# Control Strategies for Reconfigurable PV Modules

Devyani Salokhe

TU Delft

# Control Strategies for Reconfigurable PV Modules

by

Devyani Salokhe

to obtain the degree of Master of Science in Sustainable Energy Technology

at the Delft University of Technology,

to be defended publicly on Wednesday August 18, 2021 at 3:00 PM.


| | | | |
|---|---|---|---|
| Student number: | 5046238 | | |
| Project duration: | November 16, 2020 - August 18, 2021 | | |
| Thesis committee: | Dr. Rudi Santbergen | Assistant Professor, PVMD | Supervisor |
| | Prof. Dr. Arthur Weeber | Professor, PVMD | |
| | Dr. Patrizio Manganiello | Assistant Professor, PVMD | |
| | Dr. Gautham Ram Chandra Mouli | Assistant Professor, DC&S | External committee member |
| | Dr. Malte Ruben Vogt | PostDoc, PVMD | Daily supervisor |


*This thesis is confidential and cannot be made public until December 31, 2023.*

TU Delft

# Preface

This thesis concludes my Sustainable Energy Technology master's program at Delft University of Technology. The experience has been filled with great learning at the backdrop of some unprecedented times.

First, I would like to thank everyone who played a key role in the progress and development of this thesis. I would like to thank *Dr. Patrizio Manganiello*, *Dr. Rudi Santbergen* and *Dr. Malte Ruben Vogt* for their contribution to the project. Every interaction with you all was insightful and valuable for developing the research work further. I am also grateful for all the support provided by *Andres Calcabrini*. Your work and guidance has been instrumental for the thesis project.

I also wish to extend my gratitude towards my family and friends for being a constant moral support throughout my studies. I am forever indebted to my parents and my brother for believing in me and supporting me in all my endeavours. And of course a special thank you to my friends in Delft, who made this city feel like a home away from home!

As I conclude this chapter of my life, I am extremely grateful for the knowledge I have acquired during this journey. I hope to utilise it and do my part towards the betterment of the world.

*Devyani Salokhe*
*Delft, August 2021*

# Abstract

PV systems in urban environments frequently get shaded by nearby objects, which greatly reduces their yield. A potential solution for such situations is the use of reconfigurable PV modules.

Reconfigurable PV modules are a type of modules which can reconfigure themselves under varying illumination conditions to ensure optimum yield at all times. The individual PV cells of the module are grouped into reconfigurable units/ cell blocks which can be interconnected in various ways to form a variety of unique configurations. The modules have an inbuilt algorithm which is responsible for controlling the reconfiguration process.

There are various kinds of reconfiguration algorithms found in literature, each working with varying input parameters and operating principles. One such algorithm is the short circuit current sensing (SCCS) algorithm developed by the PVMD research group at TU Delft. The algorithm is meant for series-parallel connected reconfigurable modules and uses each cell block's short circuit current as input. It is a synchronous reconfiguration algorithm, which means it runs at regular time intervals and not when there is actually a change in the module's irradiance conditions. Therefore, the aim of this thesis was to develop a robust reconfiguration algorithm with a shade detection mechanism.

Consequently, numerous versions of a reconfiguration algorithm with shade detection mechanism were defined and tested using a simulation framework for modelling reconfigurable PV modules. It was observed that the reconfiguration algorithm that performed the best in terms of yield, reconfiguration count and accuracy was the one referred to as the "CC4AP+IR algorithm". The algorithm used the module's and a single cell block's operating current and voltage as inputs for the shade detection and its highest DC yield for a single day was calculated to be 98.9% of an ideal scenario's yield. In comparison to the SCCS algorithm being run at 5 minute interval, it ran only 34 times in the day while the SCCS algorithm ran 176 times in the day. Hence, despite having a low reconfiguration count, the DC yield of the new algorithm was comparable to the SCCS algorithm.

The new algorithm was also tested using real-life data from an experiment involving reconfigurable modules. This experiment indicated that if the module were to be run using the new reconfiguration algorithm instead of the SCCS algorithm, it would have run only 6% of the time that the SCCS algorithm ran, but have a comparable DC yield. The significantly lower reconfiguration count is relevant because in practice, the PV modules are connected to power converters which will not be able to tolerate the high fluctuations in output owing to frequent reconfigurations.

The final output of this thesis is in the form of a simulation framework for reconfigurable PV modules and a new asynchronous reconfiguration algorithm with a shade detection mechanism.

# Contents

# 1

# Introduction

The world is relying increasingly on renewable energy resources, out of which solar energy plays a significant role. According to International Energy Agency, solar PV was set to have the fastest growth amongst all renewable energy technologies in 2020 [1]. They also published figures which show that in 2019, one-fifth of all renewable capacity installed was in the form of small and medium-scale commercial/residential PV systems [1].

One of the major differences between utility-scale PV systems and residential/commercial PV systems is that the residential systems are installed in environments with a lot of objects casting shadows on the modules. Due to this, their energy output is reduced greatly. A solution for this is the use of reconfigurable PV modules, which adapt themselves to provide optimum energy yield in all irradiance conditions.

The aim of this thesis is to develop a robust control strategy for such reconfigurable PV modules. The PVMD research group has already developed a control strategy for series-parallel type connected reconfigurable modules - the SCCS reconfiguration algorithm [2]. During this thesis, the objective is to further develop this control strategy, particularly in terms of a sensing mechanism which will help detect partial shading conditions.

In this chapter, section 1.1 introduces the concept of reconfigurable modules and summarizes the findings of the literature review done to understand the existing control strategies for reconfigurable PV modules. Based on the findings of the literature review, section 1.2 defines the scope of the thesis.

## 1.1. Literature review

This section serves as an introduction to reconfigurable modules. Subsection 1.1.1 gives a background of various technologies used for addressing partial shading in PV systems. Subsection 1.1.2 goes into the details of one such technology - reconfigurable PV modules. The next subsections - 1.1.3, 1.1.4 and 1.1.5 - give an overview of the different control strategies for reconfigurable PV systems with respect to their inputs, operating principle and switching circuit.

### 1.1.1. Improving shade tolerance of PV systems

Conventional PV modules have certain drawbacks when installed in urban settings. Modules installed on buildings are often partially shaded by nearby objects and end up receiving non-uniform irradiation. Although there have been varied ways of quantifying power loss due to shading in literature, various case studies have estimated shading losses as high as 10% in PV systems installed in urban areas [3]–[5].

In case of partial shading, a PV module's power output would be limited by its most shaded cell, but this would lead to a disproportionate decrease in power output [6]. Instead, in commercially available modules, a complete block of shaded cells are bypassed by the bypass diode. Bypass diodes are also important for mitigating the effects of partial shading, such as hot-spot formation [7]. In conventional

modules, the PV cells are divided into three blocks, each connected to a bypass diode. This means that just one shaded cell could result in the module's power output reducing to two third of its unshaded value [8].

Therefore, additional ways to address the problem of partial shading have been found in literature - such as module level power electronics (MLPE), parallel connection of PV arrays and reconfigurable PV arrays [2]. Each technology has a set of advantages and disadvantages, which are discussed below.

Module level power electronics (MLPE) can be classified into two categories: micro inverters and power optimisers [9]–[13]. MLPE enable MPPT tracking of modules independently. However, the disadvantage with MLPE is the higher cost and complexity of the systems.

Connecting PV modules in combinations other than an all-series topology has also shown to be beneficial for shade tolerance. Examples of such topologies are parallel (P), series-parallel (SP), total-cross-tied (TCT), honeycomb (HC) and bridge-linked (BL) (see figure 1.1). There is a wide range of literature studying the performance of these topologies under non-uniform irradiance conditions which show the significance of parallel connected topologies in limiting mismatch losses due to partial shading [14]–[18]. A disadvantage of these topologies is the high system current due to parallel connected modules, which will result in high power conversion losses.

A third type of solution for non-uniform irradiance conditions is the use of reconfigurable PV arrays [19], [20]. There are various types of reconfigurable PV systems in literature, most of them being for series-parallel (SP) or total-cross-tied (TCT) topology-based PV arrays [21]–[24]. These arrays will reconfigure in such a way that their power output is maximised in non-uniform illumination conditions.

The concept of reconfiguration could also be applied to module level, where the individual PV cells act as reconfigurable units. This would increase the granularity, and consequently shade tolerance, of the PV system. Subsection 1.1.2 will discuss the concept of reconfigurable modules further.

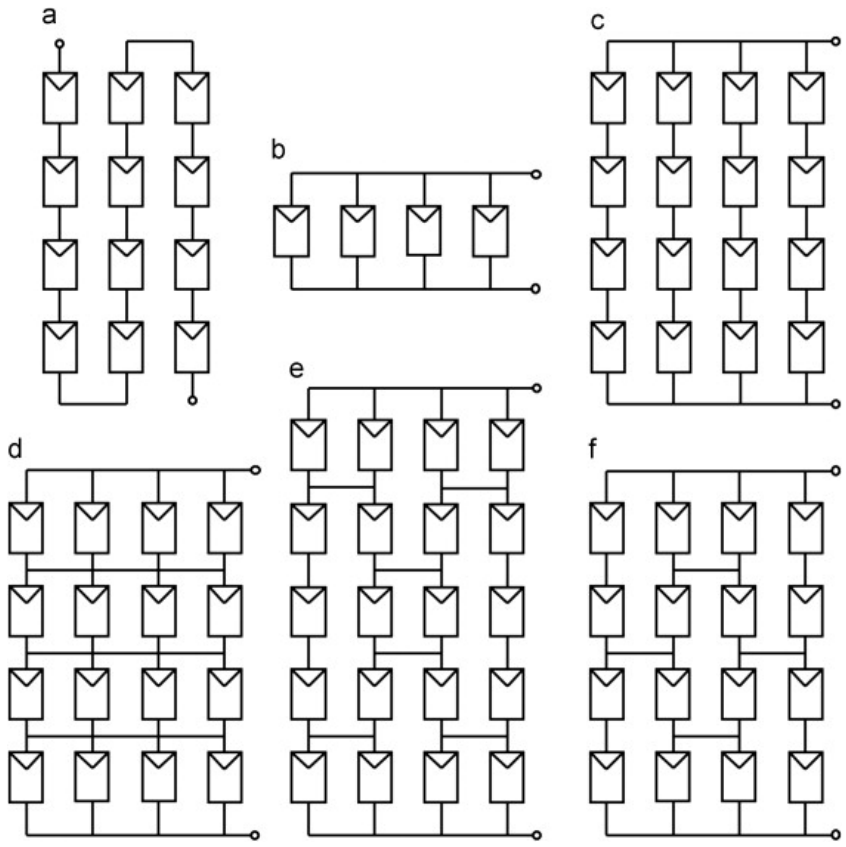



Figure 1.1: Figure showing the different types of PV array topologies: series (a), parallel (b), series-parallel (c), total-cross-tied (d), bridge-linked (e) and honeycomb (f) [25]

### 1.1.2. Reconfigurable PV modules

Reconfigurable modules are those which reconfigure in such a way that their power output is maximised in non-uniform illumination conditions. The optimum configuration is chosen based on module and weather data collected using sensors, after which the module's control strategy tries to detect if reconfiguration needs to take place. The best configuration is then determined for the module and a switching matrix is given the command to reconfigure it. The switching matrix is a set of switches which help reconfigure the module by connecting the individual reconfigurable units in the desired configuration.

The main aim of the thesis was to develop a reconfiguration algorithm which has a shade detection mechanism and triggers reconfiguration whenever required. Before this was pursued, existing reconfiguration algorithms found in literature were studied [25]–[31]. In particular, each control strategy was analysed using the following research questions:

- What is the input data that is required for the control strategy? Does it also have a shade detection mechanism?
- What is the algorithm used for deciding the optimal configuration of the PV module?
- What kind of switching matrix is required for each control strategy?

The following sections answer each of these questions for the control strategies that were studied. A point that needs to be noted is that most of these control strategies are defined for PV arrays, where reconfiguration can be done by modifying the interconnection between the individual PV modules. However, these control strategies can also be valid for the purpose of this thesis, where the individual PV cells act as the reconfigurable units in a single module.

### 1.1.3. Data acquisition

The control strategies that are studied mainly use the short circuit current, open circuit voltage and/or temperature of the modules in a PV array to determine the irradiance incident on each module. The variation lies in the number of sensors that would be required to procure this data.

Some control strategies use the current of the PV module as input data for the control algorithm. For instance, the control strategy developed in the PVMD research group by Calcabrini et al uses the short circuit current of each of the PV modules as input to the control algorithm [32]. Some more examples of control strategies with module current and temperature as input are the ones proposed by Velasco et al and Pattnaik et al [33]. Velasco et al propose a control strategy that needs irradiance of each module as input [34]. They use individual module's current and voltage to estimate the incident irradiance. The irradiance is calculated using the following equation:

$$G = \alpha * (I + I_0 * (e^{\frac{qV}{kT}} - 1)) \tag{1.1}$$

where $G$ is the estimated irradiance, $I$, $V$ and $T$ are the module's current, voltage and temperature. $\alpha$ and $I_0$ can be determined from the module short circuit current, open circuit voltage and maximum power point of the PV module (this data is available on the datasheet).

Pattnaik et al propose a control strategy in which the individual module's short circuit currents are measured and based on these values, the modules are grouped into categories and connected in series. They sense shading using a diode connected across each module - if the diode current is high, then module is shaded. If it is negligible then it is not shaded/partially shaded.

Some control strategies use the voltage of the PV module as input data for the control algorithm. Nguyen et al suggest a reconfiguration algorithm in which there is a fixed part and an adaptive part in the PV array [35]. The algorithm monitors the voltage $V$ of every row in the fixed part of the PV array. It also monitors the open circuit voltage $V_{OC,A}$ of every module in the adaptive part. Using these voltage values, the photogenerated current of each PV module is calculated using the following equations:

$$I_A = \frac{V_{OC,A}}{R_{SH}} + I_*^S * (e^{\frac{qV_{OC,A}}{akT}} - 1) \tag{1.2}$$

where $I_A$ is the photo-generated current of a PV module in the adaptive part, $V_{OC,A}$ is the open-circuit voltage of that PV module, $R_{SH}$ is the shunt resistance of the module

$$I_F = I_{out} + n * I_S * (e^{\frac{q}{akT} * (V + I_{out} R_{SM})} - 1) + \frac{V + I_{out} R_{SM}}{R_{SHM}} \tag{1.3}$$

where $I_F$ is the photo-generated current of a PV module in the fixed part, $I_{out}$ is the PV array's output current, $R_{SM}$ is the series resistance, $R_{SHM}$ is the shunt resistance of the solar cell, $I_S$ is the saturation current of the solar cell, and $n$ is the number of solar cells in the row.

### 1.1.4. Control strategy

The two most common interconnection topologies seen in reconfigurable PV arrays is series-parallel (SP) and total-cross-tied (TCT) topology [36]. Additionally, the control strategies are also different for each of the topologies. The following subsections gives some examples of control strategies for both topologies.

### Control strategy for TCT topology

TCT topology consists of PV modules connected in the form of an $m * n$ matrix ($m$ rows and $n$ columns), where $n$ modules are connected in parallel and $m$ such units are then connected in series. In case of partial shading, the row of modules which receives the least irradiation will limit the current of the entire array. Therefore, the underlying principle for all the control strategies observed for TCT topology is to choose the configuration which ensures that each of the rows receives approximately the same level of irradiation. This is done by choosing the configuration with the lowest equalization index. Equalization index can be represented as:

$$EI = max(G_{avg,1}, G_{avg,2} \ldots G_{avg,m}) - min(G_{avg,1}, G_{avg,2} \ldots G_{avg,m}) \tag{1.4}$$

where $G_{avg,1}, G_{avg,2} \ldots G_{avg,m}$ are the average irradiance values for the modules in each row of the $m * n$ PV array. An example illustrating this control strategy is shown in figure 1.2.

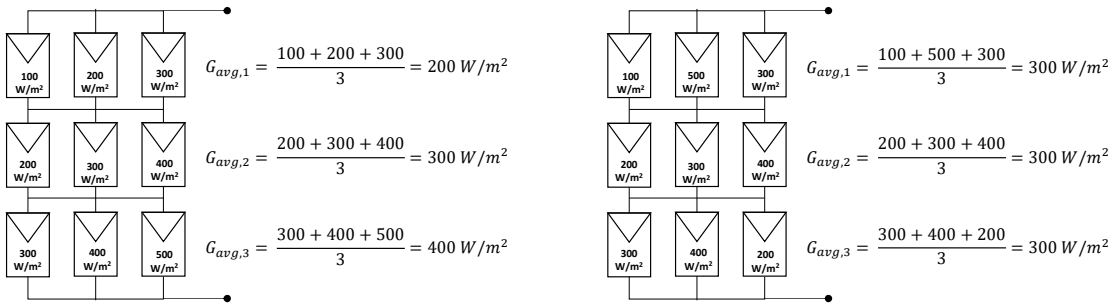



Figure 1.2: An example from [34], illustrating the reconfiguration of a 3*3 TCT PV array. The figure on the left shows the PV array before reconfiguration, the figure on the right shows the array after reconfiguration.

Velasco et al calculate the approximate irradiance received by each module using Equation 1.1 [34]. Then, considering an $m * n$ array, all possible $m * n$ configurations (which give unique power output) are determined. The configuration with the lowest equalization index is then determined as the optimal configuration. Before reconfiguring, however, the algorithm redetermines the optimal configuration 75 times. In case the optimal configuration is the same for the 75 runs, then reconfiguration is carried out. This helps in avoiding reconfigurations due to short-lived partial shading conditions, and also ensures a more efficient utilization of the switching matrix. One of the drawbacks of this method is that it needs to calculate all possible configurations of the PV array before narrowing down to the most optimal one. This could be computationally expensive for large-scale PV arrays.

Romano et al give a more efficient control algorithm as compared to the method suggested by Velasco et al because it does not need to determine all possible configurations and then narrow it down to the best one [37]. This algorithm, called the deterministic search algorithm, first arranges the modules in decreasing order of irradiance. The first $m$ modules are then placed one by one in each of the $m$ rows of the PV array. The remaining modules are then arranged in such a way that the one with the highest irradiance is placed in the row of already placed modules with the lowest irradiance. This leads to reduction of the computational time.

Nguyen et al suggest a control strategy similar to the deterministic search algorithm [35]. They calculate the photogenerated current of the modules using Equation 1.2 and Equation 1.3 and use that as input to the control algorithm. As they propose a system where most of the PV array has a fixed configuration while only a few of the modules are flexible, the number of switches and sensors required are lesser than other control strategies.

### Control strategy for SP topology

SP topology consists of PV modules connected in the form of an $m * n$ matrix ($m$ rows and $n$ columns), where $m$ modules are connected in series and $n$ such series strings are then connected in parallel. In case of partial shading, the module which receives the least irradiation will limit the current of the series string it is a part of. Therefore, all the control strategies for SP topology aim to combine similarly irradiated PV modules in one series string.

The control strategy developed by Calcabrini et al is for SP reconfigurable PV modules. As per this control strategy, the short circuit currents of the individual reconfigurable units are measured and sorted in descending order. The best module configuration is determined by comparing the difference between the short circuit currents and certain predefined threshold values [32]. For instance, in case the difference between the highest current value and lowest current value is very low, this implies that the module is receiving approximately uniform irradiation throughout. Therefore, all the reconfigurable units are connected in series. In case of partial shading condition, the difference between short circuit currents would be greater, and a different configuration will have to be selected, with the modules arranged in multiple series strings.

Some more examples of control strategies for SP reconfigurable PV modules were proposed by Alahmad et al and Pattnaik et al [38][33]. Both control strategies monitor the irradiance received by the PV modules, and classify the modules based on their irradiance levels. For instance, Pattnaik et all classify the modules as dark ($G < 400W/m^2$), grey ($400W/m^2 < G < 600W/m^2$) and bright ($600W/m^2 < G < 1000W/m^2$). They also propose monitoring the rate of change of short circuit current ($di/dt$) of the PV modules, particularly for the grey modules. In case $di/dt$ value is negative for a grey module, that means the module is transitioning from a grey to dark. Similary a positive $di/dt$ would imply that the module is transitioning from grey to bright. This is important to detect if the grey modules are transitioning to bright or dark modules. Once the modules are classified, the bright modules are connected in one series string, while the grey modules are connected in a separate series string. The dark modules are bypassed as they are generating insignificant amount of power. An example illustrating this control strategy is shown in figure 1.3.

