

Improving Electrical Performance of Reliable Solar-Powered Irrigation System

by

Thipok Rak-annouykit

S.B., Massachusetts Institute of Technology (2016)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

© 2017 Massachusetts Institute of Technology. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
May 26, 2017

Certified by.....
Prof. David J. Perreault
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
Dr. Christopher Terman
Chairman, Masters of Engineering Thesis Committee

Improving Electrical Performance of Reliable Solar-Powered Irrigation System

by

Thipok Rak-amnouykit

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 2017, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

A solar-powered irrigation system has been developed to address the lack of affordable irrigation solution for the marginal farmers in India. An MIT spinout, Khethworks, has designed an efficient water pump with low power rating, and created a low-cost irrigation system with the pump, a photovoltaic panel, and a battery. This thesis analyzes the electrical properties of such a configuration, and determines whether implementing maximum power point tracking (MPPT) can improve the system's performance. This is accomplished through modeling and conducting a system-level simulation.

The amount of electrical energy generated by the photovoltaic panel and the amount of water delivered by the pump are chosen as the key measures of the system's performance. The simulation result indicates that implementing MPPT with the current version of the Khethworks irrigation system - using lower-power panels of the 48 or 60 cell variety - would not significantly increase its performance. However, an irrigation system with higher power rating (e.g., a 72-cell panel, such as with a 320 W rating) would significantly benefit from the MPPT, and the MPPT's benefits are consistent over the variation in location and time. Based on the finding, we identify circumstances under which each of the direct load line and the maximum power point tracking approaches are preferable, and recommend an action plan to Khethworks accordingly.

Thesis Supervisor: Prof. David J. Perreault

Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank:

Prof. David Perreault, my thesis supervisor, for his continuous support and guidance throughout the duration of this research, and ultimately for sparking my interest in Power Electronics.

Dr. Wardah Inam and members of the Power Electronics Research Group, as well as **Daniel Moon** and **Michael Holachek**, for helping me in multiple ways related to this research.

Kevin Simon and **Victor Lesniewski**, the Khethworks team members, for their insights into the mechanical characteristics of the pump system and for the data from the pump's test runs.

Dr. Robert Stoner and **Dr. Reja Amatya** from the Tata center for giving me the opportunity to work on this project and learn the challenge of applying technology and design to solve problems in emerging communities

Contents

1	Introduction	21
1.1	Motivation and Objectives	22
1.2	Organization	23
2	Background	25
2.1	Marginal Farmers in East India	25
2.2	Conventional Irrigation Solution	29
2.2.1	Diesel Pumps	29
2.2.2	Electric Pumps	29
2.3	Khethworks	30
2.3.1	Pump Design for Small Farms	30
2.3.2	Electrical Inefficiency	31
3	Electrical Components	33
3.1	Photovoltaic Panel	34
3.1.1	Circuit Model	34
3.1.2	Cell Temperature T_c	36
3.1.3	Maximum Power Point	39
3.1.4	Explicit Form of v for any T_c and G	44
3.1.4.1	Open-Circuit Voltage V_{OC}	45
3.1.4.2	Photogenerated Current I_L	46
3.1.4.3	Series Resistance R_S	47
3.2	Battery	48

4	Pump System	53
4.1	Motor	54
4.2	Impeller	54
4.3	Mechanical Load	56
4.4	Pump's Operating Points	58
4.5	Test Results	59
5	Maximum Power Point Tracking	63
6	System's Operating Points	69
6.1	Case 1: Pump OFF	71
6.2	Case 2: Pump On	75
7	Theoretical Simulation	81
7.1	Input Variables	83
7.1.1	Location and Time	83
7.1.2	Irrigation System	84
7.1.3	User Preference	86
7.2	Simulation Output	87
7.3	Simulation Process	88
7.4	Simulation Results	90
7.4.1	Panel Rating	90
7.4.2	Converter Efficiency	97
7.4.3	Battery Quantity	99
7.4.4	Depth of Discharge	101
7.4.5	Location	101
7.4.6	Year	101
7.5	Summary of Findings	105
8	Conclusion and Future Work	107
8.1	Recommendation for Khethworks	107
8.1.1	Current Version of the Khethworks System	107

8.1.2	Optimization for the DLL Approach	108
8.1.3	Battery Capacity	108
8.1.4	Consideration for the MPPT Approach	109
A	Simulation Source Code	111
A.1	Simulation	111
A.1.1	Source File	111
A.1.2	Plotting Script	113
A.2	Core Functions	116
A.2.1	DLL Simulator	116
A.2.2	MPPT Simulator	120
A.3	Helper Functions	125
A.3.1	DLL Operating Point 1	125
A.3.2	DLL Operating Point 2	127
A.3.3	DLL Operating Point 3	127
A.3.4	Water Demand	128
A.3.5	Photovoltaic Panel	128
A.3.6	Pump System	139

List of Figures

2-1	Distribution of electric water pumps in India, 2000-01 [3]	26
2-2	Depth to water level during pre-monsoon season in India and Jharkhand state, 2016 [4]	27
2-3	Annual average of global horizontal irradiance in India, based on hourly estimation from 2002 to 2008 [5]	28
3-1	Diagram highlighting electrical components in the Khethworks irrigation system.	33
3-2	An equivalent circuit of a photovoltaic panel	35
3-3	Plot of the photovoltaic cell temperature as a function of time - with ambient temperature, global horizontal irradiation, and wind speed as reference.	39
3-4	Plot of (a) output current and (b) output power of a typical photovoltaic cell as a function of the cell voltage at $G = 1000 \text{ W/m}^2$, for four different cell temperatures	41
3-5	Plot of (a) output current and (b) output power of a typical photovoltaic cell as a function of the cell voltage at $T_c = 25^\circ\text{C}$, for five different irradiation levels	42
3-6	Plots showing linear trend of a photovoltaic cell's maximum output power over (a) irradiation and (b) cell temperature	43
3-7	Plots showing linear trend of a photovoltaic cell's open-circuit voltage over (a) irradiation and (b) cell temperature	45
3-8	$i - v$ curve near the vicinity of $v = V_{OC}$	47

3-9	A circuit model of a battery	50
3-10	An $i - v$ characteristic curve of a battery model in Figure 3-9	50
3-11	Plot of battery terminal voltage as a function of discharged amount at two different discharging rates for a 25.6 V, 1.4 Ah LiFePO ₄ battery	51
4-1	Diagram highlighting pump components in Khethworks irrigation system	53
4-2	Typical $\tau - \omega$ characteristic curves of a brushless DC motor at a constant PWM throttle and varying voltage levels	55
4-3	Typical $\tau - \omega$ characteristic curves of a brushless DC motor at a constant voltage level and varying PWM throttles	55
4-4	Typical characteristic curves of a centrifugal pump's impeller in (a) the torque-angular speed space and (b) the pressure-flow rate space	56
4-5	Typical $H - Q$ characteristic curve of the pump's mechanical load in Khethworks irrigation system	57
4-6	Pump's operating points in the $\tau - \omega$ space at a constant voltage level and varying PWM throttles	58
4-7	Pump's operating points in the $H - Q$ space at a constant PWM throttle and varying voltage levels	59
4-8	Plot of the motor's angular speed ω as a function of the pressure head H at 3 voltage levels and a constant PWM throttle	61
4-9	Plot of the pump's flow rate Q as a function of the pressure head H at 3 voltage levels and a constant PWM throttle	61
4-10	Plot of the motor's electrical power consumption P as a function of the pressure head H at 3 voltage levels and a constant PWM throttle	62
4-11	Plot of the pump's overall efficiency η_p as a function of the pressure head H at 3 voltage levels and a constant PWM throttle	62
5-1	A circuit diagram with a constant load connected directly to a photovoltaic panel	63
5-2	A circuit diagram with a constant load, a photovoltaic panel, and an MPPT converter	64

5-3	Graph comparing (i, v) operating points of the direct load line and the maximum power point tracking approaches. The operating points (i_1, v_1) and (i_2, v_2) correspond to the circuits in Figure 5-1 and Figure 5-2, respectively.	65
5-4	Plot of output power of JA Solar's 320 W photovoltaic panel as a function of panel voltage at $G = 1000 \text{ W/m}^2$, for five different cell temperatures	66
6-1	Definition of the power source's output current $i(t)$ and terminal voltage $v(t)$ for (a) direct load line and (b) maximum power point tracking approaches	69
6-2	Diagram showing power and signal flows between components in Khethworks irrigation system with direct load line connection	70
6-3	Diagram showing power and signal flows between some components in Khethworks irrigation system with maximum power point tracking	71
6-4	Circuit diagrams of the irrigation system with the water pump switched off, for (a) direct load line and (b) maximum power point tracking approaches	72
6-5	Graphical interpretation of the system's operating point for the direct load line approach with the pump switched off	73
6-6	Graphical interpretation of the system's operating point for the maximum power point tracking approach with the pump switched off	74
6-7	Graphical interpretation of the system's operating point for the direct load line approach with the pump switched off, showing the special case where the electrical power drawn from the photovoltaic panel is less than its maximum output capability	74
6-8	Circuit diagrams of the irrigation system with the water pump switched on, for (a) direct load line and (b) maximum power point tracking approaches	75
6-9	Diagram showing the approximation scheme for the pump's motor state	76

6-10	Graphical interpretation of the system's operating point for the direct load line approach with the pump switched on	77
6-11	Graphical interpretation of the system's operating point for the maximum power point tracking approach with the pump switched on . . .	78
6-12	Graphical interpretation of the system's operating point with the pump switched on, showing the change in voltage level v as a result of changing irradiation level G	79
7-1	Overview of the system-level theoretical simulation	82
7-2	Default value of the water demand throughout the year	86
7-3	Detailed overview of the system-level theoretical simulation at a time instance t	89
7-4	Plot of electrical energy generated in one day of operation by the irrigation system, for three different photovoltaic panel's power ratings	94
7-5	Plot of water delivered in one day of operation by the irrigation system, for three different photovoltaic panel's power ratings, with the assumption that the pump is switched off when the total amount of water delivered reaches the target value	95
7-6	Plot of water delivered in one day of operation by the irrigation system, for three different photovoltaic panel's power ratings	96
7-7	Plot of electrical energy generated and water delivered in one day of operation by the irrigation system, for three different MPPT converter's efficiency	98
7-8	Plot of water delivered in one day of operation by the irrigation system, for the systems with one battery pack and four battery packs	100
7-9	Plot of electrical energy generated and water delivered in one day of operation by the irrigation system in February 2010, for three different locations	102

7-10 Plot of electrical energy generated and water delivered in one day of operation by the irrigation system in May 2010, for three different locations 103

7-11 Plot of electrical energy generated and water delivered in one day of operation by the irrigation system in the month of May, for three years 104

List of Tables

2.1	Comparison between commercial pump models and Khethworks pump [1]	32
3.1	Calculation of R_G at different values of cell temperature for Tata Solar Power 260W multi-crystalline photovoltaic panel	48
7.1	Locations of the irrigation system, for the theoretical system-level simulation	83
7.2	Default values of the system-level simulation's input variables	90
7.3	Solar energy harvested (in Watt-hour) by the irrigation systems implementing the direct load line and the maximum power power tracking approaches, on selected days in May 2010	91
7.4	Water delivered (in m^3) by the irrigation systems implementing the direct load line and the maximum power power tracking approaches, on selected days in November 2010	93

List of Code Listings

A.1	Source file of the system-level simulation	111
A.2	Example of the system's performance plotting script	113
A.3	Simulator for an irrigation system with the direct load line approach and 260-W photovoltaic panel	116
A.4	Simulator for an irrigation system with the maximum power point tracking approach and 260-W photovoltaic panel	120
A.5	<code>findOP_DLL1.m</code>	125
A.6	<code>findOP_DLL2.m</code>	127
A.7	<code>findOP_DLL3.m</code>	127
A.8	<code>waterDemand.m</code>	128
A.9	<code>pvCurve.m</code>	128
A.10	<code>pumpCurve.m</code>	139

Chapter 1

Introduction

Irrigation is critical to the productivity of farmers in India, especially those who own only small pieces of land. A common practice among these farmers is to cultivate only during the monsoon season, when the rainwater is plentiful. Many villages in East India have access to shallow groundwater, but individual farmers cannot afford water pumps and the associated fuel cost. Moreover, the power grid does not provide reliable electricity to the farming areas [1]. Recognizing this challenge, a group of MIT graduates, with support from the MIT Tata Center for Technology and Design, founded a start-up called Khethworks to provide affordable irrigation solution to the marginal farmers in East India.

For this specific irrigation system, the cost of production is a major design constraint. As of 2015, the commercial pumps available in the India market were not optimized for irrigating small plots of land. Therefore, Khethworks developed its own pump prototype. The pump is efficient and small enough to be powered by a photovoltaic panel, thus saving the fuel or electricity cost. Because the size and power rating of the Khethworks pump are smaller than those of other commercial models, the entire irrigation system can be manufactured at a cost that the marginal farmers find affordable.

The Khethworks irrigation system consists primarily of a water pump, a photovoltaic panel, and a battery. The photovoltaic panel harvests electrical energy from

the solar irradiation and powers the water pump. The battery balances the mismatch between the power generated by the photovoltaic panel and the power consumed by the pump. The latest version of the Khethworks system, however, has not been optimized for electrical performance. The design team decided to minimize electronics cost by connecting the pump and the battery directly to the photovoltaic panel. This irrigation system has been proved functional in the field trial and is now under a pilot run in Jharkhand, India.

1.1 Motivation and Objectives

Given large variations in atmospheric temperature and solar irradiation level over the operation hours, the direct load line (DLL) approach may become very ineffective at drawing electrical energy from the photovoltaic panel, especially compared to the maximum power point tracking (MPPT) approach. Over the lifetime of the system, the increase in the solar energy harvested and the water delivered may outweigh the increase in electronics cost. If we were to keep the target water output at a constant level, implementing MPPT might also decrease the overall system cost due to the reduction in photovoltaic panel and battery sizing. These possibilities warrant a thorough electrical analysis of the system to study the benefits of MPPT implementation on the performance of solar-powered irrigation system. In Khethworks case, the study may lead to hardware improvement in its next development cycle.

This thesis presents an analysis of the solar-powered irrigation system - at the component level as well as the system level - and a theoretical simulation of the irrigation system's operation. The simulation predicts the benefits of the maximum power point tracking on the overall system's performance for many possible system configurations. Based on the simulation result, the power converter topologies suitable for the MPPT implementation are recommended to Khethworks.

1.2 Organization

This thesis is organized into the following chapters:

- Chapter 2 gives the background information on the irrigation challenges facing the marginal farmers in India and introduces the Khethworks irrigation solution, which aims to tackle this problem. Concerns about the electrical performance of the Khethworks system and its photovoltaic panel's utilization are brought up for further discussion.
- Chapter 3 analyzes the characteristics of the electrical components in the irrigation system. By making necessary assumption, the approximated quantitative model that represents each component's electrical characteristic can be derived.
- Chapter 4 characterizes the electrical and mechanical properties of the water pump in a manner similar to Chapter 3. The analysis of components that the pump system comprises, including the brushless DC motor and the impeller, leads to the connection between the pump's electrical power input and mechanical power output. Data from the pump's test runs are also presented.
- Chapter 5 details the maximum power point tracking and its effect on the photovoltaic panel's operation. A new characteristic equation representing the photovoltaic panel with the MPPT interface is derived for simulation purpose.
- Chapter 6 defines the system's operating point and other related parameters, then discusses the approximation scheme that the system-level simulation will utilize to calculate the operating point's numerical value. Graphical interpretation of the system's operating point demonstrates how the system's operation responses to the changes in its input parameters.
- Chapter 7 thoroughly covers the theoretical system-level simulation, from listing the input variables that may affect the irrigation system's performance to explaining the simulation process. At the end of this chapter, the simulation

result is analyzed and the findings on when the MPPT does and does not improve the system's performance are summarized.

- Chapter 8 identifies circumstances under which the direct load line and the maximum power point tracking approaches are preferable, and recommends the steps toward implementing the MPPT to Khethworks.

Chapter 2

Background

2.1 Marginal Farmers in East India

The small farm is the most prevalent agricultural setting in many regions of India. As per the agriculture census in 2011, the average size of holdings was 1.15 hectare. Defining marginal farms as those with less than 1.00 hectare, it turned out that as many as 67% of operational farmland holdings in India belonged to this category. East India and eastern Uttar Pradesh together have more than 31 million marginal farms [2]. These farms serve as a primary source of income for many families. They are likely to be located in rural areas with underdeveloped infrastructure and low average income. Many marginal farmers therefore cannot afford existing irrigation solutions for their farmland, despite the presence of groundwater. They often rely on monsoon and seasonal rain to cultivate their lands.

The research from MIT Department of Mechanical Engineering and its spin-off Khethworks both focus on the eastern states of India. The region was chosen as a starting point for the development of affordable irrigation solution for several reasons. First, low average income and dependence on farming indicate a strong need for irrigation solution, as well as presenting cost constraint on the final product. Figure 2-1 illustrates the prevalence of electric water pumps in Central and South India, and the lack thereof in East India. An effective irrigation solution has a potential to greatly

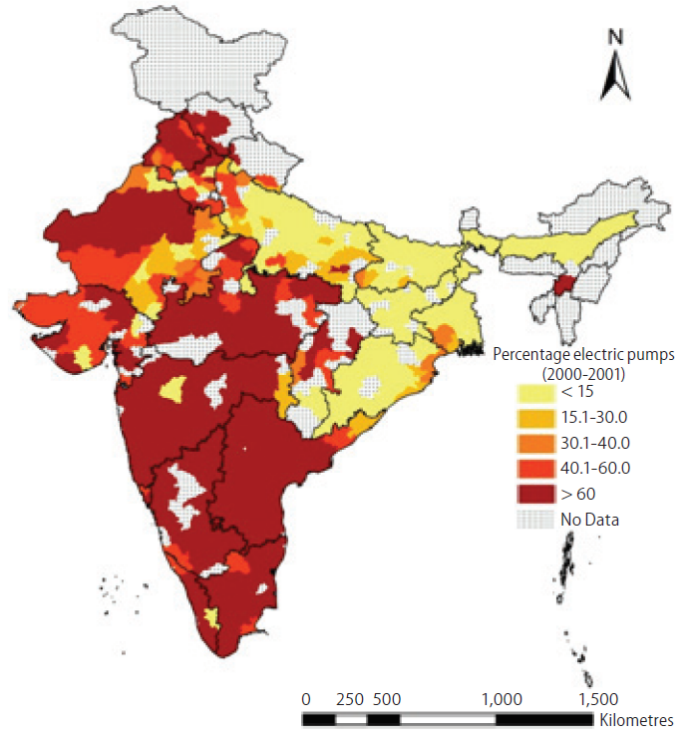


Figure 2-1: Distribution of electric water pumps in India, 2000-01 [3]

increase productivity of the farmlands and income of the farmers. Second, groundwater levels in the states of Uttar Pradesh and Jharkhand are relatively shallower than in other parts of India, as shown in Figure 2-2. While the depth to water level in Southern states can vary from 2 meters near the coast to 20 meters for hilly inland areas, Jharkhand state in particular has very stable water levels of 5-10 meters. Such shallow water level allows the use of small and cheaper pumps. Given the narrow ranges of depth to water level that the pump has to accommodate, the pump can be optimized for higher efficiency. Lastly, East India receives sufficient amounts of solar energy to provide energy for pumping of water. With more than 5.0 kWh per square meter per day in the region, solar irradiation opens up the possibility of powering the pump with photovoltaic panel. This option will help marginal farmers save on the fuel cost of diesel pumps and avoid the trouble of having unreliable access to the electrical grid.

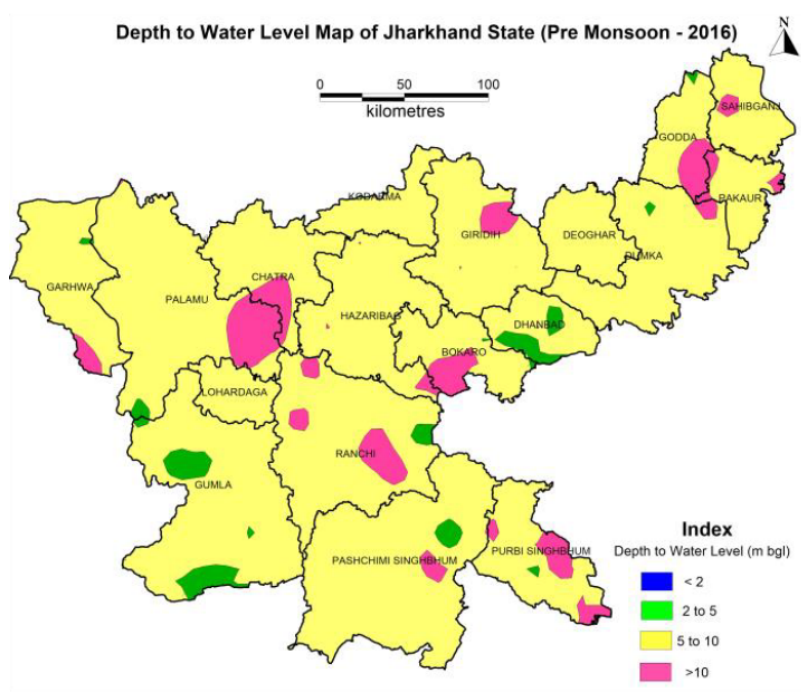
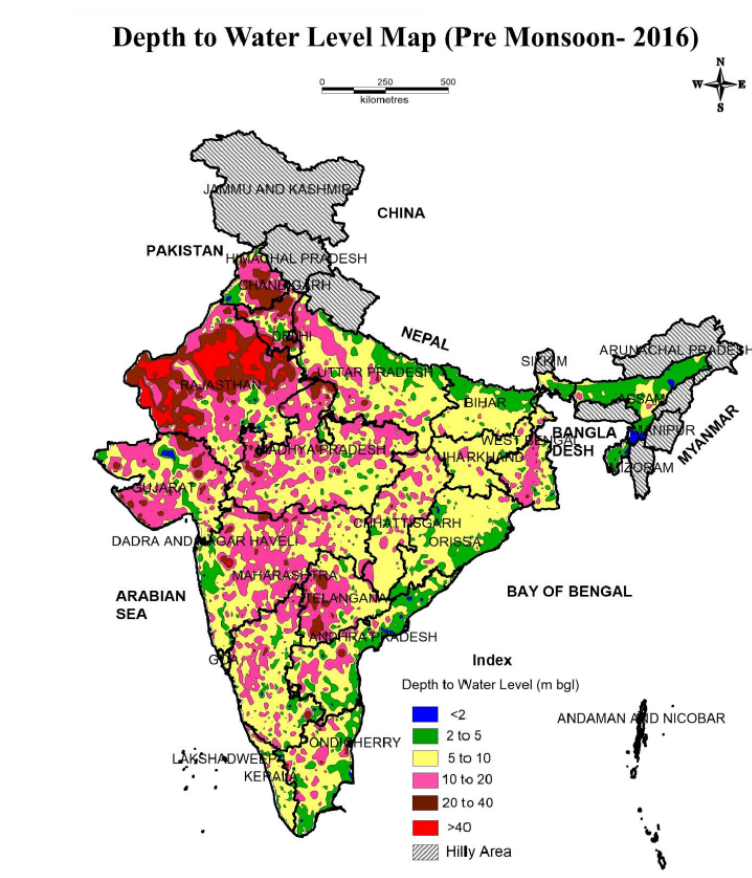


Figure 2-2: Depth to water level during pre-monsoon season in India and Jharkhand state, 2016 [4]

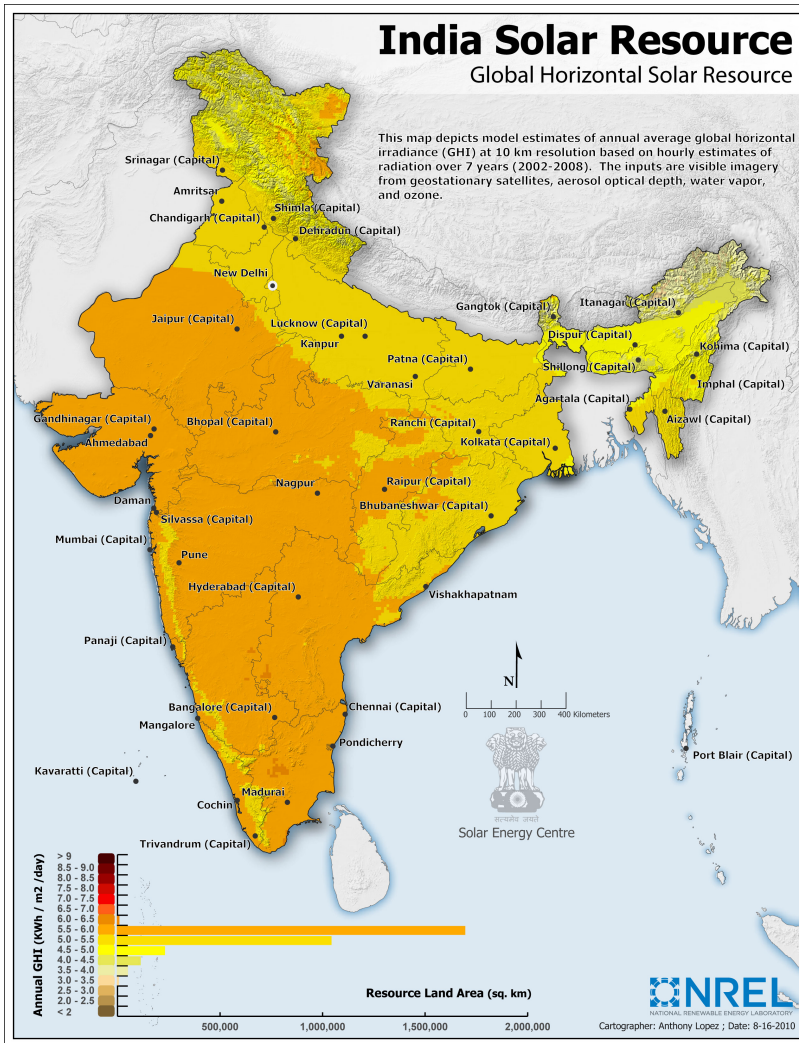


Figure 2-3: Annual average of global horizontal irradiance in India, based on hourly estimation from 2002 to 2008 [5]

2.2 Conventional Irrigation Solution

Diesel pumps and ac electrical pumps are the most prevalent irrigation solution in India. These pumps may answer the irrigation demand under certain circumstances, but when cost is strictly limited, both type of pumps fall short of being an affordable solution for marginal farmers.

2.2.1 Diesel Pumps

In India, diesel is a common energy source not only for transportation but also for areas with limited access to the electrical grid. Small diesel pumps costs somewhere between 6,000 INR to 40,000 INR and often lasts between 3 to 6 years before a maintenance or replacement is required [6]. Two main challenges of using diesel pumps to irrigate small farms are fuel cost and pump efficiency. The price of diesel fuel has increased faster than that of agricultural produce in the past decade, meaning that the farmer's adjusted income has decreased over time. Farmers in remote locations, where the demand of diesel fuel for irrigation is greater than in urban areas with reliable access to electrical grid, also face difficulty purchasing the fuel. They often need to travel about an hour to nearby town to fill bottles by themselves, or purchase bottles of diesel at an increased price per volume from some vendors in their villages. Such inconvenience increases the cost, or opportunity cost, of running diesel pumps even further. Moreover, diesel pumps designed for usage in other regions like Punjab and Maharashtra also runs less efficiently in East India because the pumps were optimized for deeper water levels. A pump suitable for usage in East India should be optimized for shallow groundwater.

2.2.2 Electric Pumps

AC electrical pumps are very popular in western and southern parts of India, largely due to the subsidized electricity cost [7]. Since the times of the Green Revolution - a period starting in the 1960s when improvement in agricultural technology led

to an increase in food production - these regions saw steep increase in agricultural productivity. Nevertheless, irrigation with electrical pumps faces several challenges in East India. The first challenge is that of electricity access. As shown earlier in Figure 2-1, the percentage of electrical pumps in East India has historically been very low compared to other regions, despite the available groundwater, because the area did not have extensive coverage of the electrical grid. Moreover, areas in East India with access to the electrical grid often face electricity reliability problems. During the peak of dry season, when electricity demand for irrigation is high, outage rate significantly increases. Because electrical issues such as transformer failure may take up to 10 days to repair, farmers have to rent diesel pumps in the meantime and their irrigation cost increases even further. Therefore, for a region without extensive and reliable power grid, ac electrical pump is not yet a cost-effective irrigation solution.

2.3 Khetworks

Khetworks is an MIT spinout that has set out to provide an affordable irrigation solution to marginal farmers, focusing on East India. Despite the lack of extensive and reliable power grid, the region receives plenty of solar irradiation. Solar power is a very viable alternative source for small water pumps. A single 1.0 m x 1.5 m solar panel can supply up to 300 Watts, which might be too little for existing commercial pumps but is sufficient for a well-optimized custom-designed pump like Khetworks'. In the dry season, when irrigation demand is at its highest, the solar irradiation is also at its peak, thus eliminating the power shortage problem that presently troubles grid-connected ac electrical pumps.

