_{off}]\mathbf{x}(t) + [k\mathbf{B}_{on} + k'\mathbf{B}_{off}]V_s(t) + [k\mathbf{C}_{on} + k'\mathbf{C}_{off}]V_f \quad (5.51)$$

$$V_o(t) = [k\mathbf{D}_{on} + k'\mathbf{D}_{off}]\mathbf{x}(t) \quad (5.52)$$

5.3.5 Small-Signal Behaviour

The equations obtained in section 5.3.4 need to be linearised. Linearisation using the perturbation technique is used here as it serves as a robust method for linearising non-linear equations. For the small-signal behaviour, perturbations are introduced as shown by equations (5.53) through (5.55), where $\hat{\cdot}$ denotes the perturbation, while the capital letter symbols denote the DC operating point.

$$\mathbf{x}(t) = \mathbf{X} + \hat{\mathbf{x}}(t) \quad (5.53)$$

$$V_o(t) = V_o + \hat{v}_o(t) \quad (5.54)$$

$$k(t) = K + \hat{k}(t) \quad (5.55)$$

It is assumed that no perturbations occur in the input voltage, V_s , since this assumption is needed to simplify the derivation process of the transfer function. Now substituting (5.53) to (5.55) into the SSAM, we obtain the model as the set formed by (5.56) and (5.57).

$$\frac{d[\mathbf{X} + \hat{\mathbf{x}}(t)]}{dt} = \left[\left(K + \hat{k}(t) \right) \mathbf{A}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{A}_{off} \right] [\mathbf{X} + \hat{\mathbf{x}}(t)] + \\ \left[\left(K + \hat{k}(t) \right) \mathbf{B}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{B}_{off} \right] V_s(t) + \left[\left(K + \hat{k}(t) \right) \mathbf{C}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{C}_{off} \right] V_f \quad (5.56)$$

$$V_o + \hat{v}_o(t) = \left[\left(K + \hat{k}(t) \right) \mathbf{D}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{D}_{off} \right] [\mathbf{X} + \hat{\mathbf{x}}(t)] \quad (5.57)$$

The core concept of the small-signal analysis is that the derivative of the steady-state component $\frac{d\mathbf{X}}{dt}$ is zero, yielding

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = \left[\left(K + \hat{k}(t) \right) \mathbf{A}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{A}_{off} \right] [\mathbf{X} + \hat{\mathbf{x}}(t)] + \\ \left[\left(K + \hat{k}(t) \right) \mathbf{B}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{B}_{off} \right] V_s(t) + \left[\left(K + \hat{k}(t) \right) \mathbf{C}_{on} + \left(K' + \hat{k}'(t) \right) \mathbf{C}_{off} \right] V_f. \quad (5.58)$$

Now simplifying we obtain

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = \left[K(\mathbf{A}_{on} - \mathbf{A}_{off}) + \hat{k}(\mathbf{A}_{on} - \mathbf{A}_{off}) + 2\mathbf{A}_{off} \right] [\mathbf{X} + \hat{\mathbf{x}}(t)] + \\ \left[K(\mathbf{B}_{on} - \mathbf{B}_{off}) + \hat{k}(\mathbf{B}_{on} - \mathbf{B}_{off}) + 2\mathbf{B}_{off} \right] V_s(t) + \left[K(\mathbf{C}_{on} - \mathbf{C}_{off}) + \hat{k}(\mathbf{C}_{on} - \mathbf{C}_{off}) + 2\mathbf{C}_{off} \right] V_f. \quad (5.59)$$

To simplify the equation above even further, we can write the following

$$\mathbf{A} = \mathbf{A}_{on} - \mathbf{A}_{off} \quad (5.60)$$

$$\mathbf{B} = \mathbf{B}_{on} - \mathbf{B}_{off} \quad (5.61)$$

$$\mathbf{C} = \mathbf{C}_{on} - \mathbf{C}_{off} \quad (5.62)$$

$$\mathbf{D} = \mathbf{D}_{on} - \mathbf{D}_{off} \quad (5.63)$$

to yield the simplified mode as the set of (5.64) and (5.65).

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = (2\mathbf{A}_{off}\mathbf{X} + 2\mathbf{C}_{off}V_f + 2\mathbf{B}_{off}V_s) + \{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\} K + \\ [2\mathbf{A}_{off} + \mathbf{AK}] \hat{\mathbf{x}}(t) + \{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\} \hat{k} + \mathbf{A}\hat{\mathbf{x}}(t) \hat{k} \quad (5.64)$$

$$V_o + \hat{v}_o(t) = 2\mathbf{D}_{off}\mathbf{X} + [\mathbf{DX}] K + [K\mathbf{D} + 2\mathbf{D}_{off}] \hat{\mathbf{x}}(t) + \mathbf{DX}\hat{k} + \mathbf{D}\hat{\mathbf{x}}(t) \hat{k} \quad (5.65)$$

The assumption can be made that the the departures of the states variables from their corresponding

steady-state values are small, which then yields the final averaging model as the set of (5.66) and (5.67).

$$0 = (2\mathbf{A}_{off}\mathbf{X} + 2\mathbf{C}_{off}V_f + 2\mathbf{B}_{off}V_s) + \{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\}K \quad (5.66)$$

$$V_o = 2\mathbf{D}_{off}\mathbf{X} + [\mathbf{DX}]K \quad (5.67)$$

The AC model of the system is given by the set of (5.68) and (5.69).

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = [2\mathbf{A}_{off} + \mathbf{AK}]\hat{\mathbf{x}}(t) + \{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\}\hat{k} \quad (5.68)$$

$$\hat{v}_o(t) = [K\mathbf{D} + 2\mathbf{D}_{off}]\hat{\mathbf{x}}(t) + \mathbf{DX}\hat{k} \quad (5.69)$$

5.3.6 Transfer Function Extraction

The transfer functions of the system can finally be created. This is done by taking the Laplace transform of (5.68) and (5.69) to yield (5.70) and (5.71), respectively.

$$s\hat{\mathbf{x}}(s) = [2\mathbf{A}_{off} + \mathbf{AK}]\hat{\mathbf{x}}(s) + \{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\}\hat{k}(s) \quad (5.70)$$

$$\hat{v}_o(s) = [K\mathbf{D} + 2\mathbf{D}_{off}]\hat{\mathbf{x}}(s) + \mathbf{DX}\hat{k}(s) \quad (5.71)$$

The $\hat{\mathbf{x}}(s)$ term can be grouped to form

$$\hat{\mathbf{x}}(s) = \{sI - (2\mathbf{A}_{off} + \mathbf{AK})\}^{-1}\{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\}\hat{k}(s), \quad (5.72)$$

which can then be substituted into (5.71) to form

$$\hat{v}_o(s) = [K\mathbf{D} + 2\mathbf{D}_{off}]\{sI - (2\mathbf{A}_{off} + \mathbf{AK})\}^{-1}\{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\}\hat{k}(s) + \mathbf{DX}\hat{k}(s). \quad (5.73)$$

Equation (5.73) can be rewritten to form (5.74).

$$\frac{\hat{v}_o(s)}{\hat{k}(s)} = [\mathbf{D}_{on}K + \mathbf{D}_{off}(1-K)]\{sI - (\mathbf{A}_{on}K + \mathbf{A}_{off}(1-K))\}^{-1}\{\mathbf{AX} + \mathbf{BV}_s(t) + \mathbf{CV}_f\} + \mathbf{DX} \quad (5.74)$$

Finally, the transfer function can be extracted by substituting the respective matrices into (5.74), which then yields the system transfer function as

$$\frac{\hat{v}_o(s)}{\hat{k}(s)} = \frac{\frac{R_{load}r_{esr}(V_f + V_s + v_x)}{L(R_{load} + r_{esr})}s + \frac{R_{load}(V_f + V_s + v_x)}{CL(R_{load} + r_{esr})}}{s^2 + \left(\left(\frac{1}{C}\right)\left(\frac{1}{(R_{load} + r_{esr})}\right) + \left(\frac{1}{L}\right)\left(r_L + r_x + \frac{R_{load}r_{esr}}{(R_{load} + r_{esr})}\right)\right)s + \frac{R_{load} + r_L + r_x}{CL(R_{load} + r_{esr})}}, \quad (5.75)$$

The values of r_x and v_x are given by (5.76) and (5.77), respectively. The parameter r_x is a resistance dependant on the duty cycle, D , and v_x is a voltage dependant on the load current.

$$r_x = Dr_{sw} + (1 - D)r_d \quad (5.76)$$

$$v_x = I_{load} (r_d - r_{sw}) \quad (5.77)$$

Table 5.5 describes the variables in equations (5.75) to (5.77). These equations will now be used to design the control strategy of the buck-converter.

Table 5.5: Transfer function variable descriptions

Parameter	Symbol	Unit
Supply voltage	V_s	V
Inductance	L	H
Output capacitance	C	F
Load Resistance	R_{load}	Ω
Diode forward voltage	V_f	V
Capacitor ESR	r_{esr}	Ω
Inductor DC resistance	r_L	Ω
Duty cycle	D	—
Load current	I_{load}	A
Switch on-resistance	r_{sw}	Ω
Diode on-resistance	r_d	Ω

5.3.7 Open-loop comparison

In order to determine to which extent the derived transfer function represents the circuit, the open-loop step responses were compared. Figures 5.19 and 5.20 depict the comparison of the output voltage waveforms.

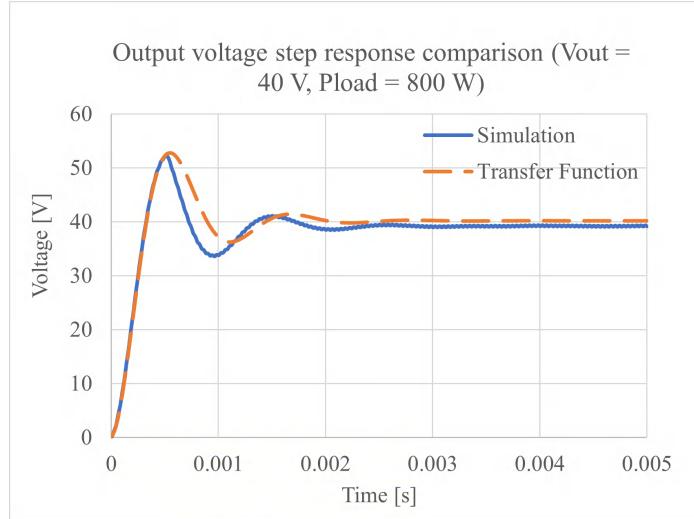


Figure 5.19: Open loop step response comparison for a 800 W load and an output voltage of 40 V

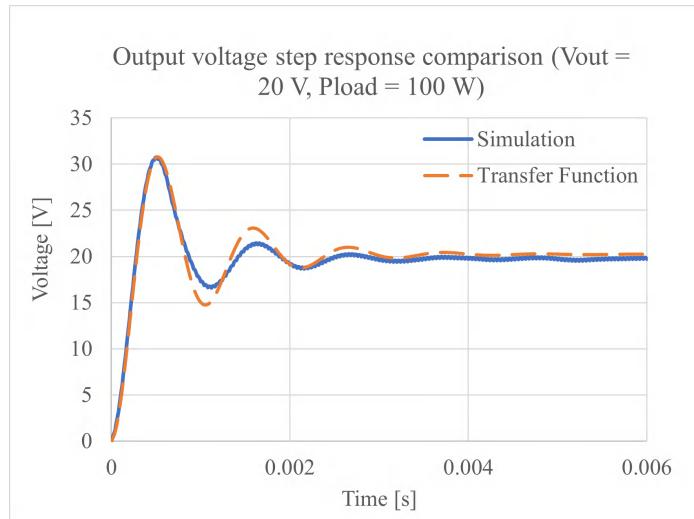


Figure 5.20: Open loop step response comparison for a 100 W load and an output voltage of 20 V

It's clear from figures 5.19 and 5.20, that the peak values and settling times are the same. The only difference is a slight deviation in the transient response between the peak time and settling time. This will have little effect on the controller design, since the overall response is the same. The differences can be credited to the additional stray components contained in the simulation, which are not included in the transfer functions.

It is important to be weary of the transfer function limitations. Specifically, the transfer functions does not consider the saturation of the output voltage, which is the supply voltage, and large differences occur at larger voltages. For example, figures 5.21 and 5.22 displays the large inaccuracy of the transfer function when a 40 W and 800 W load is connected to an output voltage of 70 V , respectively. However, its clear that the settling time still remains the same in the comparison, implying that the transfer functions can still be used for the design. In fact, the overshoot is more drastic in the transfer functions, causing a worst case design which should allow the controller to perform better in practice.

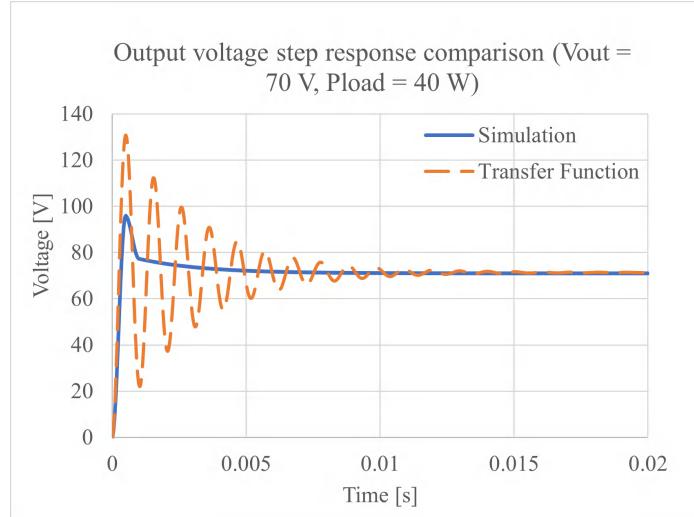


Figure 5.21: Open loop step response comparison for a 40 W load and an output voltage of 70 V

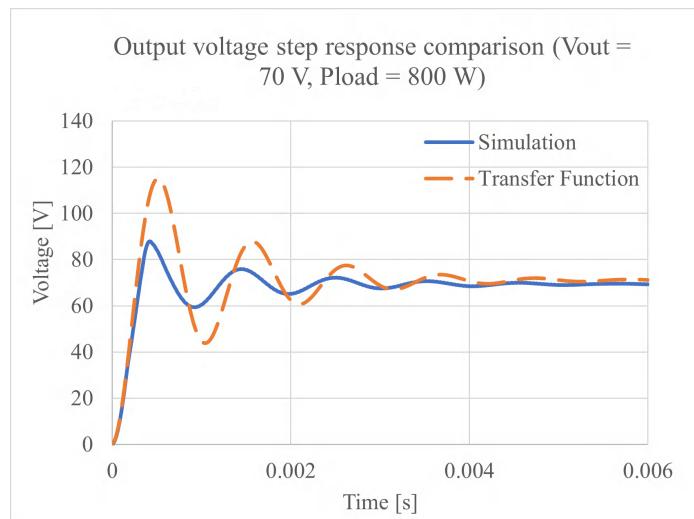


Figure 5.22: Open loop step response comparison for a 800 W load and an output voltage of 70 V

5.3.8 Controller Specifications

The operating ranges of the controller can be seen in table 5.6. Its clear from table 5.6 that the operating range of the buck converter is extremely wide, leaving a complex controller design task.

Table 5.6: Converter operating ranges

Parameter Name	Symbol	Value	Unit
Output power	P_{out}	$40 \sim 800$	W
Output voltage	V_{load}	$4.25 \sim 80.75$	V
Output current	I_{load}	$0 \sim 20$	A
Load Resistance	R_{load}	$0.2125 \sim 122.5$	Ω

The specifications of the controller was set out in the specification document contained in appendix A and are repeated in table 5.7 for convenience. It can be seen that a percentage overshoot of 0% is desired, which is crucial for the PVEM. This critically damped requirement for the PVEM is necessary, since any overshoot will cause the MPPT to observe an operating point that is not within reach of the PV module being emulated.

Table 5.7: Controller specifications

Parameter Name	Symbol	Value	Unit
Percentage overshoot	$P.O.$	0.1	%
Steady-state error	e_{ss}	1	%
Settling time	T_s	100	ms

5.3.9 Controller Design

The control system of the PVEM will make use of PI controllers. This is due to the simplicity and robustness of PI controllers. Typically, PI controllers are not designed, but rather tuned by engineers having a great deal of experience with them. Tuning a single controller manually is a simple task, but will not be sufficient for the PVEM. The I-V curves of the PVEM depends on the data that the user enters and can cover an infinite number of possibilities. Combine this with load variations, and it becomes impossible to design a single controller to meet the specifications for all conditions.

The tailored solution to the problem is to automatically tune several PI controllers by means of the genetic algorithm, whereafter the solutions will be used to create a Takagi-Sugeno fuzzy model, which can then be used to create an adaptive PI controller for the buck converter. This will allow the PI controller to dynamically adapt its constants to the changing load conditions, while always remaining within the specifications of the controller. The adaptive PI controller allows the control strategy to operate with a single mode controller, which will be a voltage mode controller in this case. Figure 5.23 displays the proposed control system. The basis of the work is obtained from [35].

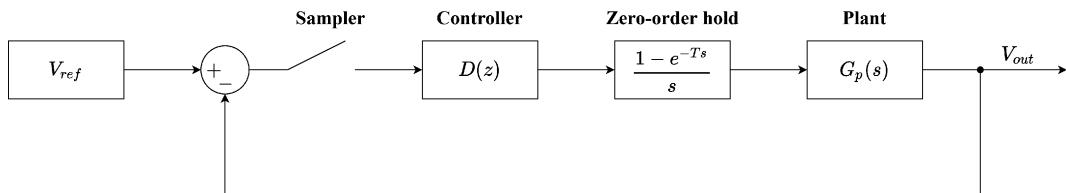


Figure 5.23: Proposed control system

It can be seen that the output voltage will be measured, subtracted from the reference voltage, sampled, and finally fed to the controller. The controller will then update the value of the zero-order hold after each sampling period in order to feed a continuous value to the plant. The value fed to the plant will be a PWM signal with a duty cycle obtained from the zero-order hold.