## 1.1.5. Switching matrix

Once the control algorithm determines the optimum configuration for a PV array, the switching matrix acts as the actuator that implements the new configuration. The switching matrix will vary for different control strategies, both in terms of the kind of switches and the number of switches required. The table below gives an overview of the number of switches that will be needed for each of the control strategies mentioned in the previous sections.

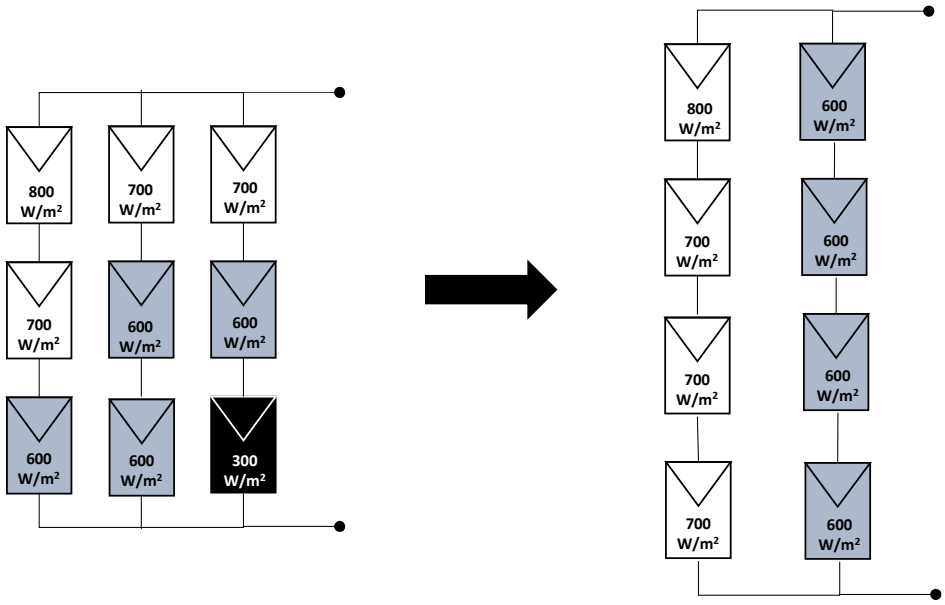



Figure 1.3: An example from [33], illustrating the reconfiguration of a 3*3 SP PV array. Figure on the left shows the PV array before reconfiguration, figure on the right shows PV array after reconfiguration. After reconfiguration it is now a 4*2 SP PV array

| Topology | Authors | Input parameters | No. of switches |
|---|---|---|---|
| TCT (considering $N_{PV}$ modules arranged in an m x n matrix, n modules in parallel per row, m such rows connected in series) | Velasco et al. [4] | Row voltage, current and temperature of the module | ***(2*N$_{PV}$ )*** m-throw switches |
| | Romano et al. [6] | Irradiance | ***(2*N$_{PV}^2$ + N$_{PV}$ )*** single throw switches |
| | Nguyen et al. [5] | Operating voltage of fixed part, open circuit voltage of adaptive part, temperature | ***(2*N$_{AB}$*m)*** single throw switches ($N_{PV}$ = no. of rows of fixed modules $N_{AB}$ = no. of reconfigurable modules) |
| SP (considering $N_{PV}$ modules arranged in an m x n matrix, n series strings in parallel, m modules per string) | Calcabrini et al. [2] | Module short circuit current | ***( (N$_{PV}$ - 1) + 0.5*(N$_{PV}$*(N$_{PV}$+1)) )*** single throw switches |
| | Alahmad et al. [7] | Irradiance | ***(2*N$_{PV}$ + 4)*** single throw + ***2*N$_{PV}$*** double throw switches |
| | Patnaik et al. [3] | Short circuit current and temperature of the module | Not specified |

## 1.2. Thesis scope and objective

Section 1.1 gave an introduction of reconfigurable modules and some examples of reconfiguration algorithms found in literature. It was observed that most algorithms relied on open circuit voltage/short circuit current of the individual reconfigurable units as input for the algorithm. However, using open circuit voltage and short circuit current would mean that the module will have to be periodically disconnected from the system for the reconfiguration algorithm to run.

Therefore, the objective of the thesis is **"To develop a robust control strategy for reconfigurable PV modules"** with a shade detection mechanism that will trigger reconfiguration whenever required. In order achieve this broad objective, the problem statement is divided into the following sub-parts:

1. **Modelling reconfigurable modules in the PVMD toolbox**
   PVMD toolbox is the PV energy yield simulation framework developed by the PVMD research group at TU Delft. The toolbox currently only allows simulation of conventional (non-reconfigurable) PV modules. Therefore, the first goal is to enable modelling reconfigurable PV modules in the toolbox. Elaborating further, this would involve first simulating partial shading/non-uniform irradi-

ance conditions necessary to observe the behaviour of reconfigurable modules, then using this data to simulate their I-V characteristics.

2. **Defining a shade detection mechanism for reconfiguration algorithm**
   Most control strategies, including the one developed in the PVMD research group [2], are synchronous and/or require short circuit current/ open circuit voltage of the modules as input. The aim is to devise an asynchronous algorithm with a shade detection mechanism, which will trigger reconfiguration only in the case of partial shading. Various module parameters will be explored for using as inputs to the new shade detection algorithm.
3. **Defining a reconfiguration algorithm using the shade detection mechanism**
   Using the shade detection mechanism defined in the previous sub-goal, a reconfiguration algorithm will be defined. The ideal algorithm should be optimised in such a way that it maximises the module yield, minimises reconfiguration count and has a good accuracy. Additionally, the number of sensors must be optimised such that a limited number of them are required for the algorithm to work efficiently.
4. **Validating developed control strategy**
   The final aim is to validate the control algorithm's efficiency by using experimental data. Its performance will be compared against the control strategy already developed in the PVMD research group. This will be used to ensure that the new algorithm indeed does function more efficiently than the already implemented control strategy.

## 1.3. Thesis outline

This chapter introduces the concept of reconfigurable modules and introduces numerous reconfiguration algorithms found in literature. The reconfiguration algorithms were studied based on their three main aspects - the input data required for the algorithms, their main operating principle and the switching circuit used for reconfiguration. The chapter also introduced two main types of reconfigurable modules in terms of their topology - the total cross tied (TCT) and series parallel (SP) reconfigurable modules (the work in this thesis is solely pertaining to reconfiguration algorithm for modules of the series-parallel topology). The chapter also established the scope of this thesis. It divided the main aim of the thesis into four sub-goals, which will be addressed in the next chapters.

The organisation of rest of the report is as follows: Chapter 2 describes the simulation framework that was established for modelling reconfigurable PV modules. Chapter 3 explores various module parameters that could be used as triggers for shade detection and also suggests a potential shade detection mechanism. Chapter 4 defines numerous reconfiguration algorithms using the proposed shade detection mechanism and determines the optimum one. Chapter 5 describes an experiment done to compare performance of the new reconfiguration algorithm with the SCCS reconfiguration algorithm. Finally, chapter 6 concludes the report by recapitulating how each sub-goal defined in section 1.2 was achieved and provides recommendations for future research.

# 2

# Modelling reconfigurable PV modules in PVMD toolbox

The current version of the PVMD toolbox is not capable of modelling reconfigurable modules and it can only simulate partial shading conditions in a limited manner (such as the ones caused by a row of modules in front). Therefore, this chapter will elaborate on the first sub-goal defined in section 1.2, which is to enable simulation of reconfigurable modules in the PVMD toolbox. Section 2.1 gives an overview of the simulation framework. The next sections describe in detail some aspects of the simulation framework - section 2.2 describes the methodology used to simulate partial shading conditions to test on the reconfigurable modules and section 2.3 describes the electrical simulation of the reconfigurable modules. Section 2.4 describes the reconfigurable module simulated in this thesis using the described framework.

## 2.1. Reconfigurable module simulation framework

The PVMD toolbox has 6 ”levels”, which model PV systems in a step-by step manner. Figure 2.1, inspired from the schematic given in Abdallah Nour El Din’s master thesis, shows a simple flowchart of the PVMD toolbox [39].

The CELL level calculates the optical properties of all the layers of the PV cell as a function of wavelength. This is done using the MATLAB-based ”GenPro4” software, which uses wave optics and ray optics to achieve this at low computational cost [40]. The output of the CELL level are the angle and wavelength dependent optical properties, which can be used to calculate the photocurrent density in the solar cell.

The MODULE level creates the sensitivity map of the PV module using ”Lux” software [41]. It uses the optical properties calculated in the CELL level as well as geometric dimensions and mounting angle of the PV module as input. Using these inputs, it calculates the sensitivity map of the module using forward ray tracing. A sensitivity map of a PV module is an indicator of its sensitivity to incident light. It can be used to calculate the Plane of Array (POA) irradiance incident on the module.

The WEATHER level then creates a sky map using irradiance data. This is done using the Perez model and module location’s DNI and DHI data. The integration of the sky map and sensitivity map gives the cell-wise POA irradiance. Using optical properties assigned at the CELL level, the WEATHER level also determines the cell-wise photocurrent density.

The THERMAL level calculates the cell-wise temperature using a fluid dynamic model. This level uses the photocurrent density calculated in the previous level, irradiance and weather conditions as input to calculate the cell temperatures.

The ELECTRIC level models the PV module’s IV curves and determines DC yield. Using the photocurrent density and the PV cell’s temperature coefficients, it uses the single diode model to generate

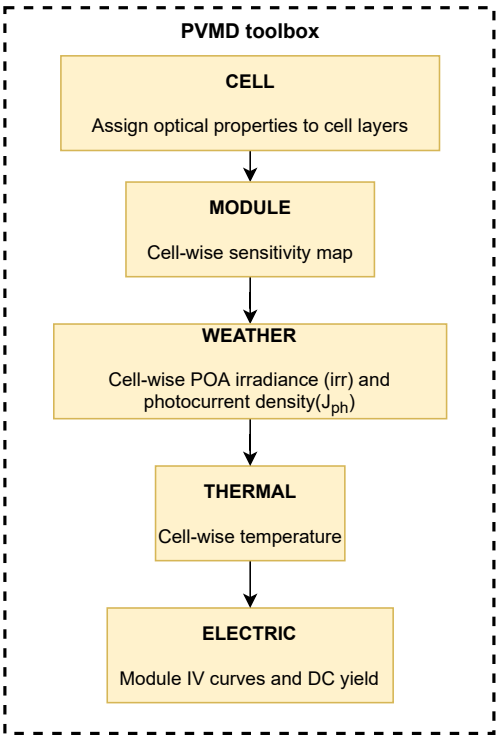



Figure 2.1: Schematic of the PVMD toolbox showing the simulation steps for a conventional (non-reconfigurable) module

numerous IV curves at varying irradiance and temperature conditions. These IV curves are used to determine the PV module's IV curve at each time step, from which the module's DC yield is calculated.

The current version of the PVMD toolbox is not equipped to simulate reconfigurable modules and test them under non-uniform irradiance conditions. Therefore, an alternate methodology was needed which would achieve what the CELL, MODULE and WEATHER level does, but for partial shading conditions. In this thesis, external softwares were used to simulate non-uniform irradiance conditions and calculate POA irradiance and photocurrent density of the PV modules (refer to section 2.2 for the detailed methodology). For electrical modelling of reconfigurable modules, a new simulation framework based on the control strategy defined in [32] was added to the ELECTRIC level to make this possible (refer section 2.3). Figure 2.2 shows a schematic summarising the new simulation framework established for modelling reconfigurable modules.

## 2.2. Simulating irradiance conditions

There are two main components of simulating non-uniform irradiance conditions. First is to simulate the PV module and its surroundings and second is to calculate the incident irradiance on the modules using the simulated scene. For this thesis, the first step of modelling the module and its surroundings was done using a 3D modelling tool called Rhinoceros 6 and the second part of calculating incident irradiance was done using MATLAB-based RADIANCE software.

Rhinoceros 6 is a CAD (Computer Aided Design) software which can be used for a wide range of applications [42]. Using Rhinoceros 6 and its plugins (Grasshopper and Ladybug), a 3D model of the PV array and its surroundings can be made and irrradiance simulation can then be done, as has been demonstrated in literature [43]. Grasshopper is a graphical algorithm editor that is used to assign optical

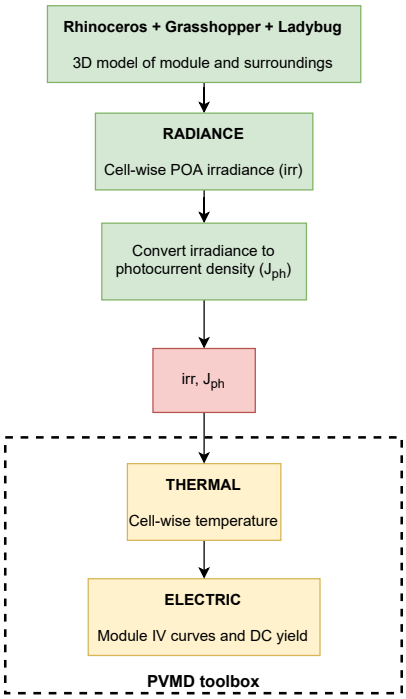



Figure 2.2: Simulation framework for reconfigurable PV modules using a combination of PVMD toolbox and external software

properties to the surfaces in the 3D model developed in Rhinoceros 6 [44]. Ladybug is a plugin which is used for modelling the irradiance conditions using the 3D geometry defined in Grasshopper and the climate file of the module's location [45].

Irradiance simulation in Grasshopper and Ladybug can be computationally intensive, so it is used only to simulate the 3D model of the module and its surroundings and assign them optical properties. The output of performing irradiance simulation in Rhinoceros+Grasshopper+Ladybug is in the form of the following data:

- A ".res" file which contains the resultant irradiance incident on each test point defined on the PV module.
- A ".sky" which contains the time duration of the simulation and its location's coordinates . It also gives the description of the light sources in the 3D model.
- Two ".rad" files containing information of the geometry and optical properties of the surfaces in the 3D model.
- A ".pts" file containing the x,y,z coordinates of each test point defined on the PV module. The test points are the points on the PV module where the incident irradiance of the PV module will be calculated.
- A ".msh" file, which is a binary file containing information about the geometry of the surroundings.

From these files, the ".rad", ".pts" and ".msh" files contain information about the geometry and optical properties of the PV module and its surroundings. These files, along with weather data, can be used as input for the MATLAB-based RADIANCE software, which is a tool for performing irradiance simulations [46]. RADIANCE does the irradiance calculation at all the test points on the PV module at a lower computational time as compared to Grasshopper and Ladybug. Additionally, since it is a MATLAB-based software, its output can be easily used for further computations in the MATLAB-based PVMD toolbox. The software employs backward-ray tracing to determine the cell-wise plane of array irradiance ($G_{POA}$). The time resolution of the irradiance data is same as that of the weather data provided.

Once the cell-wise irradiance is known, the optical properties calculated by the CELL module can

be used to accurately determine the cell-wise photocurrent. However, for the purpose of the thesis, approximate cell-wise photocurrent density was calculated using equation 2.1. Here, $rsd_f$ is the relative spectral photon flux density for AM1.5 spectrum and $G_{POA}$ is the plane of array irradiance.

$$J_{ph} = G_{POA} * cf * q * \sum rsd_f \tag{2.1}$$

Here $J_{ph}$ is the cell-wise photocurrent density ($A/m^2$), $G_{POA}$ is the cell-wise irradiance ($W/m^2$), $cf$ is the conversion factor used to account for the absorption and reflection losses in the cell, $q$ is the charge on a single electron ($1.6*10^{-19}$C) and $rsd_f$ is the relative spectral photon flux density for AM 1.5 spectrum.

The value of $cf$ was calculated as follows: Most commercial PV cells generate around 400 $A/m^2$ photocurrent density at 1000 $W/m^2$ irradiance [47]. Using this fact and substituting it in equation 2.1, the conversion factor is calculated as: $400/(1.6 * 10^{-19} * 1000 * \sum rsd_f) = 0.5358$.

## 2.3. Modelling reconfigurable PV modules

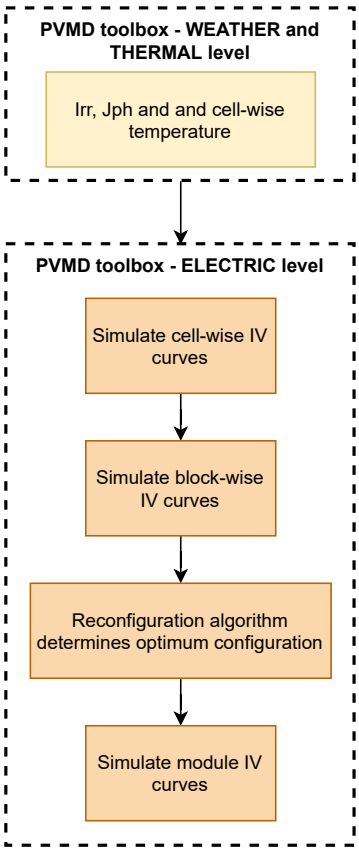



Figure 2.3: Simulation framework for electric modelling of reconfigurable modules

After calculating the cell-wise irradiance, photocurrent density and temperature, the cell-wise IV curves are determined. The PVMD toolbox does this using a single-diode model. In case of conventional modules, the PV cells are all connected in series. Hence, obtaining the module IV curve from the individual cell IV curves is relatively straightforward. On the other hand, reconfigurable modules can be interconnected in various different configurations, hence modelling of reconfigurable modules varies from that of conventional modules.

Reconfigurable modules can be made of a number of reconfigurable units or blocks, which are made of multiple cells in series/parallel. These individual cell blocks act as building blocks and can be connected in different ways to obtain multiple unique module configurations. Therefore, the first step to obtain the reconfigurable module's IV curves is to model the block-wise IV curves from the cell-wise IV curves.

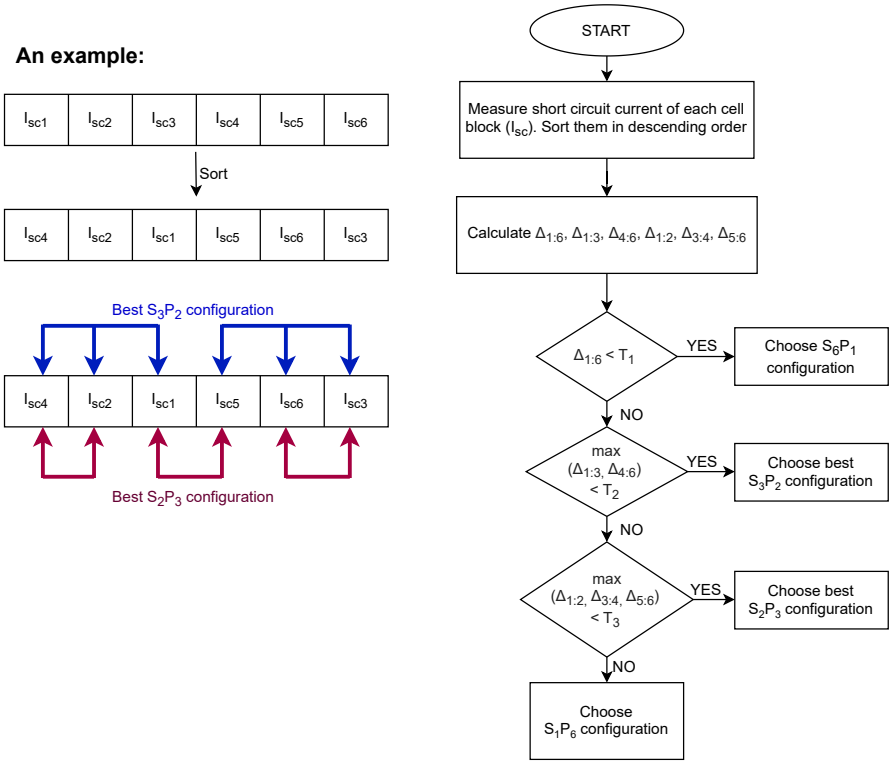



Figure 2.4: Flowchart explaining the SCCS reconfiguration algorithm with an illustrative example [2]

The block-wise IV curves will be used to model module IV-cures based on its configuration at each time step.