2.3.1 Pump Design for Small Farms

The latest Khetworks pump system consists of a photovoltaic panel, a battery, and a custom-designed motor. The photovoltaic panels supplies electrical power to the rest of the system. The motor converts electrical power into mechanical power that lifts water against a certain pressure head. The battery acts as a buffer between

those power source and sink. It can absorb excess electrical power generated by the photovoltaic panel and supply the stored energy to the motor when necessary. What we have been referring to as “the motor” so far is in fact a complex system of its own. The “motor” actually consists of three subsystems - a DC motor, an impeller, and a volute. The impeller transforms torque and angular speed, supplied by the DC motor, into the pressure and flow of the water. The volute of a centrifugal pump encases the impeller and converts kinetic energy into pressure by reducing flow rate and increasing pressure.

With the custom design approach, Khethworks’ pump has been optimized for lower flow rate and the lowest cost. Team members conducted survey in Jharkhand state with an India-based NGO Professional Assistance For Development action (Pradan) to estimate daily water need and depth of groundwater in different times of the year. Prof. Amos Winter and his students, some of which later became the founders of Khethworks, set the pump’s operating point at 50 liters per minute at 1 bar of pressure head - equivalent to a 10-meter deep well. They initially aimed for 35% efficiency and were able to achieve a peak efficiency of 29%.

As shown in Table 2.1, the Khethworks prototype as of 2015 is significantly smaller than other commercial electric pumps and diesel pumps. Other pump modules, designed for large farms and communal use, are rated for 1 or 2 horse-power (0.75 or 1.5 kW). The Khethworks module, intended for individual farms and relatively shallow water level of 5-10 meters, is rated for only one third horse power ($\tilde{250}$ W). Because of its small size, the capital cost of the entire Khethworks system is less than those of other DC pump modules by at least five times. The diesel pump may have low initial cost, but the fuel cost makes it the most expensive choice among the four modules in the long run.

2.3.2 Electrical Inefficiency

In order to reduce electronics cost and complexity, Khethworks’ original electronics design did not include MPPT circuit. The motor’s load line directly determines the voltage and current of the system’s operating point. The pump system still operates

well, but in a rather nominal manner. If the system detected that it is connected to photovoltaic panel larger than a certain power rating, it will not run. Because of this design decision, the pump system is compatible with only a small range of photovoltaic panels. Such inflexibility also makes it difficult to scale up the size in the future, unless another motor is designed and optimized to match the exact power rating. Moreover, the latest version of the control circuit has yet to implement battery management, leaving the battery exposed to several risks. Overcharging, overdraining, and improper charging rate can lead to deterioration and decay in battery capacity. Since the battery contributes to a significant portion of overall cost, a battery management system that extends battery life and saves replacement cost is highly desirable.

Model	Rotomag RS1200 Submersible [8]	Rotomag MBP60 Surface [9]	Honda WX10 [10]	Khethworks ver.2015
Pump Type	Solar DC Centrifugal	Solar DC Centrifugal	Diesel Centrifugal	Solar DC Centrifugal
Horse Power	1	2	1	1/3
PV Panel [Watts]	1,200	1,800	-	300
Max Total Head [m]	30	15	35	18
Flow Rate [litre/hour]	12,000	24,000	8,400	3,600
Pump Price [INR]	75,000	43,000	29,000	9,000
Panel Price [INR]	76,000	114,000	-	19,000
Capital Cost [INR]	151,000	157,000	29,000	28,000
Fuel Cost [INR/year]	-	-	30,000	-
Lifetime [year]	10	10	7	10
Ownership [INR/year]	15,100	15,700	36,000	2,800

Table 2.1: Comparison between commercial pump models and Khethworks pump [1]

Chapter 3

Electrical Components

To achieve an accurate system-level simulation and good understanding of the system characteristics, characteristics of individual components within the system must be quantitatively understood. In this chapter, we study and characterize electrical properties of the photovoltaic panel and battery. While the characterization is intended to be as accurate as possible, a number of approximations and assumptions are also made to reduce the complexity and made the numerical calculation possible.

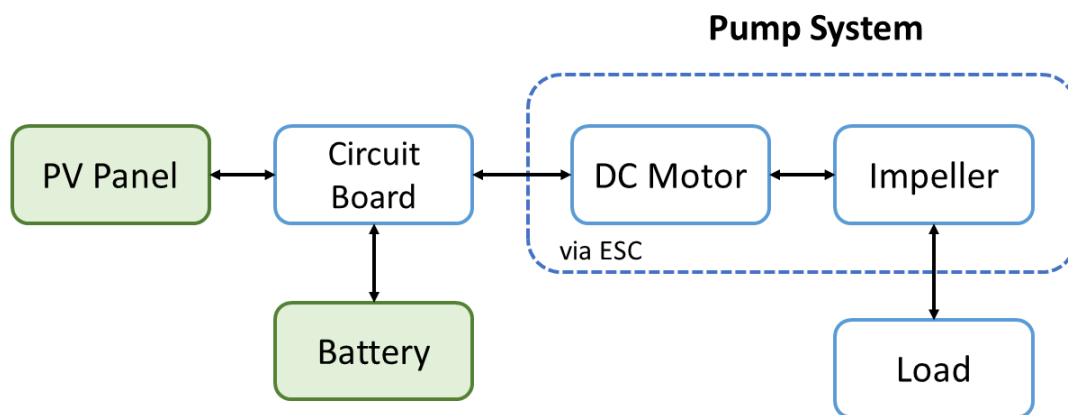


Figure 3-1: Diagram highlighting electrical components in the Khethworks irrigation system.

3.1 Photovoltaic Panel

The photovoltaic panel converts light into electrical power using certain properties of semiconductor materials. It acts as the main source of electrical power for the pump system. The amount of electrical power generated by the panel strongly depends on irradiation level of incident light on the panel's surface. Simply speaking, the stronger the irradiation, the greater the available electrical power output. The amount of power generated also depends on the photovoltaic cell's temperature and a number of intrinsic properties of the cells.

The current version of Khethworks pump system prefers a 60-cell 260-Watt photovoltaic panel, which is available locally in India from many manufacturers. A panel with power rating too large will overflow the battery and waste the excess energy. On the other hand, a panel with too small a power rating can not supply energy at a sufficient rate to the pump's motor. In this study, we will consider only photovoltaic panels with power rating between 180 Watts to 320 Watts, with greatest focus on models with 260 Watts rating. Approximation in this section, as well as the following sections, will be made based on general characteristics of these panel models to simplify characteristic equations of panel parameters.

3.1.1 Circuit Model

A photovoltaic panel is made up of an array of photovoltaic cells and can be modeled with the circuit in Figure 3-2. The photogenerated current is represented by the current source I_L . Its magnitude is a function of incident irradiation level. The N diodes connected in series represents N p-n junctions from each individual photovoltaic cell that makes up the N-cell panel. R_{SH} and R_S are the equivalent shunt and series resistance of the panel, respectively.

From the circuit above, we can write the output current i as a function of the panel voltage v :

$$i = I_L - I_0 \left\{ \exp \left[\frac{q}{nkT_c} \cdot \frac{v + iR_S}{N} \right] - 1 \right\} - \frac{1}{R_{SH}}(v + iR_S) \quad (3.1)$$

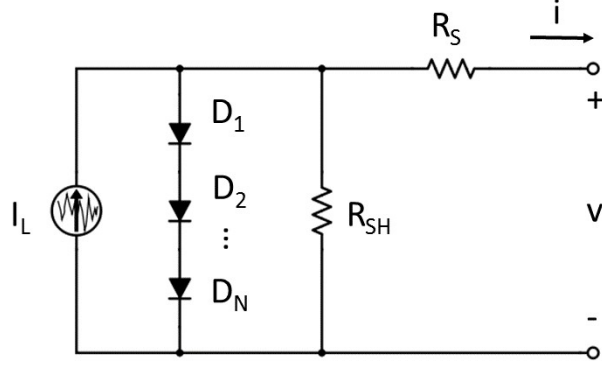


Figure 3-2: An equivalent circuit of a photovoltaic panel

In case of an ideal panel, we may approximate that the shunt resistance is indefinitely large and the series resistance is indefinitely small. With $R_S \rightarrow 0$ and $R_{SH} \rightarrow \infty$, equation (3.1) becomes

$$i = I_L - I_0 \left\{ \exp \left[\frac{qv}{nkT_c N} \right] - 1 \right\} \quad (3.2)$$

However, for the majority of photovoltaic panels of our interest, only the approximation on R_{SH} is valid. The series resistance R_S is non-negligible. Thus, the $i - v$ characteristic equation of the panel is approximately

$$i = I_L - I_0 \left\{ \exp \left[\frac{q}{nkT_c} \cdot \frac{v + iR_S}{N} \right] - 1 \right\} \quad (3.3)$$

With regards to the photovoltaic panel, our goal is to create a model or characteristic equation that can map panel voltage v to output current i for any given system conditions. Therefore, the equation (3.3) by itself does not sufficiently characterize the panel, as parameters in the equation can not be easily measured or obtained from the datasheet. For example, it is very difficult to directly measure the photogenerated current I_L and the cell temperature T_c at a given moment. Extra work is then required to express parameters in equation (3.3) in the form of easily measurable variables and

parameters available in datasheet from manufacturers.

Parameters in the datasheet are often given at certain reference conditions, such that one can easily compare performance of different photovoltaic panels. Standard Test Condition (STC), often used in laboratory setting, specifies irradiation level $G_{STC} = 1000 \text{ W/m}^2$, cell temperature $T_{c,STC} = 25 \text{ }^\circ\text{C}$, and irradiation spectrum of AM 1.5 .

The Standard Test Condition, however, does not reflect actual conditions that photovoltaic panels experience in the fields. For example, irradiation level G tends to be lower than 1000 W/m^2 . We also do not have control over cell temperature T_c but more control of the ambient temperature T_a . Therefore, manufacturers sometimes include a more practical test condition called Nominal Operating Cell Temperature (NOCT). NOCT includes more parameters significant to practical setting of the photovoltaic cell environment. It specifies irradiation level $G_{NOCT} = 800 \text{ W/m}^2$, ambient temperature $T_{a,NOCT} = 20 \text{ }^\circ\text{C}$, wind speed near the panel $v_{w,NOCT} = 1 \text{ m/s}$, irradiation spectrum of AM 1.5, and that the panel has no load.

3.1.2 Cell Temperature T_c

Cell temperature affects many properties of a photovoltaic cell. Many models in the literature attempts to estimate the cell temperature of photovoltaic cell from other easily measurable system parameters. Parameters which have significant effect on cell temperature include the atmosphere's ambient temperature T_a , irradiation level G , and wind speed v_w .

Standard model gives a simple estimation of cell temperature as a function of ambient temperature T_a and irradiation level G [11].

$$T_c = T_a + \frac{G}{G_{NOCT}} (T_{c,NOCT} - T_{a,NOCT}) \quad (3.4)$$

In the field, however, it is often observed that the wind speed has considerable effect on the panel temperature. The stronger the wind speed, the greater the heat convection near the panel surface. To derive a model of cell temperature that takes wind speed into account, we begin by balancing the input and output powers of a

photovoltaic cell in steady-state.

$$[\text{Absorbed Solar Power}] = [\text{Electric Power Out}] + [\text{Power Dissipated as Heat}] \quad (3.5)$$

Solar power absorbed by photovoltaic panel is directly proportional to the amount of incident irradiation on the panel. We assume that the coefficient of proportionality α is independent of temperature. Output electric power of the panel is also approximated to be directly proportional to the irradiation level, as demonstrated later in Section 3.1.3. However, the coefficient η is temperature dependent. Since the early years of photovoltaic research, a simple model for this output power coefficient η is [12]

$$\eta = \eta_{STC} [1 - \beta_{STC}(T_c - T_{STC})] \quad (3.6)$$

Dissipated power follows Newton's law of cooling. The heat loss rate of the panel is proportional to the difference between the cell temperature and the ambient temperature. The proportionality constant h is independent of T_c , but may be a function of other system parameters such as wind speed and humidity.

We can then formulate equation (3.5) as

$$\alpha G = \eta G + h \cdot (T_c - T_a) \quad (3.7)$$

$$T_c = T_a + \frac{\alpha}{h} \cdot G \cdot \left[1 - \frac{\eta}{\alpha}\right] \quad (3.8)$$

From equation (3.8), we would like to solve for T_c in terms of datasheet parameters and constants at STC and NOCT, such that T_c at any moment can be calculated from known input data. Proportionality coefficients α and η must be replaced with terms referring to standard reference condition such as η_{STC} and β_{STC}

At NOCT, the panel is required to have no load. Because the electric power output must be zero, we can write equation (3.7) for a photovoltaic panel at NOCT as

$$0 = \alpha G_{NOCT} - h_{NOCT}(T_{c,NOCT} - T_{a,NOCT}) \quad (3.9)$$

$$\alpha = \frac{h_{NOCT}}{G_{NOCT}}(T_{c,NOCT} - T_{a,NOCT}) \quad (3.10)$$

Substitute equation (3.10) and equation (3.6) into equation (3.8) then solve for T_c ,

$$T_c = T_a + \frac{G \cdot h_{NOCT}}{G_{NOCT} \cdot h}(T_{c,NOCT} - T_{a,NOCT}) \left[1 - \frac{\eta_{STC}}{\alpha} (1 - \beta(T_c - T_{STC})) \right] \quad (3.11)$$

$$\therefore T_c = \frac{T_a + \frac{G \cdot h_{NOCT}}{G_{NOCT} \cdot h} \cdot (T_{c,NOCT} - T_{a,NOCT}) \left[1 - \frac{\eta_{STC}}{\alpha} (1 + \beta_{STC} T_{STC}) \right]}{1 - \frac{\beta_{STC} \eta_{STC}}{\alpha} \cdot \frac{G \cdot h_{NOCT}}{G_{NOCT} \cdot h} (T_{NOCT} - T_{a,NOCT})} \quad (3.12)$$

In equation (3.12), the cell temperature T_c still carries the term α . Skoplaki et al. suggested that we may approximate $\alpha \approx 0.9$ and that the proportionality constant of heat loss rate h is a function of only the wind speed, with an empirical linear form of $h = 5.7 + 3.8v_w$ [13].

With these assumptions, we can demonstrate the second term in the denominator of equation (3.12) is negligible. For most photovoltaic panels under consideration, β_{STC} is near -0.5%/K and η_{STC} is near 15%. At NOCT where the irradiation level $G = G_{STC} = 1000 \text{ W/m}^2$ and the wind speed $v = v_{w,NOCT} = 1 \text{ m/s}$, we have

$$\frac{\beta_{STC} \eta_{STC}}{\alpha} \cdot \frac{G \cdot h_{NOCT}}{G_{NOCT} \cdot h} (T_{NOCT} - T_{a,NOCT}) = 0.19 \ll 1 \quad (3.13)$$

Thus, we may further approximate that

$$T_c \approx T_a + \frac{G \cdot h_{NOCT}}{G_{NOCT} \cdot h} \cdot (T_{c,NOCT} - T_{a,NOCT}) \left[1 - \frac{\eta_{STC}}{\alpha} (1 + \beta_{STC} T_{STC}) \right] \quad (3.14)$$

Figure 3-3 shows an example of T_c simulation over a short period of time for a city in Jharkhand state in 2011, using T_a , G , and v_w data from National Renewable Energy Laboratory (NREL) database. We note that, for this model, the cell temperature

T_c is always greater than or equal to the ambient temperature T_a . T_c increases the most during daytime, when irradiation level is high. The effect of wind speed on cell temperature can be observed from the daily peaks. On the second day, when the wind was strong, the peak cell temperature is more than 10 °C lower than that of the first day.

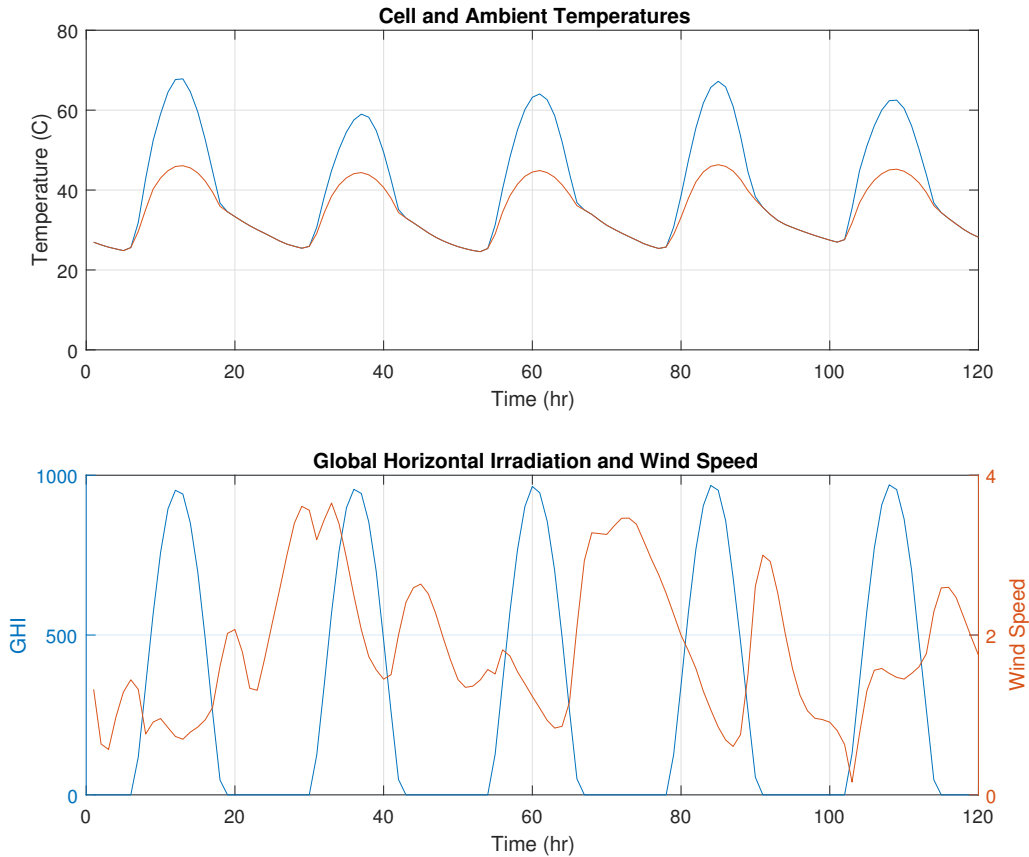


Figure 3-3: Plot of the photovoltaic cell temperature as a function of time - with ambient temperature, global horizontal irradiation, and wind speed as reference.

3.1.3 Maximum Power Point

Because of photovoltaic panel's high cost, an efficient photovoltaic system should harvest as much power as possible from the panel. Due to the current-voltage characteristic of photovoltaic panel, there exists a panel voltage at which maximum power can be drawn from the panel. For example, in Figure 3-4 (b), this voltage

corresponds to the peak of each power vs. voltage characteristic curve. The voltage, and the corresponding current value, is often referred to as the maximum power point (MPP). The maximum power point is a function of many parameters, including panel temperature, irradiation level, as well as the load impedance. Therefore, it is often too complicated to determine an explicit form of maximum power point's current and voltage. To keep the panel voltage near the maximum power point and extract as much electrical power as possible, feedback control and specific tracking algorithms must be implemented. Most Maximum Power Point Tracking (MPPT) algorithms sense the panel voltage and the output current, then actively adjust the power converter to reach, or get as close as possible to, the maximum power point.

Figure 3-4 and Figure 3-5 illustrate the $i - v$ and $p - v$ characteristics of a typical 260-Watt photovoltaic panel over variation in cell temperature and irradiation level. From Figure 3-4 (a), we see that as the cell temperature T_c increases, the short-circuit current I_{SC} also increases, while the open-circuit voltage V_{OC} decreases. Meanwhile, Figure 3-5 (a) shows that both I_{SC} and V_{OC} increase with increasing irradiation level G . Ultimately, we see that the maximum power point is dependent on both T_c and G .

Within the practical operating conditions, where $0^\circ\text{C} < T_c < 75^\circ\text{C}$ and $0 \text{ W/m}^2 < G < 1000 \text{ W/m}^2$, Figure 3-6 shows that the maximum output power is approximately a linear function of the irradiation level, as well as the cell temperature. This empirical model helps us characterize the maximum power obtainable from a photovoltaic panel without knowing the explicit form of its current-voltage characteristics.

Therefore, we characterize output power at the maximum power point $P_{MPP}(T_c)$ and $P_{MPP}(G)$ with the following equations.

$$P_{MPP}(T_c) = P_{MPP}(T_{c0}) \cdot \frac{A_T + B_T T_c}{A_T + B_T T_{c0}} \quad (3.15)$$

$$P_{MPP}(G) = P_{MPP}(G_0) \cdot \frac{A_G + B_G G}{A_G + B_G G_0} \quad (3.16)$$

The constants $A_T, B_T, A_G,$ and B_G vary with the panel model and may even vary slightly between panels of the same photovoltaic model. We may also use any cell

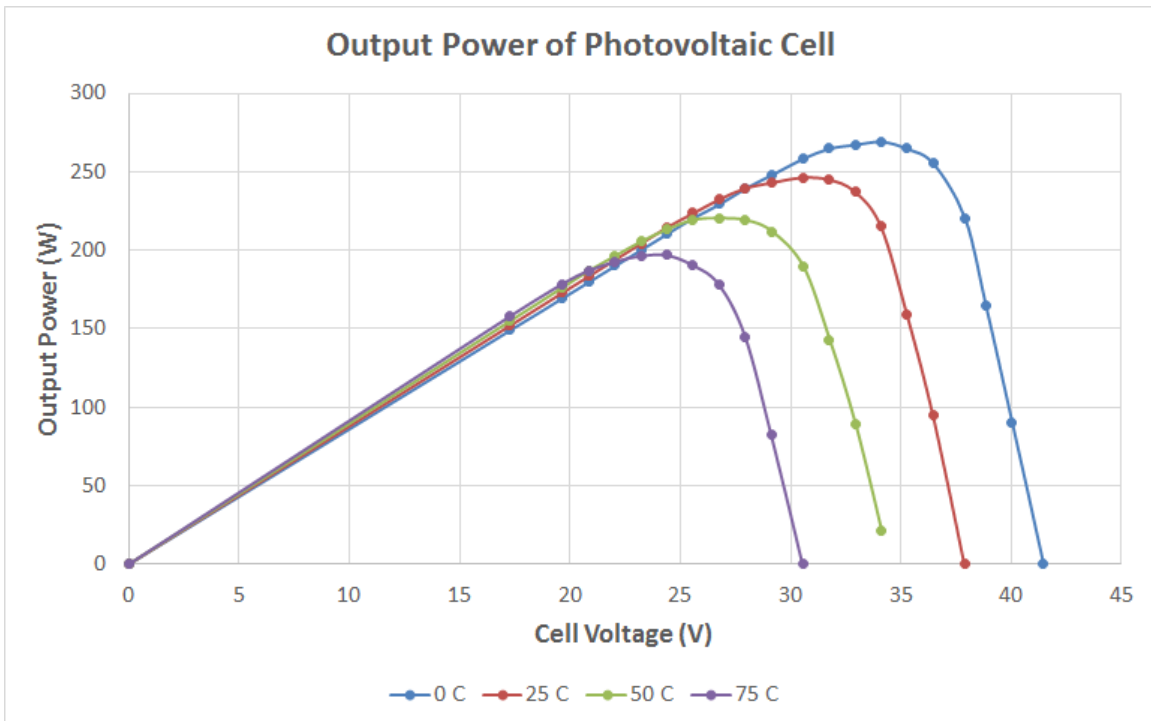
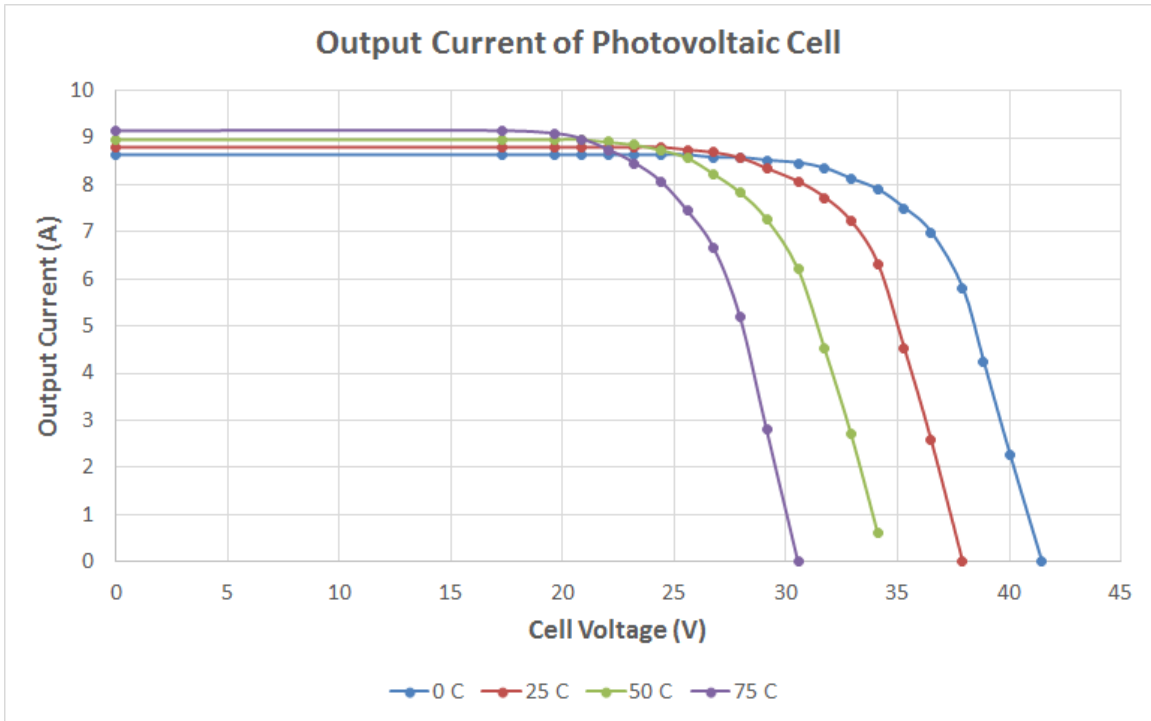


Figure 3-4: Plot of (a) output current and (b) output power of a typical photovoltaic cell as a function of the cell voltage at $G = 1000 \text{ W/m}^2$, for four different cell temperatures

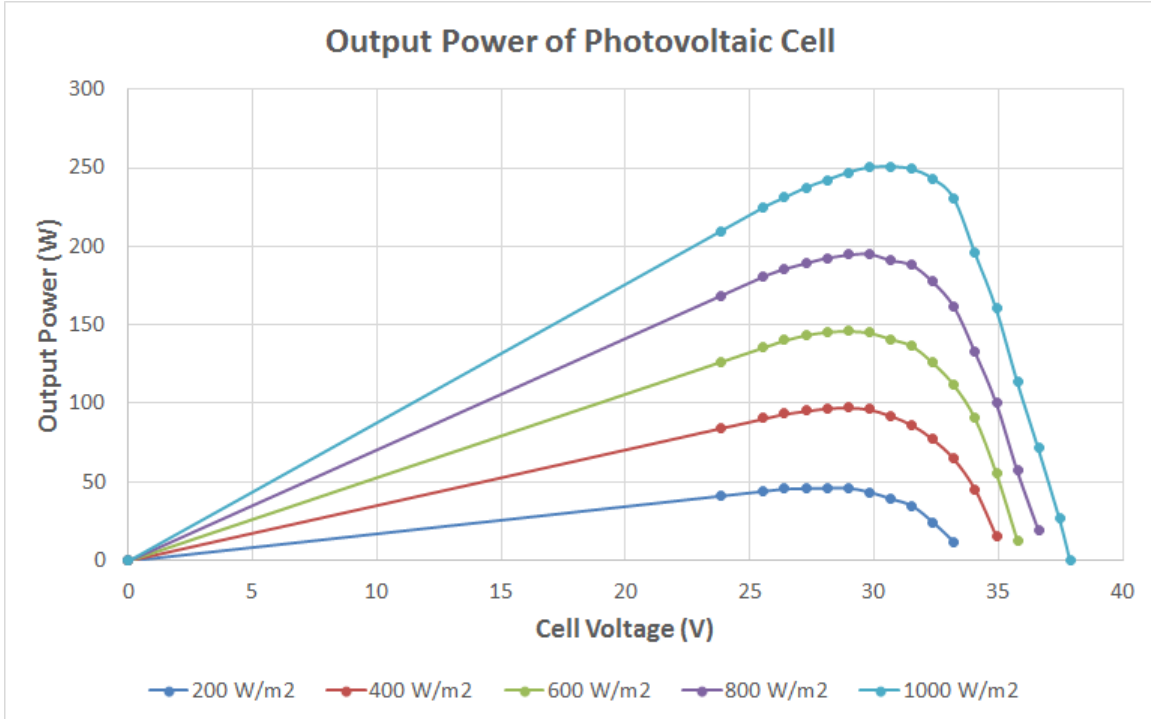
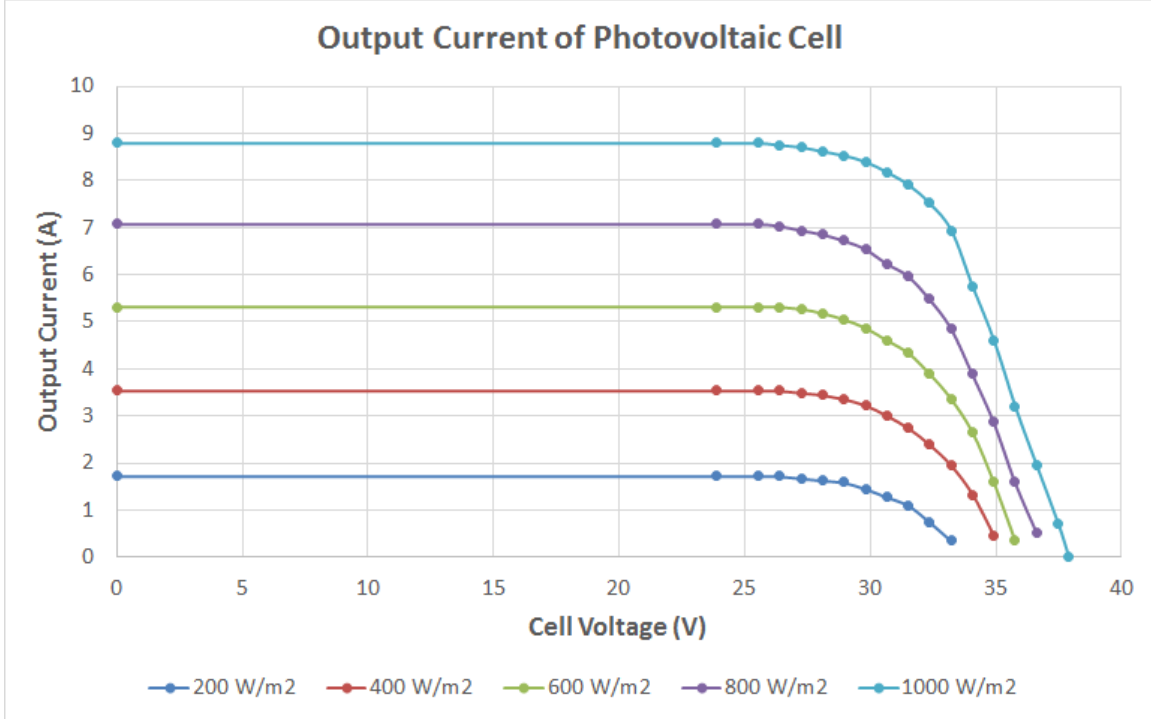


Figure 3-5: Plot of (a) output current and (b) output power of a typical photovoltaic cell as a function of the cell voltage at $T_c = 25^\circ\text{C}$, for five different irradiation levels

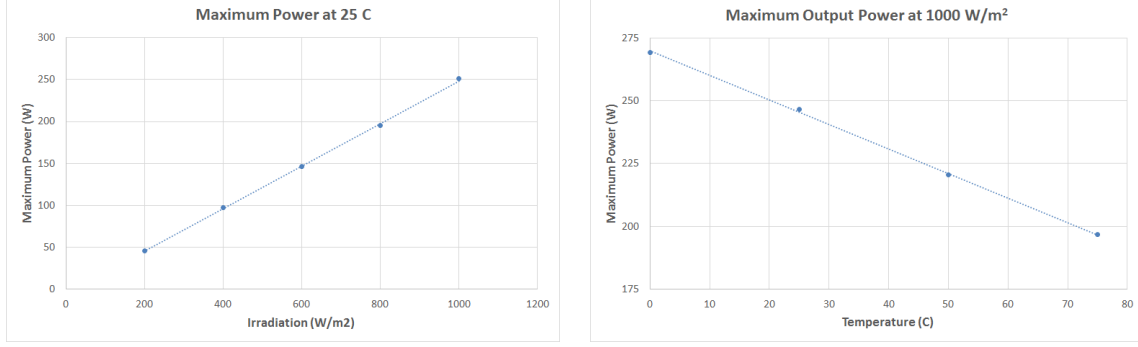


Figure 3-6: Plots showing linear trend of a photovoltaic cell's maximum output power over (a) irradiation and (b) cell temperature

temperature and irradiation level as the reference point (T_{c0}, G_0) given that we have measured the maximum output power P_{MPP} at this point.