Genetic Algorithm PI Tuning

Typical operating points for the buck converter must be selected in order to tune the PI controller. These operating points will be the minimum, median, and maximum power values at several voltages. This will yield an extensive profile of the required PI values at various operating points to satisfy the controller specifications. The more operating points are selected, the better the resolution of the fuzzy model. Table 5.8 displays the operating points selected for which the genetic algorithm must tune PI controllers.

Table 5.8: Operating Points

(a) Parameters for $V_{load} = 4.25 V$

Resistance [Ω]	Power [W]
0.452	40.0
0.289	62.5
0.213	85.0

(b) Parameters for $V_{load} = 10 V$

Resistance [Ω]	Power [W]
2.500	40.0
0.833	120.0
0.500	200.0

(c) Parameters for $V_{load} = 20 V$

Resistance [Ω]	Power [W]
10.000	40.0
1.820	220.0
1.000	400.0

(d) Parameters for $V_{load} = 30 V$

Resistance [Ω]	Power [W]
22.500	40.0
2.813	320.0
1.5	600.0

Table 5.8: Operating Points Continued

(e) Parameters for $V_{load} = 40 V$

Resistance [Ω]	Power [W]
40.000	40.0
3.810	420.0
2	800.0

(f) Parameters for $V_{load} = 50 V$

Resistance [Ω]	Power [W]
62.500	40.0
5.950	420.0
3.125	800.0

(g) Parameters for $V_{load} = 60 V$

Resistance [Ω]	Power [W]
90.000	40.0
8.570	420.0
4.500	800.0

(h) Parameters for $V_{load} = 70 V$

Resistance [Ω]	Power [W]
122.500	40.0
11.670	420.0
6.125	800.0

The varying input parameters to the voltage transfer function of equation (5.75) is the load voltage, V_{load} , and the load resistance, R_{load} . The load resistances contained in the operating parameters of table 5.8 can now be arranged from smallest to largest along with its voltage. The arranged array with the carried voltage is displayed in table 5.9.

Table 5.9: Arranged array

Index	1	2	3	4	5	6	7	8	9	10	11	12
R_{load} [Ω]	0.2125	0.289	0.452	0.5	0.833	1	1.5	1.82	2	2.5	2.813	3.125
V_{load} [V]	4.25	4.25	4.25	10	10	20	30	20	40	10	30	50
Index	13	14	15	16	17	18	19	20	21	22	23	24
R_{load} [Ω]	3.81	4.5	5.95	6.125	8.57	10	11.67	22.5	40	62.5	90	122.5
V_{load} [V]	40	60	50	70	60	20	70	30	40	50	60	70

The fuzzy model requires intervals of solutions rather than discrete values, hence the need for the arranged array. The arranged array now allows the genetic algorithm to tune a PI controller meeting the requirements for the two extreme values on the interval. For example, a PI controller will be tuned to meet the specifications for both the input parameters of index 1 and 2 from table 5.9 simultaneously, whereafter it will tune a PI controller for the input values of index 2 and 3. This process repeats until a PI controller meeting the specifications has been tuned for each index interval. The PI controller utilises the trapezoidal integration rule, yielding the equation of the controller as given by (5.78) from [36].

$$D(z) = K_d \frac{z^2 + az + b}{z(z-1)} \quad (5.78)$$

K_d , a , and b can be calculated as shown by (5.79) through (5.81), where K_P is the proportional constant, K_I the integral constant, and K_D the derivative constant. In this case K_D is zero since only a PI controller is utilised.

$$K_d = K_P + K_I \frac{T}{2} + K_D \frac{1}{T} \quad (5.79)$$

$$a = \frac{\left(K_I \frac{T}{2} - K_P - 2K_D \frac{1}{T} \right)}{K_d} \quad (5.80)$$

$$b = \frac{K_D}{TK_d} \quad (5.81)$$

After the genetic algorithm generates a solution for K_P and K_I , the values of K_d , a , and b are calculated, substituted into (5.78), and the step response is evaluated. The process was completed and the solution set for three intervals are tabulated in table 5.10. Table 5.10 is a condensed version of the full solution set, contained in table B.1 in appendix B. The abbreviation LB denotes the lower boundary of the interval while UB denotes the upper boundary of the interval. From table 5.10 it should be clear why a single controller would not suffice; i.e. the PI constants varies significantly between operating

points.

Table 5.10: Tuned PI controllers at different operating points

Interval [Ω]	K_P	K_I	P.O. LB [%]	P.O. UB [%]	T_s LB [ms]	T_s UB [ms]
$0.213 < R < 0.289$	$6.63037e - 05$	4.004932	0	0	9.50	10.00
$1 < R < 1.5$	$1.34920e - 05$	9.924234	0	0	3.50	3.75
$40 < R < 62.5$	$3.76797e - 07$	3.823601	0	0	11.50	11.25

Takagi-Sugeno Fuzzy Model Adaptive PI Controller

The Takagi-Sugeno fuzzy model is used to infer the PI controller parameters for a measured load resistance [35]. To achieve this, the PI constants must be fuzzified to obtain a relationship between any load resistance value and the PI parameters. The fuzzification is achieved with the membership functions of a fuzzy model, which can be either a triangular, sigmoid, or Gaussian curve. The Gaussian membership functions are chosen to achieve a smooth transition between the various loading conditions. The Gaussian membership function is given by

$$f(x, \sigma, c) = e^{\frac{-(x - c)^2}{2\sigma^2}}, \quad (5.82)$$

where c is the centre point of the membership function and σ is the width of the membership function. In the case of the PVEM, (5.83) and (5.84) yields the centre and width of the membership functions, respectively.

$$c = \frac{R_{load,LB} + R_{load,UB}}{2} \quad (5.83)$$

$$\sigma = R_{load,UB} - R_{load,LB} \quad (5.84)$$

Using (5.82) through (5.84), the membership functions of the tuned PI controllers from table 5.10 and B.1 were created and are depicted in figure 5.24. Moreover, figure 5.24 depicts the membership function per interval of load resistance belonging to a set of PI constants.

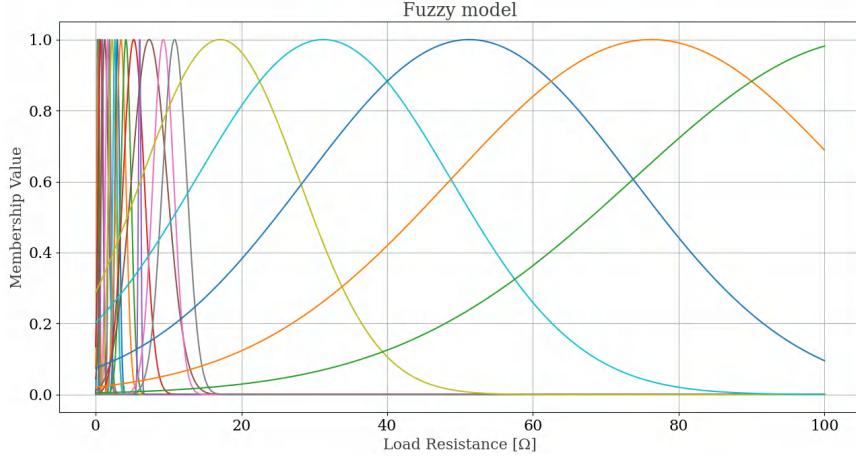


Figure 5.24: Fuzzy model

The Takagi-Sugeno equations for the defuzzification step of the PI constants are given by (5.85) and (5.86) [35].

$$K_P = \frac{\sum_{i=1}^m \mu_{A^i}(R_{load}) k_p^i}{\sum_{i=1}^m \mu_{A^i}(R_{load})} \quad (5.85)$$

$$K_I = \frac{\sum_{i=1}^m \mu_{A^i}(R_{load}) k_i^i}{\sum_{i=1}^m \mu_{A^i}(R_{load})} \quad (5.86)$$

In (5.85) and (5.86), m is the number of fuzzy-rules, $\mu_{A^i}(R_{load})$ is the membership value of the load resistance with respect to the i -th rule, and k_p^i and k_i^i is the output of the i -th rule. The membership value, $\mu_{A^i}(R_{load})$, of the i -th rule can be computed with equation (5.82). [35]

Using the Takagi-Sugeno equations, values for K_P and K_I can be obtained for any load resistance. The software developed to tune the PI controller can be found in appendix D.

Validation

To evaluate the control strategy random values of load voltages and resistances will be generated, whereafter the system will be controlled by means of the adaptive PI controller. Should all of the systems conform to the specifications, it can be said that the controller design is acceptable. To perform the validation, 5000 random sets were created and controlled with the adaptive PI controller. The results for the percentage overshoot and settling time are displayed in figures 5.25a and 5.25b respectively. It should be noted that the percentage overshoot reaches values of up to 0.015%, which is higher than the desired 0%. However, the value is extremely small and the percentage overshoot can be considered as negligible. The settling time's maximum value is 12.5 ms, which is well below the desired 100 ms. In conclusion, the adaptive PI controller performs within specification for all operating conditions and can be used in the PV emulator with confidence.

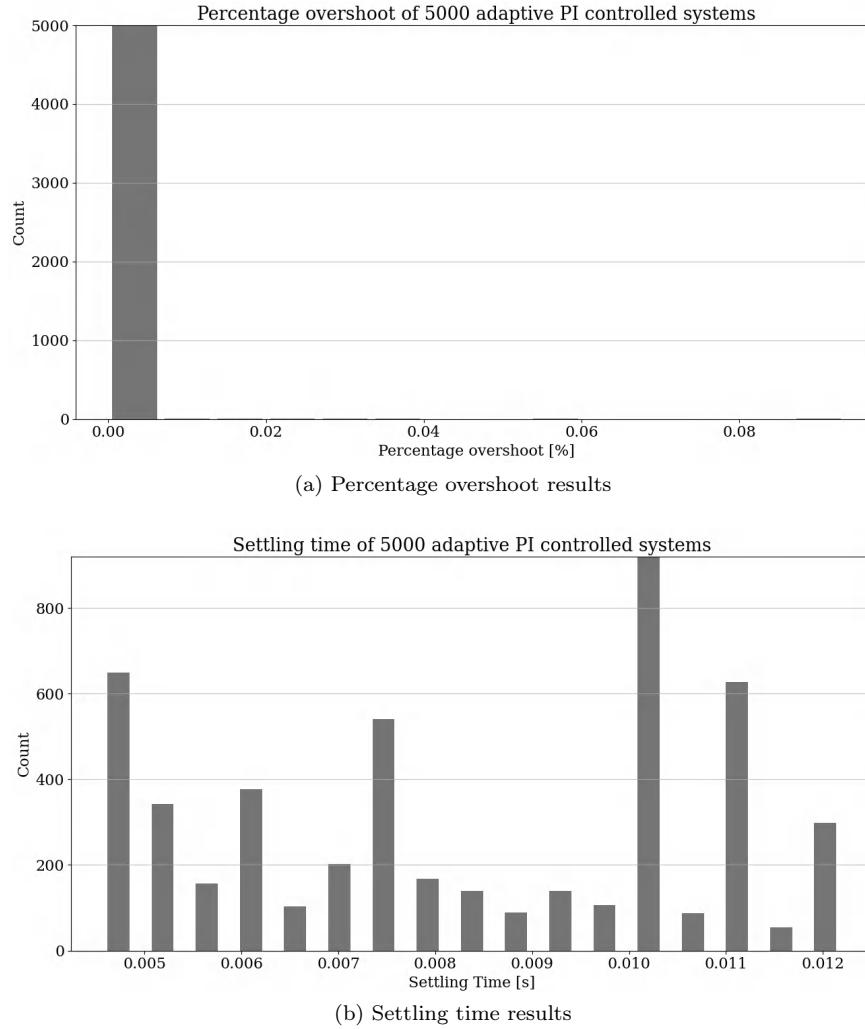


Figure 5.25: Response results of 5000 adaptive PI controlled systems

5.4 Conclusion

Firstly, the PV model was designed, which utilised the single-diode exponential model, genetic algorithm, Lambert W functions, and bisection method. Thereafter the power converter was designed which consists of a buck converter. The measurement circuits circuit was also designed. The simulations proved that the power converter works as designed. Lastly, the control strategy for the buck converter was designed which employed an adaptive fuzzy logic PI controller, yielding within-specification responses for 5000 randomly generated closed loop systems. The design phase was successful, and the implementation phase can now commence.

Chapter 6

Implementation and Evaluation

The implementation of the detailed design follows in this chapter. The implemented sub-systems are tested and evaluated with respect to their design specifications. The PV model is covered first, followed by the power converter and control strategy. Finally, the system as a whole is evaluated in the integration section.

6.1 PV Model

This section covers the implementation and evaluation of the PV model.

6.1.1 Genetic Algorithm

The software developed to compute the PV model using the Genetic Algorithm is available in appendix D. As mentioned previously, the parameters of the Genetic Algorithm must be determined using a sensitivity analysis. The test data for the sensitivity analysis is contained in appendix C. The test was performed as follows: all parameters were kept fixed except for the parameter being evaluated. The test was then repeated five times for each of the parameters, and the average time value over the five tests were used to rank the value. The results of the sensitivity analysis is contained in table 6.1.

Table 6.1: Genetic Algorithm Parameter Values

Parameter	Value
Population Size	100
Maximum Generations	25
Crossover Rate	0.8
Mutation Rate	0.05
Tournament Selection Sample	0.4

To illustrate how the genetic algorithm arrives at a solution three cases were investigated. The first is a fast convergence, taking only seven generations of the maximum twenty-five. From figure 6.1, it can be seen that the algorithm suddenly finds a solution within the tolerance after the sixth generation.

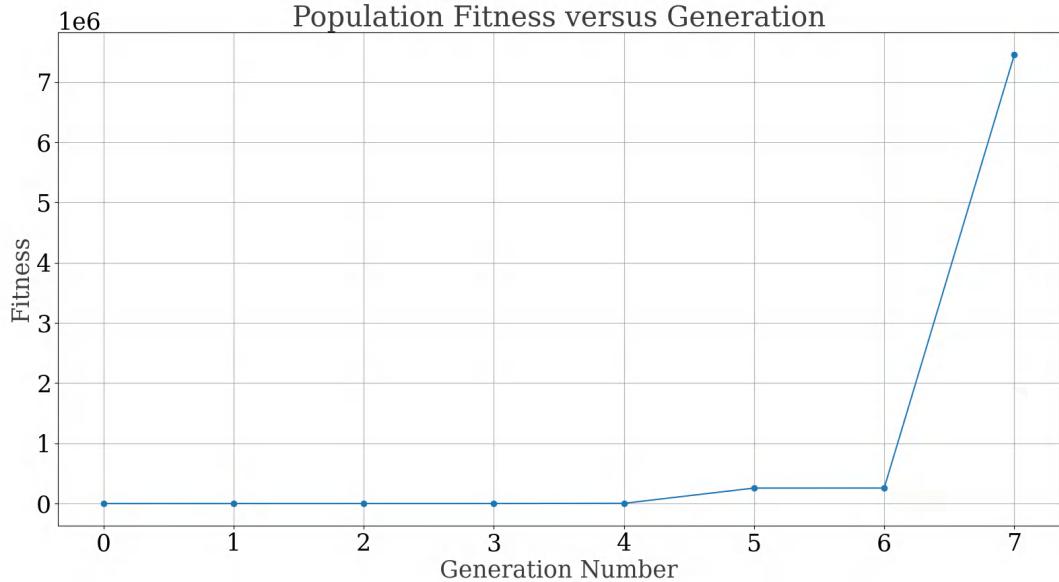


Figure 6.1: Fast convergence

The second was a medium convergence time, taking thirteen generations to converge as depicted in figure 6.2. After the tenth generation, the solution fitness exponentially increases.

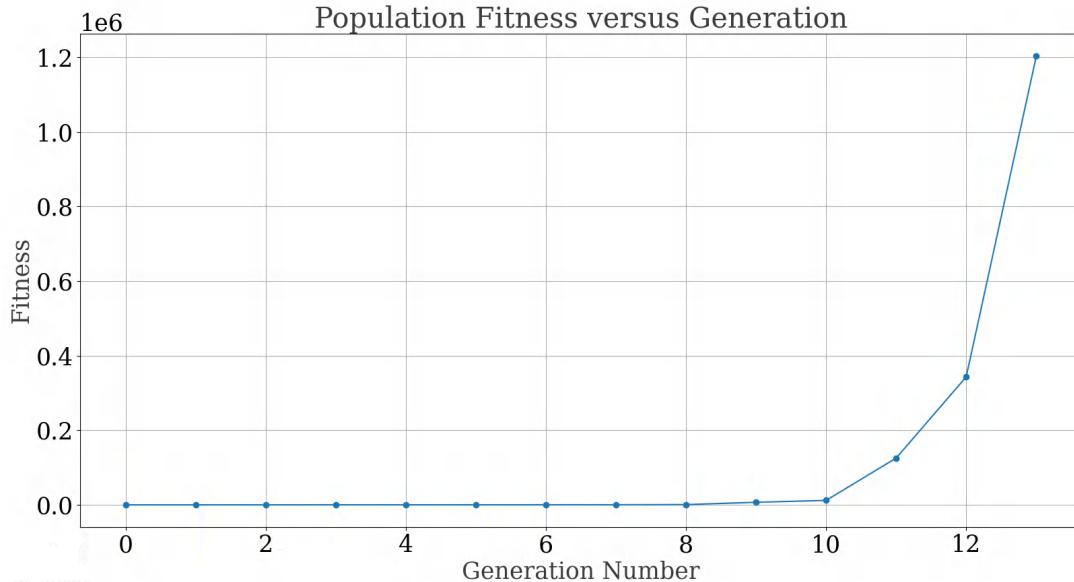


Figure 6.2: Medium convergence

The last is a solution taking all twenty five generations to converge. It can be seen from figure 6.3 that the solution fitness does not increase much after the ninth generation. The solution is also not within the tolerance after the last generation. To ensure that the solution is still within bounds, the solution is evaluated with respect to the maximum power point afterwards. If it is not within a specific tolerance of the maximum power point the solution is generated again.