Figure 2.3 is a schematic showing how the functionality of modelling reconfigurable modules was added to the PVMD toolbox. Using cell-wise irradiance, photocurrent density and temperature, the individual cells' IV curves are simulated using the existing single diode model in the toolbox. The cell-wise IV curves are then used as input to model block-wise IV curves. Additional series resistances like shading resistance (to account for metallization losses) and resistance of switches are factored in for determining the block-wise IV curves. Since the reconfigurable module could be in a different configuration in each time instant, the module IV curve will depend on the module's configuration. The optimum configuration for each time instant is chosen by the module's control strategy. The current framework uses the control strategy defined in [2] for determining the optimum configuration for each time instant. Figure 2.4 shows the flowchart of the control algorithm. Hereon it will be referred to as the short circuit current sensing (SCCS) algorithm.

The SCCS algorithm is developed in the PVMD research group at TU Delft and is used for series-parallel type of reconfigurable modules. It uses the short circuit current of each cell block as an input. Given below are the steps of the algorithm explained using an example of a module with six cell blocks:

- The short circuit current of each cell block is the input to the algorithm. For a module with six cell blocks, this would mean that there are six short circuit current values.

- Arrange the short circuit current values in ascending order. As an example, consider the sorted order in figure 2.4. Here, the lowest short circuit current is of the fourth cell block and highest short circuit current is of the third cell block.

- Calculate the Δ values using equation 2.2.

$$\Delta_{x:y} = \frac{max[I_{sc,x}, I_{sc,y}] - min[I_{sc,x}, I_{sc,y}]}{max[I_{sc,x}, I_{sc,y}]} \tag{2.2}$$

Table 2.1 shows the Δ value calculations for the example in figure 2.4. As an example, $\Delta_{1:6}$ would be calculated using the first and last element in the sorted order of short circuit currents. These Δ values are an indicator of the relative difference between the sorted short circuit currents, and hence the relative difference between the irradiance incident on their corresponding cell blocks.

| $\Delta_{x:y}$ | Value |
|---|---|
| $\Delta_{1:6}$ | $\frac{I_{sc,3}-I_{sc,4}}{I_{sc,3}}$ |
| $\Delta_{1:3}$ | $\frac{I_{sc,1}-I_{sc,4}}{I_{sc,1}}$ |
| $\Delta_{4:6}$ | $\frac{I_{sc,3}-I_{sc,5}}{I_{sc,3}}$ |
| $\Delta_{1:2}$ | $\frac{I_{sc,2}-I_{sc,4}}{I_{sc,2}}$ |
| $\Delta_{3:4}$ | $\frac{I_{sc,5}-I_{sc,1}}{I_{sc,5}}$ |
| $\Delta_{5:6}$ | $\frac{I_{sc,3}-I_{sc,6}}{I_{sc,3}}$ |

Table 2.1: Table showing the Δ value calculations for the example in figure 2.4. As an example, $\Delta_{1:6}$ would be calculated using the first and last element in the sorted order of short circuit currents

- Compare the Δ values with the predefined thresholds $T_1$, $T_2$ and $T_3$, as shown in figure 2.4. The thresholds are determined based on a trade-off between two extremes - if the thresholds are set too high, then the module will not be very sensitive to the partial shading conditions and may not deliver maximum yield. If the thresholds are set too low, then the module might be in series-parallel configuration too often. As more parallel strings means higher module current and higher Ohmic losses, this is not favourable either. The literature referred for this algorithm determine optimum values for $T_1$ at 3%, $T_2$ at 5% and $T_3$ at 8% [2]. These threshold values were used for all simulations in this thesis.
- Based on the checks mentioned in figure 2.4, choose the best configuration. For the example of a six block reconfigurable module, there are 4 types of configurations - $S_1P_6$, $S_2P_3$, $S_3P_2$ and $S_6P_1$ (refer figure 2.5). If $\Delta_{1:6}$ is less than $T_1$, then the $S_6P_1$ configuration is chosen. If $\Delta_{1:6}$ is greater than $T_1$ and $\Delta_{1:3}$, $\Delta_{4:6}$ are less than $T_2$, the best $S_3P_2$ configuration is chosen. The best $S_3P_2$ configuration for the example in figure 2.4 can be seen marked in blue arrows. Similarly, the best $S_2P_3$, $S_1P_6$ configurations are chosen based on the checks shown in figure 2.4.

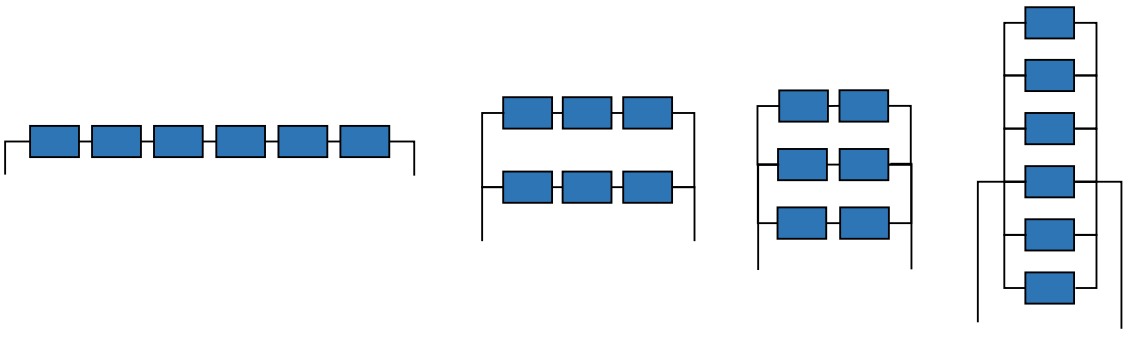


Figure 2.5: Figure showing all configuration types of a six block reconfigurable module. Each configuration type can be referred to using an $S_xP_y$ notation. Here $x$ stands for the number of cell blocks in series and $p$ stands for the number of cell blocks in parallel. In order (from left to right), this figure shows that the six block module can have a $S_6P_1$, $S_3P_2$, $S_2P_3$ or $S_1P_6$ configuration

The SCCS algorithm also has some drawbacks. Since the control algorithm requires short circuit current of each cell block as input, this leads to energy loss every time the cell blocks are disconnected to

measure the short circuit currents. Additionally, since the algorithm does not have any shade detection mechanism, it needs to be run at regular time intervals. The duration of time between two subsequent runs of the algorithm needs to be a trade-off between running the algorithm too often (even when reconfiguration is not needed) or not often, which might result in the system missing some time instants when reconfiguration was necessary.

## 2.4. Details of simulated reconfigurable module

For the purpose of this thesis, all simulations were done using a 96 cell module consisting of 6 cell blocks, which can be interconnected in different ways to obtain numerous series parallel configurations. For such a configuration, the number of electrically unique configurations of a single type can be calculated using the following equation [32]:

$$c = \frac{N!}{r! * (s!)^r} \tag{2.3}$$

where $c$ is the number of configurations of the type $S_xP_y$ ($x$ is the number of cell blocks in series and $y$ is the number of cell blocks in parallel). $N$ is the total number of cell blocks, $r$ is the number of parallel strings and $s$ is the number of cell blocks per string. A 6-block module has four types of configurations: $S_6P_1$, $S_3P_2$, $S_2P_3$ and $S_1P_6$. Using equation 2.3, there is one unique configuration for $S_6P_1$ and $S_1P_6$ each, 10 configurations for $S_3P_2$ and 15 configurations for $S_2P_3$. Therefore, there are 27 total unique configurations for a 6-block module. Figure 2.6 shows how the module's reconfigurable units are defined and figure 2.7 shows all possible ways they can be connected.

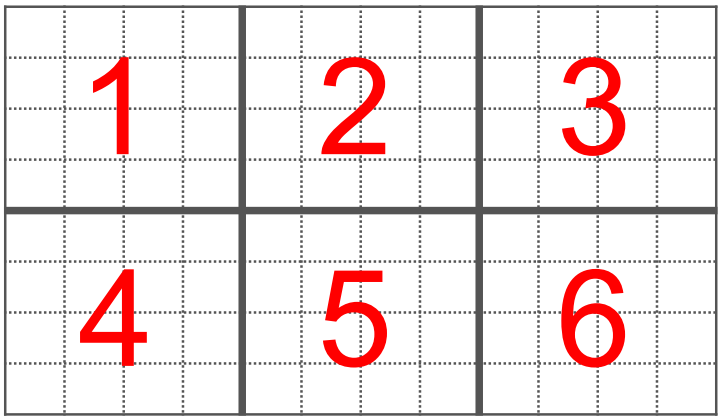



Figure 2.6: A 96 cell reconfigurable module with 6 cell blocks. The cell blocks are marked using red numbers. Each cell block is actually a set of 16 cells connected in series.

## 2.5. Conclusion

This chapter pertains to the first sub-goal of section 1.2 - establishing a simulation framework for modelling reconfigurable PV modules using Rhinoceros, RADIANCE and the PVMD toolbox. The simulation framework covers all steps from simulating non-uniform irradiance conditions to modelling the IV characteristics of the reconfigurable modules. The reconfiguration algorithm that was used for the simulating the reconfigurable modules' IV characteristics was the SCCS control algorithm. Section 2.3 described the algorithm in detail and also mentioned some of its drawbacks, which will be addressed in future chapters. Finally, the 96 cell, 6 block series-parallel reconfigurable module that was used for all simulations in this thesis was also introduced.

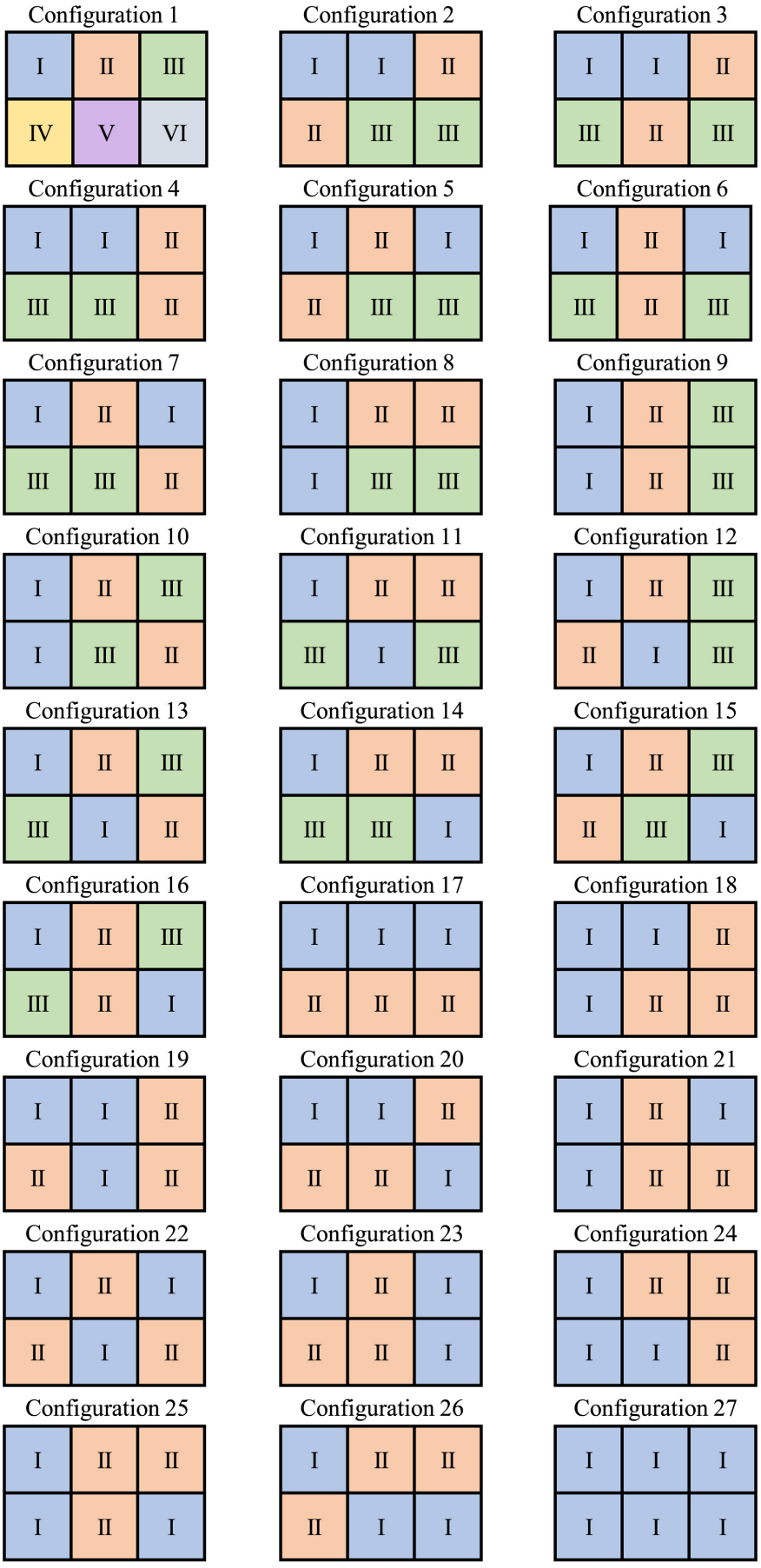



Figure 2.7: Possible configurations for the six block PV module shown in figure 2.6 [2]

# 3

# Shade detection using voltage and current

As mentioned in chapter 2, the drawbacks of the SCCS algorithm stem from one main aspect - it does not have a sensing mechanism for detecting partial shading conditions. Therefore, there is a need for a shade detection mechanism to improve the algorithm further. This chapter explores a possible shade detection algorithm that could be used for triggering the reconfiguration algorithm.

A PV module is a source of a wide range of data which could be used to detect partial shading conditions. Section 3.1 lists out the possible module parameters that could be explored for shade detection. Using this information, section 3.2 describes a possible shade detection mechanism and explains why it has potential to be an improvement over the reconfiguration algorithm described in section 2.3.

## 3.1. Control variables for detecting partial shading

Variables that can be used to detect partial shading can be non-electric (such as cell temperature) or electric (such as $V_{oc}$, $I_{sc}$, $V_{mpp}$ or $I_{mpp}$). The advantage of using electric control variables for detecting partial shading is that electric control variables such as voltage and current react instantaneously in case of any change in irradiance conditions. Therefore, only electric variables are explored in this thesis.

To quantify the correlation between partial shading and module parameters, a metric called relative irradiance difference ($\Delta irr$) was defined:

$$\Delta irr = \frac{max(irr_{min}) - min(irr_{min})}{max(irr_{min})} \tag{3.1}$$

Here $irr_{min}$ signifies the irradiance received by the most shaded cell in each reconfigurable unit of the PV module. Since the current in each reconfigurable unit is limited by its most shaded cell, that's why $irr_{min}$ was considered in the calculation of relative irradiance difference (RID).

The variation in RID with respect to relative difference in module parameters such as $V_{oc}$, $I_{sc}$, $V_{mpp}$ and $I_{mpp}$ was observed to locate any correlation between them. Here $V_{oc}$/$V_{mpp}$ signify the open circuit/MPP (maximum power point) voltage of each cell block and $I_{sc}$/$I_{mpp}$ denote the short circuit/MPP current of each cell block. The relative difference in module parameters were calculated as follows:

$$\Delta x = \frac{max(x_{block}) - min(x_{block})}{max(x_{block})} \tag{3.2}$$

Here, $x$ is $V_{oc}$, $I_{sc}$, $V_{mpp}$ or $I_{mpp}$.

There were two main purposes to explore the relationship between module parameters with the irradiance:

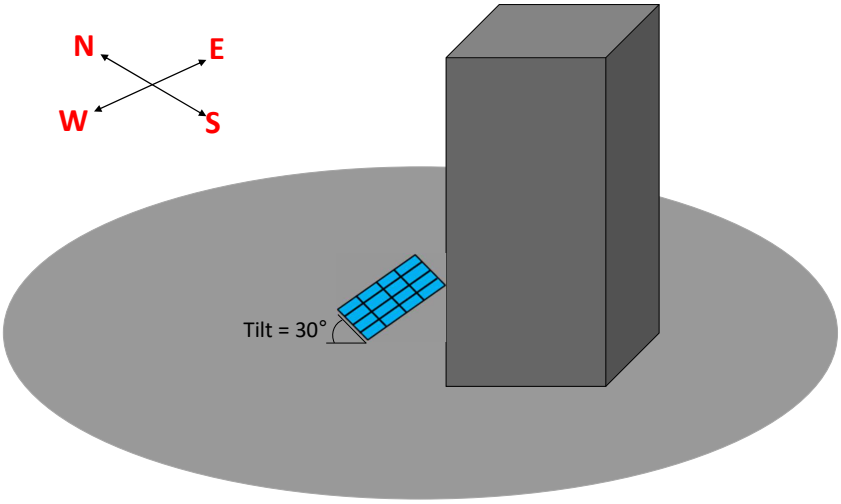



Figure 3.1: Shading scene used for observing relationship between irradiance and module parameters. The module used in the simulation is the one described in section 2.4

- First was to locate a possible ”fingerprint” in the module parameter that could help in shade detection. This could be seen in the form of a quantifiable relationship between the module parameter and irradiance, or it can be seen in the form of a threshold value for the module parameter (for instance, before reconfiguration, the parameter might be observed to be over a certain threshold value and after reconfiguration, it is always below that threshold value).
- The second purpose for exploring the relation between module parameters and irradiance was to locate a module parameter that would be a reliable indicator of the irradiance incident on the module, and hence can also be used as a control variable for reconfiguration.

The relationship between irradiance and module parameters was analysed by simulating a partial shading scene with a 96 cell PV module mounted in front of a tall concrete structure (refer figure 3.1 for the scene that was simulated) in Delft. The simulation was carried out for an entire year with 1 hour resolution weather data from Meteonorm [48].

The simulation was carried out assuming a perfect knowledge approach. At each time step, block-wise IV curves were calculated using the simulation framework discussed in chapter 2. Module IV curves were then calculated assuming an ideal reconfiguration algorithm in which the algorithm will always choose the configuration which gives the highest yield. At each time step, the simulation recorded module parameters (cell block $V_{oc}$, $I_{sc}$, $V_{mpp}$, $I_{mpp}$ and $P_{mpp}$) before and after reconfiguration. The module parameters before reconfiguration were considered as parameters for sub-optimal configurations, and their relationship with irradiance and RID was observed. Any relationship observed here will essentially help in defining a condition for partial shade detection. The next subsections elaborate on the relationships that were observed between the module parameters and RID/irradiance.

### 3.1.1. Irradiance and short circuit current

The first graph in figure 3.2 shows the relationship between the relative difference in short circuit current ($\Delta I_{sc}$) and RID. It is clear from the figure that $\Delta I_{sc}$ is equal to RID. This is because a PV cell's short circuit current varies proportionally with respect to irradiance [49]. Since the most shaded cell's short circuit current determines the cell block $I_{sc}$, the relative difference in short circuit current ($\Delta I_{sc}$) is equal to RID. Due to the clear relationship between the two, $\Delta I_{sc}$ is a reliable indicator for level of partial shading.

The second graph in figure 3.2 shows the relationship between each cell block's short circuit current ($I_{sc}$) and irradiance of the most shaded cell in that block. As mentioned above, $I_{sc}$ is proportional to irradiance and we see that reflected in this plot as well. Since there is a tangible relation between

irradiance and $I_{sc}$, we can conclude that $I_{sc}$ is a good indicator of irradiance levels.