Thus, for the simulation purpose, we will model P_{MPP} as

$$P_{MPP}(T_c) = P_{MPP}(T_{c0}) \cdot \frac{A_T + B_T T_c}{A_T + B_T T_{c0}} \cdot \frac{A_G + B_G G}{A_G + B_G G_0} \quad (3.17)$$

For most photovoltaic panels that we are interested in, $A_G \approx 0$ i.e. the maximum power is directly proportional to irradiation level. With this approximation, Equation (3.16) becomes

$$P_{MPP}(G) = P_{MPP}(G_0) \cdot \frac{G}{G_0} \quad (3.18)$$

Equation (3.17) can then be rewritten as

$$P_{MPP}(T_c, G) = P_{MPP}(T_{c0}, G_0) \cdot \frac{G}{G_0} \cdot (1 + \beta_T(T_c - T_{c0})) \quad (3.19)$$

where the coefficients $\beta_T(T_{c0}) = \frac{B_T}{A_T + B_T T_{c0}}$.

With equation (3.19), the maximum output power of a photovoltaic cell can be simply estimated from G and T_c . This equation is very useful in simulating a photovoltaic system with maximum power point tracking. However, an explicit form of $i - v$ equation is still necessary for simulation of a system where the photovoltaic panel is directly connected to the load.

3.1.4 Explicit Form of v for any T_c and G

In this subsection, we derive the explicit form of the $i - v$ equation as a function of system parameters that can be easily measured or obtained from the datasheet, such as I_{SC} , V_{OC} , and R_S .

First, we may safely estimate that the photogenerated current I_L is much larger than the p-n junction's reverse saturation current I_0 . Equation (3.3) becomes

$$i = I_L - I_0 \exp \left[\frac{q}{nkT_c} \cdot \frac{v + iR_S}{N} \right] \quad (3.20)$$

For a photovoltaic panel under open-circuit condition, $v = V_{OC}$ and $i = 0$.

$$0 = I_L - I_0 \exp \left[\frac{qV_{OC}}{nkT_cN} \right] \quad (3.21)$$

$$\therefore I_L = I_0 \exp \left[\frac{qV_{OC}}{nkT_cN} \right] \quad (3.22)$$

Substitute equation (3.22) into equation (3.3),

$$i = I_L - I_L \exp \left[\frac{q}{nkT_cN} \cdot (v + iR_S - V_{OC}) \right] \quad (3.23)$$

$$\frac{I_L - i}{I_L} = \exp \left[\frac{q}{nkT_cN} \cdot (v + iR_S - V_{OC}) \right] \quad (3.24)$$

$$\ln \left(\frac{I_L - i}{I_L} \right) = \frac{q}{nkT_cN} \cdot (v + iR_S - V_{OC}) \quad (3.25)$$

$$\therefore v = V_{OC} - iR_S - \frac{nkT_cN}{q} \ln \left(\frac{I_L}{I_L - i} \right) \quad (3.26)$$

To make equation (3.26) suitable for simulation, we must:

- construct a model of V_{OC} valid over variation in T_c and G
- derive I_L in terms of I_{SC} , V_{OC} , and R_S

- calculate R_S from data available in the datasheet

3.1.4.1 Open-Circuit Voltage V_{OC}

Figure 3-7 suggests a linear trend for the open-circuit voltage V_{OC} over both cell temperature T_c and irradiation level G . Similar to the linear approximation of maximum power output in Section 3.1.3, we characterize the open-circuit voltage V_{OC} with the following equations.

$$V_{OC}(T_c) = V_{OC}(T_{c0}) \cdot \frac{A'_T + B'_T T_c}{A'_T + B'_T T_{c0}} \quad (3.27)$$

$$V_{OC}(G) = V_{OC}(G_0) \cdot \frac{A'_G + B'_G G}{A'_G + B'_G G_0} \quad (3.28)$$

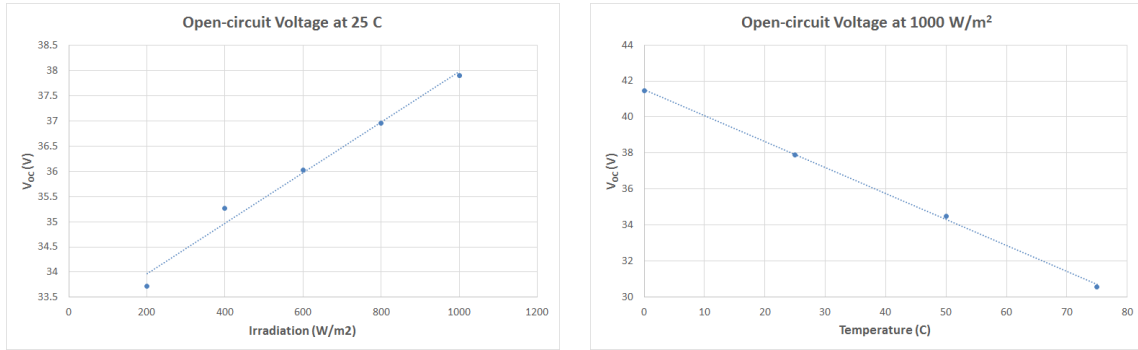


Figure 3-7: Plots showing linear trend of a photovoltaic cell's open-circuit voltage over (a) irradiation and (b) cell temperature

Thus, for the simulation purpose, we will model V_{OC} as

$$V_{OC}(T_c, G) = V_{OC}(T_{c0}, G_0) \cdot \frac{A'_T + B'_T T_c}{A'_T + B'_T T_{c0}} \cdot \frac{A'_G + B'_G G}{A'_G + B'_G G_0} \quad (3.29)$$

Equation (3.29) can be rearranged into

$$V_{OC}(T_c, G) = V_{OC}(T_{c0}, G_0) \cdot (1 + \beta'_T(T_c - T_{c0})) \cdot (1 + \beta'_G(G - G_0)) \quad (3.30)$$

where the coefficients $\beta'_T(T_{c0}) = \frac{B'_T}{A'_T + B'_T T_{c0}}$ and $\beta'_G(G_0) = \frac{B'_G}{A'_G + B'_G G_0}$

3.1.4.2 Photogenerated Current I_L

For a photovoltaic panel under short-circuit condition, we have $v = 0$ and $i = I_{SC}$. Equation (3.3) becomes

$$I_{SC} = I_L - I_0 \left\{ \exp \left[\frac{q}{nkT_c N} \cdot I_{SC} R_S \right] - 1 \right\} \quad (3.31)$$

Meanwhile, the open-circuit condition, where $i = 0$ and $v = V_{OC}$, gives

$$I_L = I_0 \left\{ \exp \left[\frac{q}{nkT_c N} \cdot V_{OC} \right] - 1 \right\} \approx I_0 \exp \left[\frac{q}{nkT_c N} \cdot V_{OC} \right] \quad (3.32)$$

For most photovoltaic panels that we are interested in, the open-circuit voltage $V_{OC} > 30\text{V}$ for $273\text{ K} < T_c < 348\text{ K}$. Assuming that number of cells in the panel $N = 60$, cell temperature $T_c = 298\text{K}$, and ideality factor $n \approx 1$, the first term of I_{SC} in equation (3.31) is

$$\begin{aligned} I_L &\approx I_0 \exp \left[\frac{q}{nkT_c N} \cdot V_{OC} \right] \\ &> I_0 \exp \left[\frac{30\text{V}}{25.7\text{mV} \cdot 60} \right] \\ &= I_0 \exp(19.455) \approx (2.813 \times 10^8) I_0 \end{aligned} \quad (3.33)$$

while the second term is

$$\begin{aligned} I_0 \left\{ \exp \left[\frac{q}{nkT_c N} \cdot I_{SC} R_S \right] - 1 \right\} &> I_0 \left\{ \exp \left[\frac{9\text{A} \cdot 1\Omega}{25.7\text{mV} \cdot 60} \right] - 1 \right\} \\ &= I_0 \{ \exp(5.837) - 1 \} \approx 342 I_0 \end{aligned} \quad (3.34)$$

$$\therefore I_L \gg I_0 \left\{ \exp \left[\frac{q}{nkT_c N} \cdot I_{SC} R_S \right] - 1 \right\} \quad (3.35)$$

Thus, we may safely assume that the photogenerated current $I_L \approx I_{SC}$ for the range of conditions under which our photovoltaic panels will operate.

3.1.4.3 Series Resistance R_S

In this subsection, we derive an estimated form of the series resistance in terms of the photogenerated current I_L and the slope at x-intercept of the $i - v$ characteristic curve - both of which can be obtained from the datasheet.

Consider the $i - v$ curve of a photovoltaic panel in the region near the x-intercept $(i, v) = (0, V_{OC})$, we define another point on the curve $(i', v') = (\delta i, V_{OC} - \delta v)$ as illustrated in Figure 3-8. This point locates very close to the x-intercept, such that $\delta i \ll I_{SC}$ and $\delta v \ll V_{OC}$

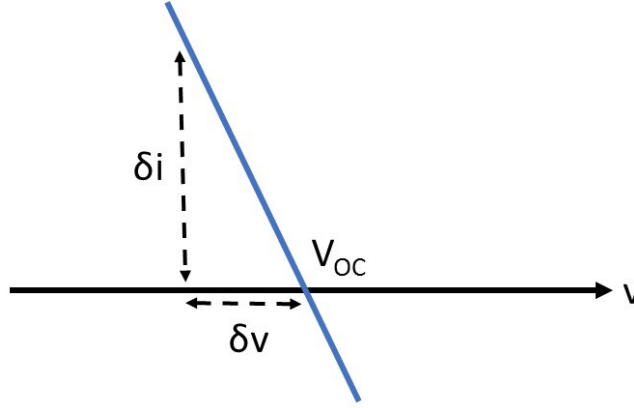


Figure 3-8: $i - v$ curve near the vicinity of $v = V_{OC}$

From equation (3.3), we have

$$\begin{aligned}
 \delta i &= I_L - I_0 \left\{ \exp \left[\frac{q}{nkT_c} \cdot \frac{V_{OC} - \delta v + \delta i R_S}{N} \right] - 1 \right\} \\
 &= I_L - I_0 \exp \left[\frac{qV_{OC}}{nkT_c N} \left(1 - \frac{\delta v}{V_{OC}} + \frac{\delta i R_S}{V_{OC}} \right) \right] - I_0 \\
 &\approx I_L - I_L \exp \left[\frac{qV_{OC}}{nkT_c N} \left(-\frac{\delta v}{V_{OC}} + \frac{\delta i R_S}{V_{OC}} \right) \right] \tag{3.36} \\
 &\approx I_L - I_L \left[1 + \frac{qV_{OC}}{nkT_c N} \left(-\frac{\delta v}{V_{OC}} + \delta i R_S \right) \right] \text{ with Taylor expansion} \\
 &= I_L \cdot \frac{q}{nkT_c N} (\delta v - \delta i R_S)
 \end{aligned}$$

Solve equation (3.36) for δi ,

$$\delta i \approx \left(\frac{\frac{qI_L}{nkTN}}{1 + \frac{qI_L}{nkTN}R_S} \right) \delta v \quad (3.37)$$

By measuring the slope of $i - v$ characteristic curve at $v = V_{OC}$, the panel's series resistance R_S can be calculated from the following equation.

$$|s| = \frac{\delta v}{\delta i} = \frac{\frac{qI_L}{nkTN}}{1 + \frac{qI_L}{nkTN}R_S} \quad (3.38)$$

$$\therefore R_S = \frac{1}{|s|} - \frac{nkTN}{qI_L} \quad (3.39)$$

T_c (K)	$ s $ (Ω^{-1})	I_{SC} (A)	R_S (Ω)
273	1.637	8.63	0.447
298	1.729	8.80	0.403
323	1.642	8.97	0.422
348	1.990	9.14	0.305

Table 3.1: Calculation of R_S at different values of cell temperature for Tata Solar Power 260W multi-crystalline photovoltaic panel

In this study, we approximate that R_S remains constant over variation in cell temperature T_c and irradiation level G . Table 3.1 gives an example of R_S calculation for a model of the photovoltaic panels used by Khethworks.

3.2 Battery

In the Khethworks system, the battery is connected between the power source and the power sink of the pump system, acting as an energy buffer. While we are able to control the pump's rotational speed and electrical power consumption, the amount of electrical power generated by the photovoltaic panel changes with the unpredictable irradiation level. The battery absorbs excess energy from the panel or supplies extra energy to the motor as the power generated by the photovoltaic panel fluctuates

above or below the motor’s power consumption. When the pump is turned off, the photovoltaic panel directly charges the battery.

Most solar installations like off-grid facilities and solar farms use deep-cycle, lead-acid batteries. The cost per capacity of lead-acid batteries are cheaper than that of lithium-ion ones. Deep-cycle capability also allows for greater performance compared to a lithium-ion battery of the same size. However, lead-acid batteries are relatively heavier and require regular maintenance to preserve the battery life. For the flooded-cell type, the electrolytes must be routinely refilled to ensure that the battery’s plates are fully submerged. After consulting practitioners and manufacturers in India, Khethworks decided to forgo this popular option and installed sealed batteries for their irrigation system. Sealed batteries minimize the maintenance required from users, who are mostly farmers, and ensure portability, meeting the design goal that the entire system should be easy to carry from a farmer’s residence to his/her plot of land.

The original analysis was conducted with sealed lead-acid (SLA) battery as the battery of choice. To keep the pump running from 9am to 5pm, it was estimated that the battery must be at least 136 watt-hour in size, corresponding to 5.7 Ah capacity for a 24 V model [14]. However, due to cost and size constraints, the latest version of Khethworks system implements a 25.6 V, 1.4 Ah LiFePO₄. This smaller model is also capable of buffering energy and assisting the motor, but it is not optimal to power to motor directly with only this battery.

If this 25.6 V, 1.4 Ah LiFePO₄ battery acts as the only source of power to a DC motor with 260 W rating, equal to that of the photovoltaic panel, it will last as follows:

$$\frac{25.6\text{V} \cdot 1.4\text{Ah}}{260\text{W}} = 0.138 \text{ hr} \approx 8.3 \text{ min} \quad (3.40)$$

Note that, in practice, any battery should not be discharged down to 0% of its capacity to prevent deterioration. In the system design, we limit discharging to no less than 40% of the fully charge state. Consequently, the time which one LiFePO₄ battery can power the motor is even less than the value calculated in equation (3.40).

The chosen battery model can therefore only serve as an auxiliary source of power. It can aid the photovoltaic panel during transition time when the panel's output power begins dropping below the motor's power consumption. Preventing the motor from sudden change in speed or coming into a sudden stop helps preserve its bearing from corrosion.

A simple Thevenin equivalent circuit model for battery is shown in Figure 3-9. V_{bat} is the actual battery voltage and r_{bat} is the effective series resistance. Here, the current i is defined as current flowing into the battery such that, in charging state, the battery's terminal voltage v will be greater than V_{bat} .

$$v = V_{bat} + i \cdot r_{bat} \quad (3.41)$$

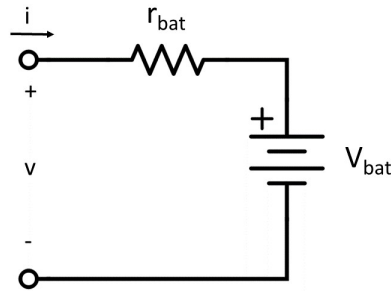


Figure 3-9: A circuit model of a battery

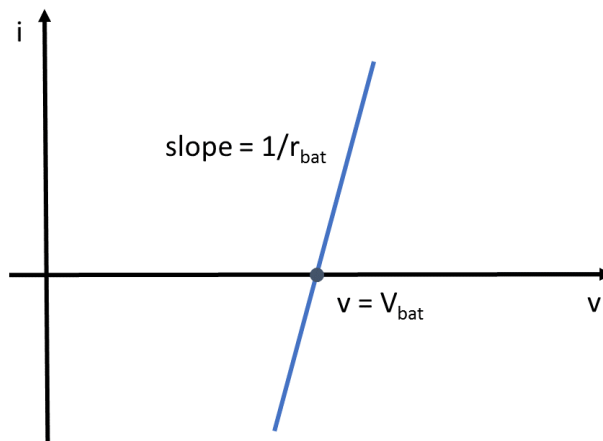


Figure 3-10: An $i - v$ characteristic curve of a battery model in Figure 3-9

The system-level simulation will include only LiFePO_4 batteries, with the number of battery cells in the system as an input variable. Other battery types and chemistries will not be considered. Lithium-ion batteries such as LiFePO_4 have high charging and discharging efficiencies. The voltage V_{bat} remains almost constant across the range of battery's state. As shown in Figure 3-11, the voltage across battery terminals v drops less than 1 V from fully charged state to almost fully discharged state. Hence, we will estimate that the battery voltage V_{bat} of LiFePO_4 does not depend on the battery's state of charge. The datasheet gives $V_{bat} = 25.6 \text{ V}$ and $r_{bat} = 0.25 \Omega$, which are also confirmed by experiment [15].

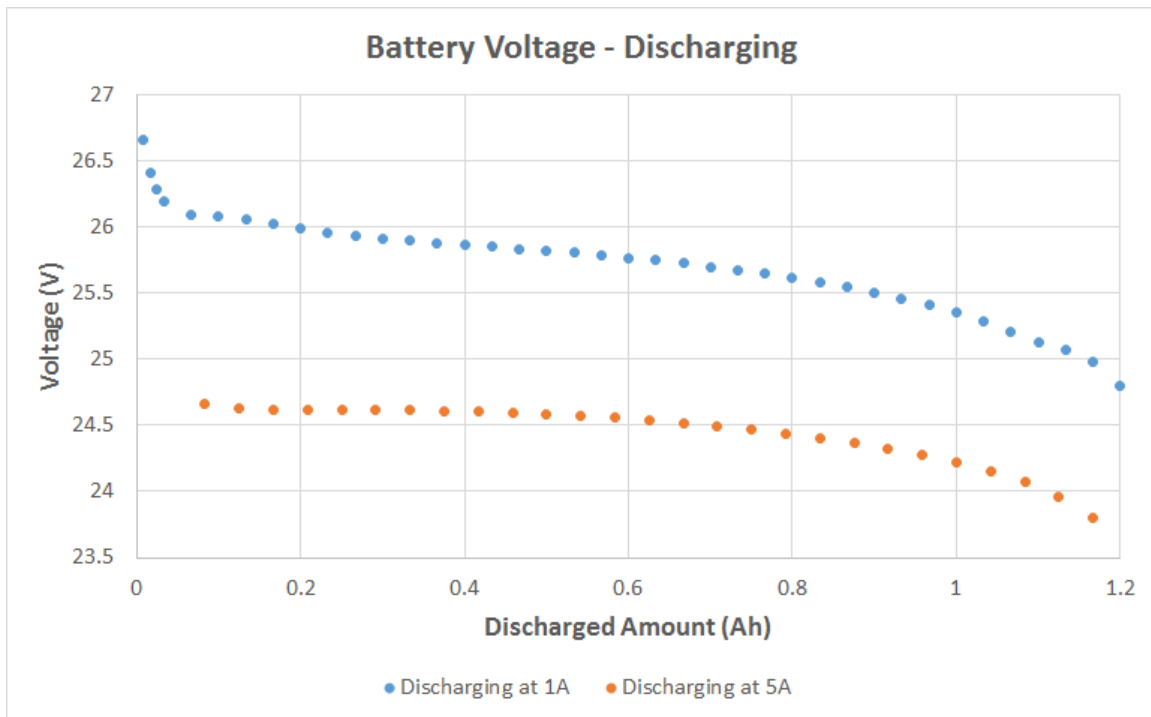


Figure 3-11: Plot of battery terminal voltage as a function of discharged amount at two different discharging rates for a 25.6 V, 1.4 Ah LiFePO_4 battery

After many cycles, we expect a change in the battery voltage, series resistance, and capacity. To delay the deterioration, battery must be kept within the optimal operating condition. Over-charging and over-draining can be prevented by monitoring the battery voltage. In the meantime, the battery current should also be monitored to keep the charging and discharging rates within limit.

Chapter 4

Pump System

The water pump is an important subsystem of Khethworks' irrigation system. It consists primarily of a brushless DC motor and an impeller. The pump's main function is to convert electrical input power into mechanical output power that pushes water against some pressure head, resulting in an upward flow of water. In this chapter, we study each pump component and make necessary approximation to characterize the pump and derive its quantitative model.

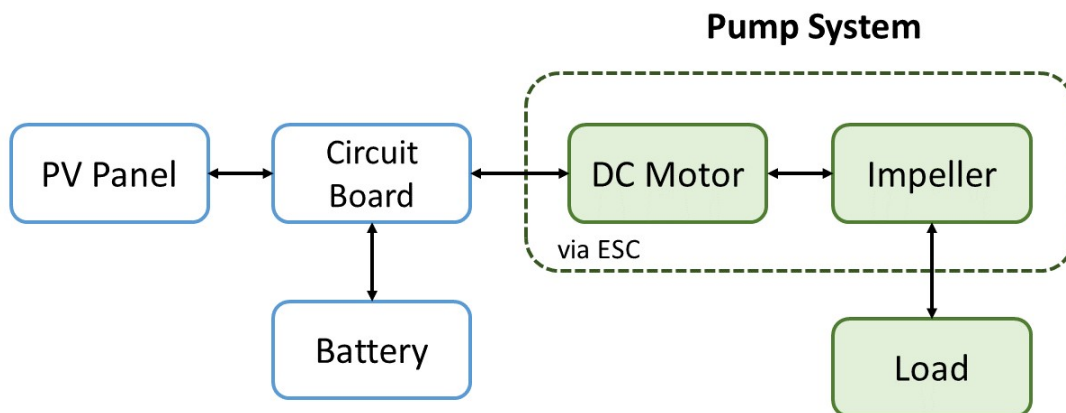


Figure 4-1: Diagram highlighting pump components in Khethworks irrigation system

4.1 Motor

Khethworks' irrigation system utilizes a brushless DC motor for its custom-designed pump. Brushless DC motors are becoming more popular for compact and portable applications [16]. With no commutator and brushes, there is less friction and the motor can achieve higher angular speed. Brushless motors also often have higher efficiency, lower electrical noise, and less EMI than that of brushed motors. The lack of physical commutator, however, requires brushless motor to have a separate controller, known as electronic speed controller (ESC). An ESC takes in input power from the source and a pulse width modulated signal (PWM) from a microcontroller unit. It manages the phase of voltage and current going into each winding. In addition to acting as an electronic commutator, the PWM control signal acts as a throttle to the brushless motor. Together, the input voltage and the PWM signal determine the motor's angular speed.

With voltage level and PWM signal as inputs, the brushless DC motor rotates at an angular speed ω and exerts torque τ . As illustrated in Figure 4-2, the motor's torque decreases linearly as its speed increases. Increasing the voltage level pushes the $\tau - \omega$ curve away from the origin, such that the torque exerted by the motor will increase if we hold the angular speed constant.

In Figure 4-3, the $\tau - \omega$ curve responds to increasing PWM throttle level in a similar manner. Holding the speed constant and increasing the PWM throttle will increase the torque. Note that despite following the same trends, the lines at different PWMs are not as parallel as those at different voltage levels in the figure above.

4.2 Impeller

An impeller is another rotating part of the centrifugal pump. The impeller receives its torque and angular speed from the attached motor and generates the flow by accelerating water outward from its center. Together with a volute, the impeller delivers a flow rate Q of the water against a certain pressure head H . It maps a

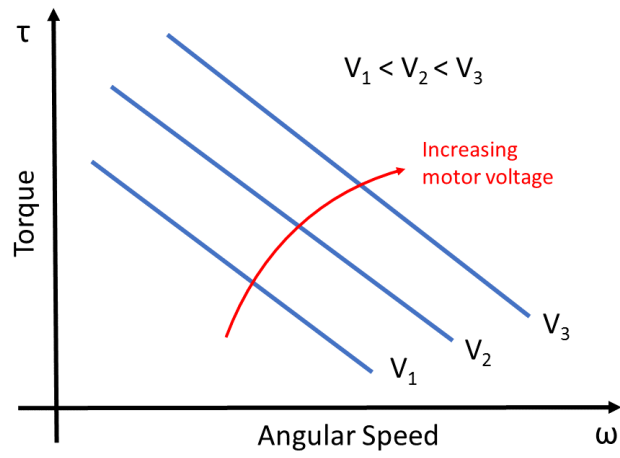


Figure 4-2: Typical $\tau - \omega$ characteristic curves of a brushless DC motor at a constant PWM throttle and varying voltage levels

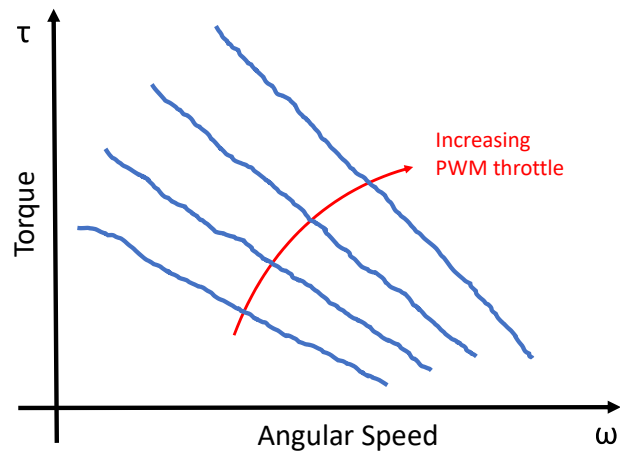


Figure 4-3: Typical $\tau - \omega$ characteristic curves of a brushless DC motor at a constant voltage level and varying PWM throttles

point (τ, ω) in the torque-speed space to a point (H, Q) in the pressure-flow rate space. This mapping depends on the impeller's geometry and may or may not be unique. Figure 4-4 shows typical characteristic curves in both spaces. Every point on the $\tau - \omega$ curve has at least one corresponding points on the $H - Q$ curve. These four parameters - torque, angular speed, pressure head, and flow rate - sufficiently describe the impeller's state when the pump is connected to its power source and mechanical load.