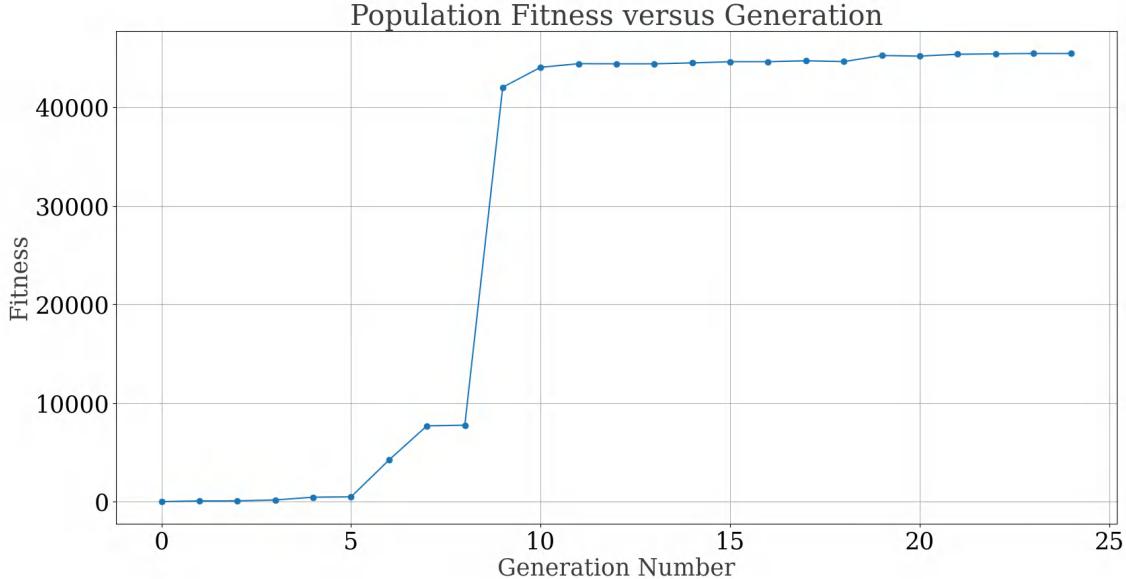


Figure 6.3: Slow convergence

6.1.2 I-V Curves

The accuracy of the generated I-V curves will be evaluated by investigating the calculated V_{oc} , V_{mp} , I_{sc} , and I_{mp} points. Two points are provided on a typical PV module datasheet. The parameters at STC and NOCT. The parameters at STC will be entered by the user as mentioned in the following subsection. The points can then be translated to the NOCT conditions and its accuracy can be evaluated at NOCT. It is important to remain unbiased in evaluating the the accuracy, thus PV modules from various manufacturers will be used. The PV module selected will be the lowest power output from an identified range, to keep things consistent. Tables 6.2 through 6.5 depicts the NOCT results and their respective errors.

Table 6.2: JA Solar 525 W (JAM72S30 525-550/MR) [4]

	Datasheet	Calculated	Error [%]
V_{oc} [V]	46.05	46.03	0.043
V_{mp} [V]	38.36	38.51	0.391
I_{sc} [A]	10.97	11.03	0.547
I_{mp} [A]	10.35	10.31	0.386

Table 6.3: Trina Solar 430 W (Trina Tallmax 430-450W)

	Datasheet	Calculated	Error [%]
V_{oc} [V]	46.00	46.15	0.326
V_{mp} [V]	38.00	38.16	0.421
I_{sc} [A]	9.03	9.04	0.111
I_{mp} [A]	8.56	8.60	0.467

Table 6.4: Canadian Solar 295 W (KuPower CS3K-295|300|305|310P) [4]

	Datasheet	Calculated	Error [%]
V_{oc} [V]	36.70	36.76	0.163
V_{mp} [V]	30.20	30.53	1.093
I_{sc} [A]	7.72	7.73	0.130
I_{mp} [A]	7.26	7.34	1.102

Table 6.5: LONGi Solar 525 W (LR5-72HPH 525~545M) [4]

	Datasheet	Calculated	Error [%]
V_{oc} [V]	45.98	45.99	0.022
V_{mp} [V]	38.36	38.61	0.652
I_{sc} [A]	11.04	11.04	0.000
I_{mp} [A]	10.23	10.31	0.782

From table 6.4, it can be seen that the highest error was 1.102%, which is acceptable since the tolerance on the parameters are often 3% in many datasheets. The error is far below this value, implying that the modelling of the I-V curves are satisfactory.

6.1.3 GUI

The implementation of the graphical user interface can be seen in figure 6.4, which corresponds to I/F 1.1 from chapter 4. The implementation was completed using Python, specifically the PyQt5 library. The code for the GUI can be found in appendix D. The GUI is divided into two main sections; the user input section, and the plotting section. The user input section captures the STC parameters of a PV module along with the temperature and irradiation the user wants to emulate at. The plotting section contains two plots, the I-V curve and the P-V curve. On both curves, the live operating point of the PVEM is displayed to view the operating point at all times. Also, two buttons are available. The first

of which is the ‘Plot Curves’ button. This button captures the input provided by the user and plots the I-V and P-V curves for the PV module, using the PV model class.

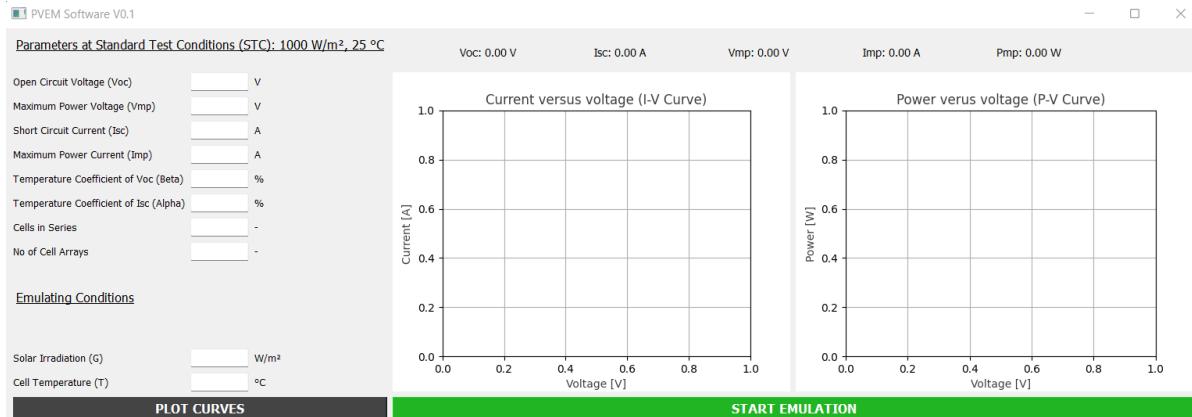


Figure 6.4: Graphical User Interface Implementation

Figure 6.5 depicts the GUI after the user has entered a PV module’s data into the appropriate fields and pressed the ‘Plot Curves’ button. A red x is displayed on both the I-V and P-V curves, which indicates the maximum power point of the PV module.

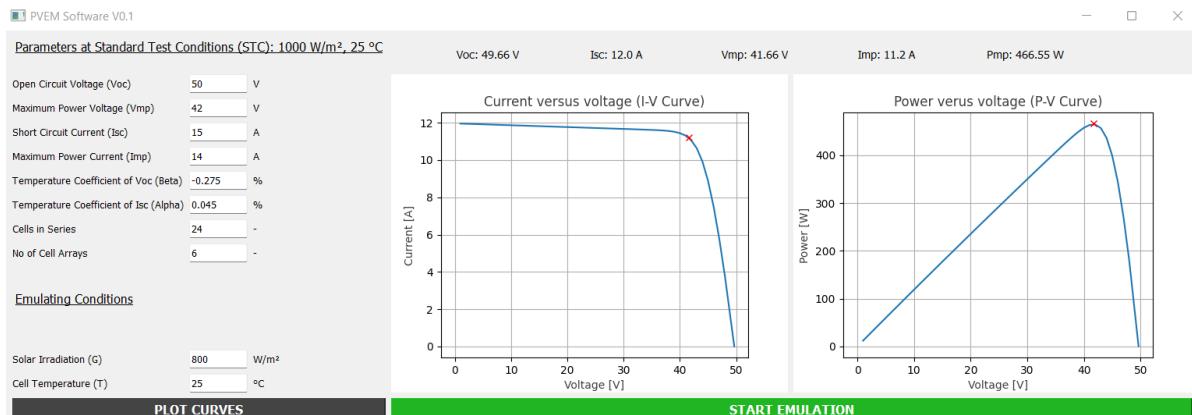


Figure 6.5: GUI after data has been entered and ‘Plot Curves’ button pressed

Once the user has pressed the ‘Plot Curves’ button, the ‘Start Emulation’ button is enabled. After pressing the ‘Start Emulation’ button, the computer establishes a serial connection with the PV microcontroller. The computer then transmits the voltages at which the PV emulator should operate corresponding to predetermined resistances. The lookup table on the PV emulator is now populated and the PV emulator starts the emulation by controlling the output at the operating point specific to the load connected. The microcontroller transmits the operating point back to the computer after a timer has elapsed, updating the graphs. Figure 6.6 depicts the GUI during the emulation phase. The button ‘Start Emulation’ from figure 6.5 has changed to ‘Stop Emulation’ since pressing the button now would stop the emulation process. The PWM signal at the output of the PV emulator will also stop after pressing this button. The dot indicated on the graphs in figure 6.6 depicts the operating point of the PV emulator on both the I-V and P-V graph.

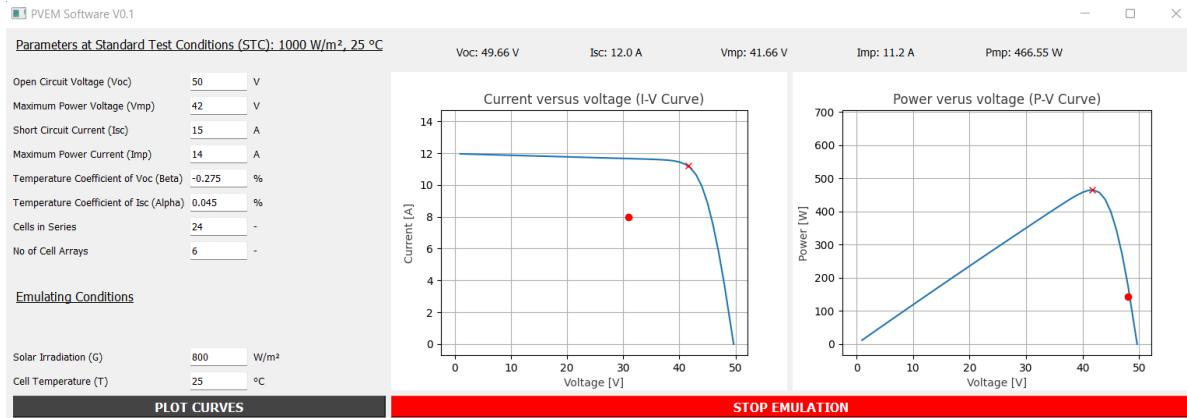


Figure 6.6: GUI during emulation

6.1.4 Serial Communication

The serial communication is an integral part of the PV emulator, since it allows the lookup table on the microcontroller to be populated as well as the visualisation of the data. The serial communication must happen in a synchronous manner to ensure that the right data is visualised at the right time. The first state of the controller is the receive state, where it waits upon incoming data. The microcontroller receives the voltage values of the I-V curve corresponding to various resistance values and populates the lookup table. After each value is sent to the controller, the controller sends it back to the PC to ensure that the correct value was received on the controller. At the end of the lookup table data, the PC transmits a stop sequence, which signals the completion of the data transmission to the controller. The controller then switches states, entering the emulation state. In this state, the controller writes data to its serial buffer at a fixed rate, while the PC reads from the serial buffer at the same rate. This allows for synchronous communication, which is then used to visualise the live operating points of the PV emulator.

6.2 Power Converter

The implemented power converter is depicted in figure 6.7. The functional units (F/U) corresponds to those set out in chapter 3.

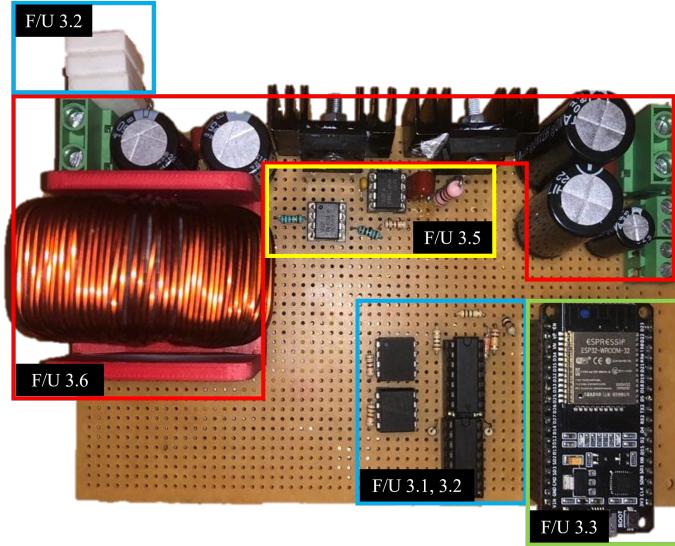


Figure 6.7: Implemented Power Converter

6.2.1 Buck converter

The buck converter was can be seen in figure 6.7, where F/U 3.6 represents the buck converter, and F/U 3.5 represents the driving circuitry. A few tests were performed on the buck converter to determine if it is within its design specification. The efficiency of the buck converter, along with the output ripple voltage are the specifics that are determined first. Figure 6.8 depicts the switching waveforms of the buck converter at a power output of 854.41W, which is 6.8% above the design specification of 800 W. The circuit was also designed to output a maximum power of 800W, and at a maximum designed output voltage of 70V, this yields 11.429 A. From figure 6.8, the output voltage is at 71V, with a current of 12.0339 A. Thus, the circuit is able to output 70V at 800W.

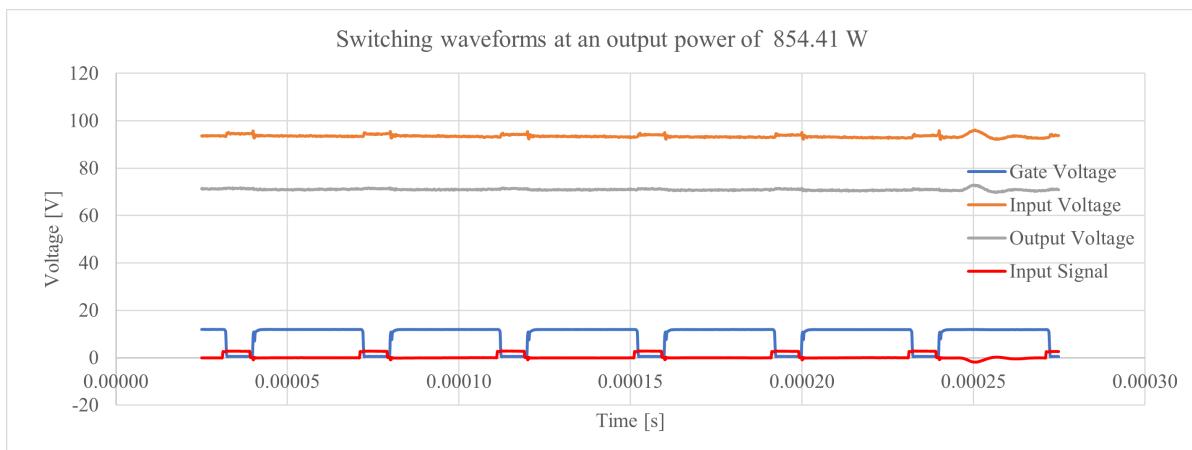


Figure 6.8: Switching Waveforms at 900.18 W

The output current design was 20A. This yields a voltage of 40V at 800W. Figure 6.9 depicts the output waveforms at 934.65W with a current of 17.82993 A flowing. This was the highest current achievable with the 2.9Ω resistance available. A resistance of 2Ω is required to test the PV emulator

at 40 V, 20 A, but this resistance value could not be achieved with the available resistors. Also, the voltage could not be raised anymore as the power limit would reach 1176 W, which would raise the input current above the 12A limit of the input source. The circuit was also not designed to reach power levels this high. However, at a current of 17.82993 A, the circuit performed as expected.

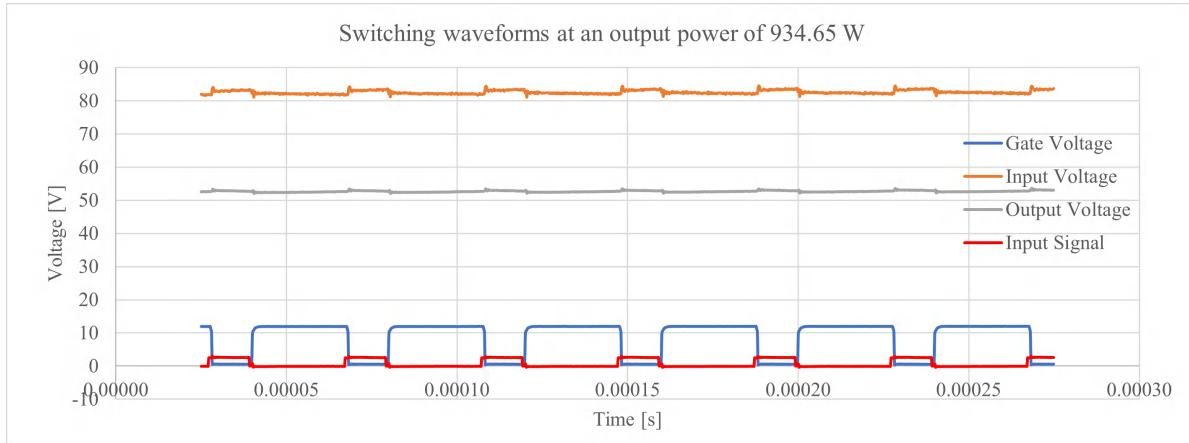


Figure 6.9: Switching Waveforms at 934.65 W

The figure below depicts the circuit at relatively low power draw. From this load connected the ripple voltage can be calculated. While ignoring the ripple on the input voltage, the output voltage reaches a maximum value of 13.6842 V, while the minimum value reaches 13.565 V. This results in a ripple voltage of 0.437%, which is below the design value of 0.5%.