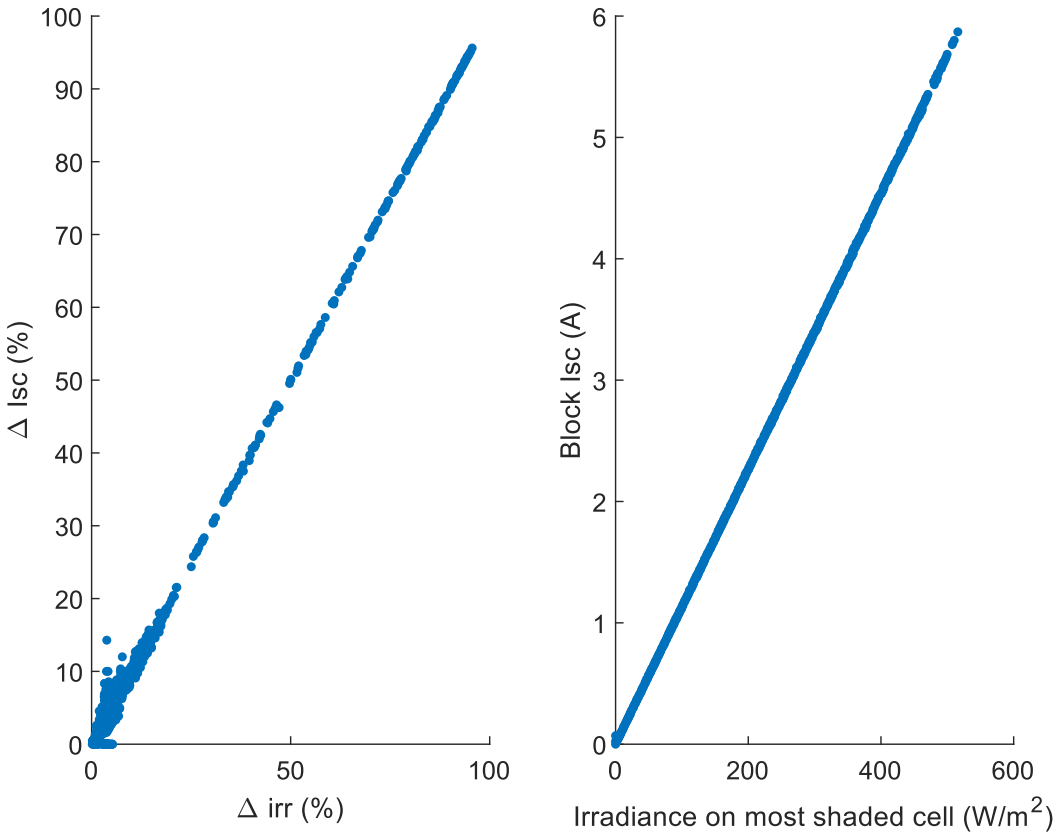



Figure 3.2: Variation in $I_{sc}$ with respect to irradiance

The short circuit current sensing algorithm described in section 2.3 uses cell block $I_{sc}$ as an input variable for reconfiguration. However, a disadvantage of using $I_{sc}$ as an input variable is that of energy loss during $I_{sc}$ measurement. Additionally, the algorithm will have to be synchronous in nature and will not be able to detect partial shading conditions without the PV module being disconnected.

### 3.1.2. Irradiance and open circuit voltage

The first graph in figure 3.3 shows the relationship between the relative difference in open circuit voltage ($\Delta V_{oc}$) and RID. The graph shows that $\Delta V_{oc}$ varies very little compared to RID. This is because $V_{oc}$ has a logarithmic relation to irradiance and varies very little with respect to irradiance [49]. In figure 3.3, RID varying between 0-100%, $\Delta V_{oc}$ only varies in the 0-10% range. Based on these observations, $\Delta V_{oc}$ is not suitable for using as a trigger for shade detection.

The second graph in figure 3.3 shows the relationship between each cell block's open circuit voltage ($V_{oc}$) and irradiance of the most shaded cell in that block. It shows that $V_{oc}$ is also not a good indicator of irradiance level as there is no quantifiable relationship between the two. For example, voltage of 11V could mean that irradiance is anywhere between 0-600 $W/m^2$.

### 3.1.3. Irradiance and MPP current

The top two graphs in figure 3.4 shows the relationship between the relative difference in MPP current and RID. It is clear from the graph that before reconfiguration, $\Delta I_{mpp}$ does not show a quantifiable relationship with RID. Additionally, on comparing it with the $\Delta I_{mpp}$ values after reconfiguration, it does not indicate any threshold dividing the $\Delta I_{mpp}$ values of optimum and sub-optimal configurations. Based on these observations, $\Delta I_{mpp}$ does not appear to be a reliable indicator of partial shading conditions.

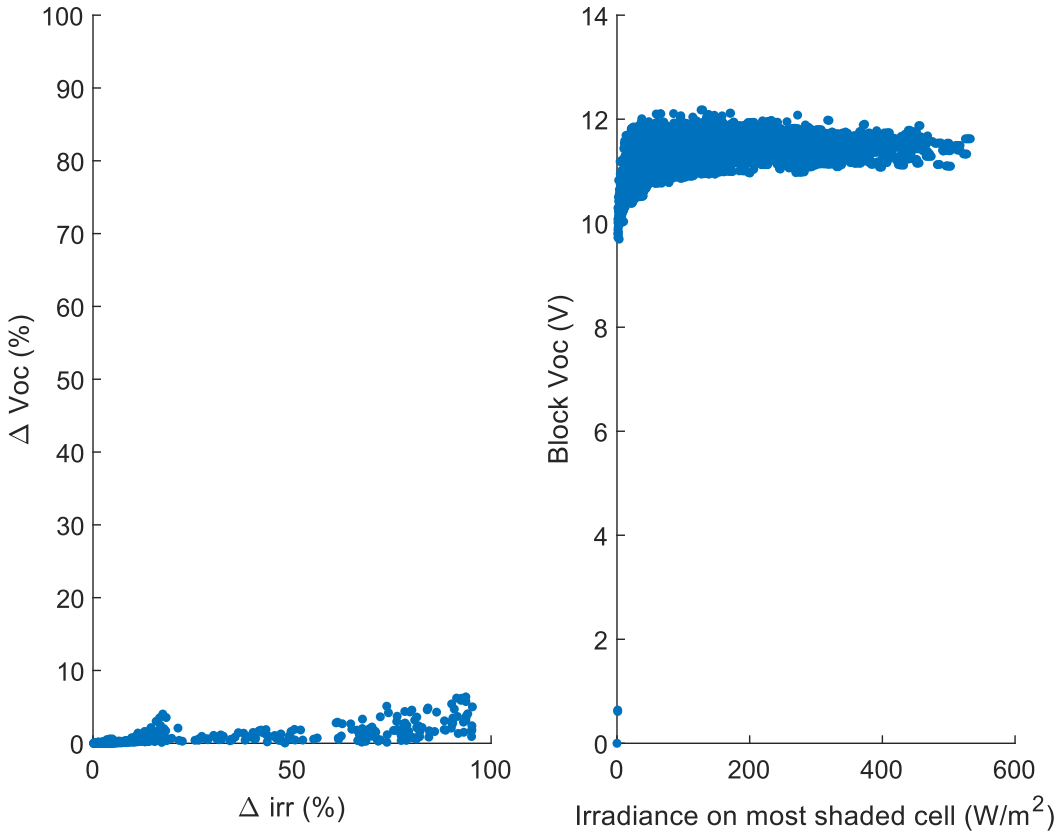



Figure 3.3: Variation in $V_{oc}$ with respect to irradiance

The bottom two graphs in figure 3.4 show the relationship between $I_{mpp}$ (before reconfiguration) and irradiance of the most shaded block. As expected, some datapoints show a proportional relationship with irradiance. However, there are some outliers which are for situations when two (or more) cell blocks are in series and the most shaded cell block in that series will limit the current of all other cell blocks in that series. Due to this, cell block's $I_{mpp}$ is not a reliable indicator of irradiance incident on it.

### 3.1.4. Irradiance and MPP voltage

The top two graphs in figure 3.5 show the relationship between the relative difference in $V_{mpp}$ and RID. Similar to $V_{oc}$, it is clear that there is neither a discernible relationship between the two, nor is there any threshold dividing the $\Delta V_{mpp}$ values of optimum and sub-optimal configurations. Therefore, $\Delta V_{mpp}$ is not suitable for using as a trigger for shade detection.

The bottom two graphs in figure 3.5 show the relationship between each cell block's MPP voltage ($V_{mpp}$) and irradiance of the most shaded cell. Similar to $V_{oc}$, it is clear that $V_{mpp}$ is not a suitable indicator of irradiance level. For example, voltage of 10V could mean that irradiance is anywhere between 0-600 $W/m^2$.

### 3.1.5. Irradiance and power

The top two graphs in figure 3.6 show the relationship between the relative difference in $P_{mpp}$ and RID. Similar to $I_{mpp}$ and $V_{mpp}$, there is neither a discernible relationship between $\Delta P_{mpp}$ and RID, nor is there any threshold dividing the $\Delta P_{mpp}$ values of optimum and sub-optimal configurations. Therefore, $\Delta P_{mpp}$ is not suitable for using as a trigger for shade detection.

The bottom two graphs in figure 3.6 shows the relationship between each cell block's power output and irradiance of the most shaded cell in that block. From the graph, you can see that $P_{mpp}$ appears to be

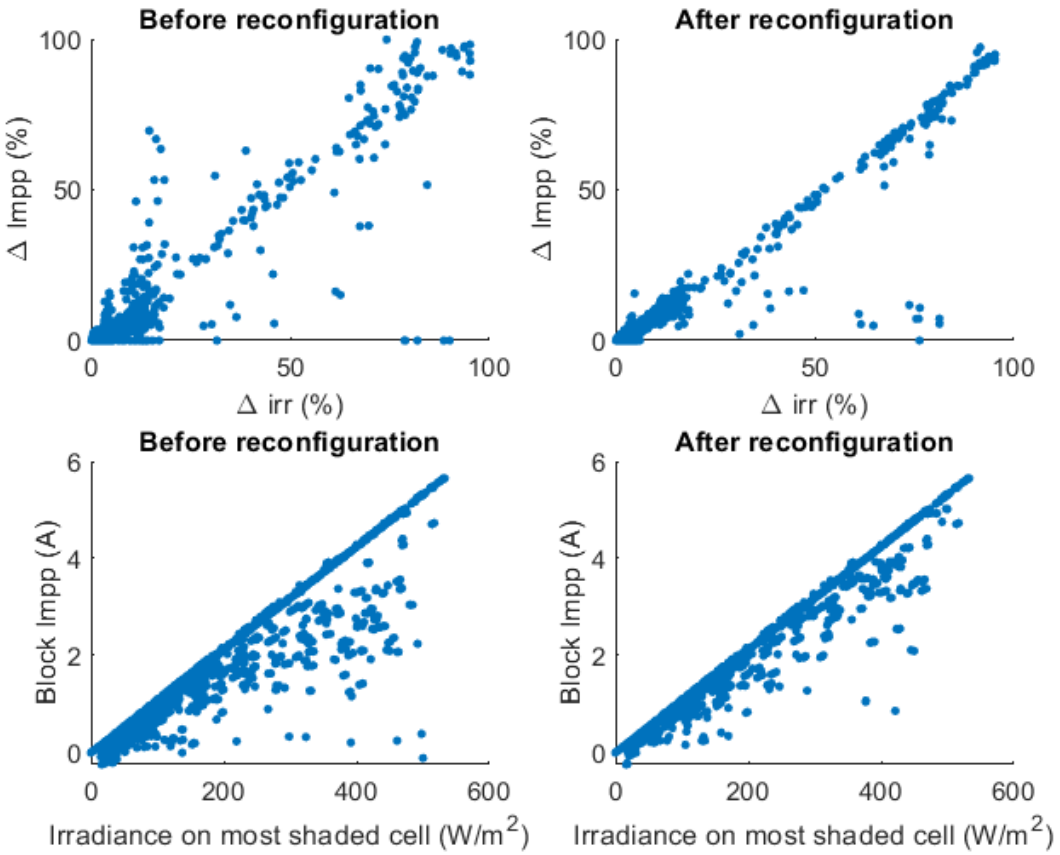



Figure 3.4: Variation in $I_{mpp}$ with respect to irradiance

varying proportionally with respect to irradiance. However, there are some outliers which correspond to situations when there are two (or more) cell blocks in series and the most shaded cell block in that series will limit the current (and hence power) of all other cell blocks in that series. Due to this, there is no quantifiable relationship between $P_{mpp}$ and irradiance and it cannot be used as an indicator for irradiance levels.

## 3.2. Shade detection mechanism

From the above analysis, if cell block parameters were to be used as indicators for partial shading or irradiance, only $I_{sc}$ is a clear indicator of irradiance levels and $\Delta I_{sc}$ can be used as an indicator for partial shading. However, this would mean that the algorithm will be synchronous in nature and will only run periodically. The aim of the thesis is to develop an algorithm that gets triggered due to the occurrence of a partial shade condition.

Therefore, an alternate approach suggested in [35] is used for shade detection. The shade detection and reconfiguration algorithm described in [35] is for TCT connected reconfigurable modules. For the shade detection part of the algorithm, the voltages of one of the blocks $V_1$ and the complete module's voltage $V_{mod}$ is monitored. In case of uniform illumination conditions, $V_{mod} = m * V_1$, where m is the number of cell blocks. In case of non-uniform illumination conditions, $V_{mod} \neq m * V_1$. Therefore, the reconfiguration algorithm is triggered if $|V_{mod} - m * V_1| > \delta V$, where $\delta V$ is a predefined threshold value. One of the advantages of this mechanism is the requirement of only two voltage sensors for partial shade detection. Inspired by this concept, a new shade detection mechanism for series-parallel connected modules is developed in this thesis.

This modified mechanism uses module voltage, module current and one cell block's voltage and current as inputs for shade detection. Similar to the mechanism described in [35], the voltage of one of the

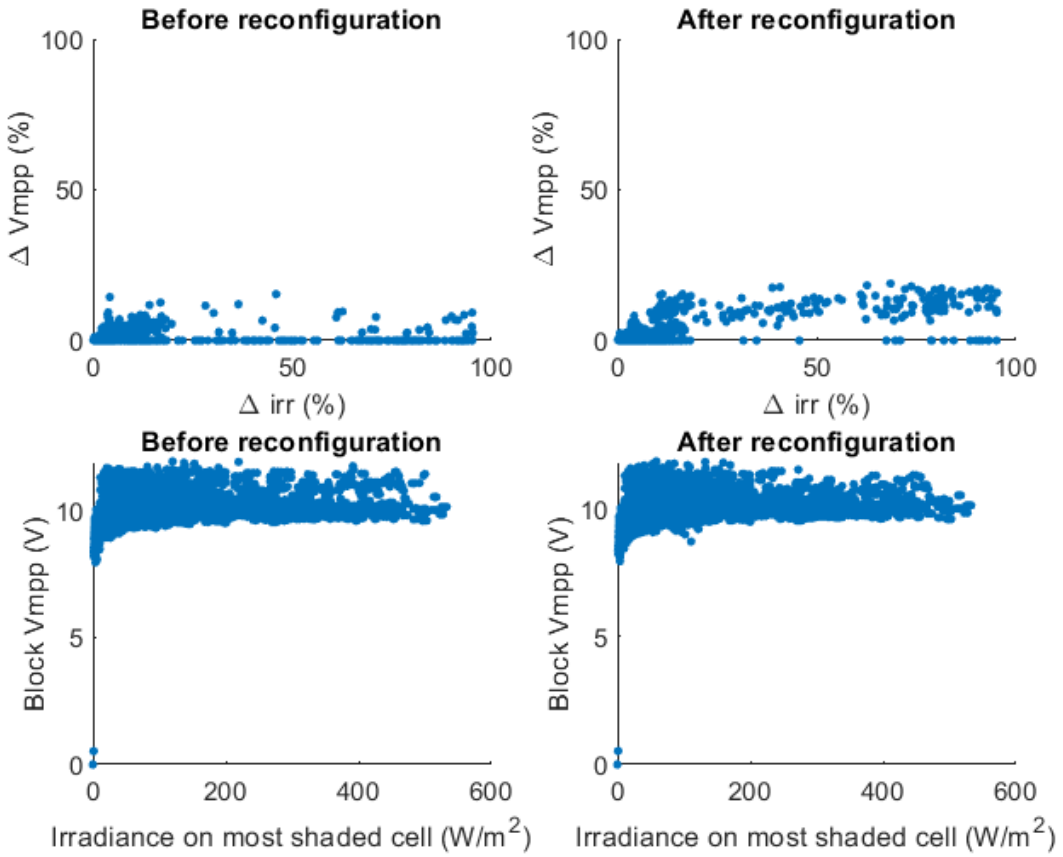



Figure 3.5: Variation in $V_{mpp}$ with respect to irradiance

cell blocks as well as the module voltage are continuously monitored. Additionally, the cell block's current and module current are also monitored. Using these module parameters, two types of checks are performed - a voltage check and a current check - to detect partial shading. Equations 3.3 and 3.4 define the voltage and current checks. If either of these checks is true, then the module will reconfigure.

$$\left|\frac{V_{mod} - b * V_{block1}}{V_{mod}}\right| > \Delta V_{threshold} \tag{3.3}$$

$$\left|\frac{I_{mod} - n * I_{block1}}{I_{mod}}\right| < \Delta I_{threshold} \tag{3.4}$$

Here $b$ stands for the number of cell blocks per string and $n$ stands for number of parallel strings in the configuration.

The voltage check will detect if there is non-uniform illumination condition and trigger the reconfiguration algorithm. This means that if the cell blocks' voltages differ from each other (over a range decided by $\Delta V_{threshold}$), the module will reconfigure. The current check helps in situation where the module is in a series-parallel configuration, particularly an all-parallel configuration. The module will be in an series-parallel configuration when the RID is high and the individual cell block's currents are unequal. In case the RID drops and the cell block currents are close to each other (within a range decided by $\Delta I_{threshold}$), then the current check will ensure that the module reconfigures out of the series-parallel configuration. Figure 3.7 shows a flowchart explaining the new shade detection algorithm.

Since the proposed shade detection mechanism does not depend on $V_{oc}$ or $I_{sc}$ measurement, there is no energy loss during input data collection for the control algorithm. Additionally, it only needs two

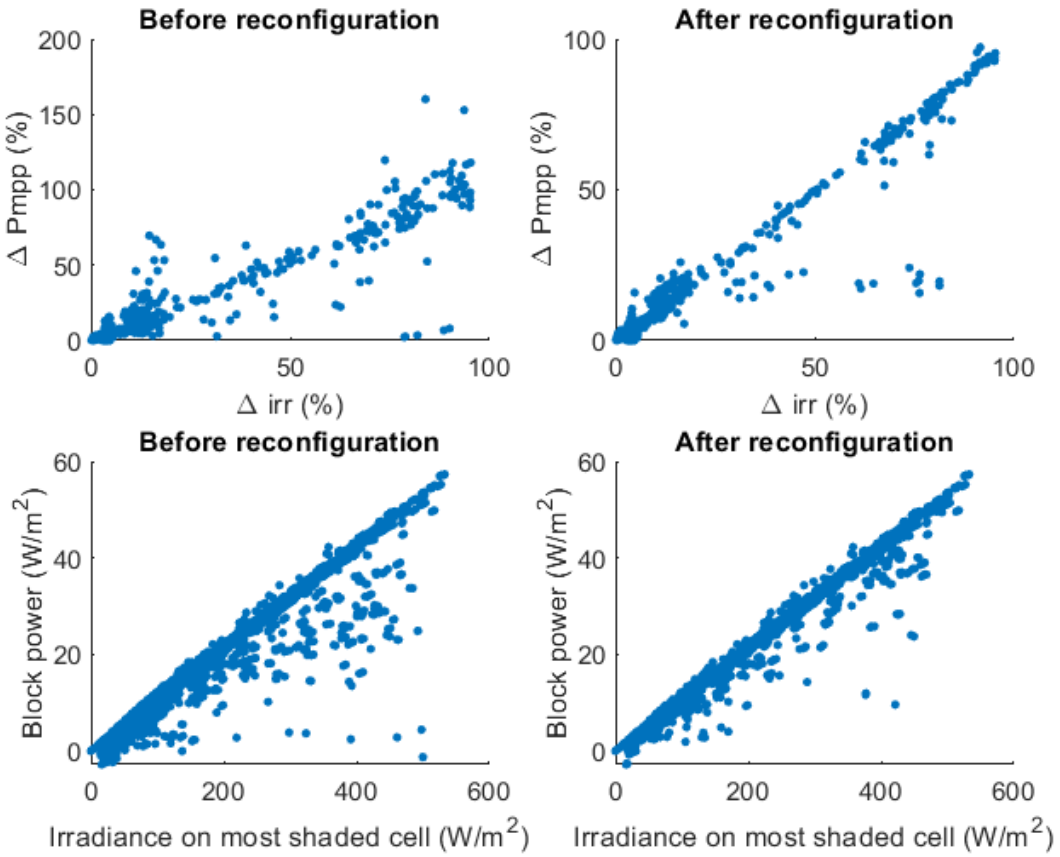



Figure 3.6: Variation in $P_{mpp}$ with respect to irradiance

current sensors and two voltage sensors, irrespective of the number of cell blocks in the module. Lastly, the sensing mechanism ensures in principle that the reconfiguration algorithm does not run when it's not required and also ensures that no partial shading conditions are missed.