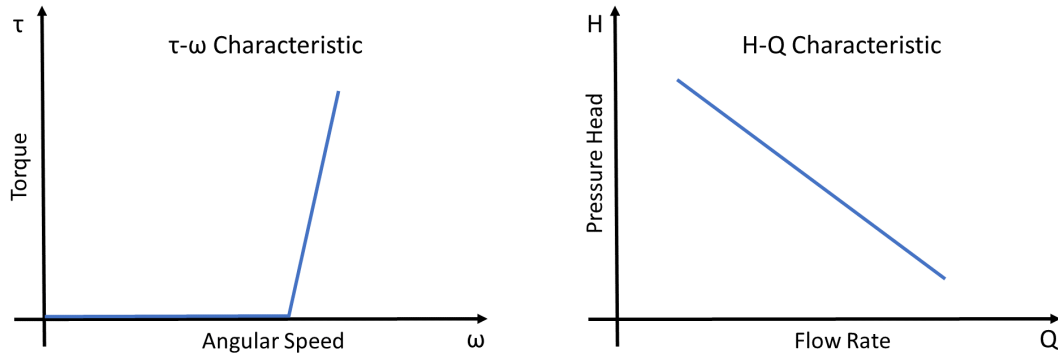


Figure 4-4: Typical characteristic curves of a centrifugal pump's impeller in (a) the torque-angular speed space and (b) the pressure-flow rate space

4.3 Mechanical Load

A pump's mechanical load is the pressure head against which the pump must push working fluid. In the case of our irrigation system, the centrifugal pump lifts water up from a well and delivers it at a certain flow rate through some pipes.

The pressure head H consists of three terms. First, the static pressure at the end of the volute p . Second, the velocity head of flow $\frac{1}{2}\rho\nu^2$, where ρ is the density of the fluid and ν is the fluid's velocity at the volute outlet. Lastly, the elevation head ρgz , where g is the acceleration due to gravity and z is the elevation difference between the inlet of the pump and the point where the fluid reaches atmospheric pressure. In our system, z is simply the elevation difference between the farm's ground level and the water level in the well.

$$H = p + \frac{1}{2}\rho v^2 + \rho g z \quad (4.1)$$

Higher flow rate requires the water to flow faster. Therefore, as the flow rate Q increases, the velocity head as well as the total pressure head increases. For the application in this study, however, static pressure and velocity head tend to be significantly smaller than elevation head. With a slow outlet velocity of about 1 m/s and the well depth of about 10 m,

$$\frac{1}{2}\rho v^2 = \frac{1}{2}(1000\text{kg/m}^3)(1\text{m/s})^2 = 0.5\text{kPa} \quad (4.2)$$

$$\rho g z = (1000\text{kg/m}^3)(9.8\text{m/s}^2)(10\text{m}) = 98\text{kPa} \quad (4.3)$$

As shown in Figure 4-5, the total pressure head H is therefore approximately the elevation head $\rho g z$ and independent of the flow rate Q .

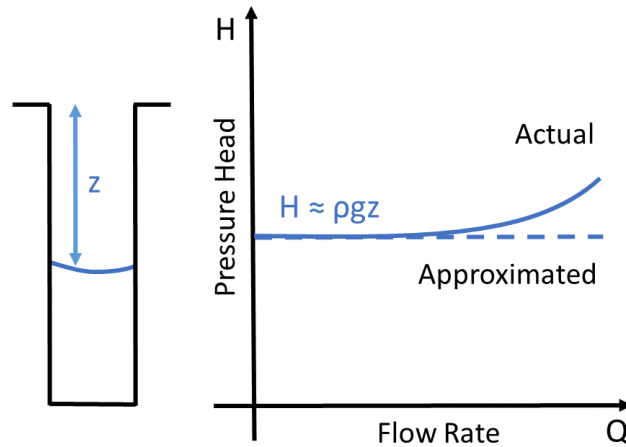


Figure 4-5: Typical $H - Q$ characteristic curve of the pump's mechanical load in Khethworks irrigation system

4.4 Pump's Operating Points

Combining characteristic curves of the motor, impeller, and load, the pump operating points in both $\tau - \omega$ and $H - Q$ spaces can be determined. Some of the pump parameters, such as the motor's angular speed ω , will be sensed by the control board and serve as input for the control algorithm. The system's control will be discussed in details in Chapter 6.

Figure 4-6 illustrates how the operating point (τ, ω) is determined from the intersection between characteristic curves of brushless DC motor and impeller. As the PWM throttles increase, the pump's torque and angular speed also increase. Consequently, more mechanical power is transferred to the flowing water.

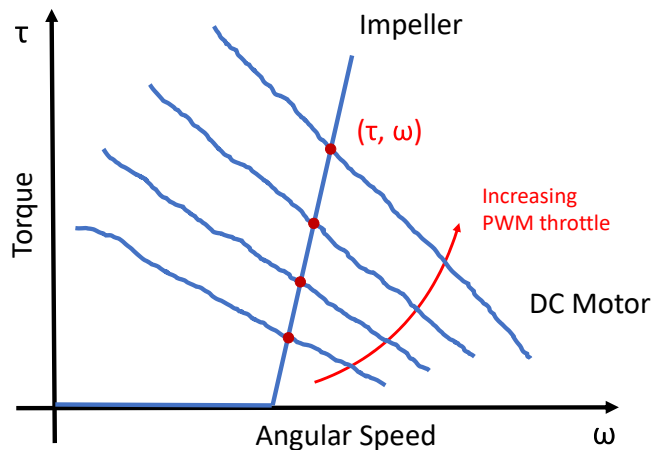


Figure 4-6: Pump's operating points in the $\tau - \omega$ space at a constant voltage level and varying PWM throttles

Similarly, in Figure 4-7, the operating point (H, Q) is determined from the characteristic curves of impeller and pump's load line. As the motor voltage increases, the operating point moves rightward along the load line. Because of the load characteristic, the pump can deliver water at greater flow rate while the pressure head remains approximately constant.

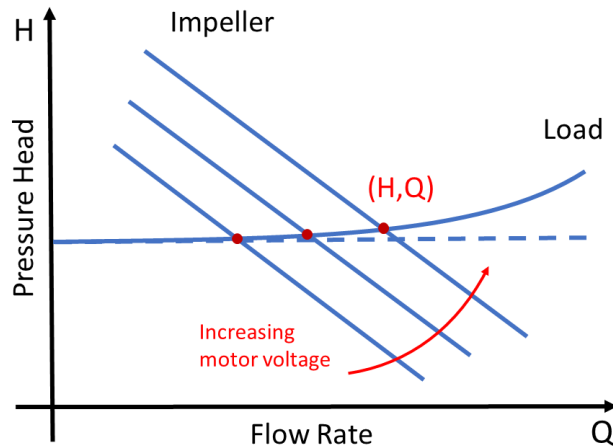


Figure 4-7: Pump’s operating points in the $H - Q$ space at a constant PWM throttle and varying voltage levels

4.5 Test Results

While the qualitative trend or dependency of the pump’s operating point on a certain parameter can be predicted, it is very difficult to pinpoint the exact location of the operating point with only theoretical calculation. To formulate an accurate pump model for system-level simulation purpose, the pump must be characterized empirically.

In the actual irrigation system, the motor’s voltage and input current depend greatly on the $i - v$ characteristics of other electrical components. Components that affect the system’s (i, v) operating point include photovoltaic panel and battery, as well as the MPPT converter if the maximum power point tracking scheme is implemented. In order to characterize the pump’s operation, however, various pump parameters are measured as a function of pressure head H at three different motor voltage levels. The PWM throttle is kept constant by fixing the electronic speed control’s dial throughout the experiment. By varying the pressure head, we imitate the varying depth of water wells where the pump will be used.

The following four pump parameters are measured/calculated:

- Angular speed ω of the motor and the impeller is directly measured.

- Flow rate Q is directly measured.
- Electrical power consumed by the motor $P = iv$, where v is the motor's voltage and i is the motor's input current.
- Overall efficiency of the pump $\eta_p = \frac{H \cdot Q}{i \cdot v}$

This pump characterization experiment was set up and conducted by Kevin Simon, a Ph.D. candidate at MIT's Department of Mechanical Engineering.

The overall efficiency η_p indicates how well the pump converts its electrical input power into mechanical power of flowing water. Figure 4-11 suggests that η_p is a function of the pressure head and the motor's voltage level.

$$\eta_p(v, H) = \frac{\text{Mechanical Power Out}}{\text{Electrical Power In}} = \frac{H \cdot Q}{i \cdot v} \quad (4.4)$$

In the system-level simulation, we assume that the well depth changes at a significantly slower rate than other parameters. For example, while the motor's parameters and the pump's flow rate change on a scale of seconds, the well depth often remains unchanged throughout the day. Therefore, at each moment in the simulation, H is treated as a constant and η_p is considered a function of only v .

Because the goal of the irrigation system is to deliver as much water as possible, we aim to maximize

$$[\text{Daily Water Delivered}] = \int_{\text{day}} Q(H, v) dt \approx \int_{\text{day}} Q(v) dt \quad (4.5)$$

The electrical characteristic of the pump system is modelled from the motor's power consumption $P(H, v)$. The pump system's $i - v$ characteristic is

$$i = \frac{P(H, v)}{v} \approx \frac{P(v)}{v} \quad (4.6)$$

Note how the motor's input current i is approximately a function of only the motor voltage v when the pressure head H is assumed constant.

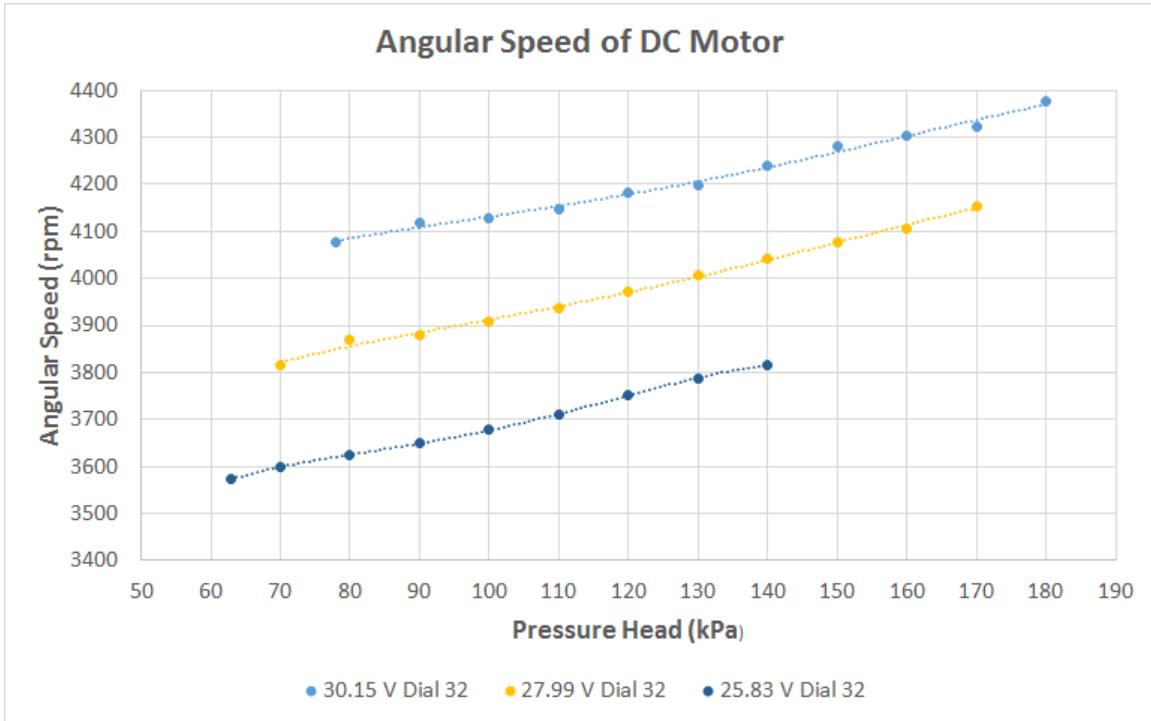


Figure 4-8: Plot of the motor’s angular speed ω as a function of the pressure head H at 3 voltage levels and a constant PWM throttle

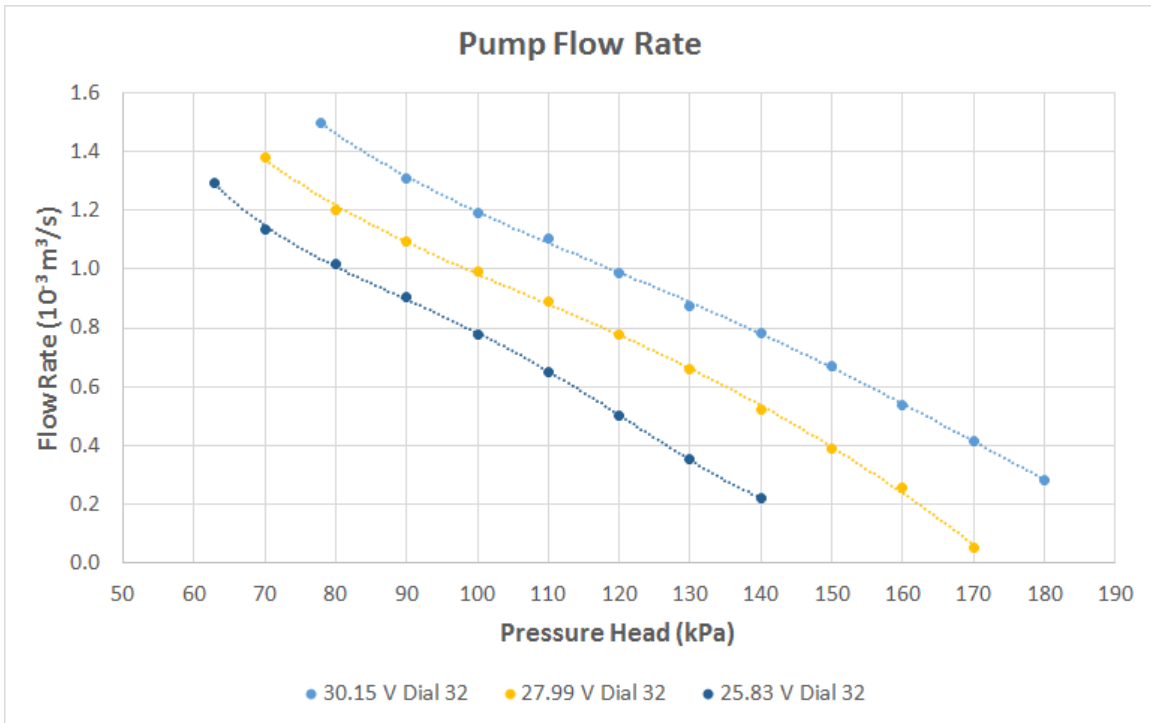


Figure 4-9: Plot of the pump’s flow rate Q as a function of the pressure head H at 3 voltage levels and a constant PWM throttle

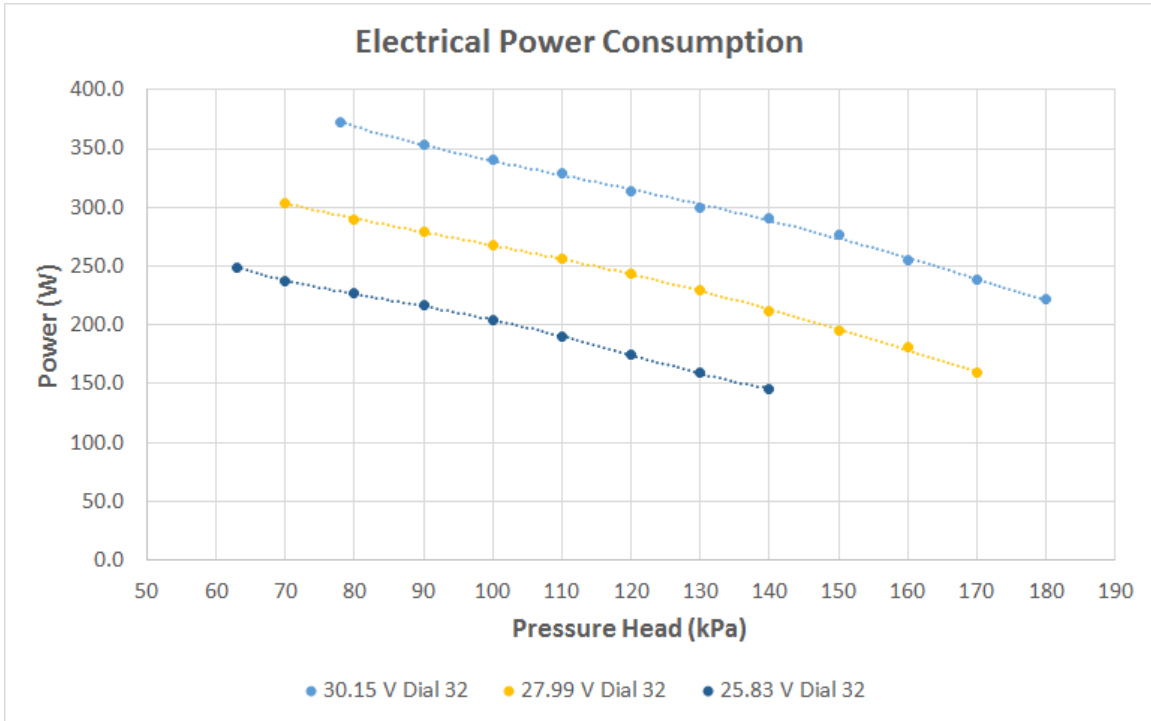


Figure 4-10: Plot of the motor’s electrical power consumption P as a function of the pressure head H at 3 voltage levels and a constant PWM throttle

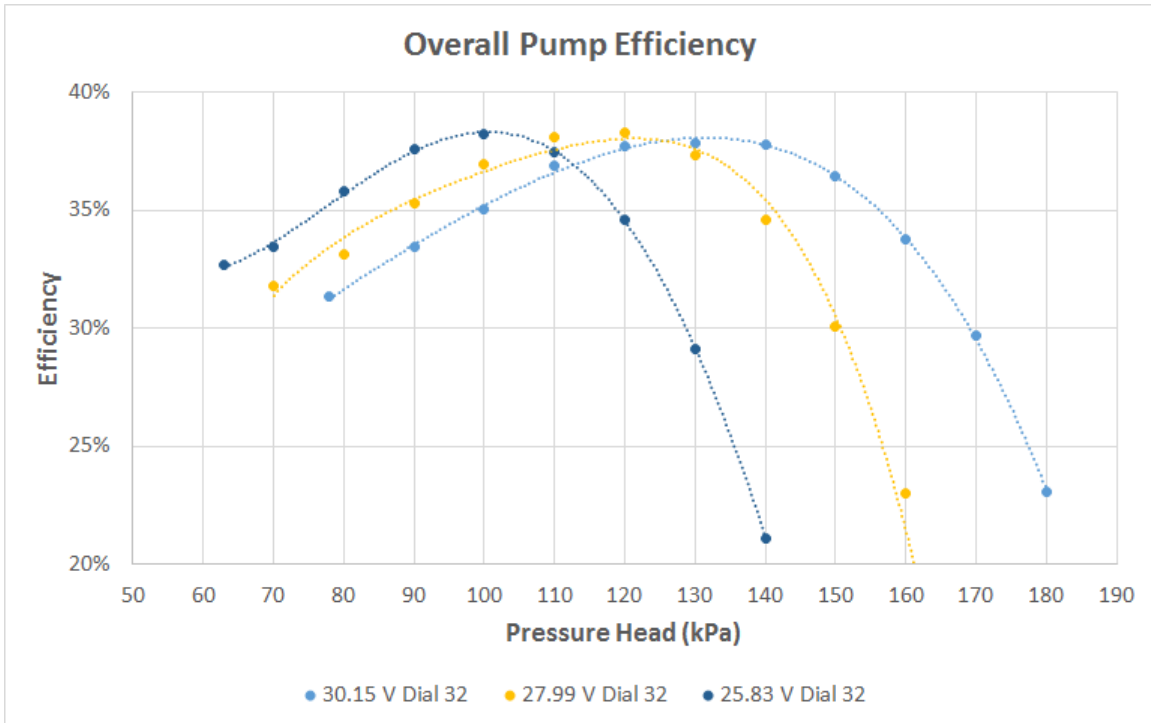


Figure 4-11: Plot of the pump’s overall efficiency η_p as a function of the pressure head H at 3 voltage levels and a constant PWM throttle

Chapter 5

Maximum Power Point Tracking

Because of photovoltaic panel's electrical characteristic, there exists a point on its $i-v$ curve where the electrical power drawn from the panel is maximized. Section 3.1.3 discusses several properties of a photovoltaic panel's maximum power point. While the maximum power P_{MPP} can be approximated with a linear model, the maximum power point (i_{MPP}, v_{MPP}) is often too complicated to be explicitly calculated in terms of panel parameters such as irradiation level G and cell temperature T_c . Therefore, Maximum Power Point Tracking (MPPT) algorithms are often implemented to keep the panel's operating point as close to the maximum power point as possible.

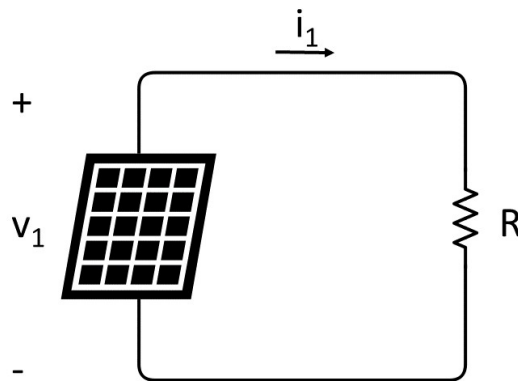


Figure 5-1: A circuit diagram with a constant load connected directly to a photovoltaic panel

In Figure 5-1, a constant load R is directly connected to a photovoltaic panel. The

operating point must satisfy (i, v) characteristics of both the photovoltaic panel and the load. As illustrated in Figure 5-3, the panel's (i, v) curve is similar to what we have seen in the earlier chapters, and the constant load's (i, v) curve is a straight line with a slope of $1/R$. The operating point of this direct load line (DLL) approach (i_1, v_1) locates at the intersection of the two curves. Because we have no control over either curve, (i_1, v_1) is not guaranteed to be the maximum power point of the photovoltaic panel.

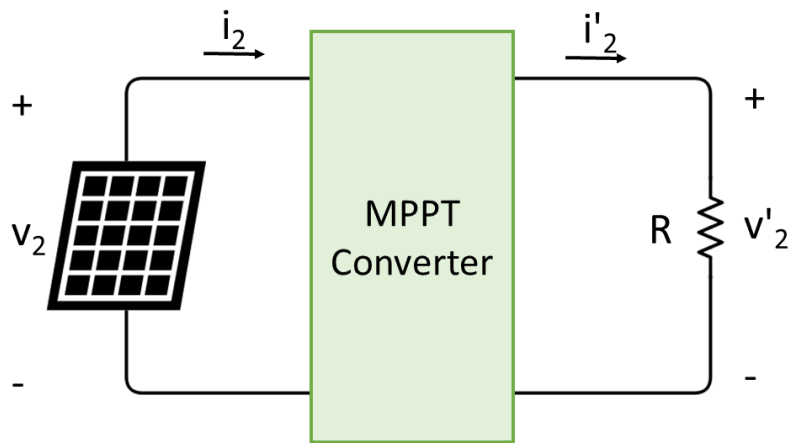


Figure 5-2: A circuit diagram with a constant load, a photovoltaic panel, and an MPPT converter

To achieve maximum output power, a DC-DC power converter is added to interface between the photovoltaic panel and the load, as illustrated in Figure 5-2. By changing the converter's voltage conversion ratio, the slope of the load line as seen as by the photovoltaic panel can be adjusted. The conversion ratio is chosen such that the new operating point (i_2, v_2) locates right at the maximum power point of the panel's $i - v$ characteristic curve.

Assuming that the MPPT converter is lossless,

$$\frac{(v'_2)^2}{R} = P_{MPP} \quad (5.1)$$

The converter's duty cycle should be set such that

$$\text{Voltage Conversion Ratio} = \frac{v'_2}{v_2} = \frac{\sqrt{P_{MPP}R}}{v_2} \quad (5.2)$$

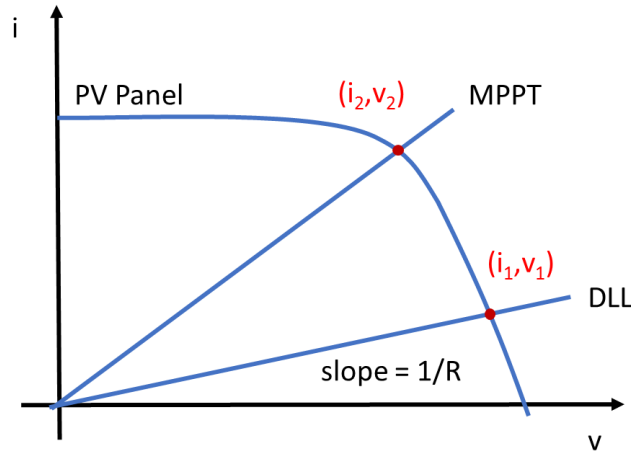


Figure 5-3: Graph comparing (i, v) operating points of the direct load line and the maximum power point tracking approaches. The operating points (i_1, v_1) and (i_2, v_2) correspond to the circuits in Figure 5-1 and Figure 5-2, respectively.

A photovoltaic system that draws maximum power from the panel at an instance can become very inefficient if it does not track the maximum power point as the system parameters change. For example, the maximum power point of JA Solar’s 320 W photovoltaic panel in Figure 5-4 at irradiation level $G = 1000 \text{ W/m}^2$ and cell temperature $T_c = 25 \text{ }^\circ\text{C}$ corresponds to $v = 37.8 \text{ V}$. If the cell temperature increases to $T_c = 55 \text{ }^\circ\text{C}$ but the system’s operating point is fixed at $v = 37.8 \text{ V}$, the panel’s new output power will be 206 W. This is only 73% of the panel’s maximum power $v_{MPP} = 283 \text{ W}$ at $G = 1000 \text{ W/m}^2$ and $T_c = 25 \text{ }^\circ\text{C}$. A power converter with MPPT capability is necessary for utilizing full potential of the photovoltaic panel. Without MPPT, the photovoltaic panel has to be oversized to accommodate power demand across the ranges of irradiation levels and cell temperatures.

As discussed in Chapter 4, the $i - v$ characteristic of our pump system is more complicated than that of a constant load. To derive a model for a photovoltaic panel with MPPT converter in system-level simulation, the converter is assumed to always bring the panel to its maximum power point. This assumption is based on the fact

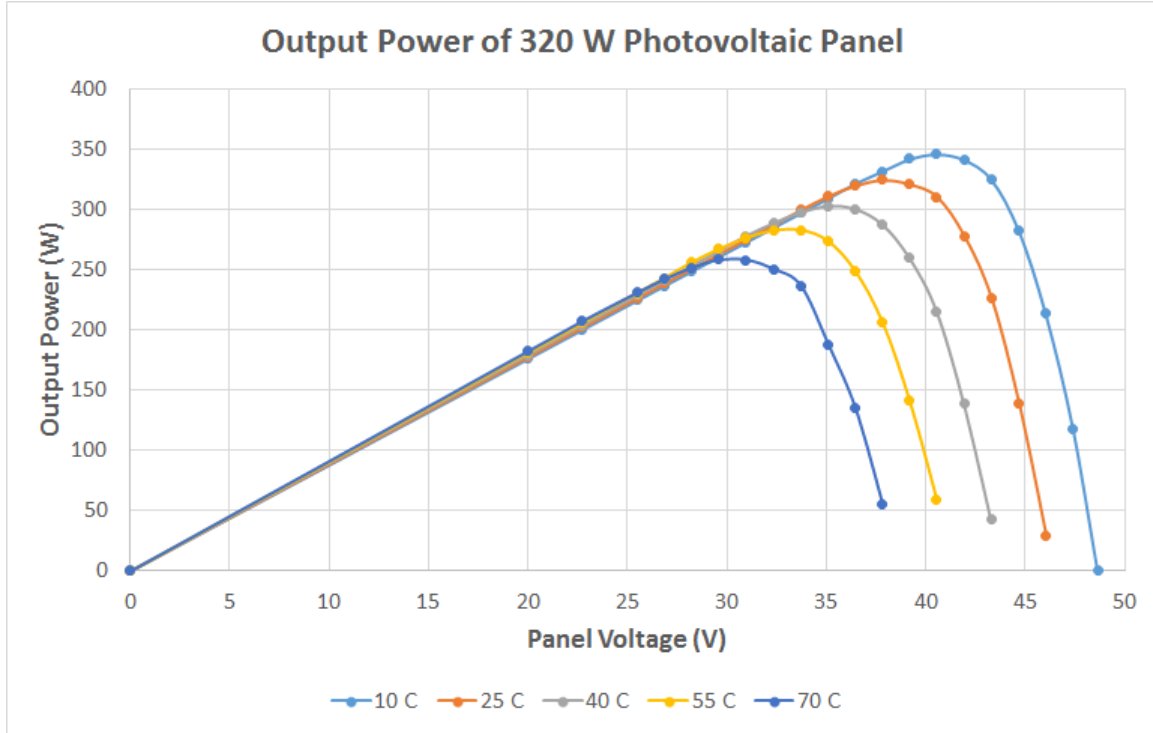


Figure 5-4: Plot of output power of JA Solar’s 320 W photovoltaic panel as a function of panel voltage at $G = 1000 \text{ W/m}^2$, for five different cell temperatures

that MPPT converter can track the maximum power point in the order of seconds, while system parameters such as irradiation level and cell temperature take minutes to vary.