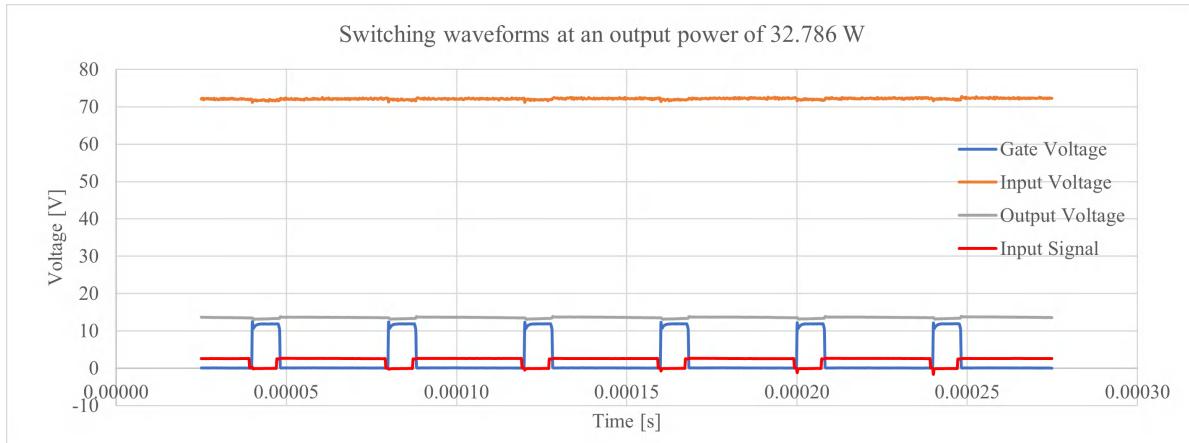


Figure 6.10: Switching Waveforms at 32.786 W

Lastly, the switching propagation delay was investigated. The driver circuitry and optocoupler creates a delay of $1.6 \mu s$ as depicted in figure 6.11. Since the switching period is $40 \mu s$ the switching delay equates to 4% of the switching period. The switching delay could possibly affect the control strategy if it were larger, but due to it being extremely small, it should have no effects.

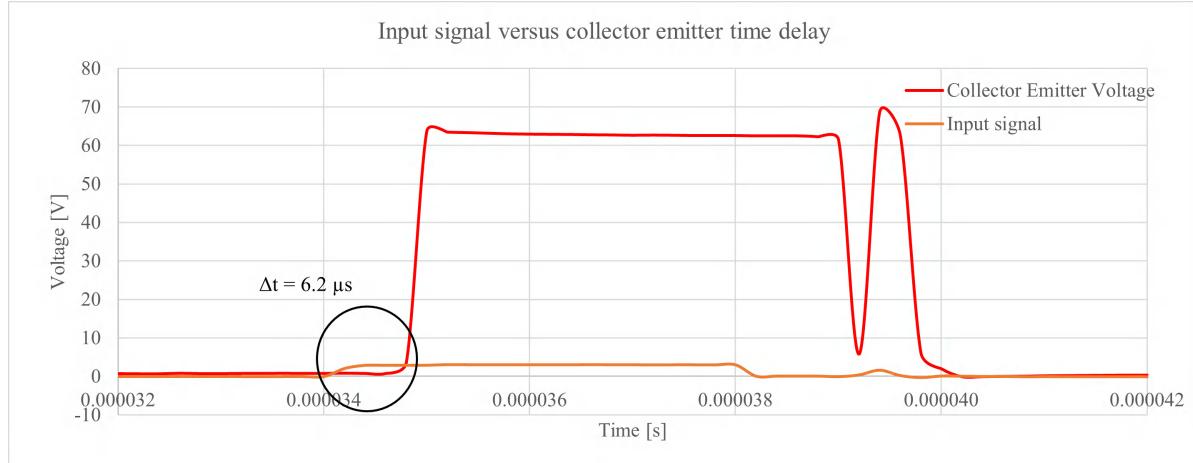


Figure 6.11: Switching delay

Efficiency

The efficiency of the circuit was tested by measuring both the input and output current and voltages at various power levels and duty cycles. Table 6.6 depicts the test results, and it can be seen that the efficiency of the circuit always remains above 82.79%, implying that the target of 75% efficiency as set out in appendix A is met.

Table 6.6: Efficiency test results

Duty Cycle	V_{in} [V]	I_{in} [A]	V_{out} [V]	I_{out} [A]	P_{in} [W]	P_{out} [W]	η [%]
0.1841	71.8	1.0	13.22	4.4966	71.80	59.4450	82.79%
0.1892	71.6	0.5	13.55	2.4196	35.80	32.7862	91.58%
0.2804	90.1	2.5	25.26	8.5918	225.25	217.0298	96.35%
0.3728	89.2	4.5	33.25	11.3095	401.40	376.0417	93.68%
0.4653	88.8	7.0	41.32	14.0544	621.60	580.7287	93.42%
0.5559	87.9	10.0	48.86	16.6190	879.00	812.0067	92.38%
0.6408	81.8	12.5	52.42	17.8299	1022.50	934.6450	91.41%

Figure 6.12 depicts a plot of the efficiency versus duty cycle for the buck converter. The efficiency suddenly spikes at about 30% duty cycle, whereafter it linearly decreases.

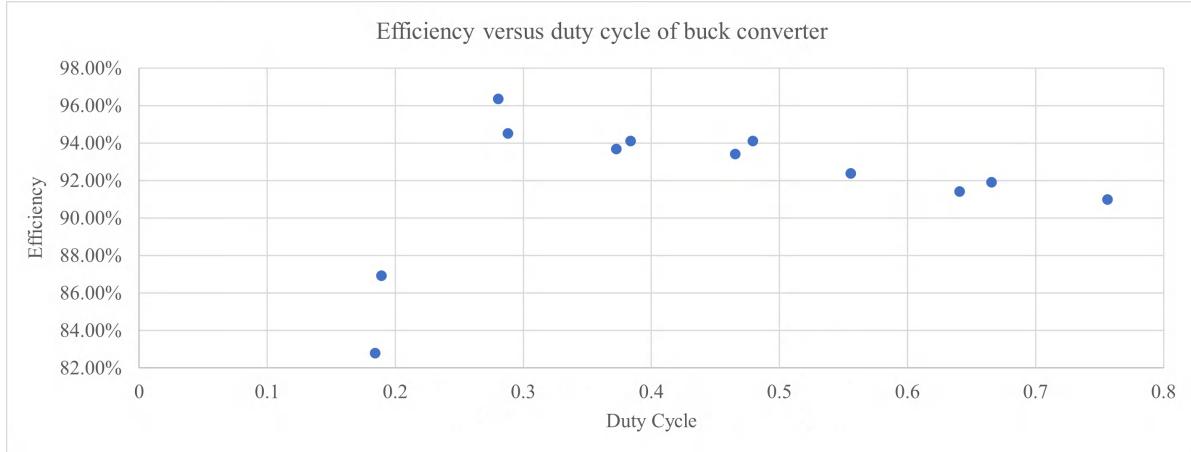


Figure 6.12: Efficiency versus duty cycle

6.2.2 Measurement Circuit

The measurement circuit was constructed and corresponds to F/U 3.1 and F/U 3.2 in figure 6.7. To implement the control strategy of the PVEM, the measurement circuits must be characterised. The voltage sensor was characterised first, followed by the current sensor.. To fully characterise the voltage sensing circuit, the output voltage was adjusted over the full range of the sensor, and the corresponding sensor output voltage was measured with the controller. The results for the voltage sensor is depicted in figure 6.13.

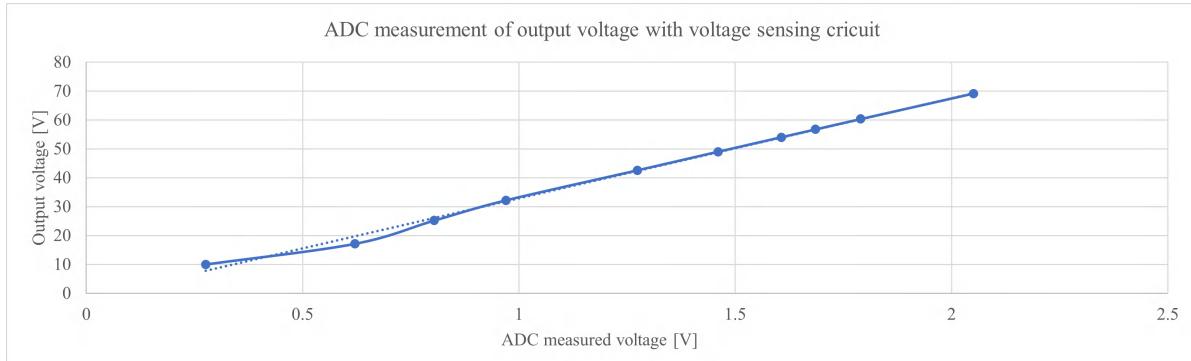


Figure 6.13: Voltage sensor characterisation

From figure 6.13 it is clear that the behaviour of the voltage sensor is fairly linear as expected and a regression line can now be fitted to the datasets. The equation for the voltage sensor is given by

$$V_{out} = 29.309375V_{sense} + 0.990643, \quad (6.1)$$

where V_{sense} represents the sensed voltage and V_{out} the calculated output voltage. Next, the current sensing circuit can be characterised in a similar fashion. The results are depicted in figure 6.14.

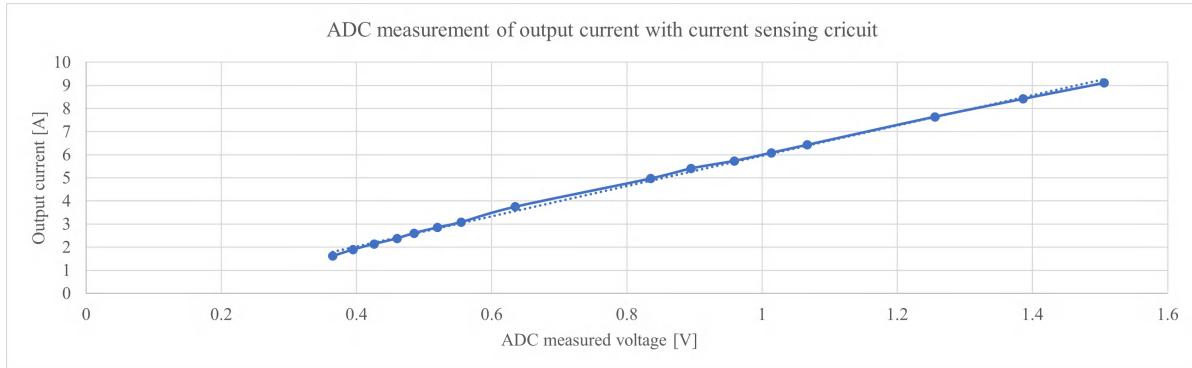


Figure 6.14: Current sensor characterisation

After fitting a linear regression line to the data, the resulting equation is given by

$$I_{out} = 5.022587I_{sense} + 0.465176, \quad (6.2)$$

where I_{out} represents the calculated output current.

6.2.3 Open-loop response comparison

The open-loop response of the buck converter was measured and compared with the transfer function of equation 5.75.

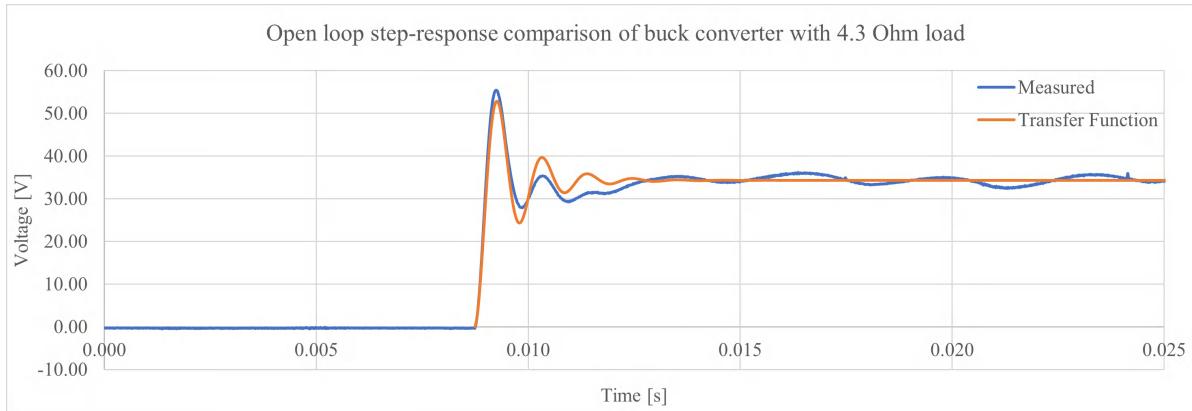


Figure 6.15: Open loop comparison

Figure 6.16 depicts the open-loop comparison of the step response. It is clear that the transfer function accurately models the measured response in terms of its overshoot and settling time. The overshoot is almost identical, but the transfer function displays a bit more oscillations during the transient period. The effects are more drastic for smaller loads, but it should be noted that the overshoot and setting time is modelled fairly accurately in both figures 6.16 and 6.17. The only large difference is the oscillations. Using the transfer function for the controller design will provide a controller able to control at worst case, which implies that if the controller works for the transfer function it will control the practical circuit.

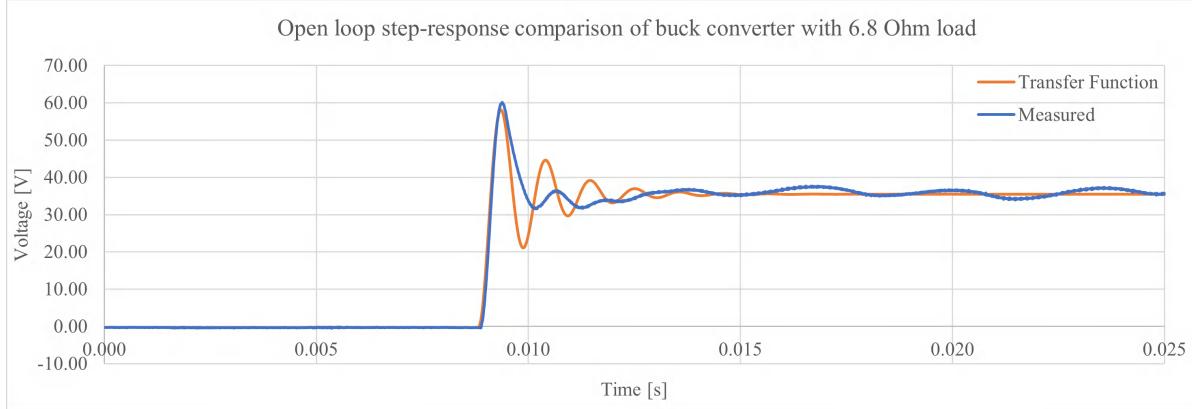


Figure 6.16: Open loop comparison

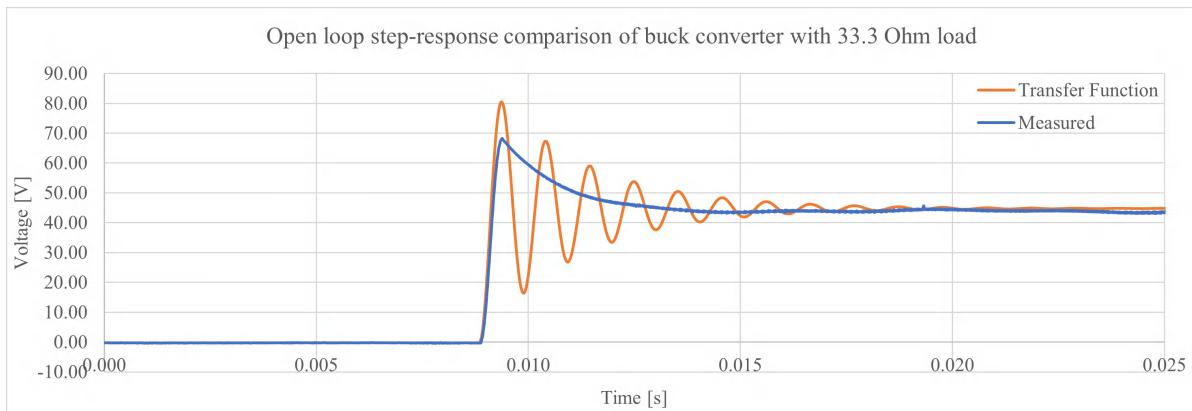


Figure 6.17: Open loop comparison

6.3 Control Strategy

The implementation of the control strategy is covered in this section.

6.3.1 Digital PI Controller Implementation

To implement the PI controller, a filter structure such as the Gray-Markel structure or Direct Form structures can be utilised. However, it is simpler to derive the difference equation from first principles. The equation for a PI controller in the w domain is given by (6.3), obtained from [36].

$$D(w) = K_P + \frac{K_I}{w}, \quad (6.3)$$

The equation for w is specific to the integration rule used, which is trapezoidal integration in this case, yielding

$$w = \frac{2(z-1)}{T(z+1)}. \quad (6.4)$$

After substituting (6.4) into (6.3), equation 6.3 can be rewritten to form

$$D(z) = \frac{M(z)}{E(z)} = K_P + K_I \frac{T(z+1)}{2(z-1)}, \quad (6.5)$$

which then yields the final form of the difference equation as

$$M(z) = M(z)z^{-1} + \left(K_P + K_I \frac{T}{2} \right) E(z) + \left(K_I \frac{T}{2} - K_P \right) E(z)z^{-1}. \quad (6.6)$$

From equation 6.6, $M(z)$ represents the output of the controller, $E(z)$ the error signal, and z^{-1} a storage element. More specifically, a variable multiplied by z^{-1} indicates that the previous value of the variable must be used. Thus, it is clear that the PI controller takes into account the previous output of the controller, the error, and the previous error value. A block diagram of equation 6.5 is depicted in figure 6.18, where ZOH is the zero-order hold. The purpose of the zero order hold is to feed a constant value to the PWM generator on the controller, which gets updated after each sampling period. The zero-order hold effectively ‘remembers’ the output of the controller for an entire sampling period.