However, in order for the shade detection mechanism to be able to detect all shade detection conditions and also prevent nuisance triggers, it needs to be optimised. The optimisation of the mechanism needs to be done for two main aspects: determining the optimum $\Delta I_{threshold}$ and $\Delta V_{threshold}$ and determining which of the cell blocks will act as the most optimum sensor block. The next chapter discusses the optimization process further.

## 3.3. Conclusion

This chapter pertains to the second sub-goal of section 1.2 - exploring various module parameters and their potential as shade detection triggers/irradiance level indicators for using as inputs in a reconfiguration algorithm. The parameters that were explored were - cell blocks' short circuit current, cell blocks' open circuit voltage, cell blocks' MPP current/voltage and cell blocks' MPP power. From this, it was concluded that only the individual cell blocks' short circuit current is a reliable shade detection triggers and irradiance level indicator.

However, using the cell blocks' short circuit currents as input would mean that the reconfiguration algorithm will be synchronous. Therefore, an alternate shade detection mechanism inspired by reference [35] was proposed. In this, a combination of the module's and a single cell blocks' operating voltage and current is used as an input to the shade detection mechanism. The next chapter discusses the process followed to optimise the algorithm.

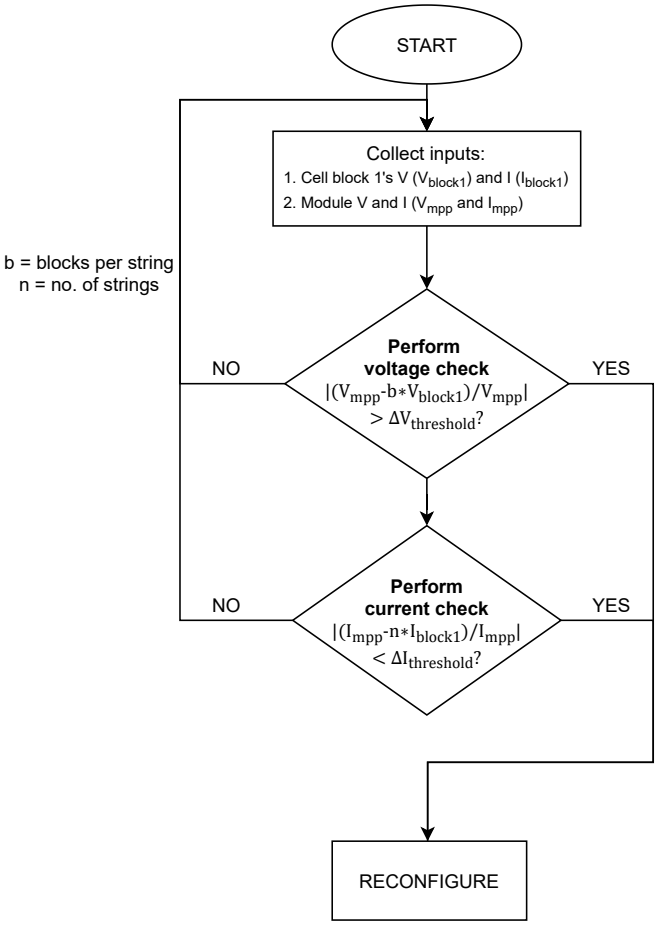



Figure 3.7: Flowchart explaining the new sensing algorithm

# 4

# Optimising the shade detection mechanism

Chapter 3 introduced a shade detection mechanism which relies on the module's and a cell block's operating current and voltage. It also mentioned that it needs to be optimised for two main aspects - choosing the optimum threshold values and optimising the position of the sensor block. This chapter will discuss the process followed for optimising the said shade detection mechanism.

It should be noted that the shade detection mechanism introduced in section 3.2 only detects partial shading condition. Therefore, a reconfiguration mechanism also needs to be defined, which will be triggered when partial shading is detected. A possible approach for reconfiguration can be switching the module between all its possible configurations and choosing the one with the highest yield. From hereon, this approach will be referred to as the ideal reconfiguration mechanism (it is named this way because in theory, this approach would always choose the optimum configuration). The shade detection mechanism and ideal reconfiguration mechanism in totality will be referred to as the reconfiguration algorithm.

Section 4.1 explains the optimisation procedure followed for the above mentioned reconfiguration algorithm. As the results from section 4.1 point out that this algorithm can be further improved, section 4.2 discusses possible improvements to the algorithm, analyses their performance and determines the best version of the algorithm. Finally, section 4.3 discusses the second component of the optimisation process - optimising the position of the sensor block for the shade detection mechanism.

## 4.1. Optimising the reconfiguration algorithm

The shading scene used for the optimisation is the same as the one described in section 3.1, that is, a single 96 cell module installed in front of a tall concrete structure in Delft. The weather data used is one minute resolution and due to computational limitations, the simulation is run for a single day only. The day chosen for the simulation is 4th May because in the year long simulation performed in section 3.1, 4th May had on of the highest reconfiguration counts for the given shading scene. Therefore, optimising the thresholds for this day would imply that the algorithm has been optimised for a scenario where the reconfiguration algorithm is frequently needed.

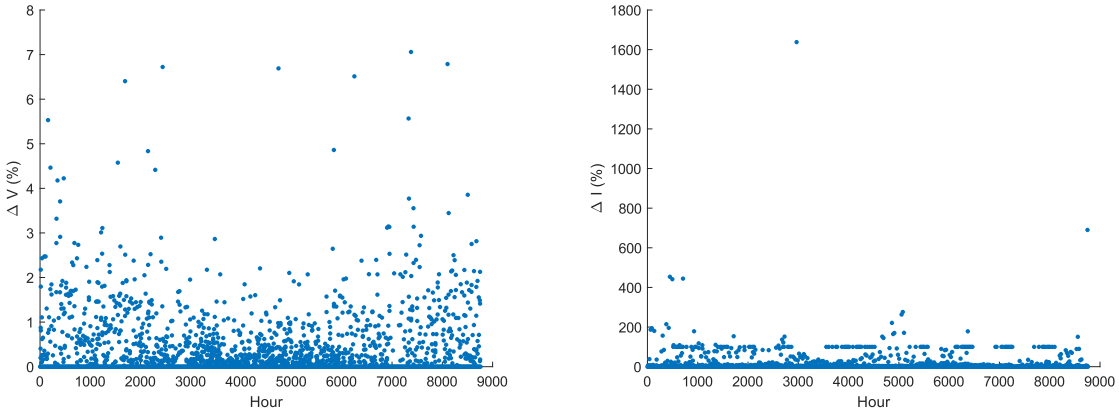
### 4.1.1. Establishing a range for thresholds of the shade detection mechanism

To determine the optimum value of $\Delta I_{threshold}$ and $\Delta V_{threshold}$, first a range was established for both the values. This was done by observing the values of $\Delta I$ and $\Delta V$ for the simulation scenario described in section 3.1.

As mentioned in section 3.1, the simulation generates the cell block and module voltage and current at each time step, before and after reconfiguration. The value of $\Delta I$ and $\Delta V$ at each time step was calculated from the voltage and current values of the module before reconfiguration (here $\Delta V$ is referring

to the LHS of equation 3.3 and $\Delta I$ is referring to LHS of equation 3.4). This way, the range in which $\Delta I$ and $\Delta V$ vary before reconfiguration was observed.

Figure 4.1 shows distribution of the $\Delta I$ and $\Delta V$ values for the duration of the simulation. It is observed that $\Delta V$ varies between 0% and 7.06% while $\Delta I$ varies between 0% and 1637.9%.




(a) $\Delta V$ values before reconfiguration

(b) $\Delta I$ values before reconfiguration

Figure 4.1: $\Delta V$ and $\Delta I$ values before reconfiguration

Therefore during optimisation, $\Delta I_{threshold}$ and $\Delta V_{threshold}$ were varied between the said intervals to determine their optimum values. For deciding which specific $\Delta I_{threshold}$ and $\Delta V_{threshold}$ values to use, first their percentile distribution was calculated. Based on this, some $\Delta I_{threshold}$ and $\Delta V_{threshold}$ values were selected. Tables 4.1 and 4.2 show the threshold values that were used for algorithm optimisation.

| Percentile | $\Delta I_{threshold}$ (%) |
|---|---|
| 50 | 0.00 |
| 60 | 0.09 |
| 70 | 0.63 |
| 80 | 1.32 |
| 90 | 2.57 |
| 100 | 1637.95 |

Table 4.1: Table showing the values of $\Delta I_{threshold}$ used during algorithm optimisation

### 4.1.2. Optimising the thresholds of the shade detection mechanism

During the optimisation process, $\Delta I_{threshold}$ and $\Delta V_{threshold}$ values were varied between the intervals derived from section 4.1.1. For each combination of the threshold value, the DC yield of the module was calculated. The optimum threshold values were the ones that corresponded to the highest module yield.

The losses for the new reconfiguration algorithm also needed to be determined for calculating DC yield. Since the shade detection mechanism needs operating voltage and current values, the module will not have to be disconnected from the system and thus no DC yield losses. For the reconfiguration mechanism, the module is switched between all its possible configurations and the one with the highest yield is chosen. For this, the maximum power point of each configuration is determined by sweeping the module IV curve. Since the module will not be connected to the system during this process, the energy lost during this time was considered as reconfiguration losses.

The time it takes to perform a single IV sweep can vary for different cell technologies, and may take anywhere between 40 to 100 milliseconds [50] [51]. Since the module that was simulated has 27 possible configuration (see figure 2.7), every time reconfiguration is triggered, it was assumed that the algorithm will spend almost 3 seconds to reconfigure to its optimum configuration (assuming that

| Percentile | $\Delta V_{threshold}$ (%) |
|---|---|
| 86 | 0.04 |
| 87 | 0.07 |
| 88 | 0.11 |
| 89 | 0.15 |
| 90 | 0.19 |
| 91 | 0.26 |
| 92 | 0.35 |
| 93 | 0.44 |
| 94 | 0.55 |
| 95 | 0.71 |
| 96 | 0.89 |
| 97 | 1.12 |
| 98 | 1.46 |
| 99 | 2.02 |
| 100 | 7.06 |

Table 4.2: Table showing the values of $\Delta V_{threshold}$ used during algorithm optimisation

a single IV sweep takes 100 milliseconds). The fraction of DC yield that will be lost owing to the reconfiguration process will vary depending on the number of times reconfiguration is triggered. The highest losses would be when reconfiguration were to happen every minute, in which case 5% of the DC yield will be lost due to reconfiguration.

In addition to the optimisation procedure the reconfiguration algorithm's performance was also compared to some reference cases. For that purpose, the following scenarios were also simulated:

- **Perfect knowledge scenario** - In this scenario, an ideal situation was assumed in which the module is always reconfigured to its optimum configuration at every time step. Another important assumption here was that there are no losses during reconfiguration. Comparing the new reconfiguration algorithm with this scenario would help in understanding how close (or far) its performance is in terms of an ideal situation.

- **No reconfiguration** - A scenario assuming a conventional PV module. In this scenario, the module is always in an all-series configuration. This scenario will give an idea of how much more efficient the new reconfiguration algorithm is with respect to conventional modules.

- **SCCS algorithm run at 5 minute interval** - In an ideal scenario, the SCCS algorithm would be run as often as possible, say at 1 minute time intervals. In practice, however, a PV module will usually be connected to a power converter which will not be able to tolerate frequent power fluctuations that might occur if the SCCS algorithm were to run per minute. Therefore, duration between subsequent runs of the algorithm will depend on the power fluctuations the converter can tolerate, and will possibly be in the range of 5-15 minutes. Therefore, an additional scenario was simulated where the SCCS algorithm is run at 5 minute time interval, as the yield calculated in this scenario will be much closer to real-life application of the algorithm. One additional thing to note here is that each run of the algorithm takes up to 150 milliseconds - from measuring the cell blocks' short circuit currents to the actual reconfiguration [2]. Therefore, whenever the algorithm runs, 0.25% of the DC yield for that will be deducted as reconfiguration losses.

Figure 4.2 shows the result of the optimisation process. All yield values are calculated as a percentage of the DC yield in the perfect knowledge scenario. For varying threshold values, the new reconfiguration algorithm's yield varied in the range of 94.44-99.07% and the average yield was 95.82%. The threshold values corresponding to the maximum yield were 0.7% for $\Delta V_{threshold}$ and 0% for $\Delta I_{threshold}$. In comparison, the SCCS algorithm's yield was 99.13% and for the no reconfiguration scenario it was 95.3%.

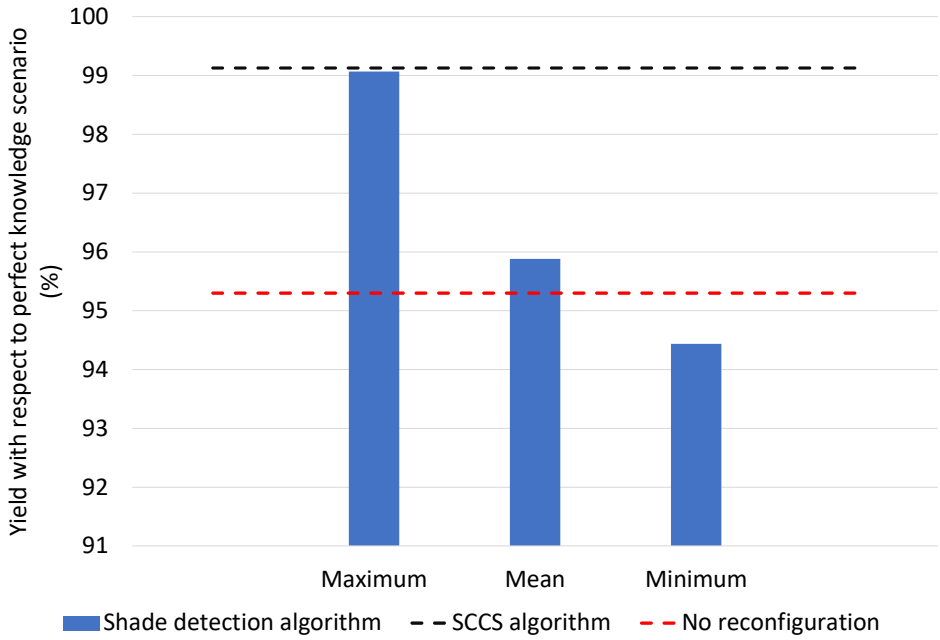



Figure 4.2: Comparison of the reconfiguration algorithm's DC yield with a conventional module's yield (no reconfiguration) and SCCS algorithm's yield

## 4.2. Modifying the reconfiguration algorithm

In section 4.1, the reconfiguration algorithm that was optimised had some key drawbacks, which could possibly be limiting its performance:

- **Current check is causing nuisance triggers**
  From the current check expression (see equation 3.4) it can be inferred that that the optimum $\Delta I_{threshold}$ value should be low enough such that it minimises the number of nuisance triggers. The fact that optimum value for $\Delta I_{threshold}$ was found to be 0% implies that any non-zero value of the threshold was causing a lot of nuisance triggers and therefore limiting the module yield (see figure 4.3).

- **High losses due to the reconfiguration mechanism**
  As explained in section 4.1.2, if the algorithm were to trigger reconfiguration every minute, the module will be reconfigured exactly as per the perfect knowledge scenario but 5% of the module's yield would be lost during reconfiguration. As a consequence, the module's yield will be 95% of the perfect knowledge scenario. This is also visible in figure 4.2, where the DC yield for the algorithm was observed to be as low as 94.5% for certain values of $\Delta I_{threshold}$ and $\Delta V_{threshold}$. In comparison, the module in the no reconfiguration scenario also produced around 95% yield as compared to the perfect knowledge scenario. This indicates that in case of poorly selected threshold values causing frequent reconfiguration triggers, the algorithm may lead to high percentage of losses and not offer any added advantage over using conventional modules.

- **Shade detection mechanism failing to detect situations when module should be in all parallel configuration**
  To investigate if the shade detection mechanism is failing to detect any specific reconfiguration situations, the configurations chosen by the algorithm were compared with the configurations chosen in the perfect knowledge scenario at each time step. This way, it could be observed which configuration was being missed the most by the reconfiguration algorithm.

  Averaging over all the simulations done for different threshold values, it was observed that the all-

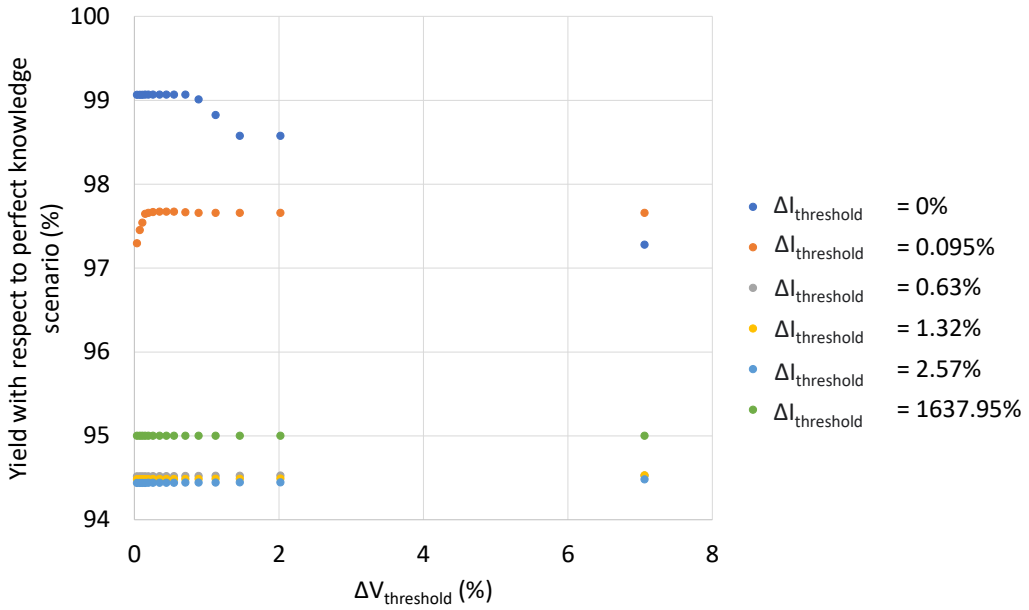



Figure 4.3: Plot showing the DC yield of the algorithm with respect to perfect knowledge scenario, for varying values of $\Delta V_{threshold}$ and $\Delta I_{threshold}$. It is clear from the figure that for a non-zero value of $\Delta I_{threshold}$, the yield is much lower than when $\Delta I_{threshold}$ is 0%

parallel configuration was the one which was missed the most (see figure 4.4). This implies that in some instances where the module would have generated the highest yield by connecting all its cell blocks in parallel, the shade detection mechanism had failed to detect this and reconfigure the module accordingly.

The conclusion from this was that this reconfiguration algorithm had three potential sources for inefficiency - first was an inefficient current check, second was high energy loss during reconfiguration and third was the inability for the algorithm to detect situations when the module should be in an all-parallel configuration. A number of modifications were proposed to tackle these issues, which can be classified into two categories - first category of improvements aimed to improve the shade detection mechanism and the second category aimed to reduce the energy loss during reconfiguration.

To improve the shade detection mechanism, the following alternatives were explored:

- **Performing current check only when module is in an all-parallel configuration**
  The current check was found to be a cause for lot of nuisance triggers and hence causing a lot of energy loss due to unnecessary running of the reconfiguration algorithm. However, the current check cannot be completely eliminated because it is especially important in a situation where the module is in an all-parallel configuration, where a voltage check would not be relevant. Therefore, a possible modification to the shade detection mechanism is to perform the current check only when the module is in an all parallel configuration. The voltage check will be performed irrespective of the module configuration.