At the maximum power point, output power of the MPPT converter is

$$P_{OUT} = \eta_M P_{MPP}(G, T_c) \quad (5.3)$$

where η_M is the converter’s efficiency and $0 < \eta_M < 1$

Therefore, the MPPT converter replaces the photovoltaic panel’s original $i - v$ characteristic curve with a new constraint

$$\eta_M P_{MPP}(G, T_c) = iv \quad (5.4)$$

For cases where maximum power consumption by the rest of the system is less than $\eta_M P_{MPP}$, such as when the pump is off and the panel only charges the battery, then the power supplied by the panel via MPPT converter will be equal to the total

power consumption of all other components in the system.

Chapter 6

System's Operating Points

Knowing the electrical characteristic of each component in the irrigation system, we can calculate the system's electrical operating point. Because the goal of this study is to maximize the photovoltaic panel's utility, the two most important parameters are the output current i and terminal voltage v of the system's power source. These two parameters are defined differently in the direct load line (DLL) and maximum power point tracking (MPPT) cases. As shown in Figure 6-1, for the DLL approach, the power source is simply the photovoltaic panel. On the other hand, in the MPPT case, the power source consists of the panel and the MPPT converter. The system's electrical operating point is defined as $(i(t), v(t))$.

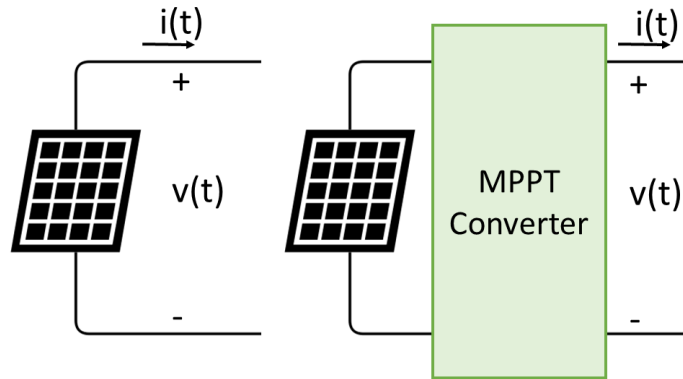


Figure 6-1: Definition of the power source's output current $i(t)$ and terminal voltage $v(t)$ for (a) direct load line and (b) maximum power point tracking approaches

Note that there are also other time-dependent system state parameters, including the battery charge, the battery input current, and the pump's input current. Knowing the system state $S(t)$, we calculate the system's operating point $(i(t), v(t))$, then update the system state to $S(t + \Delta t)$ and calculate the operating point at the next time instance $(i(t + \Delta t), v(t + \Delta t))$.

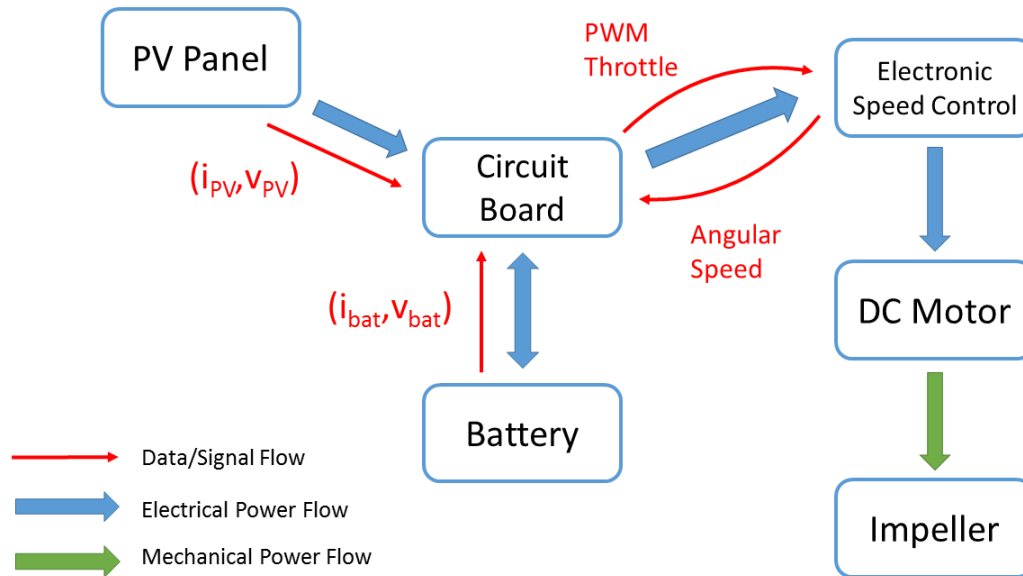


Figure 6-2: Diagram showing power and signal flows between components in Khethworks irrigation system with direct load line connection

The circuit board connects the panel, the battery, and the pump. It controls the switches that connect and disconnect component from the main circuit. For example, when the pump switch is off, the electronic speed control and the rest of the pump system should be disconnected from the power source. The control board also senses voltage and current values at various locations in the system and regulate the brushless DC motor's angular speed. The motor's voltage is approximately equal to that of the system operating point, assuming no voltage drop from the panel to the motor. As the operating point's voltage changes, the control board senses the changing angular speed of the motor and adjust the PWM throttle input signal of the electronic speed control to reach the target angular speed.

The control scheme of the MPPT case is similar to that of the DLL case. However,

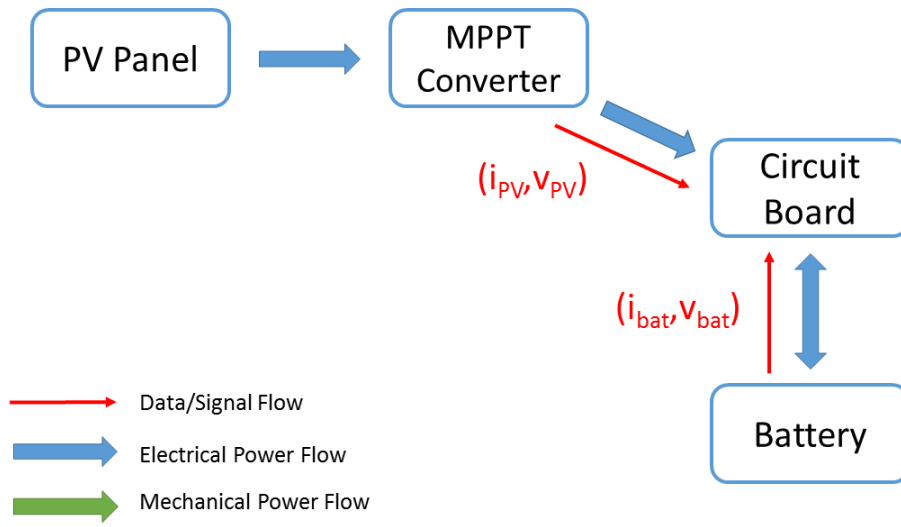


Figure 6-3: Diagram showing power and signal flows between some components in Khethworks irrigation system with maximum power point tracking

with the MPPT converter serving as an interface between the photovoltaic panel and the rest of the system, the power source's $i - v$ characteristics depends on both the converter and the panel, as explained earlier in Chapter 5.

6.1 Case 1: Pump OFF

When the pump is switched off, only the battery is left connected to the power source. Therefore, the power source's output current is equal to the battery input current. For convenience, the current leaving the power source, such as the photovoltaic panel in Figure 6-4 (a) or the MPPT converter in Figure 6-4 (b), is defined as positive. On the other hand, for other components such as the battery and the pump, the current going into the component is defined as positive.

For the direct load line approach, the system's operating point (i, v) must satisfy the following system of equations:

$$\begin{aligned}
\text{PV Panel: } v &= V_{OC} - iR_S - \frac{nkT_cN}{q} \ln\left(\frac{I_L}{I_L - i}\right) \\
\text{Battery: } v &= V_{bat} + i_{bat}r_{bat} \\
\text{Current: } i &= i_{bat}
\end{aligned}
\tag{6.1}$$

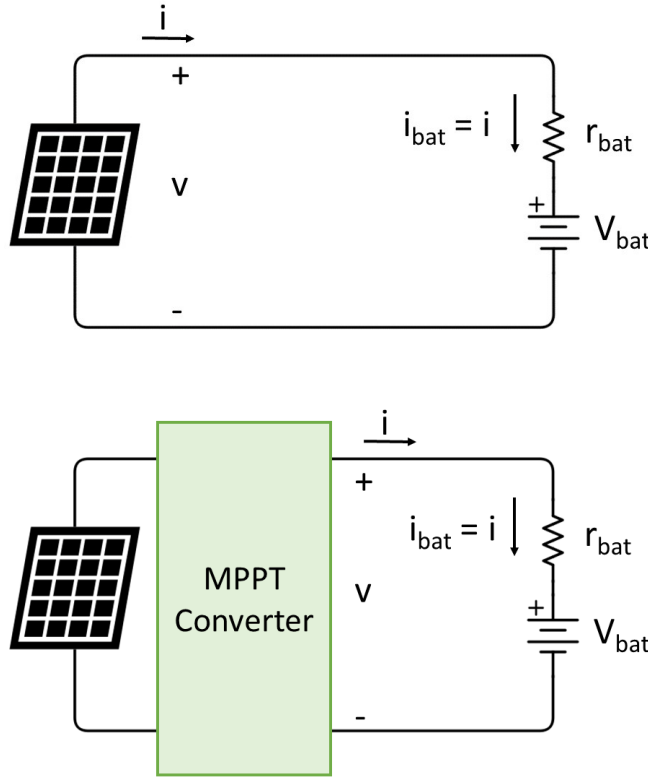


Figure 6-4: Circuit diagrams of the irrigation system with the water pump switched off, for (a) direct load line and (b) maximum power point tracking approaches

Figure 6-5 illustrates the graphical interpretation on the $i \times v$ plane of the system of equations in Equation (6.1). The $i - v$ characteristic curves of the photovoltaic panel and the battery intersect at the system's operating point.

Because explicitly solving the system of equations is difficult, the red arrows in Figure 6-5 demonstrates a possible iterative method that can be utilized to obtain the numerical values of i and v . Starting from $v_1 = V_{bat}$, we locate the point (i_1, V_{bat}) on the panel's characteristic curve. Knowing i_1 , we find the point (i_1, v_2) on the battery's characteristic curve. With v_2 , we return to the panel's curve to locate the point (i_2, v_2) . After a number of repetition, the iteration will converge to the system's operating

point (i, v) .

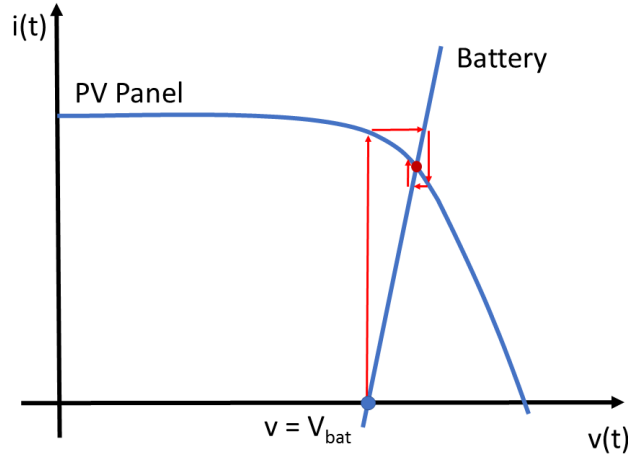


Figure 6-5: Graphical interpretation of the system's operating point for the direct load line approach with the pump switched off

For the maximum power point tracking approach, the system's operating point (i, v) must satisfy the following system of equations:

$$\begin{aligned}
 \text{MPPT: } i &= \frac{\eta_M P_{MPP}}{v} \\
 \text{Battery: } v &= V_{bat} + i_{bat} r_{bat} \\
 \text{Current: } i &= i_{bat}
 \end{aligned} \tag{6.2}$$

Similar to the direct load line case, the system's operating point is at the intersection between the power source's and the battery's $i - v$ characteristic curves. Although the MPPT converter's $i - v$ characteristic curve is valid at all values of v , in the actual system, a limit has been imposed on the allowed range of the converter's output voltage v to protect the battery and the pump's motor from over-voltage and under-voltage conditions. As shown in Figure 6-6, the red vertical dotted lines represent the allowed range of the MPPT converter's output voltage.

Under certain conditions where the photovoltaic panel's maximum output power P_{MPP} is very high, the system's voltage according to Equation (6.2) might exceed the maximum allowed value. In such case, as shown in Figure 6-7, the MPPT algorithm will hold the converter at its maximum output voltage. The new system's operating

point corresponds to a new output power level $\eta_M P'_{MPP}$ from the power source. While $P'_{MPP} < P_{MPP}$ and the photovoltaic panel is not fully utilized, the battery is still charged at the highest allowed power rating.

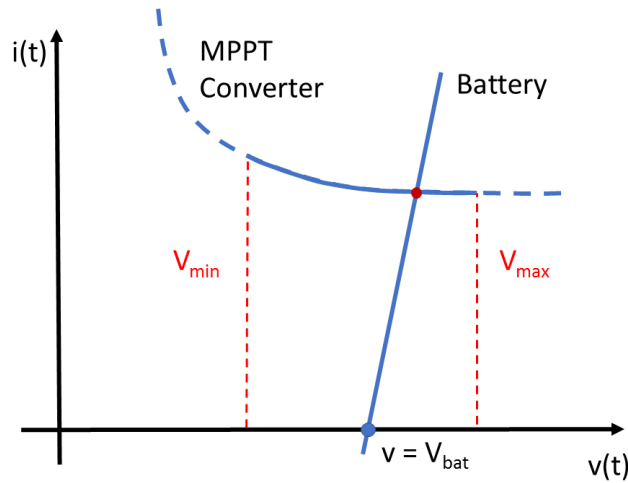


Figure 6-6: Graphical interpretation of the system's operating point for the maximum power point tracking approach with the pump switched off

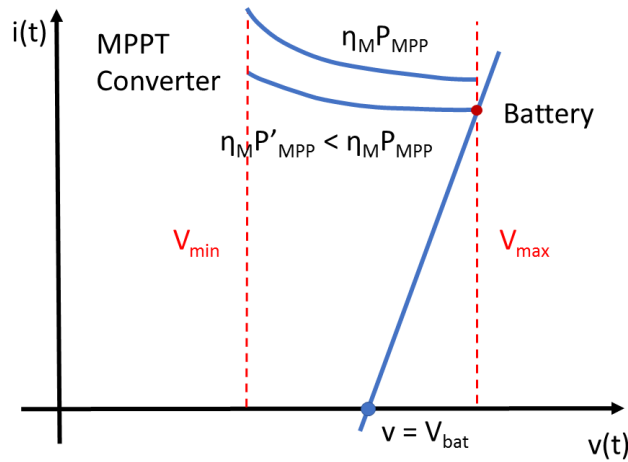


Figure 6-7: Graphical interpretation of the system's operating point for the direct load line approach with the pump switched off, showing the special case where the electrical power drawn from the photovoltaic panel is less than its maximum output capability

6.2 Case 2: Pump On

When the pump is switched on, both the battery and the pump's motor are connected to the power source. With the addition of the pump as a second load, the current leaving the power source is equal to the sum of the battery's and the pump's input currents. While the power source's output current i and the pump's input current i_m are always greater than or equal to zero, the battery's input current i_{bat} can be both positive and negative.

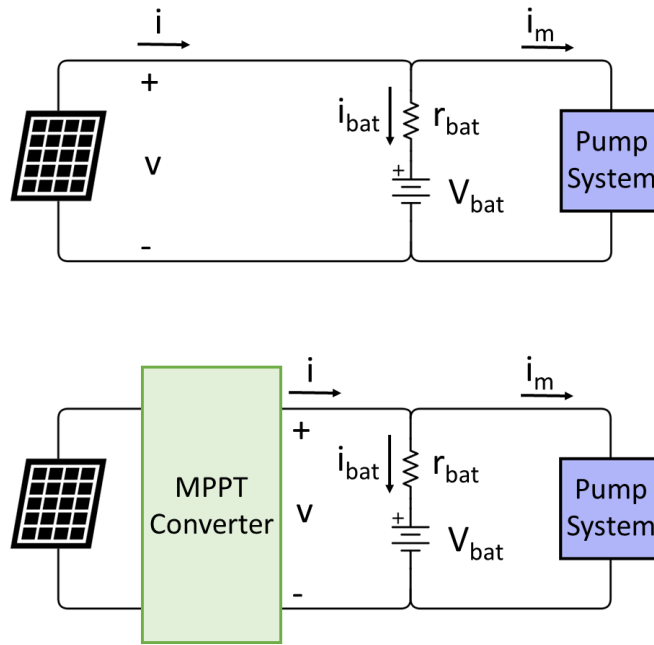


Figure 6-8: Circuit diagrams of the irrigation system with the water pump switched on, for (a) direct load line and (b) maximum power point tracking approaches

For the direct load line approach, the system's operating point (i, v) must satisfy the following system of equations:

$$\begin{aligned}
 \text{PV Panel: } v &= V_{OC} - iR_s - \frac{nkT_cN}{q} \ln\left(\frac{I_L}{I_L - i}\right) \\
 \text{Battery: } v &= V_{bat} + i_{bat}r_{bat} \\
 \text{Pump: } i_m &= \frac{P(H, v)}{v} \\
 \text{Current: } i &= i_{bat} + i_m
 \end{aligned} \tag{6.3}$$

As discussed in Chapter 4, the pump’s electrical power consumption $P(H, v)$ is available empirically for only some values of motor’s terminal voltage v and pressure head H . To solve equation (6.3) for (i, v) , it is necessary for the motor state to be approximated. For a value of pressure head H , which is assumed to be constant for each day of the pump’s operation, the motor voltage v is split into intervals.

Recall from equation (4.6) that when the pressure head H remains constant over time, the motor’s input current is a function of only the motor’s voltage v .

$$i_m = \frac{P(H, v)}{v} = \frac{P(v)}{v} \quad (6.4)$$

Therefore, the approximation scheme, illustrated in Figure 6-9, assumes a constant value of the motor’s input current i_m for any motor’s voltage v in each interval.

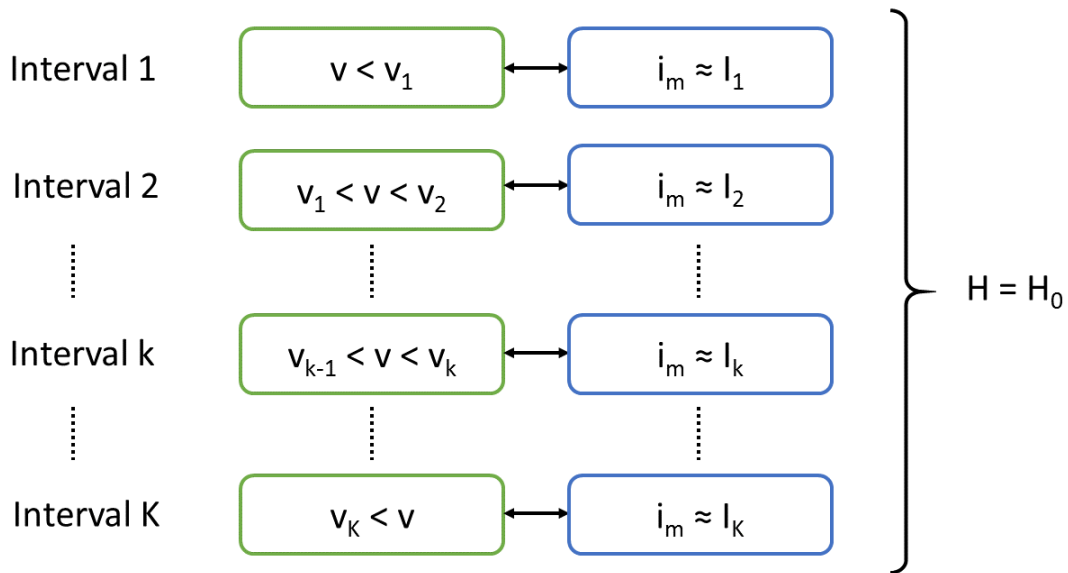


Figure 6-9: Diagram showing the approximation scheme for the pump’s motor state

To solve for the system’s operating point, the motor’s voltage $v(t)$ is initially assumed to be in the same interval as the voltage of the previous time instance $v(t - \Delta t)$. If this assumption of $v(t)$ turns out to be wrong, the system-level simulation’s algorithm will scan through every possible voltage interval, calculate the corresponding value of $v(t)$, and locate the correct interval of the motor’s state.

Suppose the motor's terminal voltage is in the k^{th} interval where $i_m \approx I_k$, equation (6.3) for the direct load line approach can be written as:

$$\begin{aligned}
 \text{PV Panel: } v &= V_{OC} - iR_S - \frac{nkT_cN}{q} \ln\left(\frac{I_L}{I_L - i}\right) \\
 \text{Battery: } v &= V_{bat} + i_{bat}r_{bat} \\
 \text{Pump: } i_m &\approx I_k \\
 \text{Current: } i &= i_{bat} + i_m
 \end{aligned} \tag{6.5}$$

The graphical interpretation of equation (6.5) strongly resembles that of Case 1, when the pump is off. In Figure 6-10, the approximately constant motor's input current results in a new load line for the battery and the motor combined. The new load line is located to the left of the battery's load line on the v -axis and is parallel to the original load line. Oftentimes, turning on the pump leads to an increase in the operating point's voltage v . However, for the direct load line approach, changing the value of v does not guarantee an increase in the electrical power drawn from the photovoltaic panel.

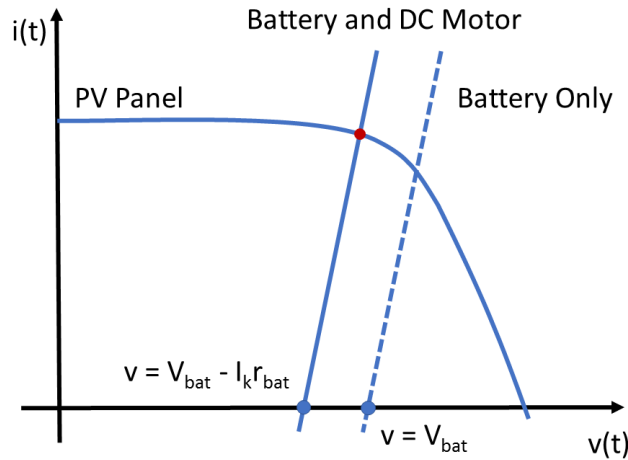


Figure 6-10: Graphical interpretation of the system's operating point for the direct load line approach with the pump switched on

For the maximum power point tracking approach, using the approximation scheme for the motor's terminal voltage described earlier, the system's operating point (i, v) must satisfy the following system of equations:

$$\begin{aligned}
\text{MPPT:} \quad i &= \frac{\eta_M P_{MPP}}{v} \\
\text{Battery:} \quad v &= V_{bat} + i_{bat} r_{bat} \\
\text{Pump:} \quad i_m &\approx I_k \\
\text{Current:} \quad i &= i_{bat} + i_m
\end{aligned} \tag{6.6}$$

Similar to the direct load line approach in Figure 6-10, switching on the water pump shifts the overall load line and tends to increase the operating point's voltage v . However, because every point on the MPPT converter's $i - v$ characteristic curve corresponds to the maximum output power $\eta_M P_{MPP}$ from the converter, changing the value of v does not affect the utilization of the photovoltaic panel.

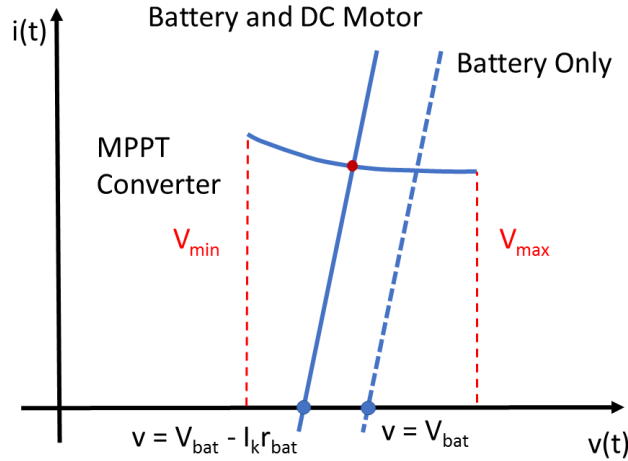
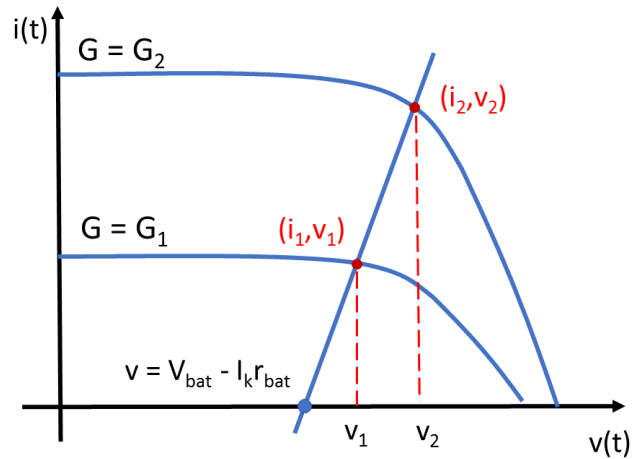


Figure 6-11: Graphical interpretation of the system's operating point for the maximum power point tracking approach with the pump switched on

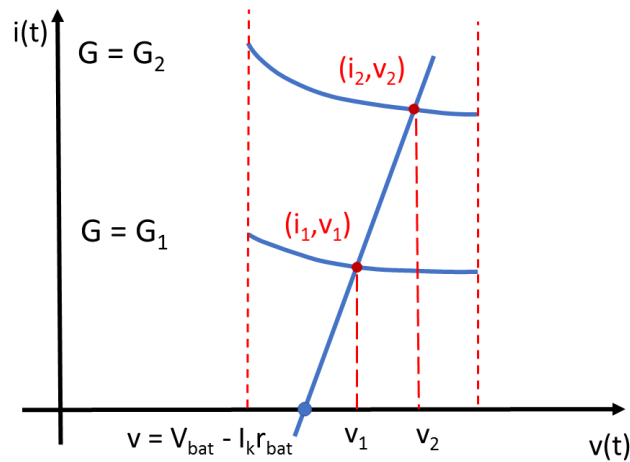
For both DLL and MPPT approaches, the graphical interpretation of the system's operating point also reflects the change in the battery's state - between charging and discharging. Figure 6-12 (a) and (b) demonstrate a change in the operating point's location as a result of changing irradiation level. In both subfigures, as the irradiation level G increases from G_1 to G_2 , the $i - v$ characteristic curve of the power source shifts away from the origin in the $+i$ and $+v$ directions to reflect the increasing P_{MPP} . Consequently, the system's operating point moves from (i_1, v_1) to (i_2, v_2) .

If $v_1 < V_{bat} < v_2$, then the battery was discharging at $G = G_1$ to supply additional

power to the pump's motor. After the irradiation level increases to $G = G_2$, the photovoltaic panel - as well as the MPPT converter for the MPPT approach - can generate sufficient electrical power for the pump's motor and charge the battery with the excess power.



(a) Direct load line approach



(b) Maximum power point tracking approach

Figure 6-12: Graphical interpretation of the system's operating point with the pump switched on, showing the change in voltage level v as a result of changing irradiation level G

Chapter 7

Theoretical Simulation

The system-level theoretical simulation is the first step of determining whether the Maximum Power Point Tracking is worth implementing. Recall that the MPPT implementation is an attempt to increase the total solar energy harvested by the photovoltaic panel and the total water output of the irrigation system. However, the benefits come at the price of increase system cost, and - possibly - reduced efficiency if the MPPT system is not sufficiently efficient at a given operating point. The simulation aims to quantify these benefits and observe the effects on other state parameters.