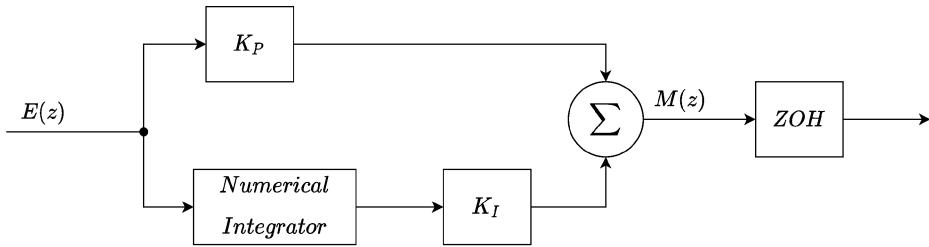


Figure 6.18: Digital PI controller block diagram

It should also be noted from equation 6.5 that the integral component adds the storage element, z^{-1} , to the controller, and its dependence on the previous value can lead to a phenomenon called *integral windup*.

Integral windup

Integral windup occurs when the error value accumulates in the controller, which typically leads to oscillations and overshoot until the error is unwound. Integral windup typically occurs when the change in set-point is extremely large or when the controller is unable to perform control due to a broken loop. The latter point is the main point of concern for the PVEM, since the controller can enter idle states during receiving data and will be unable to perform control. The solution is to implement conditional integration, which removes the integral component below a threshold value. Thus, when the controller measures the output voltage of the controller as a value below 4.25 V, which is the lowest output voltage for the PVEM, the controller does not perform integration, changing equation 6.6 to equation 6.7. After the output voltage is above the threshold, the controller equation switches from 6.7 to 6.6.

$$M(z) = M(z)z^{-1} + K_P E(z) \quad (6.7)$$

6.3.2 Digital Filter Design

The ADC of the ESP32 is extremely inaccurate for quick consecutive measurements. The inaccuracy of the measurements will greatly affect the output of the controller, and hence the output of the circuit.

To overcome this one of two techniques can be used. An averaging filter can be applied to the data, or a low-pass filter can be implemented. The factor that affects the decision of the filter to be used the most is computation speed. The averaging filter must sample more than once to determine the average of a number of samples. The sampling function of the microcontroller has a conversion time of close to $100 \mu s$. This equates to 10% of the available computation time. However, two measurements are made, actually resulting in 20% of the computation time. Thus a maximum of 4 samples (leaving space for additional computation) can be taken before the average is calculated. This is an extremely small amount of samples, and at least 10 would be needed to smoothen the ADC inaccuracies. The solution that remains is thus a low-pass filter. The low-pass filter only requires the sample and the previous sample to produce in an accurate and stable ADC measurement. A single-pole low-pass infinite impulse response (IIR) filter will be designed as it is simple to design and implement.

The equation of a low-pass IIR filter is given by

$$y[n] = dx[n] + (1 - d)y[n - 1], \quad (6.8)$$

where d is the decay factor of the filter. The equation of the decay factor is given by

$$d = e^{-2\pi f_{cn}}. \quad (6.9)$$

The parameter f_c represents the normalised cut-off frequency. The sampling frequency of the system is $1 kHz$. The cut-off frequency of the filter will be set to $100 Hz$ to eliminate most of the noise. The normalised cut-off frequency can now be calculated as

$$f_{cn} = \frac{f_c}{f_s}, \quad (6.10)$$

where f_c is the cut-off frequency in hertz and f_s is the sampling frequency. This allows d to be calculated as

$$d = e^{-2\pi f_{cn}} = e^{-2\pi f_c/f_s} = e^{-2\pi 100/4000} \approx 0.85.$$

Substituting d into (6.9) yields (6.8) as

$$y[n] = 0.85x[n] + 0.15y[n - 1].$$

6.3.3 Embedded Software Implementation

The embedded software implementation of the PV emulator was completed in C, with the controller being a DOIT ESP32 Devkit V1. The main criteria to be kept in mind while developing the software is the speed of the algorithms. The computational efficiency is crucial as the sampling period is quite quick for the amount of calculations needed during runtime.

The direct calculation method was evaluated and compared to a lookup table, but the lookup table proved far superior in speed. Since a lookup table was used, a search algorithm was necessary to find the operating point within the lookup table as fast as possible. The binary search algorithm is known to be the fastest search algorithm with a complexity of $O(\log(n))$, thus it was implemented. Also, the slowest part of the software is the functions reading the analog pins of the controller. The conversion

time is roughly $100 \mu s$, and two of these measurements must be taken, one for voltage and one for current.

Where possible, floating point division operations were avoided and substituted with multiplication as it is a faster operation. Finally, as discussed previously, a digital filter as opposed to an averaging filter was selected. Figure 6.19 displays a flow diagram of the code running on the controller. The flow diagram is a digestible version of the embedded software implementation included in appendix D.

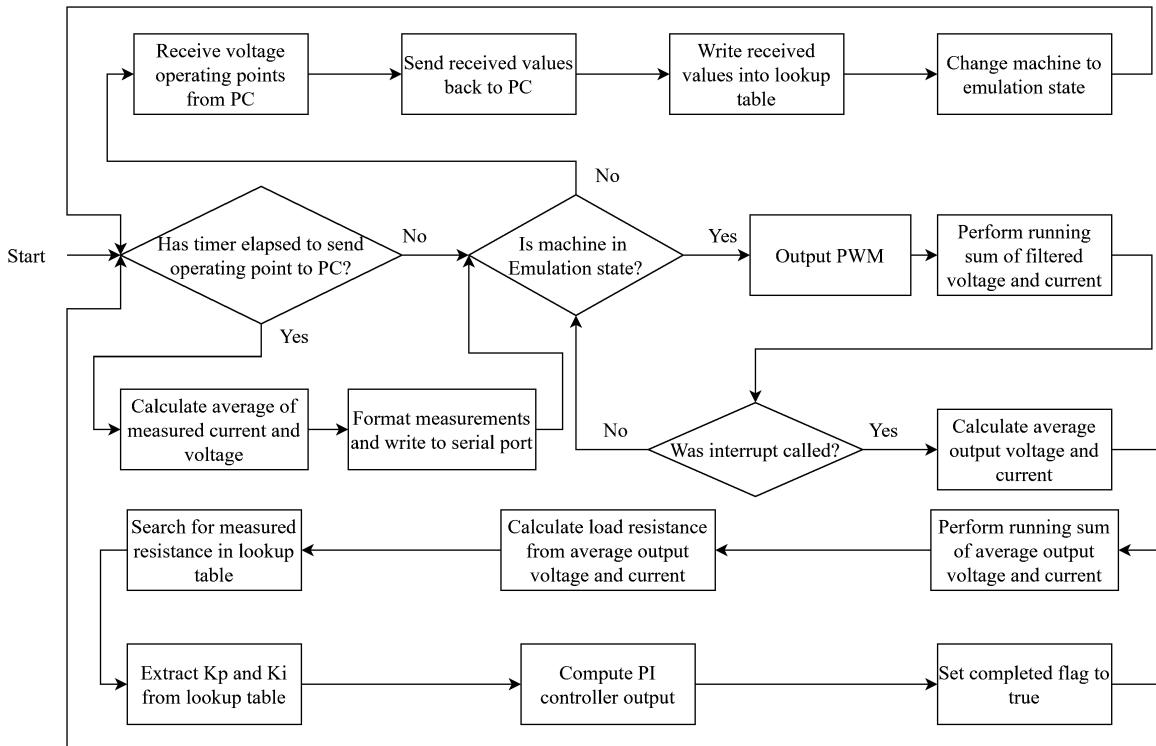


Figure 6.19: Embedded software flow diagram

6.4 Integration

The PV model, power converter and control strategy were now integrated to create the PV emulator system. Figure 6.20 depicts a high-level graphical representation of the PV emulator.

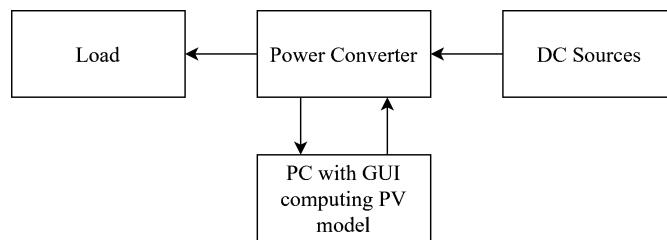


Figure 6.20: Integration

The load used in testing PV emulator was a large set of 64Ω resistors in parallel. The resistances

could then be switched in and out to change the load on the circuit. Figure 6.21 depicts the lab setup of the resistances in parallel. Each cart contains six $64\ \Omega$, $650\ W$ resistors. The lowest resistance achievable was about $2.5\ \Omega$, which is lower than the critical resistance (resistance at MPP) for most PV modules. This allows the PV emulator to be tested in both the constant current and constant voltage regions.



Figure 6.21: Variable resistive load

To evaluate the system as a whole, the I-V curve of an arbitrary PV module was selected whereafter the steady-state values of various loads were measured. The load resistance was varied from $10.2\ \Omega$ to $2.5\ \Omega$, which resulted in operating points in the constant current and voltage regions. This is crucial as the controller is a voltage controller, which can cause instability in the constant current region. However, the PV emulator performed very well within both regions. This is due to the fine segmentation in both regions, ensuring that large jumps between reference points do not occur. A visual representation of the operating points of various loads on an I-V curve are depicted in figure 6.22.

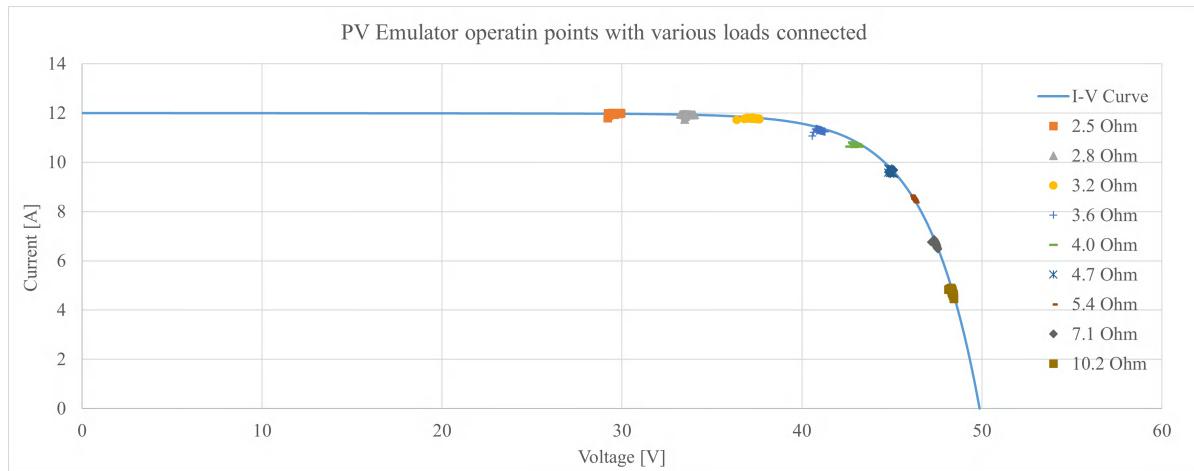


Figure 6.22: Operating Points integration

Figure 6.23 depicts the average errors of the operating points with respect to their reference voltages

from figure 6.22. It is clear that the error remains 0.5% and lower for the tested reference values. This is within the design specification of 1% as set out in section 5.3 and the system specification requirements. The errors mainly occur due to the inaccuracy of the ESP32's ADC along with the high sampling rate. If the sampling rate of the controller was slower, more samples could be averaged before it is fed to the input of the controller.

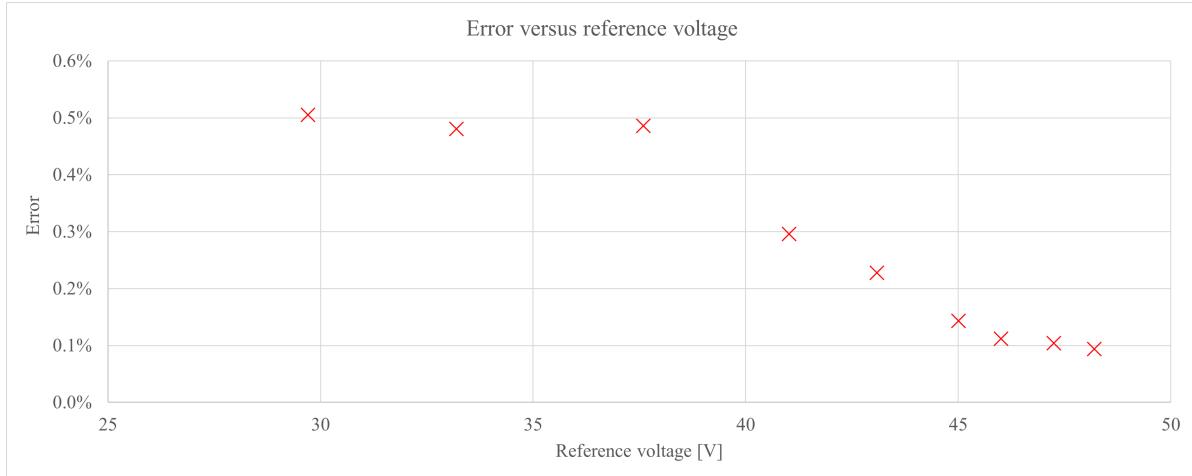


Figure 6.23: Operating Points errors

6.4.1 Step-response

The step-response of the buck converter was evaluated to determine if its operation is within specification. In the detailed design chapter, the sampling frequency was selected as 4 kHz, however, upon implementing the controller, it was found that this sampling was too fast. Figure 6.24 depicts the instability of the 4 kHz sampling frequency. It was found that reducing the sampling frequency to 1 kHz, and reducing the PI constants by a factor 10 of yields a much more stable system. PI controllers are often tuned upon implementation, thus the modifications are acceptable.

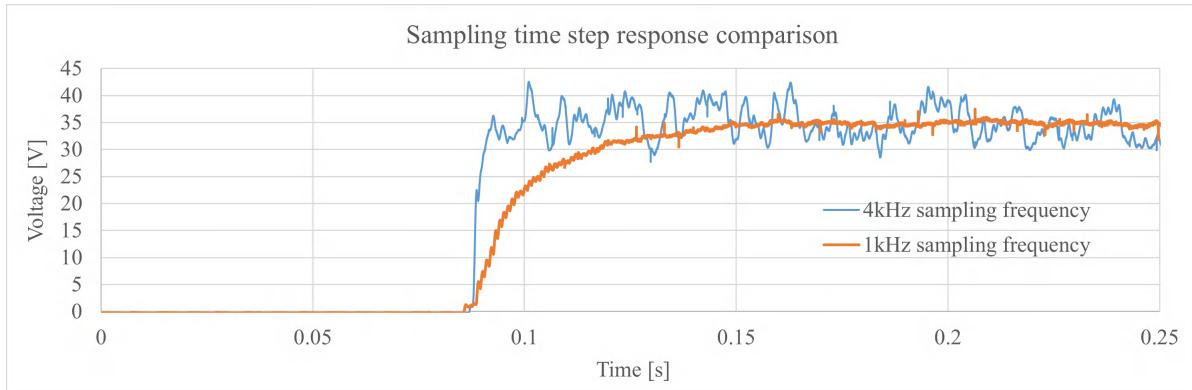


Figure 6.24: Sampling time step response comparison

The step response of the system under several loads are now evaluated. The first is the response when a 11.6Ω load is connected, which is depicted in figure 6.25. It is clear that the system is critically damped, and the settling time is found to be 89.74 ms.

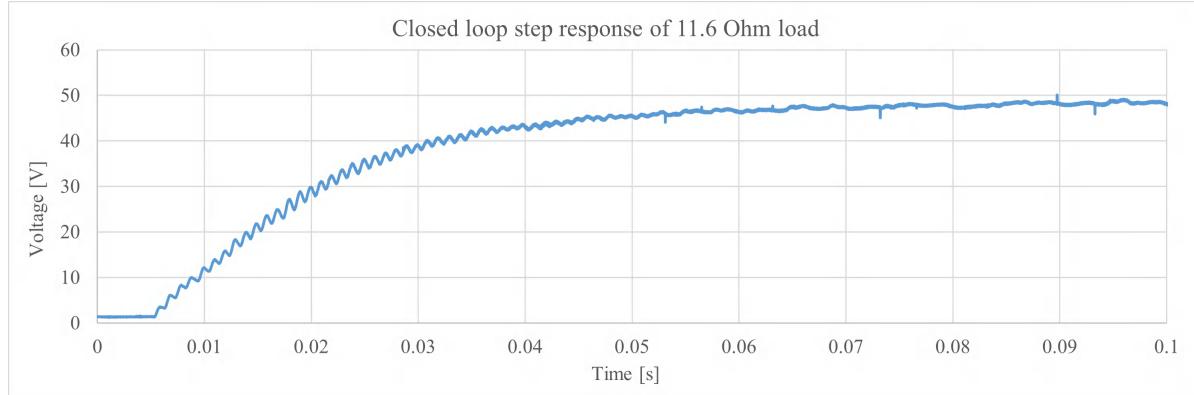


Figure 6.25: Step response of PV emulator with 11.6Ω load

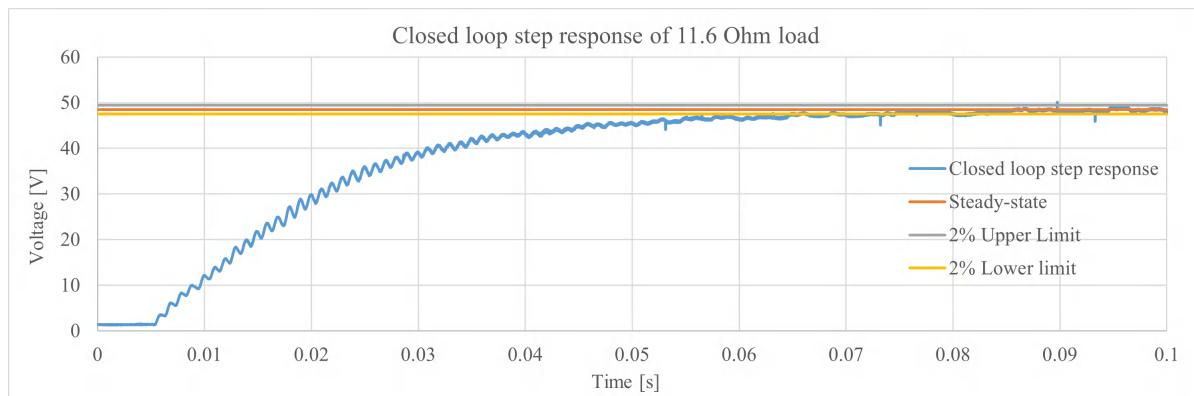


Figure 6.26: Step response of PV emulator with 11.6Ω load and boundaries indicated

Figure 6.27 shows the step-response of the system with a 5.0Ω load connected. The response of the system is faster, yielding a settling time of $68.98 ms$. The system also presents no overshoot, as desired.