- **Introduce a new current check**
  An alternate current check was defined to see if it can perform better than the old current check. From hereon, to differentiate between the two kinds of current checks, the new current check will be referred to as the $I_{mpp}$ check and old current check will be referred to as the $\Delta I$ check. The steps for the $I_{mpp}$ check are shown in figure 4.5. For this check, each cell block's current is measured and then arranged in ascending order. The consecutive current values are then paired and their relative difference is calculated. If all these relative differences are lower than a predefined threshold $\Delta I_{mpp,threshold}$, then the current check is true and reconfiguration is triggered. Similar to $\Delta I$ check, the performance of $I_{mpp}$ check will be analysed in two ways - running the

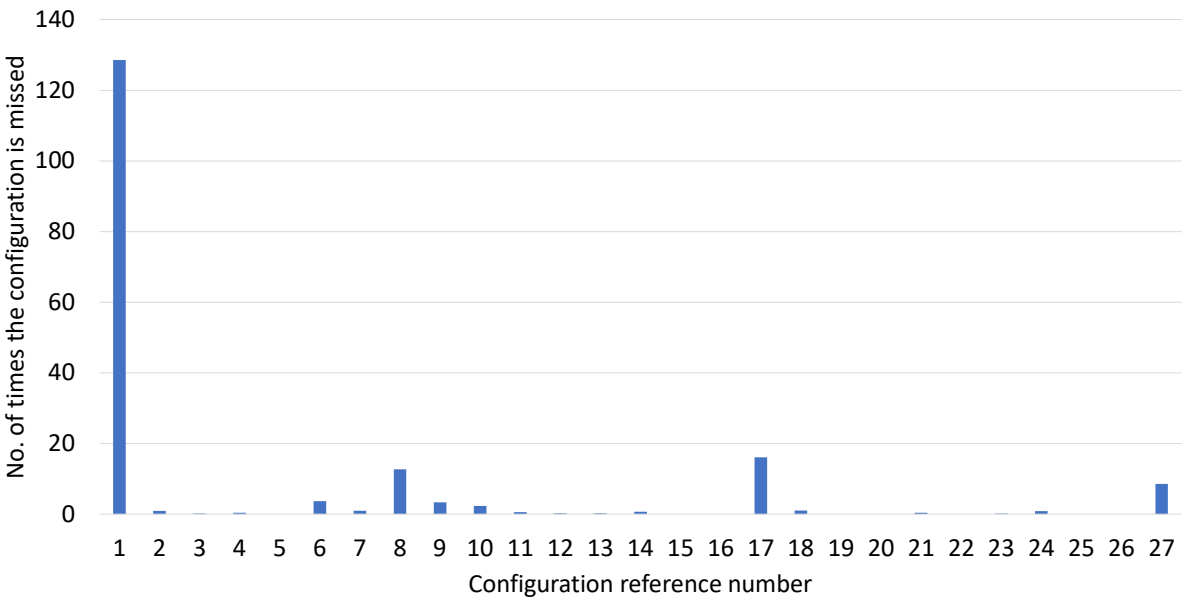



Figure 4.4: The average number of times a certain type of configuration was missed by the algorithm. Refer figure 2.7 to see which configuration number corresponds to which configuration

check irrespective of module configuration and running the check only when the module is in an all-parallel configuration.

To improve the reconfiguration performance, the following alternatives were explored:

- **Never reconfiguring the module to an all-parallel configuration**
  One of the drawbacks of the reconfiguration algorithm was that it failed to reconfigure to an all-parallel configuration whenever necessary. Due to this, the module's yield might be limited. On the other hand, reconfiguring the module to an all-parallel configuration also implies that there will be increased Ohmic and power conversion losses due to higher module current when it is in an all-parallel configuration. Therefore, a possible algorithm can be explored where the module never goes to an all-parallel configuration. If the resultant loss in yield is less than the Ohmic losses encountered in the all-parallel configuration, then the reconfiguration algorithm does not need to reconfigure to an all-parallel configuration.

- **Using the SCCS algorithm for reconfiguration**
  In the original shade detection algorithm, reconfiguration is done by switching between various possible configurations and choosing the one with the highest yield. Since this is a very time intensive process and leads to a high fraction of energy loss, an alternative approach would be to use the SCCS algorithm for reconfiguration. Since the SCCS algorithm only takes 150 milliseconds per run, this would greatly reduce the energy loss during reconfiguration. As concluded in chapter 3, short circuit current is the best indicator of incident irradiance and therefore SCCS algorithm has another advantage of being a reliable reconfiguration algorithm (since it uses the cell blocks' short circuit current as input).

Based on the different ways listed to improve the shade detection/reconfiguration mechanism, eight modifications of the reconfiguration algorithm were defined and compared to see which is the most efficient one. The modifications of the algorithm are given in the table 4.3. To be able to easily differentiate between the modifications, they were each assigned a name and acronym based on their salient features. The first half of the acronym conveys what modification has been made to the original shade detection mechanism while the second half of the acronym conveys which reconfiguration mechanism is employed.

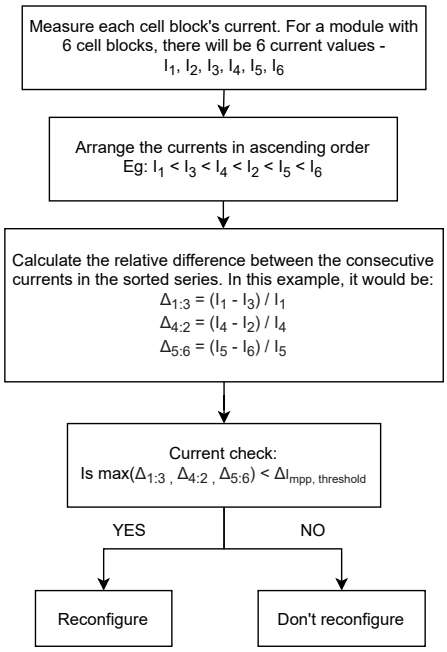



Figure 4.5: Flowchart demonstrating the $I_{mpp}$ current check

### 4.2.1. Comparing the algorithm modifications

For comparing the algorithm modifications, the same shading scene described in section 3.1 was simulated and the module yield for a single day was calculated for each modification. For the simulations, $\Delta I_{threshold}$ and $\Delta V_{threshold}$ was varied in the range established in section 4.1.1. One thing to be noted is that $\Delta I_{threshold}$ is never set to 0% for algorithms in which module can be in an all-parallel configuration; because in that case current check will never get triggered. Also, for algorithms with $\Delta I_{mpp}$ check, $\Delta I_{mpp,threshold}$ was varied from 1% to 10%.

The next three subsections discuss the performance of the algorithms based on three main metrics - the DC yield of the module under a particular algorithm, the number of times the algorithm triggers reconfiguration (referred to as the reconfiguration count) and the accuracy of the algorithm.

#### DC yield

Figure 4.6 shows the module yield for all the modifications over varying values of $\Delta I_{threshold}$, $\Delta V_{threshold}$ and $\Delta I_{mpp,threshold}$. Each modification's highest, mean and lowest yield (with respect to the perfect knowledge scenario yield) is shown in the figure. Two more reference lines can be seen in the figure, one is of the yield for the SCCS algorithm running every 5 minutes and other is for the scenario assuming a conventional module (with no reconfiguration).

The **OSD + IR** algorithm is the original reconfiguration algorithm proposed in section 4.1. The average yield for this algorithm is 95.21% of the perfect knowledge scenario, which is close to the yield in the no reconfiguration scenario. The maximum yield for this algorithm is 97.67% of perfect knowledge scenario and corresponds to $\Delta V_{threshold}$ of 0.44% and $\Delta I_{threshold}$ of 0.09%. Refer section 4.2 for a detailed analysis of this algorithm.

The **OSD + SCCS** algorithm shows the highest yields for all values of $\Delta V_{threshold}$ and $\Delta I_{threshold}$. The average yield is 98.9% of the perfect knowledge scenario and the maximum yield is 99.08% (corresponding to highest $\Delta I_{threshold}$ value of 1637.95%).

As compared to other modifications, the **OSD + IRwoAP** algorithm shows the lowest average yield of 94.96%. In this algorithm, the shade detection mechanism is similar to that of the OSD + IR algorithm, but the module is never reconfigured to an all-parallel configuration. The maximum yield is 98.5% of the perfect knowledge scenario and corresponds to $\Delta V_{threshold}$ of 0.26% and $\Delta I_{threshold}$ of 0%.

| Algorithm name | Shade detection mechanism | Reconfiguration mechanism |
|---|---|---|
| Original shade detection<br>+<br>Ideal reconfiguration<br><br>(OSD + IR) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$\Delta I$ check (equation 3.4)<br>(irrespective of module configuration) | Switch between all possible configurations and choose the one with highest yield |
| Original shade detection<br>+<br>SCCS reconfiguration algorithm<br><br>(OSD + SCCS) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$\Delta I$ check (equation 3.4)<br>(irrespective of module configuration) | Reconfiguration using SCCS algorithm |
| Original shade detection<br>+<br>Ideal reconfiguration without<br>all parallel configuration<br><br>(OSD + IRwoAP) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$\Delta I$ check (equation 3.4)<br>(irrespective of module configuration) | Switch between all possible configurations (except the all parallel configuration) and choose the one with highest yield |
| Original shade detection<br>+<br>SCCS algorithm without<br>all parallel configuration<br><br>(OSD + SCCSwoAP) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$\Delta I$ check (equation 3.4)<br>(irrespective of module configuration) | Reconfiguration using SCCS algorithm, but never reconfigure to an all parallel configuration |
| OSD, but current check for<br>all-parallel configuration only<br>+<br>Ideal reconfiguration<br><br>(CC4AP + IR) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$\Delta I$ check (equation 3.4)<br>(only for all-parallel configuration) | Switch between all possible configurations and choose the one with highest yield |
| OSD, but current check for<br>all-parallel configuration only<br>+<br>SCCS algorithm<br><br>(CC4AP + SCCS) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$\Delta I$ check (equation 3.4)<br>(only for all-parallel configuration) | Reconfiguration using SCCS algorithm |
| OSD, but with new current check<br>for all-parallel configuration<br>+<br>Ideal reconfiguration<br><br>(NCC + IR) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$I_{mpp}$ check (figure 4.5)<br>(only for all-parallel configuration) | Switch between all possible configurations and choose the one with highest yield |
| OSD, but with new current check<br>for all-parallel configuration<br>+<br>SCCS algorithm<br><br>(NCC + SCCS) | Voltage check (equation 3.3)<br>(irrespective of module configuration)<br><br>$I_{mpp}$ check (figure 4.5)<br>(only for all-parallel configuration) | Reconfiguration using SCCS algorithm |

Table 4.3: A list of all possible modifications of the reconfiguration algorithm. Each modification has been assigned a name and acronym. In further text, the modifications will be predominantly referred to using the acronyms.

In the **OSD + SCCSwoAP** algorithm the average yield is 98% of the perfect knowledge scenario. Additionally, the yield is never lower than the ”no reconfiguration” scenario. The maximum yield is 98.4% of perfect knowledge scenario and corresponds to the $\Delta I_{threshold}$ of 1637.95%.

In the **CC4AP + IR** algorithm the average yield is one of the highest amongst all modifications and the maximum yield is 98.9% of the perfect knowledge scenario and corresponds to $\Delta V_{threshold}$ of 0.89% and $\Delta I_{threshold}$ of 0.63%. Additionally, the yield is never lower than the ”no reconfiguration” scenario.

The **CC4AP + SCCS** algorithm is similar to the CC4AP + IR algorithm in terms of the shade detection mechanism, but the reconfiguration is done using SCCS algorithm. Therefore, the reconfiguration energy loss is lower in this case, which is reflected in the average yield, which is higher than the CC4AP + IR algorithm. Consequently, the yield is never lower than the ”no reconfiguration” scenario. The highest yield is 99.01% of perfect knowledge scenario and corresponds to $\Delta V_{threshold}$ of 0.19%

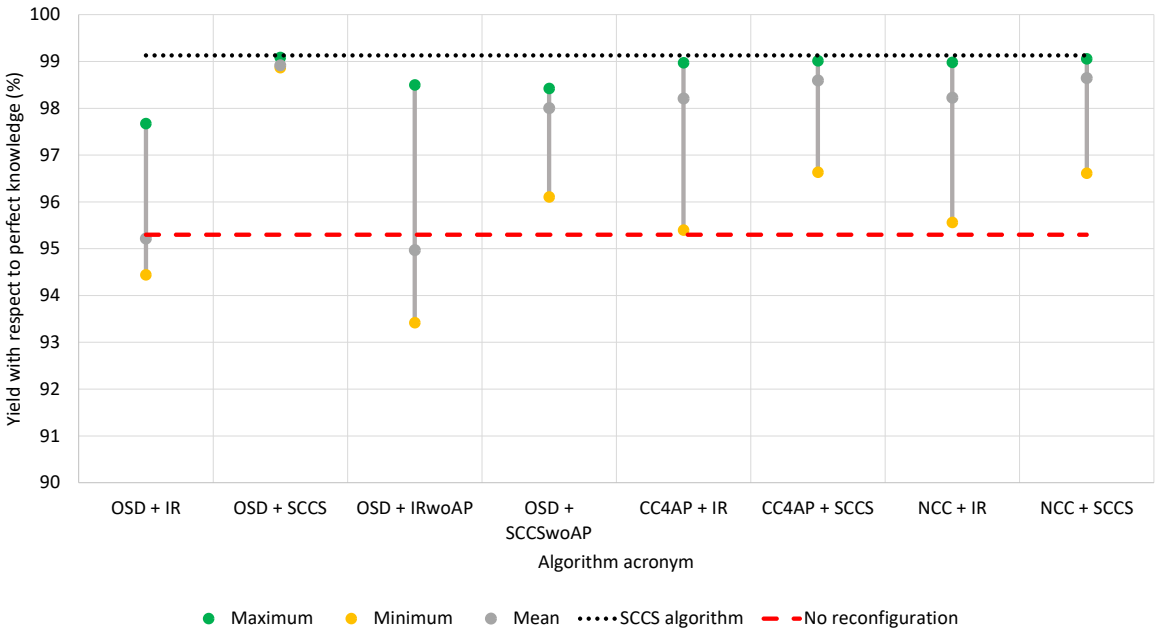



Figure 4.6: Yields of all the algorithm modifications. Each modification's highest, mean and lowest yield (with respect to the perfect knowledge scenario yield) is shown.

and $\Delta I_{threshold}$ of 0.63%.

The yield range for the **NCC + IR** algorithm is comparable to that of the CC4AP + IR algorithm. Similar to that of the CC4AP + IR algorithm, the yield is never lower than the "no reconfiguration" scenario. The average yield is 98.2% of the perfect knowledge scenario and the maximum yield is 98.9% of the perfect knowledge scenario, corresponding to $\Delta V_{threshold}$ of 0.89% and $\Delta I_{mpp,threshold}$ of 1%.

The **NCC + SCCS** algorithm is similar to the NCC + IR algorithm in terms of the shade detection mechanism, but the reconfiguration is done using SCCS algorithm. Therefore, the reconfiguration energy loss is lower in this case, which is reflected in the average yield, which is higher than the NCC + IR algorithm. Additionally, the yield is never lower than the "no reconfiguration" scenario. The highest yield is 99.05% of perfect knowledge scenario and corresponds to $\Delta V_{threshold}$ of 0.19% and $\Delta I_{mpp,threshold}$ of 8%.

## Reconfiguration count

Figure 4.7 shows the number of times reconfiguration happens for each modification. Similar to the yield plot, the highest, lowest and average reconfiguration count is shown in the figure. As a benchmark, the number of times reconfiguration happens in the perfect knowledge scenario is also shown. It needs to be noted that the maximum/minimum reconfiguration count does not always correspond to the reconfiguration count corresponding to the highest yield for each modification. Therefore, the reconfiguration count corresponding to the highest yield is also shown for reference.

One thing to note here is that the reconfiguration count for all the algorithms is limited by the number of daylight minutes. Therefore, for any of the algorithms discussed below, the maximum reconfiguration count would be 881, which corresponds to the total number of daylight minutes for 4th May, the day used in this particular simulation.

The average reconfiguration count for the **OSD + IR** algorithm is 1.3 times the perfect knowledge scenario. Additionally, the reconfiguration count corresponding to the maximum yield is equal to the lowest extreme of the count range. This implies that the algorithm triggers reconfiguration more often than is required, and to maximise yield the module should reconfigure as less as possible.

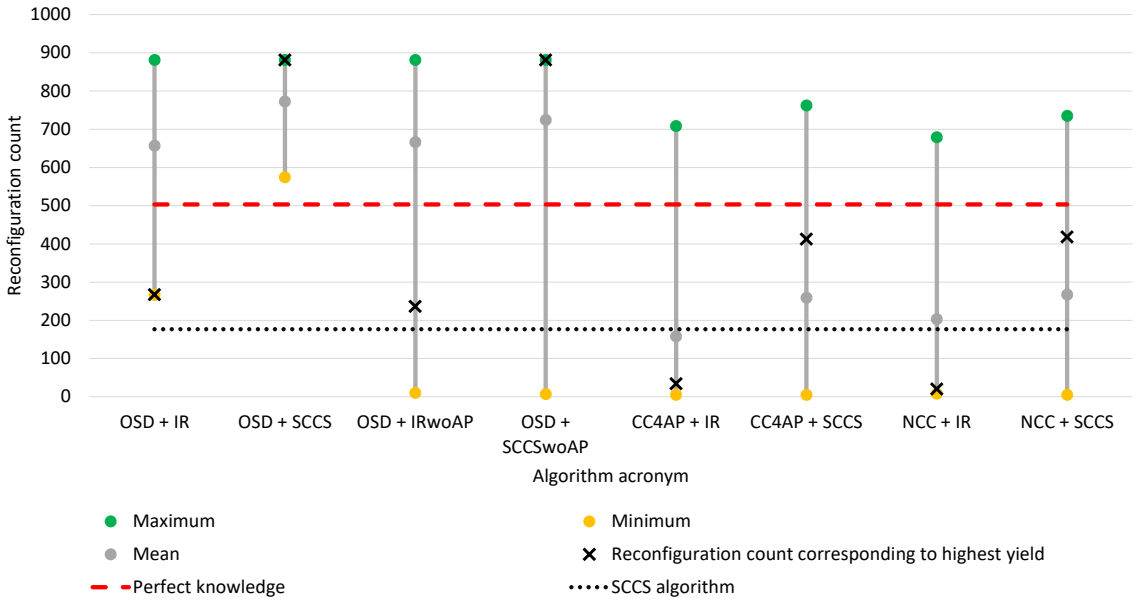



Figure 4.7: Reconfiguration count of all the algorithm modifications

The reason for the exceptionally high yields of the **OSD + SCCS** algorithm can be deduced from looking at the reconfiguration count for this algorithm. On an average, the reconfiguration count is approximately 1.5 times the perfect knowledge scenario. The reconfiguration count corresponding to the maximum yield is 881, which means that the yield is maximised when the algorithm triggers reconfiguration as often as possible, which in this case would mean that it runs every minute.

The reconfiguration count for the **OSD + IRwoAP** algorithm varies over a wide range. The average reconfiguration count is 1.3 times the perfect knowledge scenario. The reconfiguration count corresponding to the highest yield is approximately half of the perfect knowledge scenario. The high reconfiguration count and optimum $\Delta I_{threshold}$ of 0% indicate that the current check causes nuisance triggers and this algorithm functions optimally only with a voltage check.

Similar to the OSD + IRwoAP algorithm, the reconfiguration count varies over a wide range for the **OSD + SCCSwoAP** algorithm. However, unlike the OSD + IRwoAP algorithm, the reconfiguration count corresponding to the highest yield is equal to 881, which means that the yield is maximised when the algorithm triggers reconfiguration as often as possible, which would be every minute.

The average reconfiguration count for the **CC4AP + IR** algorithm is a lot lower than the previous modifications - it is 0.3 times the perfect knowledge reconfiguration count. Additionally, the reconfiguration count corresponding to the highest yield is only 34. For perspective, the module reconfigures 503 times in the perfect knowledge scenario and 176 times for the SCCS algorithm. This implies that although this algorithm reconfigures the module only a fraction of the time, the DC yield is still comparable to the SCCS algorithm, which is running every five minutes.

The average reconfiguration count for the **CC4AP + SCCS** algorithm is 0.5 times of the reconfiguration count in the perfect knowledge scenario. Additionally, the reconfiguration count corresponding to the highest yield is quite close to the perfect knowledge scenario. If the optimum thresholds of algorithms ”CC4AP + IR” and ”CC4AP + SCCS” are compared, it is observed that $\Delta V_{threshold}$ is lower for CC4AP + SCCS. This could imply that due to low reconfiguration losses, this algorithm will not lose as much energy due to nuisance triggers as CC4AP + IR. Therefore, the algorithm maximises its yield at a lower $\Delta V_{threshold}$ than the CC4AP + IR algorithm.