The flowchart in Figure 7-1 shows an overview of the theoretical simulation. The simulation input can be divided roughly into three categories. First, every target location of the irrigation system has its geographical and atmospheric data, such as the solar irradiation level and atmospheric temperature. Second, the electrical and mechanical characteristics of the irrigation system itself is also an input to the system-level simulation. The characteristics of each system component have been studied and analyzed in Chapter 3, Chapter 4, and Chapter 5. Lastly, due to the fact that the irrigation system will eventually be owned and operated by the marginal farmers, user behaviors also greatly affect the system's performance and must be accounted for in this simulation.

Combining all pieces, the simulation calculates the system's state parameters as a function of time. While each state parameter appears to be associated with only

one system component, such as the flow rate associated with the pump system, all state parameters are indeed affected by all components of the system. Chapter 6 thoroughly discusses the system's operating point, which is a crucial state parameter. The simulation output consists of parameters related to the system's performance. Some of the state parameters are directly counted as simulation output, while other output parameters are derived from the state parameters.

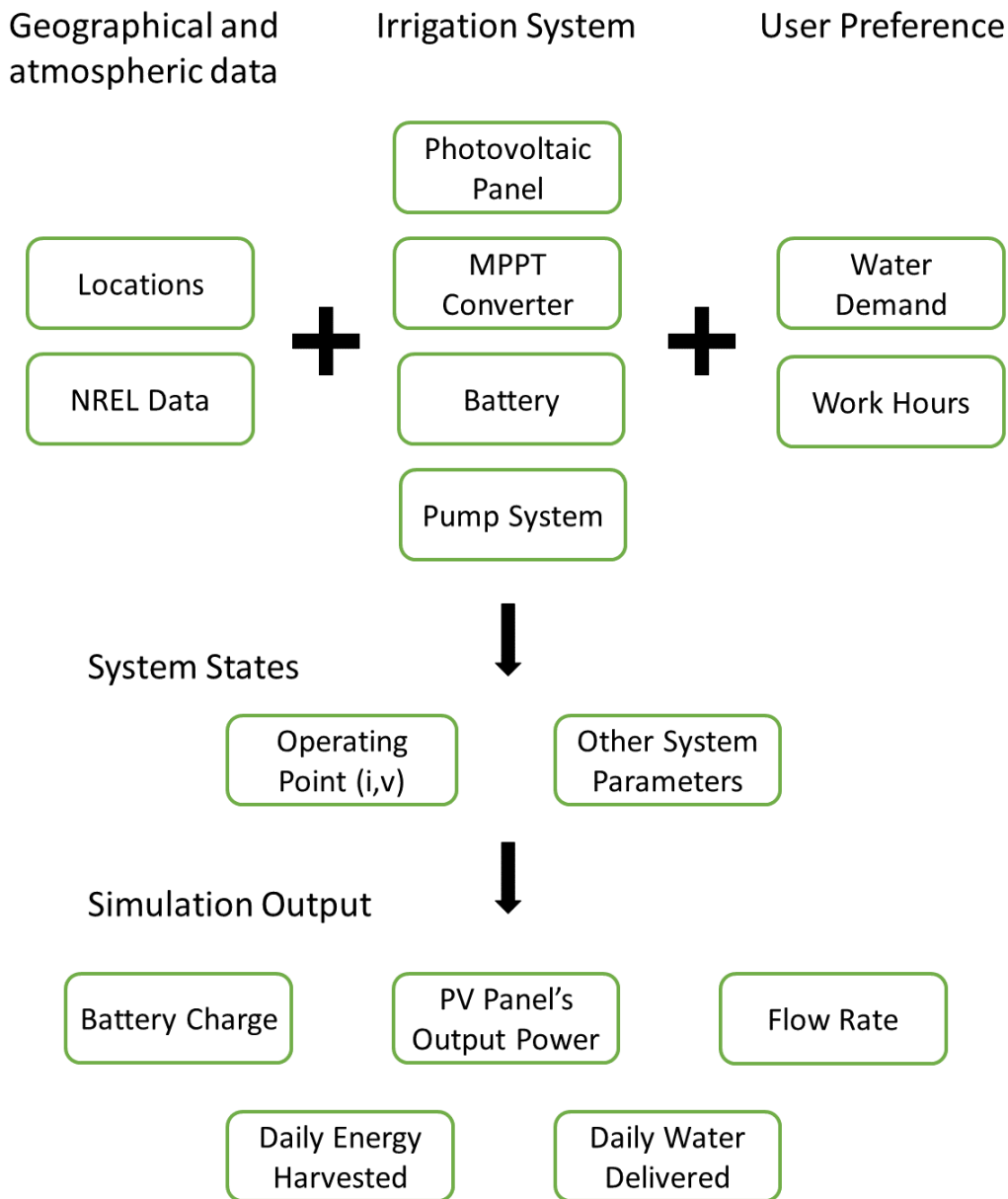


Figure 7-1: Overview of the system-level theoretical simulation

7.1 Input Variables

Input variables are the system variables that can be configured or adjusted to affect the simulation output. As the architect of the irrigation system, we have almost absolute control over a number of variables related to the system’s hardware and firmware. However, assumptions based on available data and the Khethworks team’s field survey are also necessary to obtain or estimate the numerical values of some input variables. In this section, the input variables of the system-level simulation are listed and discussed. The default and alternative values of each input variable are also given.

7.1.1 Location and Time

For simulation purpose, the locations where the irrigation system will operate are chosen in accordance with Khethworks’ current and future pilot sites.

Location	Geographic Coordinate
Chakradharpur, Jharkhand	22° 55′ N, 86° 25′ E
Kolkata, West Bengal	22° 55′ N, 88° 35′ E
Ghatgaon, Odisha	21° 35′ N, 85° 85′ E

Table 7.1: Locations of the irrigation system, for the theoretical system-level simulation

The geographical and atmospheric data of each location in Table 7.1 are obtained from the National Renewable Energy Laboratory (NREL) database. These data are available between the year 2002 and 2011. Despite running a year-long simulation, certain times of the year are more important than others. For example, the irrigation system is often not needed during the monsoon season and is absolutely demanded during the peak of the summer.

For each location and year, the following data are obtain from the NREL database:

- **Global horizontal irradiance G** - Global horizontal irradiance (GHI) is the amount of solar irradiation received by a surface horizontal to the ground. Because the photovoltaic panel is assumed to lie almost horizontal to the ground,

the solar irradiation level incident to the panel is approximately equal to the GHI.

- **Atmospheric Temperature T_a** - The atmospheric temperature for the purpose of this simulation is the dry-bulb temperature, defined as the temperature of air from a thermometer shielded from moisture and radiation.
- **Wind Speed v_w** - The wind speed near the ground is an input to the photovoltaic panel's cell temperature model.

To determine the pressure head H of the pump at each location, we approximate, as explained in Section 4.3, that the pressure head is a function of only the well's depth. According to the map of the depth to groundwater level in India, it turns out that this depth is close to 10 meters for all three chosen locations [4]. With the default value of 10 meters, the depth-to-groundwater is still considered a variable in the simulation.

7.1.2 Irrigation System

There are several variables in the specification of the irrigation system. Some of the variables are intrinsic properties of the components, while others are configurable parameters. Note that the pump system, including the electronic speed control, the brushless DC motor, and the impeller, is treated as having fixed properties in this simulation. Adjusting the pump system's parameters requires new pump design, which is out of the scope of this study.

The input variables associated with the irrigation system are as follows:

- **Photovoltaic Panel Model** - The photovoltaic panel model specifies not only not the panel's power rating but also other critical properties such as the maximum power points at various irradiation levels and cell temperatures. From the simulation's point of view, selecting the panel model is equivalent to selecting which set of (i, v) characteristic curves to use in the simulation. The default photovoltaic panel is a 260-Watt model from Tata Solar Power's TP250

series. The two alternatives are a 180-Watt model from Sharp's NU Series and a 320-Watt model from JA Solar's JAP6 series.

- **MPPT Converter's Efficiency** - The efficiency of the MPPT converter plays a critical role in determining the performance of MPPT implementation. Although the converter design in later chapter aims to maximize the converter's efficiency, the default value for this simulation is assumed to be 0.95, with the alternative values of 0.98, for the case of well designed converters, and 0.90, for the case of poorly designed converters.
- **Battery Model** - Similar to the photovoltaic panel model, selecting the battery model is equivalent to selecting the Thevenin-equivalent circuit for the simulation. Although the batteries analyzed in this study is limited to LiFePO_4 , the battery voltage V_{bat} and the equivalent series resistance r_{bat} still vary with the battery model. As stated in the analysis of battery characteristics, the default value of V_{bat} is 25.6 V and that of r_{bat} is 0.25 Ω .
- **Battery Quantity** - Given a battery model of choice, the next step is to determine how many packs of the battery should be installed in the system. The default quantity of battery is 1, with the alternative values of 2 and 4.
- **Depth of Discharge** - Because any battery should not be discharged to 0% of its capacity, the depth of discharge must be specified. One of the control board's function is to measure the battery's terminal voltage, determine its level of charge, and disconnect the battery from the rest of the system when the board detected that the battery has reached its draining limit. For this simulation, the default depth of discharge is 0.8, allowing the battery to operate between 20% and 100% of its capacity. The alternative value for the depth of discharge is 0.4. Typically, a smaller depth of discharge corresponds to a longer battery life.

7.1.3 User Preference

The preference of the marginal farmers who operate Khethworks' irrigation system also affect its performance. The default values of variables in this subsection are based on user feedback from Khethwork's pilot site in Jharkhand.

- **Water Demand** - Figure 7-2 shows the default value of water demanded by the marginal farmers, based on Khethworks' estimation for the Jharkhand area. In this plot, the day of the year starts from January 1. According to the estimation, the farmers demand the highest amount of water from March to May, smaller amount from September to February, and no water at all during the monsoon season. It is also assumed that, once the total amount of water delivered has reached the daily demand, the farmers will switch off the pump and let the battery charge up for the next day of operation.

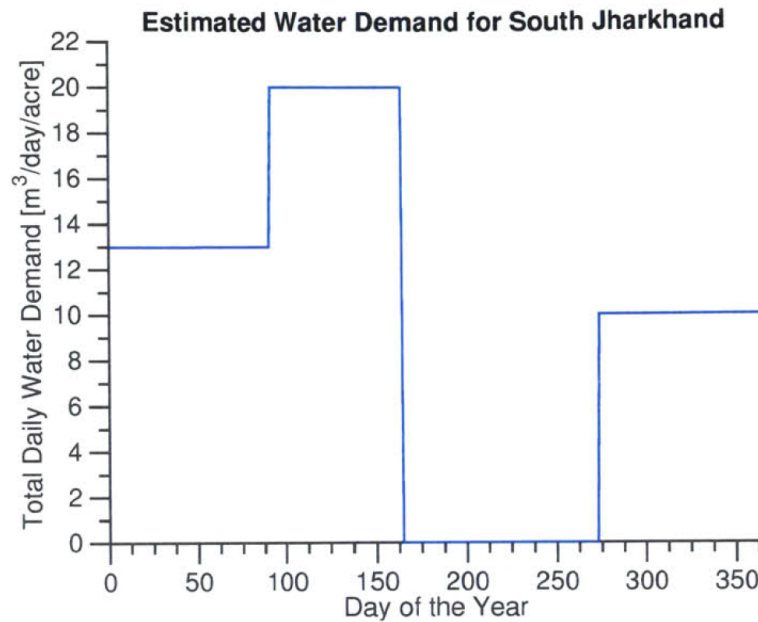


Figure 7-2: Default value of the water demand throughout the year

- **Work Hours** - In most cases, the farmers will bring the Khethworks' irrigation system with them to their plots of land at the beginning of the day, let the system operate throughout the work hours, and bring it back to their residences

at the the end of the day. Thus, it is assumed that the system operates during most of the user’s work hours. The default work hours are assumed to be from 9am to 5pm. Despite using fixed value of the work hours throughout the year, the simulation can also run with seasonal-dependent work hours.

7.2 Simulation Output

To quantitatively measure the irrigation system’s performance, the following five output parameters are calculated and plotted:

- **Power Output of the Photovoltaic Panel** - The electrical power generated by the photovoltaic panel indicates the instantaneous utilization of the panel. For the direct load line approach, the power drawn from the panel can be calculated from the operating point’s voltage and current.

$$P_{PV}(t) = i(t) \cdot v(t) \tag{7.1}$$

For the maximum power point tracking approach, the power generated by the panel is equal to the panel’s maximum output power and depends on the irradiation level and the cell temperature.

$$P_{PV}(t) = P_{MPP}(t) \tag{7.2}$$

- **Daily Energy Harvested** - The main metric to determine how well the photovoltaic panel has been utilized is the amount of electrical energy it can generate over a period of time. One of the optimization goals is to harvested as much solar energy per unit power rating or unit price of the solar panel. Choosing a period of one day, the amount of energy harvested can be calculated from the photovoltaic panel’s power output $P_{PV}(t)$.

$$[\text{Daily Energy Harvested}] = \int_{day} P_{PV}(t) dt \tag{7.3}$$

- **Battery’s Charge Level** - While the battery utilization is not the main focus of this simulation, the battery’s charge level is tracked to measure how the irrigation system maintains the pump’s performance as the solar irradiation level varies. The battery’s charge level $q_{bat}(t)$ is one of the system’s state parameters.
- **Flow Rate** - Because delivering water is the goal of the irrigation system, the pump’s flow rate reflects the system’s performance at each time instance. Similar to the battery’s charge level, the flow rate $Q(t)$ is a system’s state parameter. The derivation of $Q(t)$ was discussed earlier in Chapter 4.
- **Daily Water Delivered** - The amount of water delivered by the pump is another main metric of the irrigation system’s performance. To measure whether the irrigation system meets the daily water demand of the users, the amount of water delivered in one day must be calculated from the flow rate.

$$[\text{Daily Water Delivered}] = \int_{day} Q(t)dt \quad (7.4)$$

7.3 Simulation Process

After the location and the year have been selected, strings of multiple atmospheric parameters for every hour of the year are obtained from the NREL database. Then, for each day of operation, the user preference sets the daily input variables such as the water demand and the work hours.

An hour, however, is too large to be a unit time scale for this system-level simulation because many input variables, as well as system state, may change within a hour’s time. For example, the user might turn the pump’s switch in between the hour. Moreover, as shown in Section 3.2, if the battery is directly connected to and charged by the photovoltaic panel at full power, it will become fully charged in less than 10 minutes.

Therefore, despite having only hourly data of the atmospheric input variables, each hour is divided in multiple time steps. At each time instance, the system’s operating point is calculated the system’s state parameters are updated. In this simulation,

the number of time steps in an hour is set to a default value of 10, but can still be adjusted for more accurate simulation result.

Figure 7-3 summarizes the simulation process at a single time instance, from the input variables to the output parameters. The atmospheric parameters serve as input variables to different components of the irrigation system, but all components are connected. The components interact to determine the system's operating points and state parameters, shown in the dotted box. Some output parameters are simply the state parameters that have already been calculated, while others require further calculation from the state parameters.

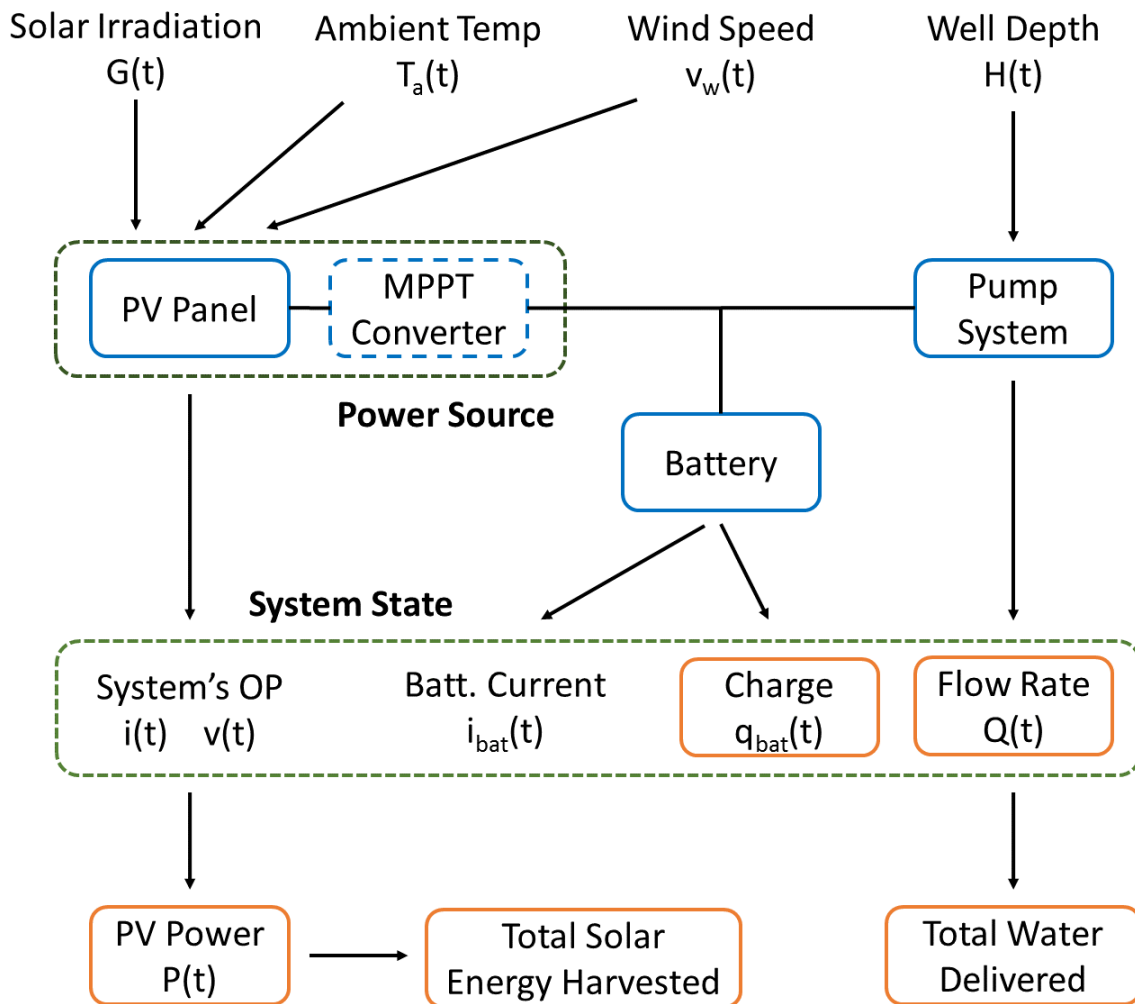


Figure 7-3: Detailed overview of the system-level theoretical simulation at a time instance t

7.4 Simulation Results

From many possible variations of the input variables, those with significant effect on the system's performance are selected and interpreted in this section. Some of the selected plots illustrate important changes to the performance as a direct result of the change in specific input variables. On the other hand, other plots are selected to demonstrate that some input variables have negligible effects on the performance.

Unless specified otherwise in the figures, the following default values are assumed for the input variables of the simulations in this section.

Input Variable	Default Value
Location	Chakradharpur, Jharkhand
Year	2010
Month	May
PV Panel	Tata Solar Power's 260W
Converter Efficiency	0.95
Battery Chemistry	LiFePO ₄ at 1.4 Ah
Battery Model	25.6 V and 0.25 Ω
Battery Quantity	1
Depth of Discharge	0.8
Water Demand	As shown in Figure 7-2
Work Hours	9am to 5pm

Table 7.2: Default values of the system-level simulation's input variables

7.4.1 Panel Rating

The power rating of the photovoltaic panel is the input variable with the greatest impact on the irrigation system's performance, as well as the greatest impact on determining whether the maximum power point tracking should be implemented. The daily energy harvested for three different power ratings of the photovoltaic panel is shown in Figure 7-4. For the irrigation system with a small photovoltaic panel, such as the 180-Watt and 260-Watt models, implementing MPPT does not significantly increase the panel's electrical power output and harvested energy because the original system has already been optimized to operate efficiently with the direct load line

approach. The battery and the pump system in this DLL system are chosen such that the system's operating point stays near the photovoltaic panel's maximum power point over the practical ranges of irradiation level and cell temperature [1]. However, in the system with a larger photovoltaic panel, the difference between the daily energy harvested of the two approaches becomes more noticeable.

Table 7.3 displays the daily energy harvested data from selected days in May 2010 for three power ratings of photovoltaic panel. For the 180-Watt panel, the change in energy harvested is even negative, due to the assumed 5% electrical power loss in the MPPT converter. On the contrary, for the 320-Watt panel, implementing the MPPT results in 26.7% increase in the daily energy harvested. This increase corresponds to the 23% increase in the panel's effective power rating from 260 W to 320 W. Therefore, the direct load line approach is no longer an effective way to utilize the photovoltaic panel in an irrigation system with high power rating, and the maximum power point tracking is recommended.

Approach	Day 5	Day 10	Day 15	Day 20	Day 25	Average
DLL	806	1082	1151	1000	1151	1038
MPPT	842	984	1143	860	1144	995
Change	4.49%	-9.09%	-0.70%	-14.00%	-0.61%	-4.18%

(a) 180-Watt Photovoltaic Panel

Approach	Day 5	Day 10	Day 15	Day 20	Day 25	Average
DLL	1089	1394	1566	1318	1545	1382
MPPT	1197	1442	1664	1431	1666	1480
Change	9.92%	3.44%	6.26%	8.57%	7.83%	7.06%

(b) 260-Watt Photovoltaic Panel

Approach	Day 5	Day 10	Day 15	Day 20	Day 25	Average
DLL	1144	1496	1655	1344	1657	1459
MPPT	1514	1799	2071	1797	2063	1848
Change	32.34%	20.25%	25.14%	33.71%	24.50%	26.70%

(c) 320-Watt Photovoltaic Panel

Table 7.3: Solar energy harvested (in Watt-hour) by the irrigation systems implementing the direct load line and the maximum power power tracking approaches, on selected days in May 2010

The system with the direct load line approach is unable to fully utilize its photovoltaic panel in the high power rating cases due to the change in panel characteristics. All of the photovoltaic panel models used in this simulation are multi-crystalline (polycrystalline) silicon, where a number of photovoltaic cells are connected in series. The 180-Watt panel model used in this simulation consists of 48 cells, while the 260-Watt and the 320-Watt models consists of 60 and 72 cells, respectively. As the power rating and the number of photovoltaic cells increase, the panel's maximum power point generally moves toward higher voltage value. Consequently, the system's operating point of the DLL approach is further from the vicinity of the panel's maximum power point and the panel's utilization decreases.

Figure 7-5 presents the total amount of water delivered by the irrigation system after each day of operation. Once the amount of water delivered reaches the daily demand for the day, the pump is assumed to be switched off. The flat portion of the daily water delivered curve corresponds to the period of days where the water demand is met.

In general, the daily water delivered curves closely resemble the daily energy harvested curves. For example, in Figure 7-4, all of the three daily energy harvested curves for the month of February experience two downward spikes around Day 9 and Day 17. Similar behavior is also exhibited by all of the three daily water delivered curves for the same month in Figure 7-5.

As expected, the irrigation system with a larger photovoltaic panel can meet the water demand more consistently than one with a smaller panel. In May 2010, while the system with the 180-Watt panel can not reach the water delivery goal on any day of the month, the system with the 320-Watt panel has no problem delivering 20 m³ per day. It is also worth noting that the amount of water supplied by the latest version of Khethworks system, with the 260-Watt photovoltaic panel, meets the demand level almost throughout the year, and is slightly lacking only for the months with higher demand like May.

Suppose the daily water target is lifted and the pump is kept operating throughout the user's work hours to maximize the amount of water delivered in each day, as

shown in Figure 7-6, the daily water delivered curves are observed to take an upward shift from their counterparts in Figure 7-5. The comparison between the DLL and MPPT approaches in Table 7.4 indicates that the benefit of implementing MPPT on the amount of water delivered increases with the irrigation system’s power rating. When the amount is not limited by the water target, the MPPT improves the daily water delivered by as much as 24.9% for the system with 320-Watt panel, but by only 1.2% and 2.7% for those with 180-Watt and 260-Watt panels, respectively.

Approach	Day 5	Day 10	Day 15	Day 20	Day 25	Average
DLL	10.93	10.93	10.93	7.56	9.54	9.99
MPPT	11.04	10.45	10.48	8.46	10.11	10.11
Change	1.01%	-4.39%	-4.12%	11.89%	5.38%	1.19%

(a) 180-Watt Photovoltaic Panel

Approach	Day 5	Day 10	Day 15	Day 20	Day 25	Average
DLL	15.78	14.83	14.05	10.79	13.45	13.78
MPPT	15.55	14.72	14.68	11.65	14.13	14.15
Change	-1.46%	-0.74%	4.48%	7.97%	5.06%	2.66%

(b) 260-Watt Photovoltaic Panel

Approach	Day 5	Day 10	Day 15	Day 20	Day 25	Average
DLL	15.74	15.10	14.79	11.29	14.06	14.20
MPPT	19.44	18.43	18.41	14.58	17.76	17.72
Change	23.51%	22.05%	24.48%	29.14%	26.32%	24.85%

(c) 320-Watt Photovoltaic Panel

Table 7.4: Water delivered (in m³) by the irrigation systems implementing the direct load line and the maximum power power tracking approaches, on selected days in November 2010

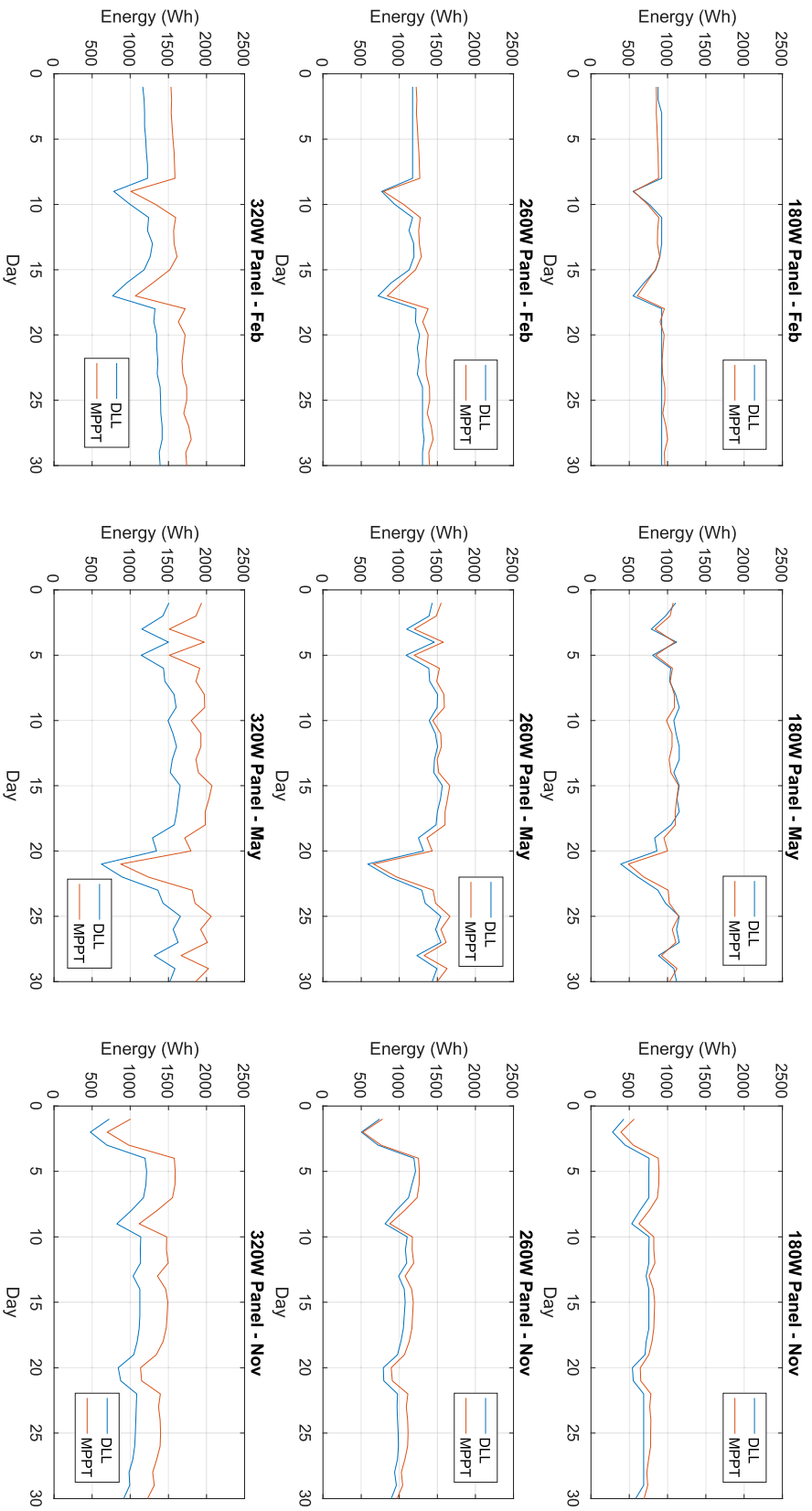


Figure 7-4: Plot of electrical energy generated in one day of operation by the irrigation system, for three different photovoltaic panel's power ratings