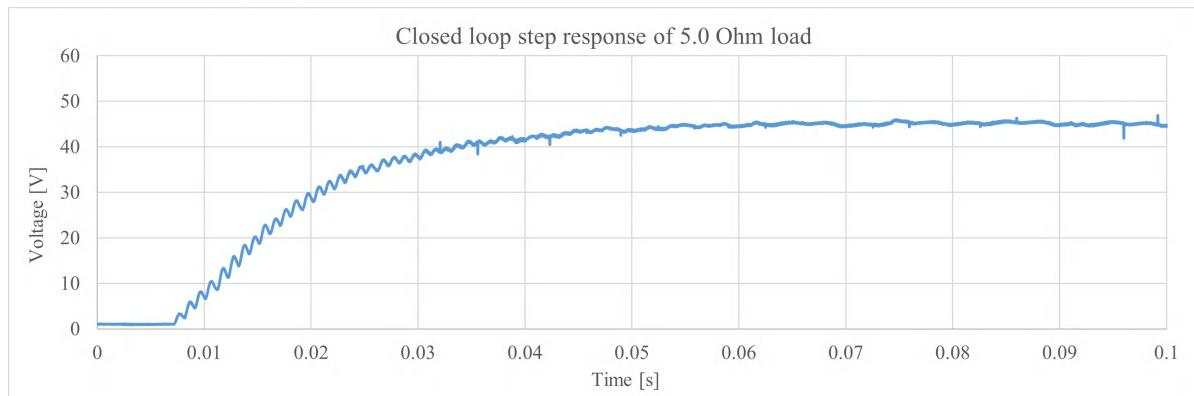


Figure 6.27: Step response of PV emulator with 5.0Ω load

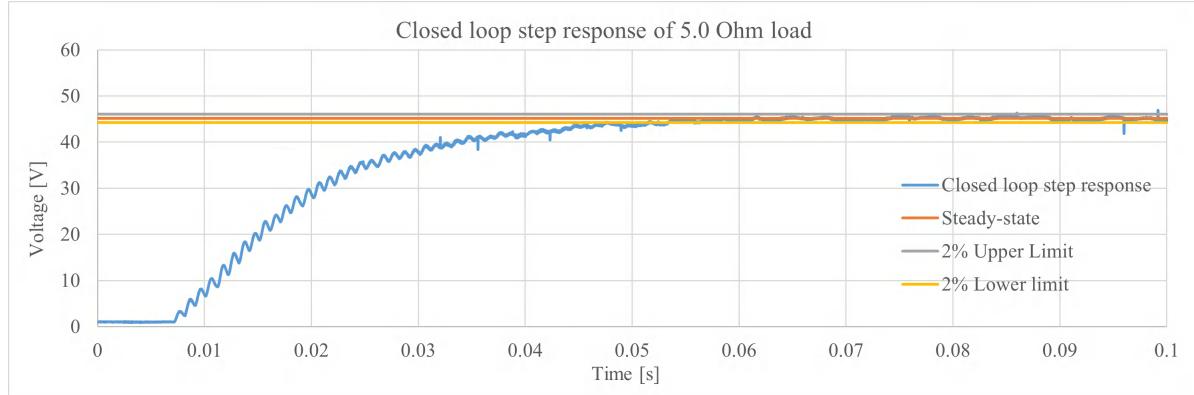


Figure 6.28: Step response of PV emulator with 5.0Ω load and boundaries indicated

Lastly, a 3.2Ω load is connected to the system, presenting the step-response displayed in figure 6.29. The system has a settling time of $78 ms$, along with 0% overshoot.

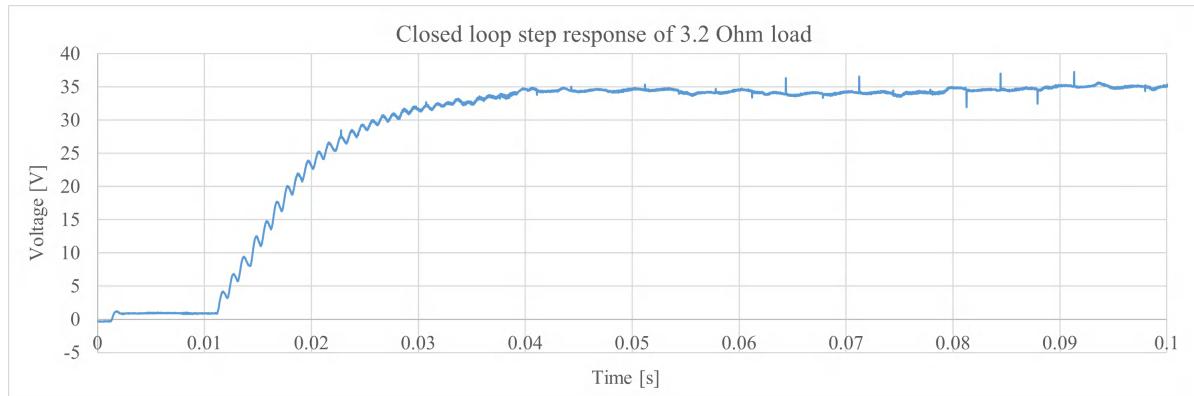


Figure 6.29: Step response of PV emulator with 3.2Ω load

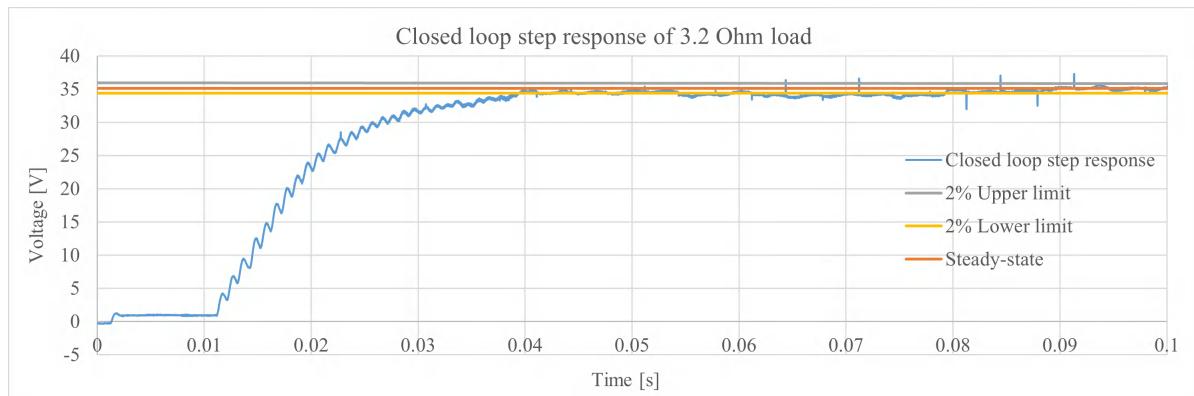


Figure 6.30: Step response of PV emulator with 3.2Ω load and boundaries indicated

After the evaluation, it is clear that the system is within the design specification (100 ms settling time, 0% overshoot) set out in appendix A, and chapter 5.

6.4.2 Dynamic Response

The dynamic response of the system is crucial to evaluate as the system is constantly subject to load variations. The dynamic response must also have a settling time of less than 100 ms , while its overshoot must also be zero. Figure 6.31 depicts the response of the system when the load is decreased. The system presents no overshoot and its settling time is 80.58 ms .

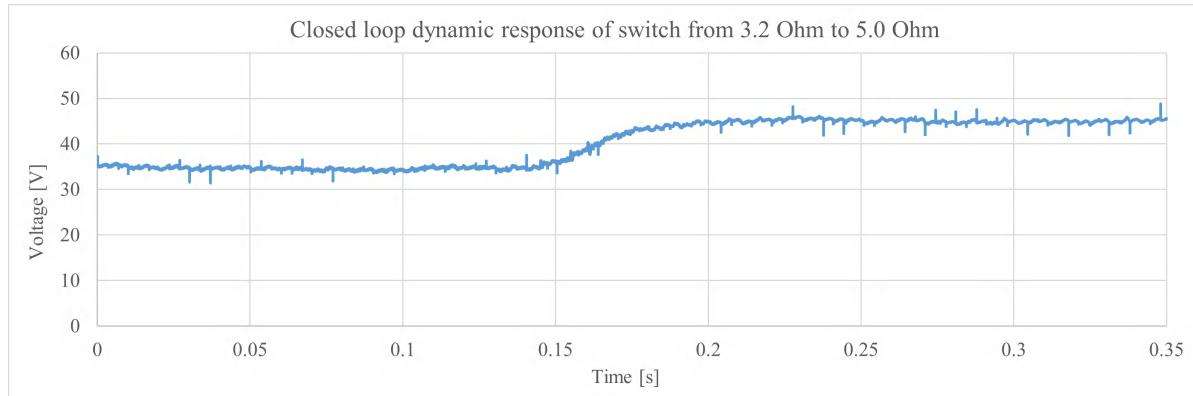


Figure 6.31: Dynamic response when load is decreased

Figure 6.32 displays the response of the system upon increasing the load on the system. Once again the overshoot is zero, while the settling time is 90.08 ms .

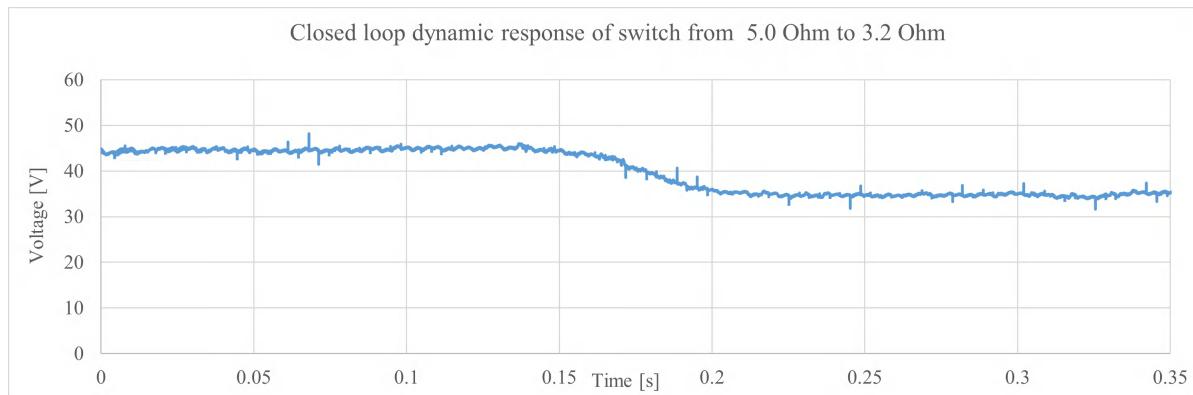


Figure 6.32: Dynamic response when load is increased

For both cases the system performs within specification, implying that implementation is successful

6.5 Conclusion

The implementation of all sub-systems pertaining to the PV emulator, along with their integration and evaluation was covered in this chapter. It is clear that the implementation was successful as all

design specifications were met. The PV emulator proved to be better than expected in various cases, yielding a steady-state error of 0.5% and less as opposed to the desired 1% value. The overshoot of the system was 0% in all cases, yielding a critically damped system as desired. The settling time was also within specification for all loads. The adaptive PI controller did exactly what it was supposed to do, controlling the output of the PV emulator at the reference value under any loading condition.

Chapter 7

Conclusions and Recommendations

This chapter concludes the project by providing the overview and analysis of the entire project.

7.1 Project Overview and Conclusion

7.1.1 Chapter 1

This chapter covered a background of the PV industry and the need for a PV emulator during the testing of MPPT devices. The factors affecting PV modules were also discussed. The problem statement, objectives, scope definition, deliverables, and project methodology followed.

7.1.2 Chapter 2

A discussion of existing research in the PV emulator field followed, and the definition of a PV emulator was derived. The PV emulator was broken down into three main components, namely; the PV model, power converter, and control strategy. For each of the main components various options were investigated and initial trade-offs were performed. The advantages and disadvantages of the main components were evaluated to familiarise the reader with each of the options. A brief discussion of the loads connected to PV modules is also included.

7.1.3 Chapter 3

In this chapter the conceptual design of the PV emulator was completed. The functional analysis, architecture, and operational flow followed. The context of the PV emulator was defined along with its sub-systems and interfaces. The high-level design was completed during this chapter, allowing for trade-off studies to be completed.

7.1.4 Chapter 4

The preliminary design followed, where the trade-off studies for technologies utilised were performed. For each trade-off study, various criteria were defined and a function was selected for each criteria. The technologies were scored and the best solution was selected. This allows an unbiased solution to be selected, which will best fit the final solution.

7.1.5 Chapter 5

This chapter was reserved for the detailed design of the PV emulator. The PV model was designed first which included a Genetic algorithm to perform the parameter extraction of the PV model. It was clear that the PV model included implicit equations, which were then solved using the Lambert W function to solve for the implicit exponential functions. The power converter was designed next, which consisted of a buck converter, able to output 800 W, with a maximum current draw of 20 A. The driver circuitry was also designed, responsible for driving the high-side switch of the buck converter through an optically isolated barrier. To retain the isolation barrier, the measurement circuit designed utilised linear optocouplers and operational amplifiers. The result was a fully optically isolated control loop. The control strategy's design followed, with an adaptive PI controller utilising fuzzy logic. Specifically, the Takagi-Sugeno fuzzy model was used along with Gaussian membership functions to fuzzify the proportional and integral constants of the PI controller. The transfer function of the buck converter was first derived, whereafter the PI controller could be designed based on the resistive load connected to the converter. The controller was designed to conform to the specification at both ends of a resistance range, which could then be utilised for the fuzzy model. The PI controllers were designed to operate in a critically damped state to ensure that the output power of the controller never overshoots the operating point. The resulting controller conformed to the design specification of a 100 ms settling time, and 0% overshoot.

7.1.6 Chapter 6

After the detailed design of the PV emulator and its sub-systems, the implementation followed. The PV model, power converter, and control strategy were implemented, along with the interfaces required to integrate the sub-systems. The performance and accuracy of the sub-systems were then evaluated with respect to the design by performing various experiments and data analysis.

The PV model was first, where the genetic algorithm was tuned with a sensitivity analysis to select its input parameters. The resulting I-V curves of the genetic algorithm were now evaluated with respect to the NOCT parameters of PV modules. The I-V curves yielded a maximum error of 1.102%, which is acceptable as many datasheets specify that a maximum error of 3% is present in their measurements. The implementation of the GUI followed, which was developed using PyQt5. The GUI is used to capture the PV module datasheet parameters and create the I-V curves. The GUI then displays the live operating points of the PV emulator on both the I-V and P-V curves.

The power converter was implemented and evaluated next. The buck converter was tested at a power output of up to 934.65 W, which was well above the design specification of 800 W. The output voltage ripple of the circuit was within its design value, boasting a low ripple of $\pm 0.437\%$. A negligible switching delay was introduced with the driver circuitry, therefore not affecting the control strategy. Next, the efficiency of the converter was evaluated, which revealed an efficiency of 82.79%. This was 7.79% above the required value, yielding a satisfactory design. The measurement circuits were characterised in order to integrate them into the PV emulator. Finally, the open-loop response of the buck converter was investigated to determine the accuracy of the transfer function used for the controller design. The correlation between the two was very strong, with the critical parameters, overshoot and settling time, being almost identical. It was therefore established that the controller should perform within specification.

The implementation of the control strategy followed, where the trapezoidal integration PI controller difference equation was derived. The integral windup was also addressed to ensure that the error of the controller does not accumulate when the control loop is broken. A filter was designed afterwards, since it was discovered that the inaccuracies of the ADC on the controller would threaten the successful implementation of the control strategy. A description of the embedded software followed to display the optimisation required for fast execution.

The integration followed where the PV emulator finally came to life. The sub-systems were integrated and the performance of the PV emulator as a whole followed. The steady-state errors proved to be within 0.5% on average during emulator. The closed-loop step- and dynamic-responses of the system was finally evaluated as it is the most important part of the emulator. The initial design sampling time was found to be extremely unstable. This is due to the hardware limitations of the ESP32, not being able to accurately sample fast enough. PI controller must often be tuned after design, therefore it was acceptable to adjust its parameters. The sampling frequency was reduced along with the integral constants, yielding a satisfactory controller. Both the step- and dynamic-responses proved to be within specification. The adaptive PI controller, utilising the Takagi-Sugeno fuzzy model proved to be fitting for the PV emulator, controlling its output as desired.

7.2 Future Recommendations

7.2.1 Analog to Digital Converter

A discrete analog to digital converter can be added to replace the ESP32's ADC in the PV emulator. This will allow an increase in both speed and accuracy of the PV emulator, since external ADC's are typically faster and more accurate. Reading the ADC values consumed the most time in the embedded software implementation, thus freeing computation time here will allow more samples to be taken and averaged before it is fed to the PI controller. This will significantly increase the accuracy and speed of the PV emulator.