The average reconfiguration count for the **NCC + IR** algorithm is 0.4 times of the reconfiguration count

in the perfect knowledge scenario. Similar to the CC4AP + IR algorithm, the reconfiguration count corresponding to the highest yield is a very low value of 20. For perspective, the module reconfigures 503 times in the perfect knowledge scenario and 176 times for the SCCS algorithm. This implies that although this algorithm reconfigures the module only a fraction of the time, the DC yield is still comparable to the SCCS algorithm, which is running every five minutes.

The average reconfiguration count for the **NCC + SCCS** algorithm is 0.5 times of the reconfiguration count in the perfect knowledge scenario. Additionally, the reconfiguration count corresponding to the highest yield is quite close to the perfect knowledge scenario. If the optimum thresholds of algorithms NCC+IR and NCC+SCCS algorithms are compared, it is observed that $\Delta V_{threshold}$ is lower for NCC + SCCS algorithm. This could imply that due to low reconfiguration losses, this algorithm will not lose as much energy due to nuisance triggers as the NCC + IR algorithm. Therefore, it maximises its yield at a lower $\Delta V_{threshold}$ than the NCC + IR algorithm.

## Algorithm accuracy

Figure 4.8 shows the accuracy of each modification. Accuracy was calculated as follows:

$$Accuracy = \frac{C_{accurate}}{C_{total}} \tag{4.1}$$

where $C_{accurate}$ is the number of times the configuration chosen by the algorithm is the same as the configuration chosen by the perfect knowledge scenario, $C_{total}$ is the total number of time instants.

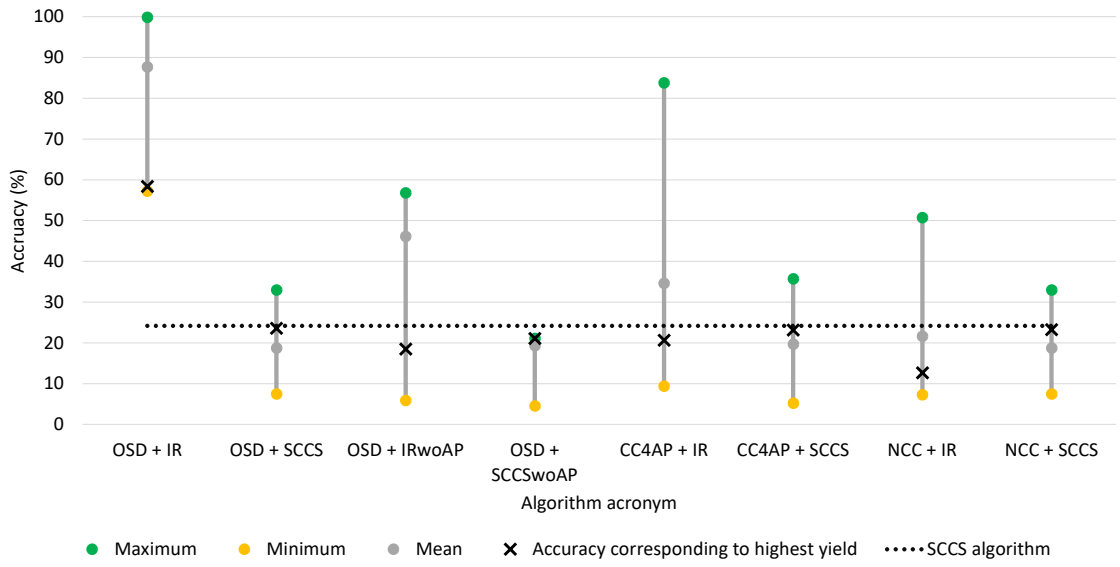



Figure 4.8: Accuracy of all the algorithm modifications

From the figure it is clear that the reconfiguration mechanism of the algorithm is the main influencing factor in its reconfiguration accuracy.

For algorithms which use the ideal reconfiguration mechanism, some of them show the highest average accuracy (in particular, algorithms **OSD + IR**, **OSD + IRwoAP** and **CC4AP + IR**). But that can be attributed to their high reconfiguration counts, which means that their shade detection mechanism almost always triggers reconfiguration and ensures that the module is configured as per the perfect knowledge scenario.

Although algorithm **NCC + IR** also uses the ideal reconfiguration mechanism, its average accuracy is much lower than the other algorithms which use the same mechanism. However, if the high average yield of this algorithm is considered (which is 98% of the perfect knowledge case) it implies that although the algorithm may not choose the accurate configuration every time, it still ensures a high DC yield.

Also, the average accuracy of this algorithm is similar to that of the SCCS algorithm, which runs every five minutes.

For algorithms which used SCCS as the reconfiguration mechanism (in particular, algorithms **OSD + SCCS**, **OSD + SCCSwoAP**, **CC4AP + SCCS** and **NCC + SCCS**), average accuracy of the algorithm was comparable to the accuracy of the SCCS algorithm. This could be attributed to the fact that all these algorithms have the same reconfiguration mechanism, that is, the SCCS algorithm.

## 4.3. Optimising the sensor block position

For the algorithms involving $\Delta V_{threshold}$ and $\Delta I_{threshold}$, one of the cell blocks of the PV module need to be assigned as a sensor block. Therefore, it needs to be analysed if the position of the sensor block on the module makes any difference on the algorithm's performance. For this, the threshold values were set to the optimum values as derived from the optimisation in section 4.2.1. The algorithm used was CC4AP + IR.

Figure 4.9 shows the result of this optimisation. The DC yield varies by 0.5% (with respect to the perfect knowledge scenario) when the sensor block is varied, but varies in within a 4% (with respect to the perfect knowledge scenario) range based on the threshold values. As can be seen, the position of the sensor block does not play a major role in influencing the algorithm's performance, as compared to the influence the values of $\Delta V_{threshold}$ and $\Delta I_{threshold}$ have.

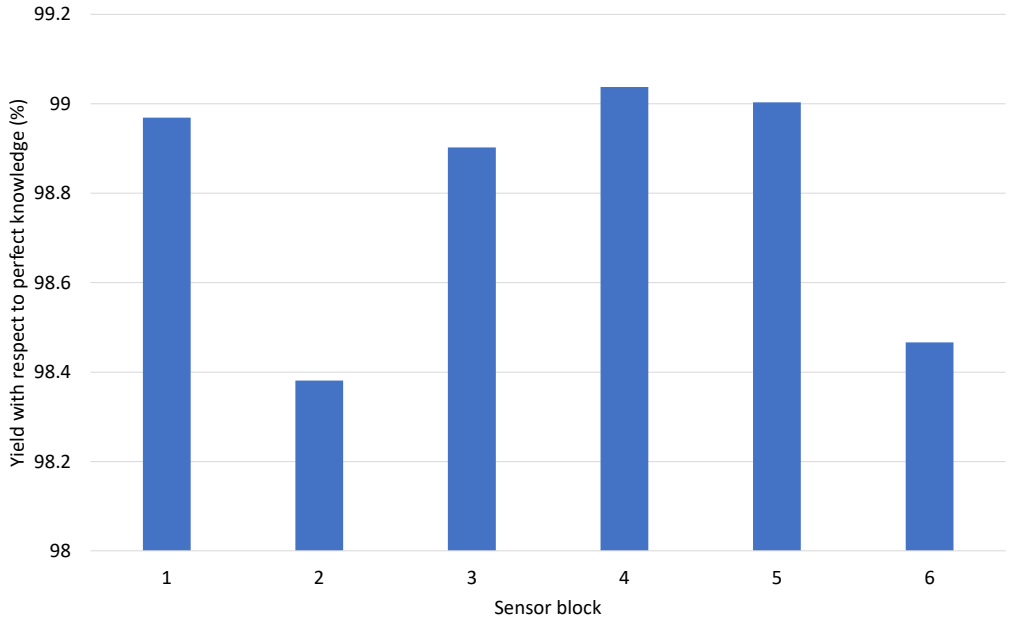



Figure 4.9: Yield for different sensor blocks

## 4.4. Conclusion

This chapter pertained to the third sub-goal of section 1.2 - optimising the new reconfiguration algorithm. Section 4.2.1 discussed numerous modifications to the new asynchronous reconfiguration algorithm. Each algorithm was analysed based on three metrics - the average DC yield of the module, the reconfiguration count and algorithm accuracy. The ideal algorithm was decided based on the one which has the highest yield, average reconfiguration count comparable to the perfect knowledge scenario and average accuracy comparable/higher than the SCCS algorithm.

In terms of yield, algorithms OSD+SCCS, OSD+SCCSwoAP, CC4AP+IR, CC4AP+SCCS, NCC+IR and NCC+SCCS had the highest average yields. Additionally, they always performed better than the conventional module.

However, algorithms OSD+SCCS and OSD+SCCSwoAP show high average reconfiguration counts

with respect to perfect knowledge scenario. Additionally, the reconfiguration count corresponding to their highest yields was equal to 881, which implied that they were running every minute. This is a drawback for the algorithms as the main aim was to have it running only when partial shading scenarios are detected.

On the other hand, algorithms CC4AP+IR, CC4AP+SCCS, NCC+IR and NCC+SCCS had lower reconfiguration counts in terms of the perfect knowledge scenario and comparable to that of the SCCS algorithm. This implies that although the algorithms were running much less frequently as compared to the perfect knowledge scenario, they were running as frequently as the SCCS algorithm and their average yields were also comparable to it.

In terms of average accuracy, algorithms CC4AP+SCCS, NCC+IR and NCC+SCCS have a comparable accuracy to the SCCS algorithm and for algorithm CC4AP+IR it is higher than SCCS algorithm.

Based on these observations, algorithms CC4AP+IR, CC4AP+SCCS, NCC+IR and NCC+SCCS appear to be the most optimum algorithms in terms of performance. In these algorithms, the shade detection mechanism performs voltage check irrespective of configuration and current check only for all-parallel configurations. The reconfiguration mechanism is either the ideal reconfiguration mechanism (IR), or the SCCS algorithm (SCCS).

One additional advantage for algorithm CC4AP+IR is that it only requires four inputs - it requires module voltage, current and a cell block's voltage, current as input for the shade detection mechanism. On the other hand, algorithms CC4AP+SCCS, NCC+IR and NCC+SCCS require eight inputs - they need module's voltage, current and single cell block's voltage, current for shade detection and each cell block's short circuit current for reconfiguration.

Therefore, it can be concluded that the CC4AP+IR algorithm is the most optimal of all the eight modifications that were defined. This conclusion is drawn based on the module's performance and the fact that it requires the least amount of input parameters. The next chapter will test this modification on actual on-site data and analyse its performance.

# 5

# Validating the reconfiguration algorithm

Chapter 4 optimised the reconfiguration algorithm. In this chapter, the algorithm is tested on a real-life PV installation. Section 5.1 describes the PV installation and the data that is collected from the same. Section 5.2 describes the simulation procedure followed to compare the performance of the new proposed reconfiguration algorithm with the existing SCCS algorithm. Section 5.3 discusses the results of the said simulation.

## 5.1. Details of the PV installation

The PVMD research group at TU Delft currently has an installation of reconfigurable PV modules (see figure 5.1). The setup is being used to test and validate reconfigurable algorithms. Similar to the module used during the simulations for this thesis, the installed modules are 6 block, 96 cell PV modules.

Figure 5.1: Photo of the reconfigurable modules installed by the PVMD research group (image courtesy: Andres Calcabrini, PVMD research group, TU Delft)

Figure 5.2 shows the a schematic of the switching circuit for this module. Each cell block is a set of sixteen 5-inch (125mm by 125 mm) solar cells connected in series. The individual cells have a maximum power of 2.72W under standard test conditions (STC). These cell blocks are connected to each other via MOSFET switches. The module requires 2 MOSFET switches per cell block to connect them to the positive and negative module terminal. It also needs bidirectional switches connecting the cell blocks with each other. The reason these switches need to be bidirectional is to avoid short-circuiting the cell blocks through the MOSFET body diodes. In the actual installation, the bidirectional switches are implemented as two back-to-back MOSFETs in series, which makes their resistance equal to twice the resistance of the single MOSFET switches. As a result of such a design, the module always has resistance equal to $2R_{sw}$ ($R_{sw}$ is resistance of a single MOSFET) in series with each cell block.

The module also has sensors to measure the cell block's and module's current and voltage. The module's voltage is being measured using an electronic load while the cell block voltages are measured using voltage dividers. The cell block has current sensors in series to measure cell block current.

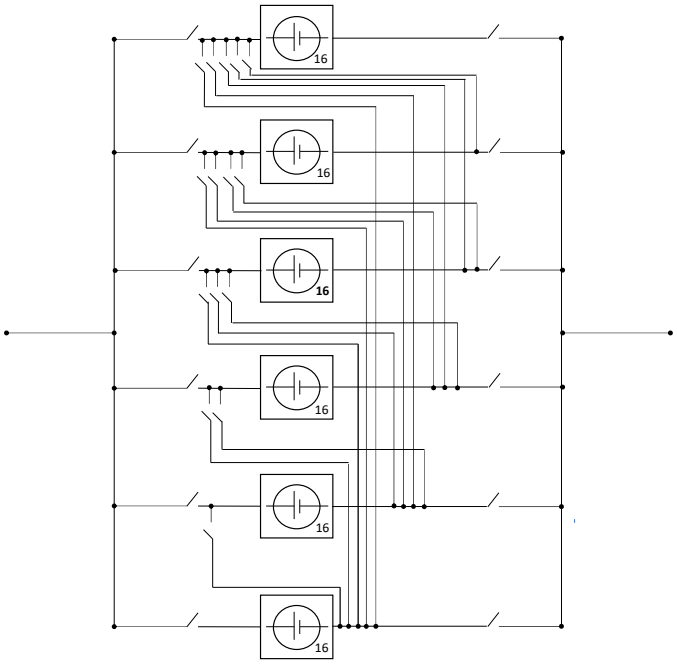



Figure 5.2: Switching circuit in the reconfigurable module [32]

## 5.2. Details of the experiment

Given the time frame of the thesis it was not feasible to validate the performance of the proposed algorithm on the actual module. Instead, using the cell and module IV curves collected via the attached sensors, the module's performance was predicted in case it were to be run using the new algorithm. Then it was compared to the performance of the SCCS algorithm (which was tested on the module).

Andres Calcabrini from the PVMD research group provided the data from an experiment conducted to test the SCCS algorithm. In this experiment, the SCCS algorithm was implemented on the installed module and its performance was monitored from 17th May 2021 to 7th June 2021. The algorithm was run at 30 second time interval and the best configuration for the module was chosen during each run. Additionally, the individual cell block's IV curves were also recorded during each run of the algorithm. These IV curves were used to calculate the yield of module for the duration of the experiment. Additionally, the IV curves were also used for simulating the behaviour of the module if it were to be operated using the new proposed algorithms.

From the four reconfiguration algorithms that were found to be promising in chapter 4, the performance of algorithms CC4AP+IR and CC4AP+SCCS were analysed in this experiment. Since algorithms CC4AP+IR and CC4AP+SCCS differ by reconfiguration mechanisms which have a extremely different magnitudes of reconfiguration losses, simulating both their performance will help in understanding if reconfiguration losses have a substantial effect on the DC yield. The optimum $\Delta V_{threshold}$ and $\Delta I_{threshold}$ values determined in section 4.2.1 were used for the simulations.

In the experiment performed on the installed PV module, the SCCS algorithm was run at 30 seconds interval. An additional scenario was simulated where the SCCS algorithm is run at 5 minute time interval,

as the yield calculated in this scenario will be much closer to real-life application of the algorithm.

## 5.3. Results

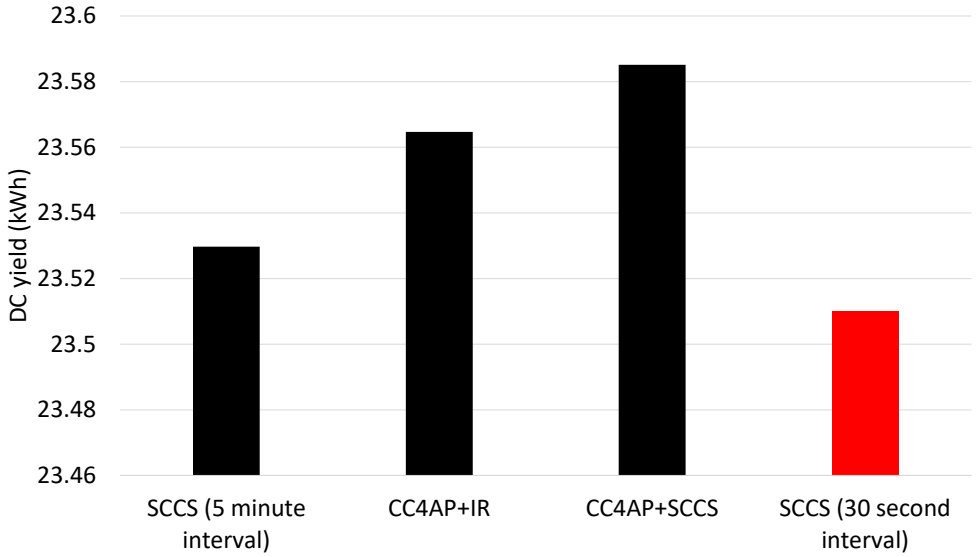



Figure 5.3: Comparing the yields of different algorithms. Marked in red is the yield of the algorithm that was actually implemented on the PV installation

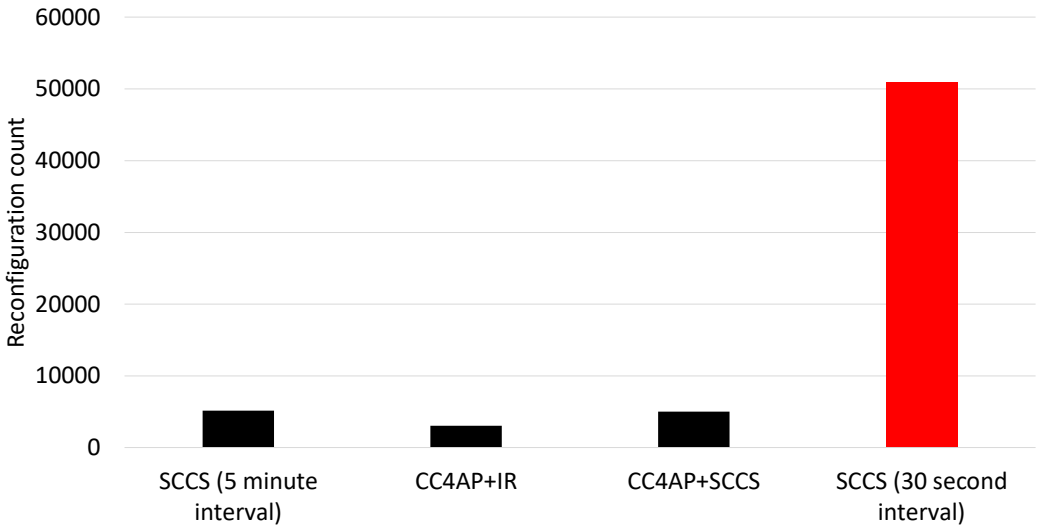



Figure 5.4: Comparing the reconfiguration counts for different algorithms. Marked in red is the reconfiguration count for the algorithm that was actually implemented on the PV installation

Figure 5.3 shows the yield of the various scenarios that were simulated. When the module is controlled using SCCS algorithm at 30 second interval, its yield is the lowest at 23.5 kWh. However, there is not a significant increase in yield in case the algorithm were run at 5 minute interval. Similarly in the case of the CC4AP+IR algorithm, the yield is higher than the SCCS algorithm, but not by a substantial margin.

However, the difference between the algorithms becomes stark when their reconfiguration counts are compared. The SCCS algorithm, when run at 30 second interval, ends up running 51,021 times. In comparison, running the same algorithm for 5 minutes interval would mean that it runs only 1/10th of the time and with a comparable DC yield. Similarly, algorithm CC4AP+IR runs only 6% of the time but

due to the low reconfiguration counts, they have lower reconfiguration losses and consequently higher yields.