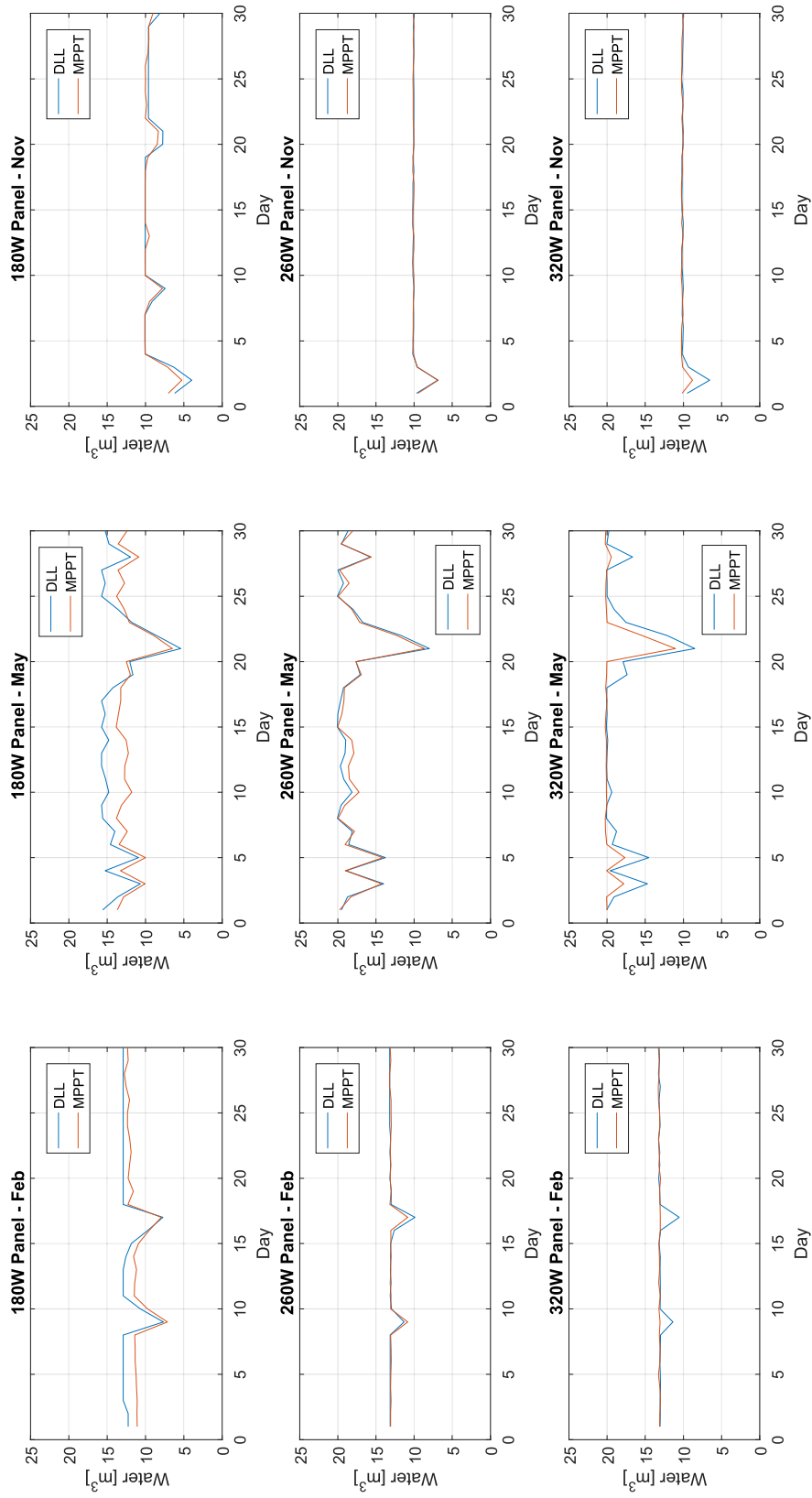


Figure 7-5: Plot of water delivered in one day of operation by the irrigation system, for three different photovoltaic panel's power ratings, with the assumption that the pump is switched off when the total amount of water delivered reaches the target value

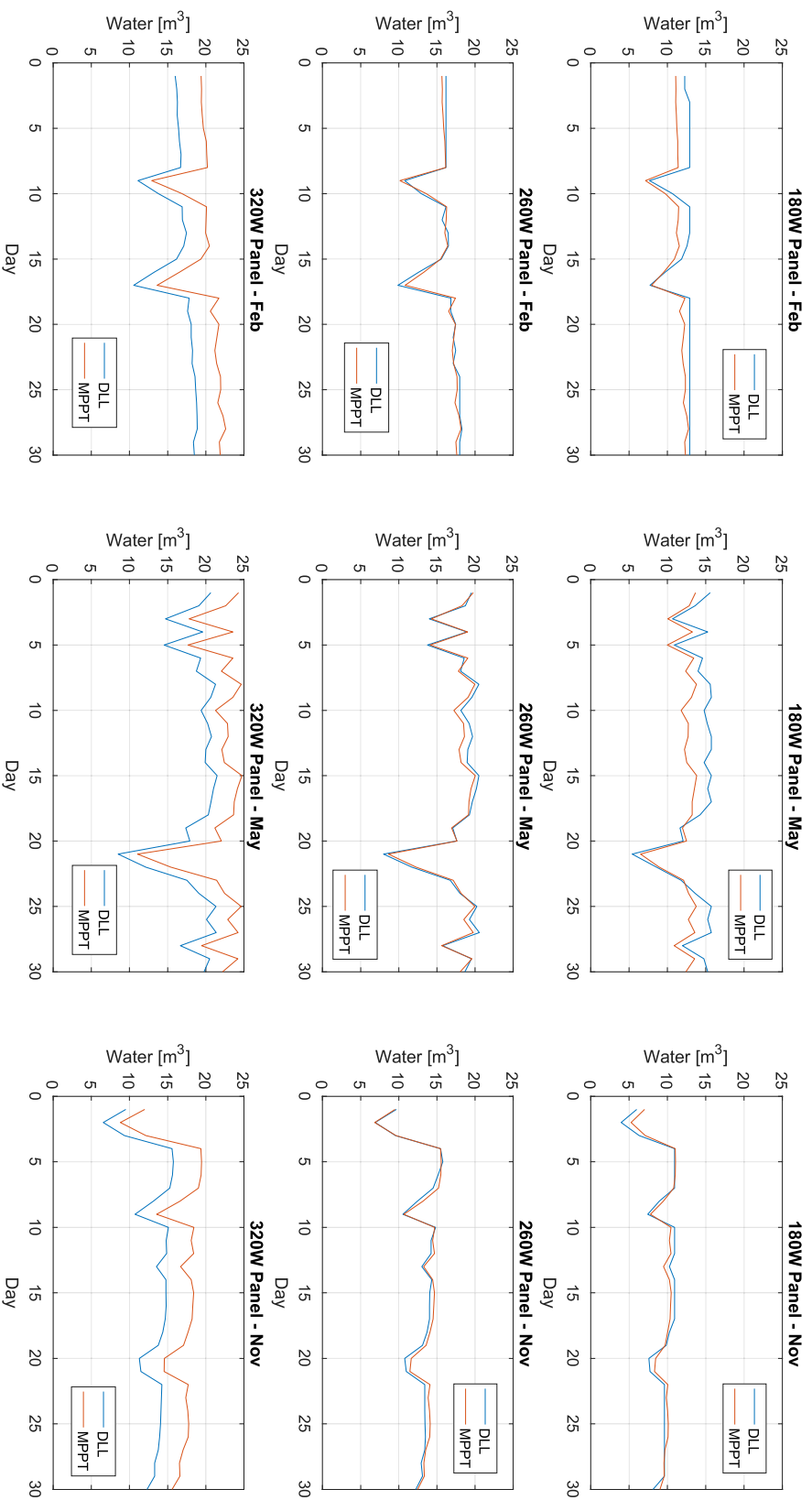


Figure 7-6: Plot of water delivered in one day of operation by the irrigation system, for three different photovoltaic panel's power ratings

7.4.2 Converter Efficiency

Other than the power rating of the photovoltaic panel, the efficiency of the MPPT converter is another critical input variable of the system-level simulation. Because of the MPPT converter's inefficiency, a portion of electrical power is dissipated as heat in this stage. The power loss reduces the amount of electrical power the pump system receives and often decreases overall performance of the irrigation system.

If the power converter in the MPPT stage is ideal and lossless, then the energy harvested by an irrigation system with MPPT approach over any period of time should always be greater than or equal to the energy harvested by a similar system with DLL approach. As shown in the left plots of Figure 7-7, the irrigation system implementing a well designed MPPT stage with 98% efficiency performs better than its DLL counterpart on both the energy harvested and the water delivered. In the default MPPT case of this simulation, where the efficiency of MPPT converter is 0.95, the amount of electrical energy generated is close to that of the system with the DLL approach.

As the MPPT converter becomes less efficient, Figure 7-7 suggests that the daily energy harvested, and consequently the amount of daily water delivered, decreases. For the case where the DLL and MPPT approaches yield comparable performance, lowering the MPPT converter's efficiency will cause the performance of the MPPT approach to become even worse than that of the DLL approach. However, it must be noted that the case with the converter efficiency of 0.9 is included in the simulation only for demonstration purpose. In practice, any power converter with efficiency coefficient as low as 0.9 should not be used to perform the maximum power point tracking. For example, if such converter supplies 250 W to the pump system, it must tolerate the heat dissipation of

$$250\text{W} \cdot \frac{1 - 0.9}{0.9} = 27.8\text{W} \quad (7.5)$$

The cost of heatsink and the risk of overheating the control board make this option unfeasible.

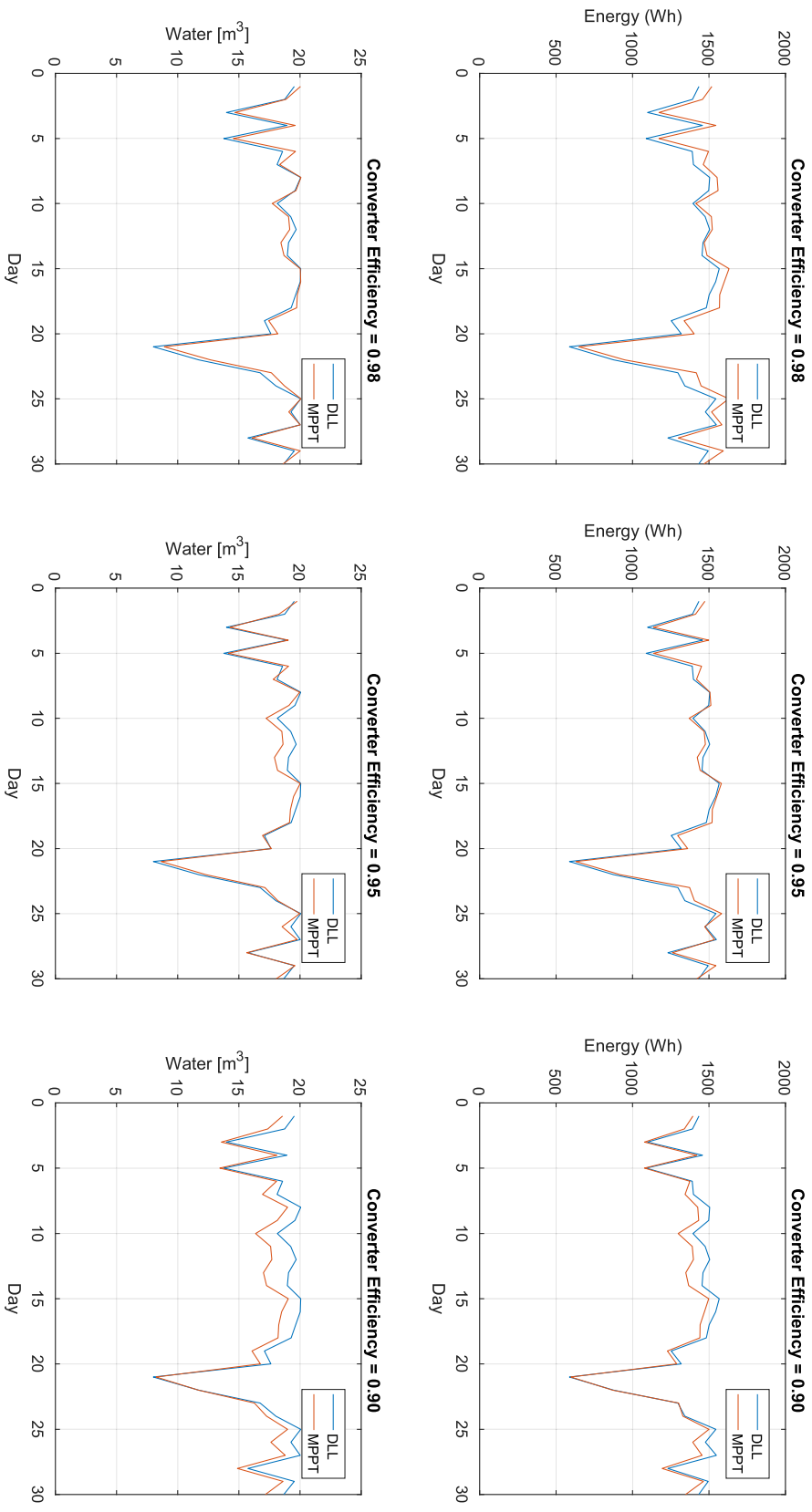


Figure 7-7: Plot of electrical energy generated and water delivered in one day of operation by the irrigation system, for three different MPPT converter's efficiency

7.4.3 Battery Quantity

In the irrigation system, the battery functions as the energy buffer between the photovoltaic panel and the pump. Adding more battery packs to the system increases the total battery capacity. Increasing the total battery capacity, however, will not significantly alter the amount of electrical energy generated by the photovoltaic panel. For the direct load line approach, the larger capacity increases the battery's capability to stabilize the system's operating point. Therefore, the power is still drawn from the photovoltaic panel around the same voltage and current levels. For the maximum power point tracking approach, the photovoltaic panel is already maintained at its maximum power point regardless of where the system's operating point is.

When the pump is switched off and the battery is already full, the opportunity to draw more electrical power from the photovoltaic panel is wasted. The larger battery capacity mitigates this problem and increases the amount of water delivered. Figure 7-8 shows that the daily water delivered increases with the number of battery packs in the system. For example, in February, the downward spikes on Day 9 and Day 17 are smaller for the system with four battery packs, indicating that the irrigation system is closer to meeting the daily water demand. Similar increase in performance is also observed from Day 5 to Day 20 in May.

Nevertheless, the benefit of increasing the number of battery packs on the amount of water delivered is relatively small compared to the benefit of implementing the MPPT discussed in the previous subsections. Even with four times the original battery capacity, the total capacity is still low - equivalent to only 30 minutes of directly powering the water pump. Therefore, adding more battery packs might not be a cost-effective option to improve the irrigation system's performance.

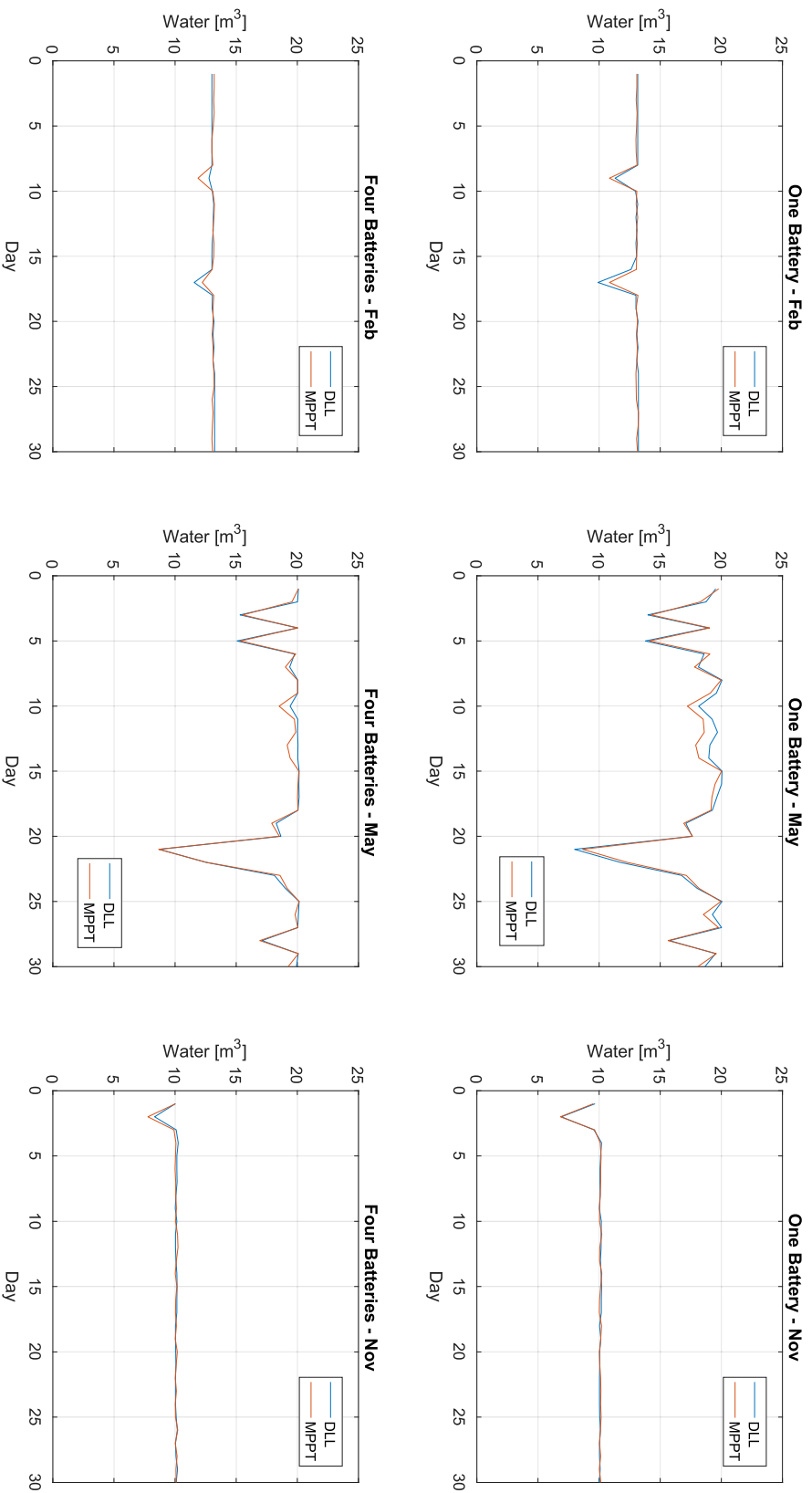


Figure 7-8: Plot of water delivered in one day of operation by the irrigation system, for the systems with one battery pack and four battery packs

7.4.4 Depth of Discharge

The effect of changing the battery's depth of discharge (DOD) on the system's performance is similar to that of changing the number of battery packs, as both changes affect the total battery capacity. For example, the total capacity of 1 battery pack with the DOD of 0.4 is equal to the capacity of 0.5 battery pack with the DOD of 0.8. Because the default value of the battery capacity is already smaller, further lowering it by decreasing the battery's depth of discharge will lower the amount of water delivered.

7.4.5 Location

The three locations chosen for this system-level simulation are in the neighboring states of Jharkhand, West Bengal, and Odisha. Due to their proximity, these states are often affected by the same weather condition. Similar trends appear not only in the atmospheric data of the three locations, but also across the system's performance parameters like the daily energy harvested and the daily water delivered. In Figure 7-9, the drops in the daily energy harvested and the daily water delivered are observed at all locations, despite the varying magnitudes.

As discussed earlier in Section 7.4.1, implementing the MPPT to the current version of the Khethworks irrigation system does not significantly increase its performance. The simulation result shown in Figure 7-9 and Figure 7-10, which correspond to the default set of the input variables, confirm that this statement holds true for all three locations. The simulation result also indicates that the irrigation system gives a consistent performance over the interested region.

7.4.6 Year

The irrigation system performs consistently not only over the variation in locations but also over the years. As shown in Figure 7-11, the system with the MPPT approach delivers approximately the same amount of water and generates slightly more electrical energy than the system with the DLL approach for all three years.

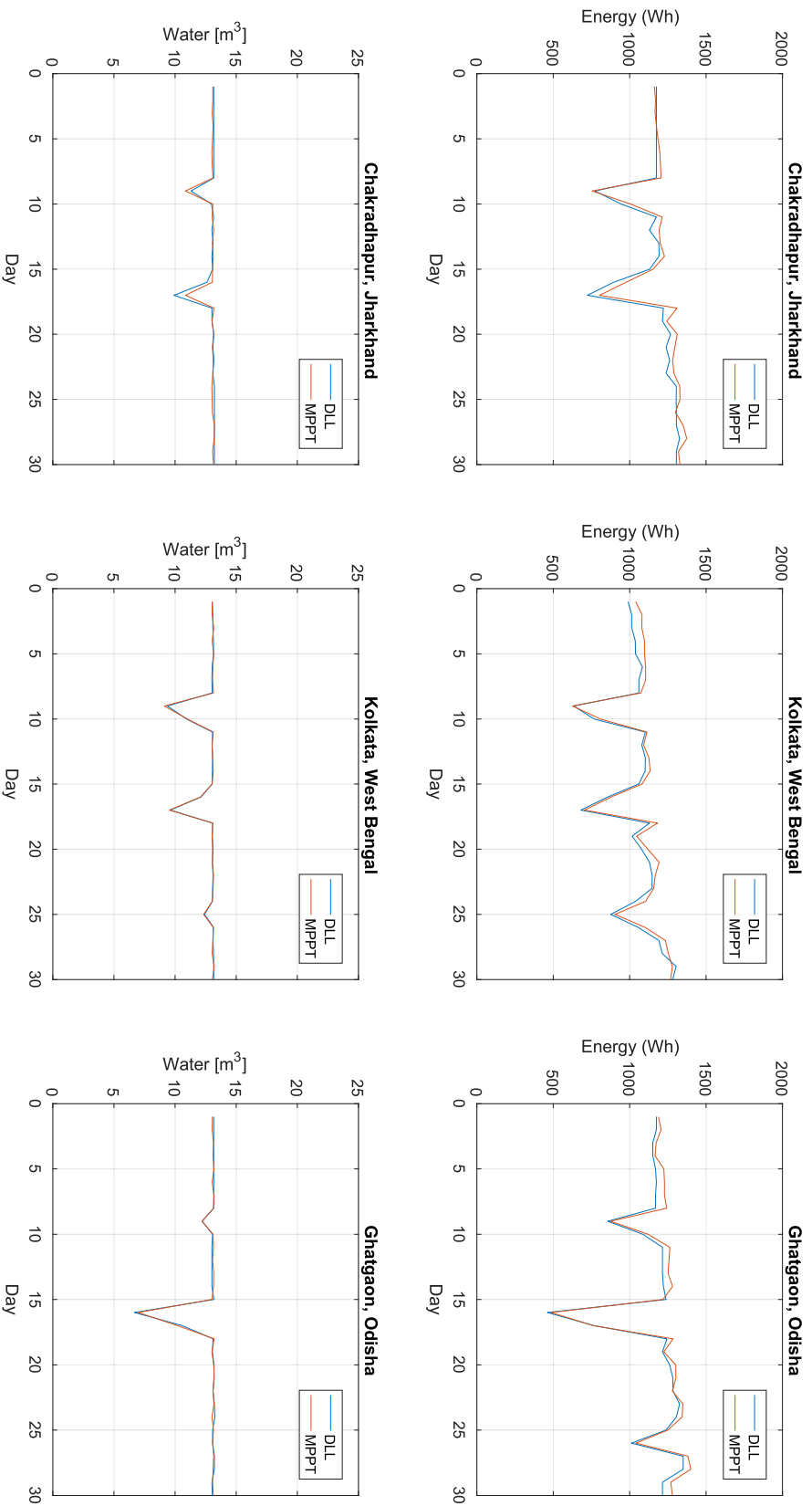


Figure 7-9: Plot of electrical energy generated and water delivered in one day of operation by the irrigation system in February 2010, for three different locations

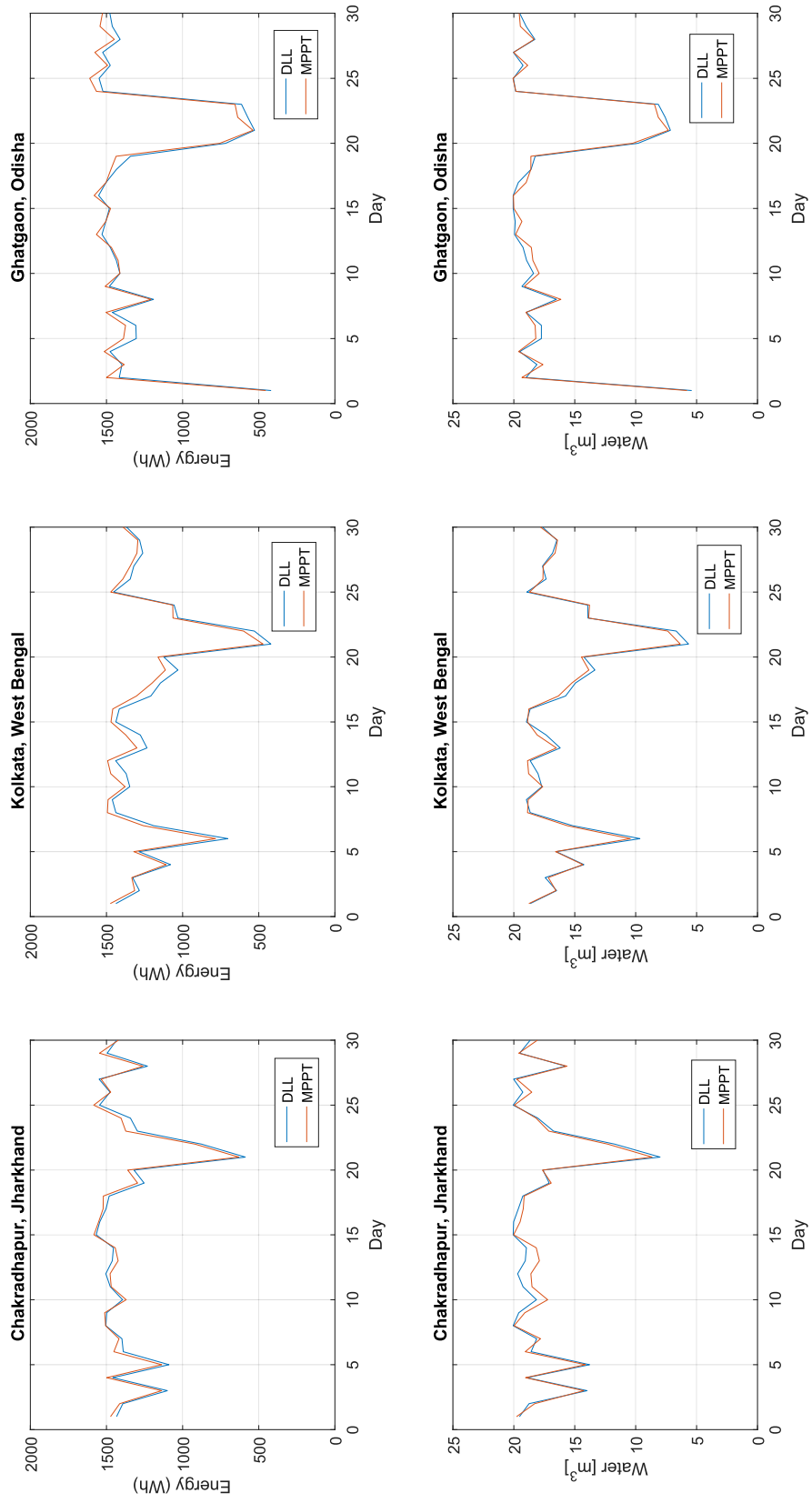


Figure 7-10: Plot of electrical energy generated and water delivered in one day of operation by the irrigation system in May 2010, for three different locations

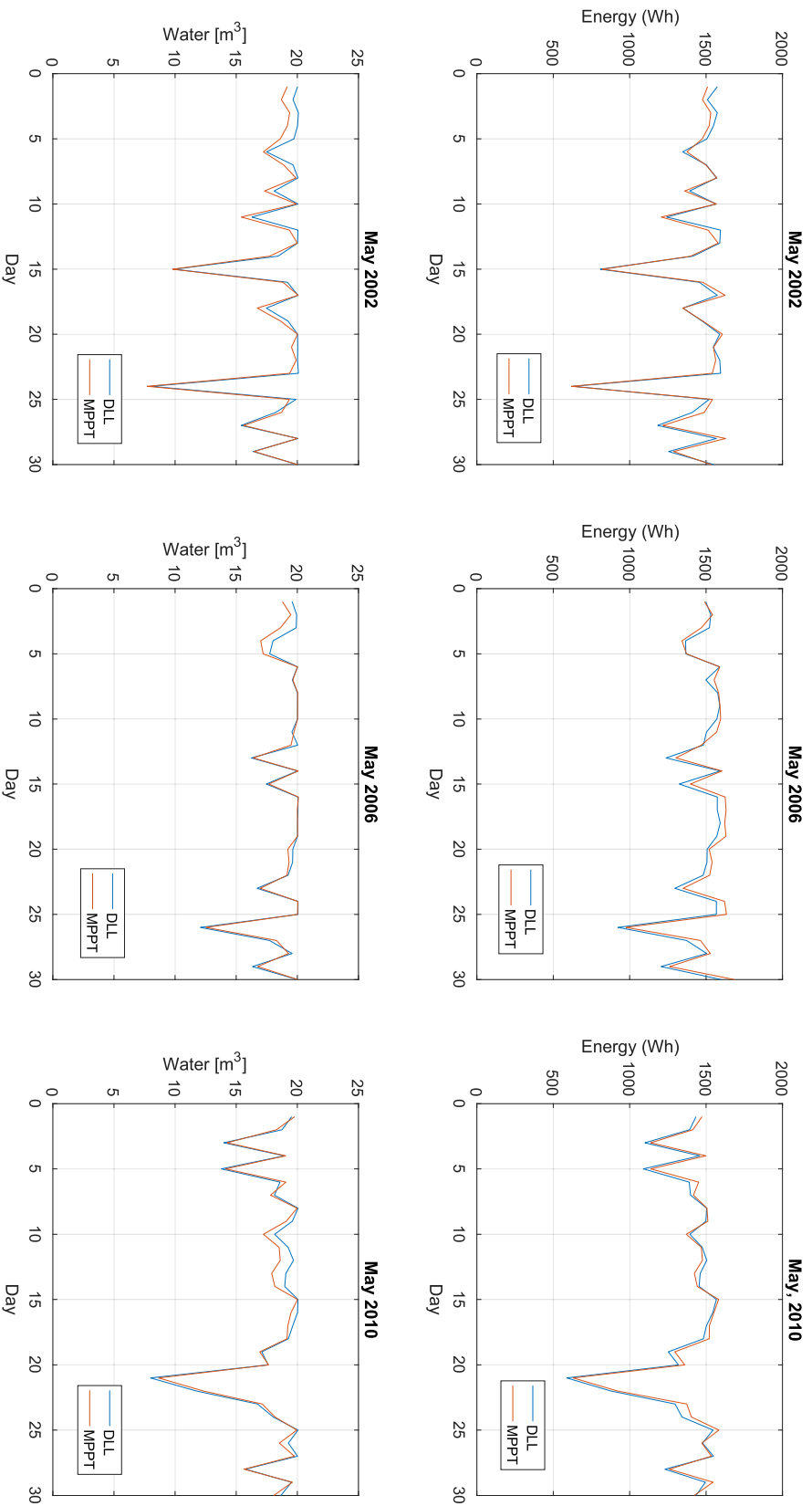


Figure 7-11: Plot of electrical energy generated and water delivered in one day of operation by the irrigation system in the month of May, for three years

7.5 Summary of Findings

- The irrigation system with high power rating benefits more from the maximum power point tracking. With the 60-cell photovoltaic panel and the pump system that are rated around 260 W, the DLL and MPPT approaches generate almost the same amount of electrical energy and a deliver similar amount of water per day.
- The MPPT converter's efficiency plays a critical role in determining the benefits of the MPPT on the irrigation system's performance. The MPPT implementation may even fail to improve the system's performance if the power converter is very inefficient.
- Increasing the total battery capacity increases the amount of water delivered by a small amount, but this increase in performance may not worth the increased battery cost.
- Changing the battery's depth of discharge is equivalent to changing the number of battery packs, as both changes directly affect the total battery capacity.
- The benefits of the MPPT approach over the DLL approach, if any, remain consistent over the irrigation system's location and the year of operation. Any finding from the simulation result is valid for all locations and years that this irrigation system is simulated.