7.2.2 Supply Circuit

It is recommended that a supply circuit is added to the PV emulator to allow it to be powered from a wall socket. This will also lead to an improvement in the accuracy of the system since the input voltage to the buck converter will be more stable. The user friendliness of the system will also improve, since a large DC source is not required to power the PV emulator.

7.2.3 Partial Shading

The PV model can be expanded to include the ability to emulate partially shaded conditions. This allows the PV emulator to emulate more realistic conditions, allowing for better testing of MPPT algorithms.

7.3 Limitations

The main limitation within the project was the lack of an MPPT which removed the ability to test the PV emulator under practical conditions. Therefore, it is recommended that the PV emulator is tested with an MPPT next. The PV emulator should perform within specification since the dynamic response of the emulator performs as designed.

7.4 ECSA Exit Level Outcomes

GA1

The problem is identified, analysed and constrained in chapter 1, while the criteria for an acceptable solution is set out in appendix A and chapter 3. The necessary information and knowledge is gathered and synthesised in chapter 2. Possible approaches to the solution of the problem were formulated in appendix A, which also included the analyses of the possible solutions. Finally the best solutions were evaluated and selected in chapters 3, 4, and appendix A. The problem and its solution is finally present in this document and through demonstrations.

GA3

The problem is identified and constrained in chapter 1, whereafter the high-level design is completed in chapter 3 to address the needs of the user. The planning of the design process occurs in chapter 1, where a methodology is selected, and then the planning continues in chapter 3 where the high level design is set out. The important issues and constraints are acknowledged and addressed throughout the design phase of the project. Chapter 2 covers the acquisition of requisite knowledge, information, and resources, while design tools are used throughout chapters 3 to 5. Also sufficient analysis, modelling, and optimisation was an integral part of the entire project. Alternatives were specifically evaluated in chapter 4, but some high-level trade-offs were performed in chapter 2. In all trade-off studies technical factors as well as economic factors were considered.

GA6

Effective written communication is demonstrated throughout this document, by using the correct structure, style, and language for the audience (my peers, supervisor, and external moderator). Concepts and explanations were graphically supported, allowing for simple explanations of algorithms, software, mathematics, circuits, functionality, etc. This document can easily be used for future reference, serving as a source of information for others involved in engineering activity. Lastly, the requirements of the target audience, set out in appendix A, were successfully met as proved in chapter 6.

GA8

The project was executed individually, starting with the problem and objective identification in chapter 1. This was followed by the methodology, allowing for strategic work. All tasks were executed effectively and on time, as the project as a whole, along with its milestones, were submitted in completion on or before the required deadlines.

GA9

This chapter, along with the oral presentations presented alongside this document, served as a reflection on own learning. Information was sourced and evaluated during the entire project, but more specifically in chapter 2. The majority of this project was based on the accessing, comprehension, and application of knowledge acquired outside of formal instruction, which lead to independent learning. The project enabled the independent operation in a complex and ill-defined context which required a lot of responsibility and initiative.

Bibliography

- [1] *2020 Snapshot of Global PV Markets*. International Energy Agency, 2020.
- [2] Aradhna Pandarum, Gaoshitwe Lekoloane and Dominic Milazi. “Trends and statistics of Solar PV Distributed Generation in South Africa”. In: (2018).
- [3] C. A. Belhadj, I. H. Banat and M. Deriche. “A Detailed Analysis of Photovoltaic Panel HotSpot Phenomena based on the Bishop Model”. In: *14th International Multi-Conference on Systems, Signals & Devices (SSD)* (2017).
- [4] *550W MBB Half-cell Module*. Rev 1.0. JA Solar. JAM72S30 525-550/MR, 2018. URL: <https://www.eeusolar.com/pdf/JAM72S30-525-550-MR.pdf>.
- [5] Wikipedia. *V-Model*. 5th Oct. 2022. URL: <https://en.wikipedia.org/wiki/V-Model>.
- [6] J. A. Simpson and E. S. C. Weiner. *The Oxford English Dictionary*. Ed. by Clarendon Press Oxford. 18th. Oxford, Clarendon Press, 1st Jan. 1989.
- [7] T. Easwarakhanthan et al. “Microcomputer-controlled simulator of a photovoltaic generator using a programmable voltage generator”. In: *Solar Cells* 17 (1st Jan. 1986), pp. 383–390.
- [8] K. Khouzam and K. Hoffman. “Real-time simulation of photovoltaic modules”. In: *Solar Energy* 56.6 (1996), pp. 521–526.
- [9] R Ayop and C.W. Tan. “A comprehensive review on photovoltaic emulator”. In: *Renewable and Sustainable Energy Reviews* 80 (May 2017), pp. 430–452.
- [10] Q. Zeng, P. Song and L. Chang. “A PHOTOVOLTAIC SIMULATOR BASED ON DC CHOPPER”. In: *2002 IEEE Canadian Conference on Electrical and Computer Engineering*. 1st Jan. 2002.
- [11] H. Metwally and A. A. S. Mohamed. *Photovoltaic Water Pumping Systems*. Ed. by University of Zagazig Zagazig. 1st ed. LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. ISBN: 978-3-659-10290-5.
- [12] N. Rajasekar, N. K. Kumar and R. Venugopalan. “Bacterial Foraging Algorithm based solar PV parameter estimation”. In: *Solar Energy* 97 (Sept. 2013), pp. 255–265.
- [13] Y.T. Seo, J.Y. Park and S.J. Choi. “A Rapid I-V Curve Generationfor PV Model-based Solar Array Simulators”. In: *IEEE* (2016).
- [14] H. Abouobaida and E. B. Said. “Practical Performance Evaluation of Maximum Power Point-Tracking Algorithms in A Photovoltaic System”. In: *International Journal of Power Electronics and Drive System (IJPEDS)* (4th Dec. 2017).
- [15] Suwito et al. “Simple and Fast Response Photovoltaic PanelEmulator using Transistor Current Source”. In: *IEEE* (1st Jan. 2021).
- [16] H. A. Khawaldeh et al. “Performance Investigation of a PV Emulator UsingCurrent Source and Diode String”. In: *IEEE* (1st Jan. 2018).
- [17] E. Hayakwong and A. Matarach. “Design of a Low-Cost and Simple Solar Emulator for Laboratory Studies”. In: *Smart Electrical Systems & Technology* (1st Jan. 2021).

- [18] A. Azizi et al. "Impact of the aging of a photovoltaic module on the performance of a grid-connected system". In: *Solar Energy* 174 (Nov. 2018), pp. 455–474.
- [19] J. Bai et al. "Characteristic output of PV systems under partial shading or mismatch conditions". In: *Solar energy* 112 (Jan. 2015), pp. 41–54.
- [20] A. B. B. Abdelghani and H. B. A. Sethom. "Modeling PV installations under partial shading conditions". In: *SN Applied Sciences* 2.627 (2020).
- [21] S. M. Azharuddin et al. "A near accurate solar PV emulator using dSPACE controller for real-time control". In: *Energy Procedia* 61 (1st Jan. 2014), pp. 2640–2648.
- [22] *SLVP089 Synchronous Buck Converter Evaluation Module, User's Guide*. Rev 1.0. Texas Instruments. SLVP089, 1998. URL: https://www.ti.com/lit/ug/slvu001a/slvu001a.pdf?ts=1646933168024&ref_url=https%253A%252F%252Fwww.google.com%252F.
- [23] S. Mishra et al. "An Extremely Low-Cost Multi-Panel PV Emulator for Research and Education". In: *IEEE* (1st Jan. 2018).
- [24] Z. Zarkov and L. Stoyanov. "Emulator of PV Panels for Laboratory Studies". In: *IEEE* (1st Jan. 2019).
- [25] M. Alaoui, H. Maker and A. Mouhsen AND. H. Hihi. "Photovoltaic emulator of different solar array configurations under partial shading conditions using damping injection controller". In: *International Journal of Power Electronics and Drive System (IJPEDS)* 11.2 (1st June 2020), pp. 1019–1030.
- [26] J. Gonzalez-Llorente et al. "Simple and Efficient Low Power Photovoltaic Emulator for Evaluation of Power Conditioning Systems". In: *IEEE* (1st Jan. 2016).
- [27] L. L. O. Carralero et al. "PV Emulator Based on a Four-Switch Buck-Boost DC-DC Converter". In: *IEEE* (4th Dec. 2019).
- [28] D. Bui et al. "DC-DC Converter Based Impedance Matching for Maximum Power Transfer of CPT System with High Efficiency". In: *IEEE* (1st Jan. 2018).
- [29] C. Das, K. Mandal and M. Roy. "Design of PV Emulator Fed MPPT Controlled DC-DC Boost Converter for Battery Charging". In: *IEEE* (1st Jan. 2020).
- [30] R. Ayop and C. W. Tan. "An Adaptive Controller for Photovoltaic Emulator using Artificial Neural Network". In: *Indonesian Journal of Electrical Engineering and Computer Science* (1st Mar. 2017).
- [31] N. Moldovan, R. Picos and E. Garcia-Moreno. "Parameter Extraction of a Solar Cell Compact Model using Genetic Algorithms". In: *Spanish Conference on Electron Devices*. 11th Feb. 2019.
- [32] M. Tripathy, M. Kumar and P.K. Sadhu. "Photovoltaic system using Lambert W function-based technique". In: *Solar Energy* (10th Oct. 2017).
- [33] M. F. N. Tajuddin et al. "State Space Averaging Technique of Power Converter with Digital PID Controller". In: *TENCON* (1st Jan. 2009).
- [34] M. M. Garg et al. "An Approach for Buck Converter PI Controller Design using Stability Boundary Locus". In: *IEEE* (1st Jan. 2018).
- [35] J. Zhang et al. "Design and Realization of a Digital PV Simulator with a Push-Pull Forward Circuit". In: *Journal of Power Electronics* (1st May 2014).

- [36] C. L. Phillips, H. T. Nagle and A. Chakrabortty. *Digital Control System Analysis & Design*. 4th. Pearson Education Limited, 2015.

Appendices

- A: The System Specification Requirements Document
- B: Controller Design Data
- C: Genetic Algorithm Sensitivity Analysis
- D: Code Repository

SYSTEM REQUIREMENTS SPECIFICATION

PV Emulator (PVEM)

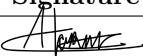
Mr. A.M. Horak

31625045

Document Identification

Project Title	PV Emulator (PVEM)
Document Number	SysRS_PVEM_V1_4
System/Subsystem	PV Emulator (PVEM)
Document Issue Date	05/10/2022
Client	Dr. M.G. Botha
Client Reference	EERI474-PVEM

Origination and Approval

	Individual Name	Signature	Date
Author	Mr. A.M. Horak		22/03/2022
Quality Assurance			
Technical Approval	Dr. M.G. Botha		
Project Manager	Prof. Febe de Wet		

Acceptance

	Individual Name	Signature	Date
Approved By	Dr. M.G. Botha		

Distribution list

Company	Individual Name	Date
North-West University	Dr. M.G. Botha	05/10/2022

Security Levels and Restrictions

Level	Description	Applicable Level
1	Strictly Confidential – not to be distributed	
2	Company Confidential – distributed inside company	
3	Client Confidential – distributed to limited clients and contractors	×
4	Public Domain – distributed freely	

Contact Information

Contact Person	Mr. A.M. Horak
Company	North-West University
Street Address	8 College Avenue, Potchefstroom, 2520
Telephone Number	0835072936
Email address	armand.horak@gmail.com

Contents

1	Introduction and Scope	1
1.1	Identification	1
1.2	Intended Use	1
1.3	Intended Users	1
1.4	Scope	1
1.5	Document Overview and Use	1
2	Applicable and Other Referenced Documents	1
2.1	Applicable documents	1
2.2	Other referenced documents	1
3	Acronyms, Abbreviations, and Meanings	2
3.1	Meanings	2
3.2	Acronyms	2
3.3	Abbreviations	2
4	Requirements	3
4.1	Identification of External Interfaces	3
4.1.1	Environmental Interface	3
4.1.2	Load Interface	3
4.1.3	Electrical Utility Interface	3
4.2	Identification of States and Modes	3
4.2.1	States	3
4.2.2	Modes	4
4.3	System Functions and Performance Requirement	4
4.3.1	Calculate PV model	5
4.3.2	Voltage Control	5
4.3.3	Steady-State	5
4.4	Relationship of States and Modes	5
4.5	System External Interface Requirements	6
4.5.1	Load Interface	6
4.5.2	Electrical Supply Interface	6
4.5.3	User Interface	6
4.5.4	Environmental Interface	7
4.6	Environmental Requirements	7
4.6.1	Operational Environment	7
4.6.2	Storage Environment	7
4.7	External Resource Requirements	7
4.8	Physical Requirements	7
4.9	Other System Qualities	7
4.10	Design and Construction Requirements	8
5	Notes	8
5.1	Verification Requirements	8
5.2	List of Safety Requirements	9

1 Introduction and Scope

This section covers an introduction to the PV emulator specifications and requirements.

1.1 Identification

This system requirements specifications acts as a guidance tool for the development of a PV emulator (PVEM) at the North-West University (NWU).

1.2 Intended Use

The intended use for the PVEM is to allow for type testing devices typically connected to PV modules. These devices are maximum power point trackers (MPPTs) or devices containing MPPTs. The PVEM is not intended to be used as a variable DC supply, but rather a non-linear supply with the output characteristics of a PV module.

1.3 Intended Users

The intended users for the PV emulator are manufacturers of MPPT devices which need to conformance test their devices without being restricted by environmental conditions pertaining to PV modules. The PVEM can also be used by anyone that needs to use PV as a source to test their equipment. Another use case is the educational sector where students can practically use the PVEM as a power source for their projects or where they can view how the ambient conditions and PV module parameters affect the I-V and P-V curves of the PV module.

1.4 Scope

The PVEM aims to emulate the terminals of a PV module by modeling it solely from the datasheet parameters, namely, V_{OC} , I_{SC} , V_{MP} , and I_{MP} at standard test conditions (STC). The PVEM then emulates the terminals of a PV module by delivering power to those terminals, based on the specified ambient conditions and connected load. The PVEM visually displays the operating point on the specific I-V or P-V curve to the user. The user determines the parameters of the PV module and enters it into the user interface contained in the PVEM.

1.5 Document Overview and Use

This document acts as a guidance tool for the specification requirements in the design of a PVEM.

2 Applicable and Other Referenced Documents

This section covers the applicable and other referenced documents pertaining to the PVEM as well as this document.

2.1 Applicable documents

None.

2.2 Other referenced documents

N/A

3 Acronyms, Abbreviations, and Meanings

This section covers the acronyms, abbreviations, and meanings along with their related definitions used in this document.

3.1 Meanings

The meanings and related definitions used in this document can be viewed below.

Meaning	Definition
Shall	expresses a binding requirement
May	expresses permissive guidance
State	expresses an observable condition of the system at any given time
Mode	expresses behavior exhibited by the system
Environment	the conditions, objects or circumstances surrounding an object or living being
Idle	when the system does not do or control anything

The Oxford English Dictionary shall be used in the interpretation of terms in Section 4, Requirements, which are not otherwise defined above.

3.2 Acronyms

The acronyms and related definitions used in this document can be viewed below.

Acronym	Definition
NWU	North-West University
GUI	Graphical User Interface
STC	Standard Test Conditions

3.3 Abbreviations

The abbreviations and related definitions used in this document can be viewed below.

Abbreviation	Definition
PV	Photo-voltaic
PVEM	PV emulator
I-V	Current-Voltage
P-V	Power-Voltage
SR-ID	System Requirement Identification Number

4 Requirements

This section covers the requirements of the system.

4.1 Identification of External Interfaces

This subsection covers the identification of external interfaces.

4.1.1 Environmental Interface

The environmental interface is the interface through which the PVEM interacts with the environment. This interface is the physical PVEM which is surrounded and influenced by environmental particles, temperatures and humidity levels.

4.1.2 Load Interface

The load interface is the interface through which power is supplied from the PVEM to the load.

4.1.3 Electrical Utility Interface

The electrical utility interface is the interface through which the PVEM is powered.

4.2 Identification of States and Modes

This subsection covers the states and modes of the PVEM.

4.2.1 States

The states of the PVEM are identified and discussed below.

Off State

In the off state the PVEM is unable to perform any functions, no power is drawn from the electrical utility interface, and no power is supplied to a connected load in this state.

On State

In the on state the PVEM draws power from the electrical utility interface, but does not perform any functions. No power is delivered at the output of the PVEM and thus no control loop regarding output power is active in the PVEM. In this state the PVEM is able to accept input parameters and user commands.

Emulating State

In the emulating state the PVEM delivers power to the output terminals. The operating point is constantly recalculated, according to the equivalent load, and used in combination with a control system to ensure that the PVEM does not deviate from the I-V curve.

Failed State

In the failed state the PVEM is unable to perform any functions until the failure that occurred is resolved.

4.2.2 Modes

The modes of the PVEM are identified and discussed below.

On Idle Mode

The PVEM waits for user commands in this mode. The user is able to enter the input parameters necessary for emulation into the GUI.

Calculation Mode

The PVEM generates the PV model and controller(s) based on the input parameters given by the user. The calculation should be completed within 1.5 seconds or less. The PVEM notifies the user via the user interface when the calculation has completed.

Display Graph Mode

The PVEM generates the I-V curve obtained from the PV model and then displays it to the user via the GUI.

Emulate Sense Mode

The PVEM waits for a stop emulating command from the user. The PVEM reads the necessary sensor values and detects if the load at the terminals of the PVEM has changed.

Determine Operating Point Mode

The PVEM determines if the operating point must move to the left/right from the PV model and sensed parameters.

Move operating point to the right Mode

The PVEM ramps up the voltage with a control loop until the new operating point has been reached.

Move operating point to the left Mode

The PVEM holds the current stable by decreasing the voltage and then slowly ramps up the voltage with a control loop until the new operating point has been reached.

Display Operating Point Mode

The PVEM displays the operating point of the PVEM on the I-V graph.

4.3 System Functions and Performance Requirement

The system functions and the related performance requirements are discussed here.