An additional thing to note here is that the CC4AP+IR algorithm's yield is not substantially lower than the CC4AP+SCCS algorithm. This could be explained by the fact that the CC4AP+IR algorithm triggers reconfiguration less frequently than the CC4AP+SCCS algorithm.

## 5.4. Conclusion

This chapter pertains to the fourth sub-goal of section 1.2 - validating the asynchronous CC4AP+IR algorithm and comparing its performance to the synchronous SCCS algorithm. It was observed that the CC4AP algorithm shows a comparable yield to the SCCS algorithm, but at a substantially lower reconfiguration count.

As pointed out earlier, the reconfigurable module cannot be controlled using a synchronous algorithm running every 30 seconds, owing to the problems it would cause with the power converter. Therefore, the significance of having an asynchronous CC4AP+IR algorithm or a shade detection mechanism is conveyed through this experiment. The algorithm ensures that the module is reconfiguring at crucial instants, hence ensuring optimum yield at less reconfiguration count.

# 6

# Conclusions and recommendations

In this chapter, the aim of the thesis will be recapitulated and the research questions answered. Additionally, the conclusions drawn from this work and recommendations for future work will also be discussed. The aim of the thesis was **"To develop a robust control strategy for reconfigurable PV modules"**. The next sections - section 6.1, 6.2, 6.3 and 6.4 - discuss each sub-goal that was defined in section 1.2 and draw conclusions from the work done to achieve them. Finally, section 6.5 discusses the recommendations for future research.

## 6.1. Modelling reconfigurable modules in the PVMD toolbox

The current version of the PVMD toolbox has limited functionality in terms of simulating partial shading and modelling reconfigurable PV modules. Therefore, the first sub-goal of the thesis was to enable modelling of reconfigurable PV modules in the PVMD toolbox and simulate partial shade conditions to test their performance.

Chapter 2 discusses the work done to achieve this sub-goal. For simulating partial shade conditions, external softwares like Rhinoceros and RADIANCE were employed. For modelling reconfigurable modules, first a simulation framework was established. As per the simulation framework, the reconfigurable module's IV characteristics could be determined using cell-wise irradiance conditions and a reconfiguration algorithm which determines its optimum configuration at each time step. The algorithm employed here was the SCCS algorithm developed in the PVMD research group.

The SCCS algorithm is a synchronous reconfiguration algorithm for series-parallel configurations, which employs short-circuit current of the module's cell blocks as input. Since it requires short-circuit current of each cell block as input, this leads to energy loss every time the cell blocks are disconnected to measure the short circuit currents. Additionally, since the algorithm does not have any shade detection mechanism, it needs to be run at regular time intervals. If the duration of time between two subsequent runs of the algorithm is longer than the duration between change in irradiance conditions, the algorithm may miss some instants when reconfiguration is necessary. It was concluded that the drawbacks of the algorithm stem from the fact that it does not have a shade detection mechanism.

## 6.2. Defining a shade detection mechanism for reconfiguration algorithm

The current control strategy has certain drawbacks, which stem from the fact that it does not have a shade detection mechanism and that the module needs to be disconnected from the power system every time the algorithm has to run. Therefore, the second sub-goal was to explore and identify alternate module parameters that could function as reliable shade detection triggers and also ensure that the module does not have to be disconnected from the system every time the algorithm runs.

Chapter 3 explored various module parameters (namely current/voltage of individual cell blocks) and their relationship with relative irradiance difference over the module. This analysis was done to see

which of them can be reliable shade detection triggers. Additionally, the relationship between these parameters and irradiance was also observed to identify if any of them can function as reliable indicators of irradiance on the individual cell blocks. It was concluded that only $I_{sc}$ can function as a reliable shade detection trigger as well as an indicator of irradiance level. However, using $I_{sc}$ as a control variable would mean that the reconfiguration algorithm wouldn't run when the module is connected to the system. Therefore, an alternate reconfiguration algorithm was developed in which the shade detection mechanism uses a combination of the module's and a cell block's operating current and voltage to trigger reconfiguration.

## 6.3. Defining a reconfiguration algorithm using the shade detection mechanism

Chapter 4 develops the new reconfiguration algorithm further by optimising the threshold values in the algorithm and also the position of the sensor block. It was observed that the best reconfiguration algorithm in terms of performance and the number of input sensors required was the CC4AP+IR algorithm, which employed a "voltage check" at all times, a "current check" when the module was in an all-parallel configuration and used the ideal reconfiguration mechanism. The average DC yield of the CC4AP+IR algorithm for a single day was calculated to be 98.2% of an ideal scenario's yield while the SCCS algorithm was 99.13% of the ideal scenario's yield. On an average, the CC4AP+IR algorithm ran 158 times over an entire day. In comparison, the SCCS algorithm (running once every 5 minutes) ran 176 times in the day. Hence, despite having a low reconfiguration count, the DC yield of the asynchronous algorithms was comparable to the SCCS algorithm.

The effect of changing the position of the sensor block on the algorithm's performance was also studied. It was observed that changing the sensor block position did not greatly affect the yield of the module. For comparison, the DC yield varied within a 0.5% range when the sensor block position was varied, but varied within a 4% range when the threshold values of the algorithm were varied.

## 6.4. Validating the developed control strategy

This sub-goal involved testing the the new reconfiguration algorithm using real-life data and comparing their performance to the SCCS algorithm.

Chapter 5 explains the methodology followed to perform the analysis. An experiment was already being run by the PVMD research group to test the SCCS algorithm. Using the data generated from the experiment (between 17th May 2021 and 7th June 2021), the module's DC yield was calculated for the SCCS algorithm and for the new reconfiguration algorithm CC4AP+IR. It was observed that the DC yield of the module was 0.2% higher for the CC4AP+IR algorithm, which is below the measurement uncertainty of 1% for the experimental data. However, on comparing reconfiguration counts, the contrast between the algorithms was a lot clearer. It was observed that the new algorithm ran merely 6% of the time compared to the SCCS algorithm, but generated a comparable DC yield.

Therefore, the new reconfiguration algorithm demonstrated to have some key benefits over the SCCS algorithm. It runs for only a fraction of the time, which means that possible power fluctuations and losses during reconfiguration are greatly reduced. The sensing of the control variables for shade detection can be done without disconnecting the module from the system, which makes the algorithm asynchronous and also ensures low reconfiguration losses. Additionally, only four sensors are required for the input data for shade detection, irrespective of the number of cell blocks in the module. In comparison, the SCCS algorithm would require one current sensor per cell block.

## 6.5. Recommendations for future work

This section discusses some recommendations which will further help in developing the new reconfiguration algorithm as well as the simulation framework in the PVMD toolbox.

In the current model of reconfigurable modules in the PVMD toolbox, additional switching resistances are considered while calculating the block-wise IV curves. However, in comparison to a conventional module, the reconfigurable module will also have some additional wiring resistances, which need to be studied and quantified.

For the reconfiguration algorithm, various module parameters were explored as potential shade detection triggers or irradiance level indicators. In future work, non-electric variables (like temperature) could also be explored for this purpose.

The reconfiguration algorithm currently implemented in the PVMD toolbox is specifically for a six block PV module with 16 cells in series. Further research could be done on making the algorithm more modular, based on the number of cell blocks in the PV module. Additionally, the algorithm's performance could also be analysed in a scenario with multiple PV modules as part of a PV array.

As discussed in the thesis, reconfigurable module have additional hardware components such as voltage/current sensors, switches etc. The modules will also have additional cost of wiring, a module-level microcontroller for MPPT and reconfiguration algorithm [52]. These components will add up to the initial cost of the module and result in the reconfigurable modules being more expensive than conventional modules. On the other hand, research has indicated that reconfigurable modules show a gain in performance over conventional modules when used in scenarios where partial shading is present majority of the time [53]. This could also help compensate for the additional investment and translate as financial gain over a longer duration of time. Therefore, the new reconfiguration algorithm needs to be tested over a longer duration of time to judge if it outperforms a conventional module under varying partial shading conditions, and compensates for the additional initial cost of the module.

# Bibliography


[1] IEA, *Global energy review 2020*. [Online]. Available: https://www.iea.org/reports/global-energy-review-2020/renewables, (accessed on 10-12-2020).

[2] A. Calcabrini, M. Muttillo, R. Weegink, P. Manganiello, M. Zeman, and O. Isabella, "A fully reconfigurable series-parallel photovoltaic module for higher energy yields in urban environments," *Renewable Energy*, 2021.

[3] K. Kurokawa, "Realistic values of various parameters for pv system design," *Renewable Energy*, vol. 15, no. 1-4, pp. 157–164, 1998.

[4] J. Leloux, L. Narvarte, and D. Trebosc, "Performance analysis of 10,000 residential pv systems in france and belgium," in *26th European Photovoltaic Solar Energy Conference and Exhibition, Hamburg*, 2011.

[5] Y. Ueda, K. Kurokawa, K. Kitamura, M. Yokota, K. Akanuma, and H. Sugihara, "Performance analysis of various system configurations on grid-connected residential pv systems," *Solar energy materials and solar cells*, vol. 93, no. 6-7, pp. 945–949, 2009.

[6] S. R. Pendem and S. Mikkili, "Modelling and performance assessment of pv array topologies under partial shading conditions to mitigate the mismatching power losses," *Solar Energy*, vol. 160, pp. 303–321, 2018.

[7] W. Herrmann, W. Wiesner, and W. Vaassen, "Hot spot investigations on pv modules-new concepts for a test standard and consequences for module design with respect to bypass diodes," in *Conference Record of the Twenty Sixth IEEE Photovoltaic Specialists Conference-1997*, IEEE, 1997, pp. 1129–1132.

[8] B. B. Pannebakker, A. C. de Waal, and W. G. van Sark, "Photovoltaics in the shade: One bypass diode per solar cell revisited," *Progress in photovoltaics: Research and Applications*, vol. 25, no. 10, pp. 836–849, 2017.

[9] S. M. MacAlpine, R. W. Erickson, and M. J. Brandemuehl, "Characterization of power optimizer potential to increase energy capture in photovoltaic systems operating under nonuniform conditions," *IEEE Transactions on Power Electronics*, vol. 28, no. 6, pp. 2936–2945, 2012.

[10] M. Z. Ramli and Z. Salam, "Performance evaluation of dc power optimizer (dcpo) for photovoltaic (pv) system during partial shading," *Renewable energy*, vol. 139, pp. 1336–1354, 2019.

[11] H. A. Sher and K. E. Addoweesh, "Micro-inverters—promising solutions in solar photovoltaics," *Energy for Sustainable Development*, vol. 16, no. 4, pp. 389–400, 2012.

[12] R. Hasan, S. Mekhilef, M. Seyedmahmoudian, and B. Horan, "Grid-connected isolated pv microinverters: A review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 1065–1080, 2017.

[13] K. Sinapis, C. Tzikas, G. Litjens, M. Van den Donker, W. Folkerts, W. Van Sark, and A. Smets, "A comprehensive study on partial shading response of c-si modules and yield modeling of string inverter and module level power electronics," *Solar Energy*, vol. 135, pp. 731–741, 2016.

[14] A. Bidram, A. Davoudi, and R. S. Balog, "Control and circuit techniques to mitigate partial shading effects in photovoltaic arrays," *IEEE Journal of Photovoltaics*, vol. 2, no. 4, pp. 532–546, 2012. DOI: 10.1109/JPHOTOV.2012.2202879.

[15] B. I. Rani, G. S. Ilango, and C. Nagamani, "Enhanced power generation from pv array under partial shading conditions by shade dispersion using su do ku configuration," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 3, pp. 594–601, 2013. DOI: 10.1109/TSTE.2012.2230033.

[16] F. Belhachat and C. Larbes, "Modeling, analysis and comparison of solar photovoltaic array configurations under partial shading conditions," *Solar Energy*, vol. 120, pp. 399–418, 2015.

[17] V. Jha and U. S. Triar, “A detailed comparative analysis of different photovoltaic array configurations under partial shading conditions,” *International Transactions on Electrical Energy Systems*, vol. 29, no. 6, e12020, 2019.

[18] N. Mishra, A. S. Yadav, R. Pachauri, Y. K. Chauhan, and V. K. Yadav, “Performance enhancement of pv system using proposed array topologies under various shadow patterns,” *Solar Energy*, vol. 157, pp. 641–656, 2017.

[19] C. Chang, *Solar cell array having lattice or matrix structure and method of arranging solar cells and panels*, US Patent 6,635,817, Oct. 2003.

[20] R. A. Sherif and K. S. Boutros, *Solar module array with reconfigurable tile*, US Patent 6,350,944, Feb. 2002.

[21] S. Vemuru, P. Singh, and M. Niamat, “Analysis of photovoltaic array with reconfigurable modules under partial shading,” in *2012 38th IEEE Photovoltaic Specialists Conference*, IEEE, 2012, pp. 001 437–001 441.

[22] M. Balato, L. Costanzo, and M. Vitelli, “Reconfiguration of pv modules: A tool to get the best compromise between maximization of the extracted power and minimization of localized heating phenomena,” *Solar Energy*, vol. 138, pp. 105–118, 2016.

[23] L. Bouselham, B. Hajji, A. Mellit, A. Rabhi, and A. Mazari, “A reconfigurable pv architecture based on new irradiance equalization algorithm,” in *International Conference on Electronic Engineering and Renewable Energy*, Springer, 2018, pp. 470–477.

[24] G. Spagnuolo, G. Petrone, B. Lehman, C. A. Ramos Paja, Y. Zhao, and M. L. Orozco Gutierrez, “Control of photovoltaic arrays: Dynamical reconfiguration for fighting mismatched conditions and meeting load requests,” *IEEE Industrial Electronics Magazine*, vol. 9, no. 1, pp. 62–76, 2015. DOI: 10.1109/MIE.2014.2360721.

[25] D. La Manna, V. L. Vigni, E. R. Sanseverino, V. Di Dio, and P. Romano, “Reconfigurable electrical interconnection strategies for photovoltaic arrays: A review,” *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 412–426, 2014.

[26] A. M. Ajmal, T. S. Babu, V. K. Ramachandaramurthy, D. Yousri, and J. B. Ekanayake, “Static and dynamic reconfiguration approaches for mitigation of partial shading influence in photovoltaic arrays,” *Sustainable Energy Technologies and Assessments*, vol. 40, p. 100 738, 2020.

[27] G. S. Krishna and T. Moger, “Reconfiguration strategies for reducing partial shading effects in photovoltaic arrays: State of the art,” *Solar Energy*, vol. 182, pp. 429–452, 2019.

[28] D. N. LP *et al.*, “Pv reconfiguration systems: A technical and economic study.,” *Journal of Electrical Systems*, vol. 13, no. 1, 2017.

[29] P. Armstrong, R. Armstrong, R. Kang, R. Camilleri, D. Howey, and M. McCulloch, “A reconfigurable pv array scheme integrated into an electric vehicle,” 2013.

[30] A. Youssef, M. El Telbany, and A. Zekry, “Reconfigurable generic fpga implementation of fuzzy logic controller for mppt of pv systems,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1313–1319, 2018.

[31] D. Nguyen and B. Lehman, “A reconfigurable solar photovoltaic array under shadow conditions,” in *2008 Twenty-Third Annual IEEE Applied Power Electronics Conference and Exposition*, IEEE, 2008, pp. 980–986.

[32] A. Calcabrini, R. Weegink, O. Isabella, and M. Zeman, “A simulation study of reconfigurable modules for higher yields in partially shaded pv systems,” Jun. 2020.

[33] B. Patnaik, J. Mohod, and S. Duttagupta, “Distributed multi-sensor network for real time monitoring of illumination states for a reconfigurable solar photovoltaic array,” Mar. 2012, pp. 106–109, ISBN: 978-1-4673-1040-6. DOI: 10.1109/ISPTS.2012.6260892.

[34] G. Velasco-Quesada, F. Guinjoan-Gispert, R. Pique-Lopez, M. Roman-Lumbreras, and A. Conesa-Roca, “Electrical pv array reconfiguration strategy for energy extraction improvement in grid-connected pv systems,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4319–4331, 2009. DOI: 10.1109/TIE.2009.2024664.

[35] D. Nguyen and B. Lehman, “An adaptive solar photovoltaic array using model-based reconfiguration algorithm,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2644–2654, 2008. DOI: 10.1109/TIE.2008.924169.

[36] A. Laudani, G. M. Lozito, V. Lucaferri, M. Radicioni, and F. R. Fulginei, “On circuital topologies and reconfiguration strategies for pv systems in partial shading conditions: A review,” *Aims Energy*, vol. 6, no. 5, pp. 735–763, 2018.

[37] P. Romano, R. Candela, M. Cardinale, V. Li Vigni, D. Musso, and E. Sanseverino, “Optimization of photovoltaic energy production through an efficient switching matrix,” *Journal of Sustainable Development of Energy, Water and Environment Systems*, vol. 1, pp. 227–236, Apr. 2013. DOI: 10.13044/j.sdewes.2013.01.0017.

[38] M. Alahmad, M. A. Chaaban, S.-K. Lau, J. Shi, and J. Neal, “An adaptive utility interactive photovoltaic system based on a flexible switch matrix to optimize performance in real-time,” *Solar Energy*, vol. 86, pp. 951–963, Mar. 2012. DOI: 10.1016/j.solener.2011.12.028.

[39] A. Nour El Din, “Improved electrical model and experimental validation of the pvmd toolbox: Extending the energy yield prediction model to tandem pv modules,” 2020.

[40] R. Santbergen, T. Meguro, T. Suezaki, G. Koizumi, K. Yamamoto, and M. Zeman, “Genpro4 optical model for solar cell simulation and its application to multijunction solar cells,” *IEEE journal of photovoltaics*, vol. 7, no. 3, pp. 919–926, 2017.

[41] R. Santbergen, V. Muthukumar, R. Valckenborg, W. van de Wall, A. Smets, and M. Zeman, “Calculation of irradiance distribution on pv modules by combining sky and sensitivity maps,” *Solar Energy*, vol. 150, pp. 49–54, 2017.

[42] *Rhinoceros - features*. [Online]. Available: https://www.rhino3d.com/features/, (accessed on 27-7-2021).

[43] J. de Sousa Freitas, J. Cronemberger, and R. M. Soares, “Using rhinoceros plugins grasshopper and ladybug to assess bipv façades in brasília,”

[44] *Rhinoceros - grasshopper*. [Online]. Available: https://www.rhino3d.com/6/new/grasshopper/, (accessed on 27-7-2021).

[45] *Ladybug tools | food4rhino*. [Online]. Available: https://www.food4rhino.com/en/app/ladybug-tools, (accessed on 27-7-2021).

[46] *About radiance*. [Online]. Available: https://www.radiance-online.org/about, (accessed on 27-7-2021).

[47] M. Green, E. Dunlop, J. Hohl-Ebinger, M. Yoshita, N. Kopidakis, and X. Hao, “Solar cell efficiency tables (version 57),” *Progress in photovoltaics: research and applications*, vol. 29, no. 1, pp. 3–15, 2021.

[48] *Meteonorm*. [Online]. Available: https://meteonorm.com/en/, (accessed on 30-4-2021).

[49] A. H. Smets, K. Jäger, O. Isabella, R. A. Swaaij, and M. Zeman, *Solar Energy: The physics and engineering of photovoltaic conversion, technologies and systems*. UIT Cambridge, 2015.

[50] J. I. Morales-Aragonés, S. Gallardo-Saavedra, V. Alonso-Gómez, F. J. Sánchez-Pacheco, M. A. González, O. Martínez, M. A. Muñoz-García, M. d. C. Alonso-García, and L. Hernández-Callejo, “Low-cost electronics for online iv tracing at photovoltaic module level: Development of two strategies and comparison between them,” *Electronics*, vol. 10, no. 6, p. 671, 2021.

[51] M. Herman, M. Jankovec, and M. Topič, “Optimal iv curve scan time of solar cells and modules in light of irradiance level,” *International Journal of Photoenergy*, vol. 2012, 2012.

[52] M. Baka, P. Manganiello, D. Soudris, and F. Catthoor, “A cost-benefit analysis for reconfigurable pv modules under shading,” *Solar Energy*, vol. 178, pp. 69–78, 2019.

[53] M.-I. Baka, F. Catthoor, and D. Soudris, “Near-static shading exploration for smart photovoltaic module topologies based on snake-like configurations,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 15, no. 2, pp. 1–21, 2016.