4.3.1 Calculate PV model

The PVEM, upon receipt of the “Generate PV Model” user command, shall generate the PV model within 5 seconds of receipt of the user command from the GUI for the specific user selected ambient conditions. [SR-ID 1](#)

4.3.2 Voltage Control

The PVEM, upon a change in load, shall control the load voltage within 20 ms of detecting a change in load at the load terminals by changing the PWM signal at the converter. [SR-ID 2](#)

4.3.3 Steady-State

The PVEM, upon a change in load, shall reach a steady-state within 100 ms of detecting a change in load at the terminals by first controlling the voltage and then converging towards the operating with a PWM signal at the converter. The final settling point must be within 1% of the operating point. [SR-ID 3](#)

4.4 Relationship of States and Modes

This section visualizes the relationship between the states and modes of the PVEM.

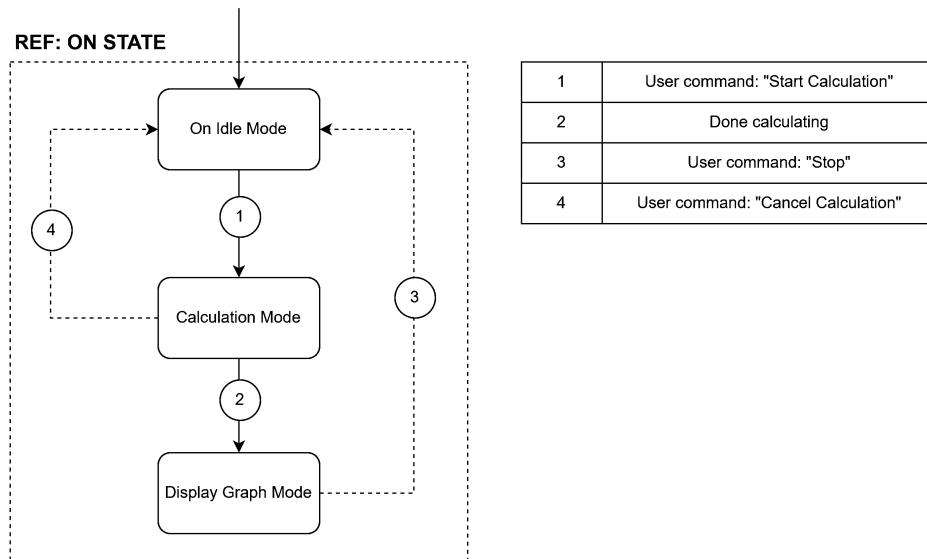


Figure 1: On State Modes

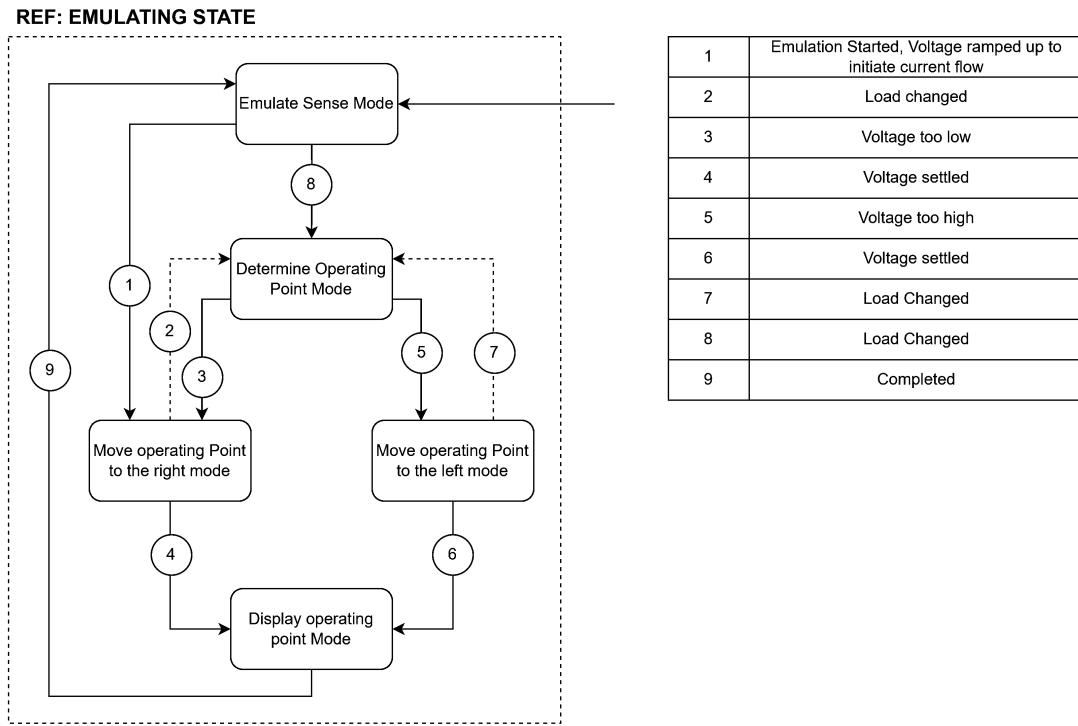


Figure 2: Emulating State Modes

4.5 System External Interface Requirements

This section contains the system external interface requirements.

4.5.1 Load Interface

The system shall supply power, up to a maximum of 800 W, to the load via the load interface. The output power was determined by a simple PV model with a 600 W solar panel at STC, and evaluated at 0°C, 1200 W/m². The maximum allowable current will be 20 A, and the maximum voltage 70 V, but never will both be at a maximum at a single instance. Thus, at 70 V a maximum of 11.43 A can be delivered and at 20 A, a maximum of 40 V can be delivered. High voltage and low voltage PV modules are available, represented by the 70 V and 20 A limits respectively. [SR-ID 4](#)

The load interface must provide a suitable and safe (not posing a shock hazard) manner for a load to be connected to the PVEM. [SR-ID 5](#)

4.5.2 Electrical Supply Interface

The system shall be powered from an 85 V DC source. The system shall be powered from a wall socket. The PVEM shall have an efficiency of at least 75%, thus, not drawing more than 1100 W from the electrical supply. [SR-ID 6](#)

4.5.3 User Interface

- The user interface shall provide a way for the user to turn the PVEM on or off. [SR-ID 7](#)
- The user interface shall provide a way for the user to input the parameters needed for the PV model. [SR-ID 8](#)

- The user interface shall visualize the I-V and P-V curves along with the operating point. [SR-ID 9](#)
- The user interface shall provide a way to indicate warnings, failure, and successful operation. [SR-ID 10](#)

4.5.4 Environmental Interface

The system shall interact with the environment via a temperature sensor and cooling mechanism. This will cause displacement of hot air out of the PVEM in order to sufficiently cool the components within the PVEM. The PVEM must be able to shut down when temperatures exceed 75°C . The operating environment must not exceed 50°C . [SR-ID 11](#)

4.6 Environmental Requirements

This section covers the environmental requirements of the PVEM.

4.6.1 Operational Environment

The system shall maintain functionality within the operational environment for all temperatures between 0°C and 50°C and humidity levels between 0% and 50% relative humidity. The PVEM must not be operated in environments that are very dusty. The PVEM must not become wet at all. [SR-ID 12](#)

4.6.2 Storage Environment

- The PVEM should be stored in an environment where the relative humidity does not exceed 50%. [SR-ID 13](#)
- The PVEM should be stored in an environment where the temperature does not exceed 50°C . [SR-ID 14](#)
- The PVEM should not be stored in direct sunlight. [SR-ID 15](#)
- The PVEM should not be used to support other objects. [SR-ID 16](#)
- The PVEM should not be stored on top of fragile/soft objects. [SR-ID 17](#)

4.7 External Resource Requirements

- The PVEM shall have an efficiency of at least 75%, implying that an 800 W converter will draw no more than $1066.66 \text{ W} + 33.34 \text{ W}$ from the electrical supply interface. [SR-ID 6](#)

4.8 Physical Requirements

- The PVEM should not pose a shock hazard when touched. [SR-ID 18](#)
- The PVEM must not be too heavy. [SR-ID 19](#)
- The PVEM must not be too large to handle. [SR-ID 20](#)
- Exposed parts of the PVEM must not exceed 5°C above the ambient temperature. [SR-ID 21](#)
- The PVEM must not make an audible sound regarding to the switching elements. [SR-ID 22](#)

4.9 Other System Qualities

- The PVEM shall be user friendly and safe to use for individuals trained to in its operation. [SR-ID 23](#)

4.10 Design and Construction Requirements

The PVEM shall consist of the following subsystems:

- The GUI
- The PV Model
- The Power Converter
- The Control System for the Power Output
- The Temperature Control Unit

The GUI is responsible for accepting the input parameters, commands, and visualizing the I-V curves and operating points of the PVEM. The GUI also communicates the necessary warning and info messages to the user. It is crucial that the GUI is user friendly and that the GUI visualizes the curves that the user wants to see such as the I-V and P-V curves. The GUI will be contained on some form of computer, therefore the computer on which the GUI is situated must consist of a display, thus have an operating system, and be able to run software being capable of running GUIs. [SR-ID 24](#)

The PV model is a mathematical model that will run some form of converging algorithms, which typically require quite powerful processing power for quick calculations. Thus, a sufficiently capable controller must be employed. [SR-ID 25](#)

The control system must be able to control the output voltage. [SR-ID 26](#)

All of the wiring contained in the sub-systems must be sufficiently insulated and protected against environmental elements. Cable management must be done properly, and no wires should come in place of any moving parts. [SR-ID 27](#)

5 Notes

This section covers the verification and safety requirements for the PVEM. The verification requirements will ultimately be used to evaluate a successful design of the PVEM, while the safety requirements place additional constraints on the design of the PVEM.

5.1 Verification Requirements

All above mentioned requirements should be addressed. The PVEM should be demonstrated and tested as a measure of verification. The testing phase should include the following requirements:

- The PVEM must successfully converge to the operating point within 100 ms of a variation in load. (Testing for [SR-ID 3](#))
- The PVEM must, when steady-state has been reached, provide an output corresponding to an operating point within 1% of the desired operating point. (Testing for [SR-ID 3](#))
- The PVEM must not consume more than 1.1 kW of power from the electrical supply interface. (Testing for [SR-ID 6](#))
- All components within the PVEM must maintain temperatures lower than 75°C during operation. (Testing for [SR-ID 11](#))
- The PVEM must calculate the PV model within 5 seconds. (Testing for [SR-ID 1](#))
- The PVEM must be able to supply 800 W of power to the load, with a maximum voltage of 70 V and a maximum current of 25 A. (Testing for [SR-ID 4](#))

- The switching elements in the PVEM must not be audible. (Testing for [SR-ID 22](#))
- The PVEM must provide output power to the load terminals only when in the emulation state.
- The PVEM must only be able to start the emulation process once the user has entered all required parameters.

Each of these verification requirements will be scored based on a balanced scorecard. The PVEM will then obtain a score for each requirement to determine how well each of these requirements were met.

5.2 List of Safety Requirements

- The PVEM must not pose a shock hazard when touched. [SR-ID 28](#)
- Any moving parts within the PVEM must not pose a danger to the surrounding environment. [SR-ID 29](#)
- The PVEM must not be a fire hazard. [SR-ID 30](#)
- No loose electrical connections may be present within the PVEM. [SR-ID 31](#)
- The PVEM must have sufficient electrical isolation between sub-circuits to ensure the safety of electrical components. [SR-ID 32](#)

Document History

Issue	Date	Status	Filename
Version 1.0	22/03/2022	Created	SysRS_PVEM_V1_0
Version 1.2	10/04/2022	Released	SysRS_PVEM_V1_2
Version 1.3	20/04/2022	Released	SysRS_PVEM_V1_3
Version 1.4	05/10/2022	Released	SysRS_PVEM_V1_4

Revision History

Issue	Date	Status	Filename
Version 1.0	22/03/2022	Created	SysRS_PVEM_V1_0
Version 1.1	27/03/2022	Edited	SysRS_PVEM_V1_1
Version 1.2	10/04/2022	Edited	SysRS_PVEM_V1_2
Version 1.3	20/04/2022	Edited	SysRS_PVEM_V1_3
Version 1.4	05/10/2022	Edited Requirements	SysRS_PVEM_V1_4

Authorization History

Issue	Date	Status	Filename
Version 1.0	22/03/2022	Created	SysRS_PVEM_V1_0
Version 1.2	10/04/2022	Authorized	SysRS_PVEM_V1_2

Appendix B

Controller Design Data

B.1 Tuned Controller Raw Data

Table B.1 displays the raw data for the tuned PI controller for each interval.

Table B.1: Tuned PI controllers at different operating points

Interval $[\Omega]$	K_P	K_I	P.O. LB [%]	P.O. UB [%]	T_s LB [ms]	T_s UB [ms]
$0.213 < R \leq 0.289$	$6.63e - 05$	4.0049	0	0	0.0095	0.0100
$0.289 < R \leq 0.452$	$7.68e - 05$	4.9758	0	0	0.0073	0.0080
$0.452 < R \leq 0.5$	$5.44e - 05$	7.0127	0	0	0.0048	0.0050
$0.5 < R \leq 0.833$	$4.99e - 05$	7.6399	0	0	0.0043	0.0050
$0.833 < R \leq 1$	$6.90e - 05$	9.9956	0	0	0.0035	0.0035
$1 < R \leq 1.5$	$1.34e - 05$	9.9242	0	0	0.0038	0.0040
$1.5 < R \leq 1.82$	$8.62e - 06$	9.8939	0	0	0.0040	0.0040
$1.82 < R \leq 2$	$8.83e - 05$	9.9369	0	0	0.0040	0.0040
$2 < R \leq 2.5$	$7.98e - 053$	9.8579	0	0	0.0040	0.0040
$2.5 < R \leq 2.813$	$5.063e - 05$	9.9623	0	0	0.0040	0.0040
$2.813 < R \leq 3.125$	$1.14e - 05$	9.9228	0	0	0.0040	0.0040
$3.125 < R \leq 3.81$	$7.23e - 05$	9.8944	0	0	0.0043	0.0040
$3.81 < R \leq 4.5$	$4.61e - 05$	9.7464	0	0	0.0050	0.0050
$4.5 < R \leq 5.95$	$7.90e - 07$	8.1779	0	0	0.0050	0.0050
$5.95 < R \leq 6.125$	$1.37e - 07$	8.1433	0	0	0.0060	0.0060
$6.125 < R \leq 8.57$	$1.34e - 08$	6.9751	0	0	0.0063	0.0063
$8.57 < R \leq 10$	$1.17e - 09$	6.5670	0	0	0.0070	0.0070
$10 < R \leq 11.67$	$9.09e - 09$	6.0650	0	0	0.0090	0.0090
$11.67 < R \leq 22.5$	$5.82e - 07$	4.8275	0	0	0.0105	0.0103
$22.5 < R \leq 40$	$3.76e - 07$	4.1662	0	0	0.0115	0.0113
$40 < R \leq 62.5$	$2.97e - 08$	3.8236	0	0	0.0120	0.0120
$62.5 < R \leq 90$	$5.31e - 08$	3.7033	0	0	0.0123	0.0123
$90 < R \leq 122.5$	$2.23e - 07$	3.5562	0	0	0.0123	0.0123

Appendix C

Genetic Algorithm Sensitivity Analysis

Table C.1: Crossover Rate Sensitivity Analysis

Crossover Rate	Time 1 [s]	Time 2 [s]	Time 3 [s]	Time 4 [s]	Time 5 [s]	Average Time [s]
0.95	0.4357	0.2819	0.9144	0.3770	2.2004	0.8419
0.9	0.5600	0.6990	0.6642	4.4095	3.0450	1.8755
0.85	0.3980	0.2062	0.2720	1.3753	0.2283	0.4959
0.8	0.2007	0.3410	0.7090	0.4798	0.3887	0.4238
0.75	1.0084	1.4535	1.0483	0.1545	1.4885	1.0306
0.7	0.2100	0.4895	0.5889	1.3662	0.2375	0.5784
0.65	0.2271	0.5323	0.6751	0.4066	0.3834	0.4449
0.6	0.3419	0.7306	0.4874	1.5944	1.9753	1.0259
0.55	0.9004	1.8515	1.8337	0.2981	1.1941	1.2155
0.5	1.2353	0.5331	0.4710	1.7645	0.2870	0.8582

Table C.2: Mutation Rate Sensitivity Analysis

Mutation Rate	Time 1 [s]	Time 2 [s]	Time 3 [s]	Time 4 [s]	Time 5 [s]	Average Time [s]
0.05	0.3276	0.5290	1.8589	0.4669	2.1448	1.0655
0.1	2.1256	2.4733	0.6910	2.3510	2.4356	2.0153
0.15	2.7542	2.8018	2.4410	2.9667	0.7903	2.3508

Table C.3: Selection Fraction Sensitivity Analysis

Selection Fraction	Time 1 [s]	Time 2 [s]	Time 3 [s]	Time 4 [s]	Time 5 [s]	Average Time [s]
0.3	2.0761	0.5910	0.6410	2.3652	2.4456	1.6238
0.35	1.6924	0.4761	1.9337	1.9840	0.5310	1.3234
0.4	0.2914	0.4800	0.9090	0.5380	0.4670	0.5371
0.45	0.3355	2.6765	1.8745	1.9699	0.3480	1.4409
0.5	0.4044	2.6441	2.3405	0.4750	0.5150	1.2758

Table C.4: Population Size Sensitivity Analysis

Population Size	Time 1 [s]	Time 2 [s]	Time 3 [s]	Time 4 [s]	Time 5 [s]	Average Time [s]
50	1.4043	0.4351	0.5260	0.2400	0.6060	0.6423
100	0.9256	0.4170	1.0182	0.2111	0.4090	0.5962
150	1.5275	0.2490	0.9600	0.5600	1.5981	0.9789
200	0.4161	0.7190	1.9289	0.5060	0.3880	0.7916
250	2.6380	0.8790	0.5170	2.7540	0.5930	1.4762

Appendix D

Code Repository

The software developed in the design and implementation of the PV emulator can be found using the link provided below. A README file is included which describes the files and their use.

<https://github.com/armand-horak/PVEmulator.git>