Chapter 8

Conclusion and Future Work

Based on the simulation result previously discussed in Chapter 7 and the feedback from Khethworks' pilot program, this chapter recommends several actions which the Khethworks' engineering team may take to improve the performance of their irrigation solution.

8.1 Recommendation for Khethworks

8.1.1 Current Version of the Khethworks System

Assuming that the control circuit can regulate the switches between components well, the latest configuration of the Khethworks irrigation system will not significantly benefit from the MPPT implementation. With the photovoltaic panel's power rating of 260 W, the MPPT approach increases the daily amount of water delivered from the DLL approach by less than 5%. If there is a need to increase the irrigation system's performance by a small percentage, it is more cost-effective to increase the photovoltaic panel's power rating, as the cost of oversizing the panel is lower than the total cost of power converter, microcontroller, and other electronics components for the MPPT implementation. Thus, the direct load line approach is recommended for the current version of the Khethworks system.

The variation in the system’s power ratings may make the irrigation solution more applicable to a larger group of users and increase Khethworks’ customer base. However, given that the first version of the product is still in its pilot stage, there are several engineering tasks with higher priority than accommodating different power ratings of the photovoltaic panel. These tasks include calibrating the control board’s voltage and current sensing, optimizing the control algorithm, and improving the battery interface.

8.1.2 Optimization for the DLL Approach

For a system with the direct load line approach, battery and photovoltaic panel selection is critical to the panel’s utilization. To maximize the photovoltaic panel’s output power, the battery’s nominal voltage must be close to the panel’s maximum-power voltage. The battery chemistry that can maintain its voltage over a wide range of charge level is also preferred. Similar to the case of the latest 260-Watt Khethworks system, the battery will regulate the load line such that the voltage of the system’s operating point remains near the battery’s nominal voltage.

However, this method of regulating the system’s operating point will become less effective as time passes. The battery’s nominal voltage tends to decrease after a certain number of charge cycles, causing the system’s operating point to shift away from the photovoltaic panel’s maximum-power voltage. Moreover, the battery’s internal resistance may increase over time, limiting the battery current as well as its ability to anchor the system’s operating point [17].

8.1.3 Battery Capacity

As the new battery packs are added in parallel to the existing ones, the Thevenin resistance of the entire battery cluster decreases. Therefore, increasing the total battery capacity will increase the stability of the system’s operating point, in addition to slightly improving the system’s performance. Although a larger battery capacity is always preferred, the battery cost is a significant portion of the overall cost. Khethworks may keep the 1.4-Ah battery pack from their latest configuration, and consider increasing

the number of battery packs only if the system's performance in the field trial fails to match the simulation result.

8.1.4 Consideration for the MPPT Approach

The simulation result indicates that this version of the Khethworks irrigation system is already well optimized for operation in East India. While it is still unclear when Khethworks would expand their product lines to accommodate users in other regions, the MPPT approach remains relevant to increasing the system's power out - either by increasing the flow rate or lifting water from deeper wells.

The irrigation system with double the power rating of the current version, for example, might not operate efficiently if we simply double the number of photovoltaic panels and keep the direct load line approach. The new pump design will likely be optimized for the new values of pressure head and flow rate. Its new current-voltage characteristic, therefore, may fail to keep the system's operating point near the voltage where maximum power can be drawn from the photovoltaic panel. Implementing the MPPT will solve the mismatch problem and keep the panel's utilization at the maximum.

Appendix A

Simulation Source Code

Due to space constraints, this appendix only displays the crucial parts of the irrigation system's system-level simulation.

A.1 Simulation

A.1.1 Source File

```
1 % main function to simulate the irrigation system
2 % 3 different PV power ratings - plot at 3 different months
3 clc;
4 clear;
5
6 %% Plot variables
7 % South Jharkhand, India is Source 1 -> S1, Year is 2010 -> Y2010
8 s1y2010_name = 'radwx_086252255_2010.csv'; % Chakradhapur, Jharkhand
9 s2y2010_name = 'radwx_088352255_2010.csv'; % Kolkata, West Bengal
10 s3y2010_name = 'radwx_085852135_2010.csv'; % Ghatgaon, Odisha
11
12 timeScale = 10; % the number of time steps in an hour
13
14 batNum = 1; % number of batteries in the system
15 dod = 0.8; % the maximum depth of discharge allowed
16 % to change the target flow, edit waterDemand.m
17
18 v_conv1 = 27.99; % the initial output voltage of MPPT circuit
19 v_conv2 = 27.99;
20 v_conv3 = 27.99;
21 eff_conv1 = 0.95; % power efficiency of MPPT circuit
```

```

22 eff_conv2 = 0.95;
23 eff_conv3 = 0.95;
24
25 var_naming = strcat('all','PV_unlimited');
26 loc_naming = 'loc01';
27
28 %% Simulation Section
29 % Read from csv file
30 [solarData.GHIWm2,solarData.DNIWm2,solarData.DHIWm2,solarData.
    DrybulbC,solarData.Pressurembar,solarData.Wspdms]...
31     = readNREL(s1y2010_name, 1, 8762);
32
33 % remove the first 2 header values
34 solarData.GHIWm2 = solarData.GHIWm2(3:8762,1);
35 solarData.DNIWm2 = solarData.DNIWm2(3:8762,1);
36 solarData.DHIWm2 = solarData.DHIWm2(3:8762,1);
37 solarData.DrybulbC = solarData.DrybulbC(3:8762,1);
38 solarData.Pressurembar = solarData.Pressurembar(3:8762,1);
39 solarData.Wspdms = solarData.Wspdms(3:8762,1);
40
41 % Temperature Model
42 % Variables from Datasheet
43 T_STC = 298;
44 eta_STC = 0.156;
45 beta_STC = -0.004048;
46 T_aNOCT = 293;
47 T_NOCT = 320;
48 G_NOCT = 800;
49 v_wNOCT = 1;
50 % Variable from NREL
51 vec_G = solarData.GHIWm2;
52 vec_v_w = solarData.Wspdms;
53 vec_T_a = solarData.DrybulbC; % Celcius
54 % Variable from Literature
55 tau_alpha = 0.9;
56 vec_h_w = 5.9 + 2.8*vec_v_w;
57 h_wNOCT = 5.9 + 2.8*v_wNOCT;
58
59 % Calculating cell temperature in Celcius
60 vec_T_c = vec_T_a + vec_G./G_NOCT*(T_NOCT-T_aNOCT)*h_wNOCT./vec_h_w
    *(1-eta_STC*(1-beta_STC*T_STC)/tau_alpha);
61
62 % Battery Parameters
63 rbat_single = 0.25; % [Ohm] battery internal resistance of one pack
64 batCap_single = 1.4; % [Ah] battery capacity of one pack
65
66 batCap = batNum*batCap_single; % [Ah] total battery capacity
67 rbat = rbat_single/batNum; % [Ohm] battery equivalent resistance
68
69 % Time Parameters
70 % yearSpan is usually 365 and daySpan is usually 24*timeScale
71 yearSpan = 365;
72 daySpan = timeScale*24;

```



```

73 dt = 1/timeScale; % [hour] the number of seconds per time step
74
75 % #1-1 Direct Load Line
76 [vec_P_DLL1,vec_E_DLL1,vec_qbat_percent_DLL1,vec_water_DLL1,
    vec_flow_DLL1]...
77     = runDLL180(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat);
78
79 % #1-2 Direct Load Line
80 [vec_P_DLL2,vec_E_DLL2,vec_qbat_percent_DLL2,vec_water_DLL2,
    vec_flow_DLL2]...
81     = runDLL260(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat);
82
83 % #1-3 Direct Load Line
84 [vec_P_DLL3,vec_E_DLL3,vec_qbat_percent_DLL3,vec_water_DLL3,
    vec_flow_DLL3]...
85     = runDLL320(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat);
86
87 % #2-1 Maximum Power Point Tracking
88 [vec_P_MPPT1,vec_E_MPPT1,vec_qbat_percent_MPPT1,vec_water_MPPT1,
    vec_flow_MPPT1]...
89     = runMPPT180(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat,
    v_conv1,eff_conv1);
90
91 % #2-2 Maximum Power Point Tracking
92 [vec_P_MPPT2,vec_E_MPPT2,vec_qbat_percent_MPPT2,vec_water_MPPT2,
    vec_flow_MPPT2]...
93     = runMPPT260(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat,
    v_conv2,eff_conv2);
94
95 % #2-3 Maximum Power Point Tracking
96 [vec_P_MPPT3,vec_E_MPPT3,vec_qbat_percent_MPPT3,vec_water_MPPT3,
    vec_flow_MPPT3]...
97     = runMPPT320(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat,
    v_conv3,eff_conv3);

```

Listing A.1: Source file of the system-level simulation

A.1.2 Plotting Script

```

1 % Plot Section - Daily Water Delivered
2 x = (1:30); % plot for 30 days
3 figure
4
5 % Case 1 - Feb
6 subplot(3,3,1);
7 xStartDay = 31;
8 plot(x,vec_water_DLL1(xStartDay+1:xStartDay+30));
9 hold on
10 plot(x,vec_water_MPPT1(xStartDay+1:xStartDay+30));
11 title('180W Panel - Feb');
12 axis([0 30 0 25]);
13 legend('DLL','MPPT');

```

```

14 xlabel('Day'); ylabel('Water [m^3]');
15 grid on
16
17 % Case 1 - May
18 subplot(3,3,2);
19 xStartDay = 120;
20 plot(x,vec_water_DLL1(xStartDay+1:xStartDay+30));
21 hold on
22 plot(x,vec_water_MPPT1(xStartDay+1:xStartDay+30));
23 title('180W Panel - May');
24 axis([0 30 0 25]);
25 legend('DLL','MPPT');
26 xlabel('Day'); ylabel('Water [m^3]');
27 grid on
28
29 % Case 1 - Nov
30 subplot(3,3,3);
31 xStartDay = 303;
32 plot(x,vec_water_DLL1(xStartDay+1:xStartDay+30));
33 hold on
34 plot(x,vec_water_MPPT1(xStartDay+1:xStartDay+30));
35 title('180W Panel - Nov');
36 axis([0 30 0 25]);
37 legend('DLL','MPPT');
38 xlabel('Day'); ylabel('Water [m^3]');
39 grid on
40
41 % Case 2 - Feb
42 subplot(3,3,4);
43 xStartDay = 31;
44 plot(x,vec_water_DLL2(xStartDay+1:xStartDay+30));
45 hold on
46 plot(x,vec_water_MPPT2(xStartDay+1:xStartDay+30));
47 title('260W Panel - Feb');
48 axis([0 30 0 25]);
49 legend('DLL','MPPT');
50 xlabel('Day'); ylabel('Water [m^3]');
51 grid on
52
53 % Case 2 - May
54 subplot(3,3,5);
55 xStartDay = 120;
56 plot(x,vec_water_DLL2(xStartDay+1:xStartDay+30));
57 hold on
58 plot(x,vec_water_MPPT2(xStartDay+1:xStartDay+30));
59 title('260W Panel - May');
60 axis([0 30 0 25]);
61 legend('DLL','MPPT');
62 xlabel('Day'); ylabel('Water [m^3]');
63 grid on
64
65 % Case 2 - Nov
66 subplot(3,3,6);

```

```

67 xStartDay = 303;
68 plot(x,vec_water_DLL2(xStartDay+1:xStartDay+30));
69 hold on
70 plot(x,vec_water_MPPT2(xStartDay+1:xStartDay+30));
71 %title('Case 2 Daily Water Delivered - Nov');
72 axis([0 30 0 25]);
73 legend('DLL','MPPT');
74 xlabel('Day'); ylabel('Water [m^3]');
75 grid on
76
77 % Case 3 - Feb
78 subplot(3,3,7);
79 xStartDay = 31;
80 plot(x,vec_water_DLL3(xStartDay+1:xStartDay+30));
81 hold on
82 plot(x,vec_water_MPPT3(xStartDay+1:xStartDay+30));
83 %title('Case 3 Daily Water Delivered - Feb');
84 axis([0 30 0 25]);
85 legend('DLL','MPPT');
86 xlabel('Day'); ylabel('Water [m^3]');
87 grid on
88
89 % Case 3 - May
90 subplot(3,3,8);
91 xStartDay = 120;
92 plot(x,vec_water_DLL3(xStartDay+1:xStartDay+30));
93 hold on
94 plot(x,vec_water_MPPT3(xStartDay+1:xStartDay+30));
95 %title('Case 3 Daily Water Delivered - May');
96 title('320W Panel - May');
97 axis([0 30 0 25]);
98 legend('DLL','MPPT');
99 xlabel('Day'); ylabel('Water [m^3]');
100 grid on
101
102 % Case 3 - Nov
103 subplot(3,3,9);
104 xStartDay = 303;
105 plot(x,vec_water_DLL3(xStartDay+1:xStartDay+30));
106 hold on
107 plot(x,vec_water_MPPT3(xStartDay+1:xStartDay+30));
108 title('320W Panel - Nov');
109 axis([0 30 0 25]);
110 legend('DLL','MPPT');
111 xlabel('Day'); ylabel('Water [m^3]');
112 grid on

```

Listing A.2: Example of the system's performance plotting script

A.2 Core Functions

A.2.1 DLL Simulator

```
1 % runDLL260 - a function to simulate Direct Load Line system
2 % PV rating is 260
3
4 function [vec_P_DLL,vec_E_DLL,vec_qbat_percent,vec_water,vec_flow
5 ]...
6     = runDLL260(vec_G,vec_T_c,timeScale,batNum,dod,batCap,rbat)
7     % PV Rating
8     PV_rating = 260;
9     % time parameters
10    yearSpan = 365;
11    % timeScale is given
12    daySpan = timeScale*24;
13    dt = 1/timeScale; % [hour] the fraction of hour per time step
14
15    % create size 2d of the new loop vector, call it matSpan (matrix
16    span)
17    matSize = [yearSpan daySpan]; % size of data matrix
18
19    % call depth fucntion to get daily depth vec 365x1
20    % approximate that the well head is a constant value for now
21    vec_H = ones(yearSpan,1)*10*1000*9.807; % [Pa] from 10m * 1000kg
22    /m^3 * 9.807m/s
23    % call waterDemand fuction to get daily targetFlow vec 365x1
24    vec_targetFlow = waterDemand(yearSpan);
25
26    % battery parameter definition and initialization
27    % dod is given
28    hystMax = 1;
29    hystMin = 1 - dod;
30    % batNum is given
31    % batCap is given
32    % rbat is given
33
34    mat_ibat = zeros(matSize);
35    mat_qbat = zeros(matSize); % TOTAL charge in the batteries
36    mat_qbat(1,1) = batCap*(hystMin+hystMax)/2; % qbat initialized
37    at minimum level
38
39    % resize vec_G, vec_T_a, vec_T_c to [365 24*timeScale]
40    mat_G = customReshape(vec_G,timeScale);
41    mat_T_c = customReshape(vec_T_c,timeScale);
42
43    % initialize zeros vec that will be used later, 365x[24*
44    timescale]
45    vec_water = zeros(yearSpan,1);
46
```

```

42 % initialize zeros mats to record simulation results, 365x[24*
timescale]
43 mat_v = zeros(matSize); % v for operating point voltage
44 mat_i = zeros(matSize); % i for PV current
45 mat_state = zeros(matSize); % motor state (0 off 1 on)
46 mat_charging = zeros(matSize); % battery state (0 dis 1
charging)
47 mat_flow = zeros(matSize);
48
49 % initialize operating voltage
50 v_prev = 28; % [V] system OP
51
52 for day = 1:yearSpan
53     daily_H = vec_H(day); % head of the water well used for the
day
54     daily_targetFlow = vec_targetFlow(day); % daily target flow
55
56     water = 0; % [m^3] % initialize total amount of water
delivered for the day
57     motorOn = 0; % initialize the motor state
58
59     startHour = 8; % first hour to turn on the motor - 8am
60     lastHour = 18; % last hour to still have motor on - 6pm
61
62     for tick = 1:daySpan
63         % retrieving values
64         G = mat_G(day,tick); % irradiation at the tick
65         T_c = mat_T_c(day,tick); % cell temp at the tick
66         qbat = mat_qbat(day,tick);
67
68         atWork = (startHour*timeScale < tick) && (tick <=
lastHour*timeScale);
69         motorON = (water < daily_targetFlow) && atWork;
70         Q = 0;
71
72         charging = mat_charging(day,tick);
73         vbat = batteryCurve(charging,qbat/batNum);
74
75         % if water delivered so far < day target and after 9am
and before 6pm
76         if motorON % motor ON
77             if G > 50
78                 % calculate system op with func 1 - all
connected
79                 [i,v,S,Q,Imotor] = findOP_DLL1(PV_rating,G,T_c,
vbat,rbat,v_prev,daily_H);
80                 ibat = i - Imotor;
81                 if qbat == batCap*hystMax && i > Imotor
82                     ibat = 0;
83                 elseif qbat == batCap*hystMin && i < Imotor
84                     % Approximate that same efficiency, same v
85                     % but less Imotor for less Q
86                     ibat = 0;

```

```

87         Q = Q*i/Imotor;
88     end
89     % update variables
90     v_prev = v;
91     charging = (i > Imotor);
92     % record variables
93     mat_v(day,tick) = v;
94     mat_i(day,tick) = i;
95     mat_ibat(day,tick) = ibat;
96     mat_state(day,tick) = S;
97     mat_flow(day,tick) = Q;
98     else % G = 0
99         % calculate system op with func 3 - motor ON and
battery
100         if qbat == batCap*hystMin
101             v = v_prev;
102             S = 0; Q = 0; Imotor = 0;
103         else
104             [v,S,Q,Imotor] = findOP_DLL3(PV_rating,vbat,
rbat,v_prev,daily_H);
105         end
106         i = 0;
107         % update variables
108         v_prev = v;
109         charging = 0;
110         % record variables
111         mat_v(day,tick) = v;
112         mat_i(day,tick) = i;
113         mat_ibat(day,tick) = i-Imotor;
114         mat_state(day,tick) = S;
115         mat_flow(day,tick) = Q;
116     end
117
118     % update water delivered because motor is ON
119     water = water + Q*3600/timeScale;
120     % update qbat and record charging state
121     qbat_new = qbat + (i-Imotor)*dt;
122     if qbat_new > batCap*hystMax
123         qbat_new = batCap*hystMax;
124     elseif qbat_new < batCap*hystMin
125         qbat_new = batCap*hystMin;
126     end
127
128     if tick < daySpan
129         mat_qbat(day,tick+1) = qbat_new;
130         mat_charging(day,tick+1) = charging;
131     else % tick == daySpan
132         if day < yearSpan
133             mat_qbat(day+1,1) = qbat_new;
134             mat_charging(day+1:1) = charging;
135         end
136     end
137

```

```

138         else % motor OFF
139             if G > 0
140                 % calculate system op with func 2 - PV panel and
battery
141                 [i,v] = findOP_DLL2(PV_rating,G,T_c,vbat,rbat,
v_prev);
142                 % update variables
143                 v_prev = v;
144                 charging = 1;
145                 % record variables
146                 mat_v(day,tick) = v;
147                 mat_i(day,tick) = i;
148                 mat_ibat(day,tick) = i;
149                 mat_state(day,tick) = 0; % motor is OFF
150                 mat_flow(day,tick) = 0;
151             else % G = 0
152                 i = 0; v = vbat;
153                 v_prev = v;
154                 charging = 0;
155                 % record variables
156                 mat_v(day,tick) = v;
157                 mat_i(day,tick) = 0;
158                 mat_ibat(day,tick) = 0;
159                 mat_state(day,tick) = 0; % motor is OFF
160                 mat_flow(day,tick) = 0;
161             end
162
163             % no change in water delivered because motor is OFF
164             % update charge and record charging state
165             qbat_new = qbat + i*dt; % increase or unchanged
166             if qbat_new > batCap*hystMax
167                 qbat_new = batCap*hystMax;
168             end
169
170             if tick < daySpan
171                 mat_qbat(day,tick+1) = qbat_new;
172                 mat_charging(day,tick+1) = charging;
173             else % tick == daySpan
174                 if day < yearSpan
175                     mat_qbat(day+1,1) = qbat_new;
176                     mat_charging(day+1:1) = charging;
177                 end
178             end
179
180         end
181     end
182
183     % update and record at the end of the day
184     vec_water(day) = water;
185 end
186
187 mat_P_DLL = mat_v .* mat_i; % power drawn from PV panel
188 mat_qbat_percent = mat_qbat ./ batCap;

```

```

189
190 % Reformating and plotting
191 vec_P_DLL = reshape(transpose(mat_P_DLL), [yearSpan*daySpan, 1]);
192 vec_qbat_percent = reshape(transpose(mat_qbat_percent), [yearSpan
*daySpan, 1]);
193 vec_flow = reshape(transpose(mat_flow), [yearSpan*daySpan, 1]);
194
195 % calculate daily energy harvest - to something with vec_Ppv
196 vec_E_DLL = sum(mat_P_DLL, 2)/timeScale; % [Wh]
197 end

```

Listing A.3: Simulator for an irrigation system with the direct load line approach and 260-W photovoltaic panel

A.2.2 MPPT Simulator

```

1 % runMPPT260 - a function to simulate Maximum Power Point Tracking
  system
2 % PV rating is 260
3
4 function [vec_P_MPPT, vec_E_MPPT, vec_qbat_percent, vec_water, vec_flow
  ]...
5     = runMPPT260(vec_G, vec_T_c, timeScale, batNum, dod, batCap, rbat,
  v_conv, eff_conv)
6     % Calculating maximum power drawn for MPPT case
7     MPPTeff = 1.0; % The efficiency of MPPT converter
8
9     % PV rating
10    P0 = 260; % [W] as rated by the datasheet
11    T0 = 25; % [Celcius]
12    G0 = 1000; % [W/m^2]
13
14    % using A+BT model, with T in Celcius
15    A_temp = 269.847; % [W]
16    B_temp = -0.9750; % [W/Celcius]
17    % using A+BT model, with G in W/m^2
18    A_irr = -5.107; % [W]
19    B_irr = 0.2536; % [m^2]
20
21    % calculate power obtainable from PV, regardless of system state
22    vec_P_MPPT_max = MPPTeff*P0*(A_temp + B_temp.*vec_T_c)/(A_temp +
  B_temp*T0)...
23    .*(A_irr + B_irr.*vec_G)/(A_irr + B_irr*G0);
24    % if entry < 0 then entry = 0;
25    vecSize = size(vec_P_MPPT_max);
26    for x = 1:vecSize(1)
27        if vec_P_MPPT_max(x) < 0
28            vec_P_MPPT_max(x) = 0;
29        end
30    end
31
32    % time parameters

```



```

33 yearSpan = 365;
34 % timeScale is given
35 daySpan = timeScale*24;
36 dt = 1/timeScale; % [hour] the number of seconds per time step
37
38 % create size 2d of the new loop vector, call it matSpan (matrix
span)
39 matSize = [yearSpan daySpan]; % size of data matrix
40
41 % call depth fucntion to get daily depth vec 365x1
42 % approximate that the well head is a constant value for now
43 vec_H = ones(yearSpan,1)*10*1000*9.807; % [Pa] from 10m * 1000kg
/m^3 * 9.807m/s
44 % call waterDemand fuction to get daily targetFlow vec 365x1
45 vec_targetFlow = waterDemand(yearSpan);
46
47 % battery parameter definition and initialization
48 % dod is given
49 hystMax = 1;
50 hystMin = 1 - dod;
51 % batNum is given
52 % batCap is given
53 % rbat is given
54
55 mat_ibat = zeros(matSize);
56 mat_qbat = zeros(matSize); % TOTAL charge in the batteries
57 mat_qbat(1,1) = batCap*(hystMin+hystMax)/2; % qbat initialized
at minimum level
58
59 % resize vec_G, vec_T_a, vec_T_c to [365 24*timeScale]
60 mat_G = customReshape(vec_G,timeScale);
61 mat_T_c = customReshape(vec_T_c,timeScale);
62 mat_P_MPPT_max = customReshape(vec_P_MPPT_max,timeScale);
63
64 % initialize zeros vec that will be used later, 365x[24*
timescale]
65 vec_water = zeros(yearSpan,1);
66
67 % initialize zeros mats to record simulation results, 365x[24*
timescale]
68 mat_v = zeros(matSize); % v for operating point voltage
69 mat_i = zeros(matSize); % i for PV current
70 mat_state = zeros(matSize); % motor state (0 off 1 on)
71 mat_charging = zeros(matSize); % battery state (0 dis 1
charging)
72 mat_P_MPPT = zeros(matSize); % actual power drawn from PV panel
73 mat_flow = zeros(matSize);
74
75 for day = 1:yearSpan
76     daily_H = vec_H(day); % head of the water well used for the
day
77     daily_targetFlow = vec_targetFlow(day); % daily target flow
78

```

```

79     water = 0; % [m^3] % initialize total amount of water
delivered for the day
80     motorOn = 0; % initialize the motor state
81
82     startHour = 8; % first hour to turn on the motor - 8am
83     lastHour = 18; % last hour to still have motor on - 6pm
84
85     for tick = 1:daySpan
86         % retrieving values
87         G = mat_G(day,tick); % irradiation at the tick
88         T_c = mat_T_c(day,tick); % cell temp at the tick
89         qbat = mat_qbat(day,tick);
90         P_MPPT_max = mat_P_MPPT_max(day,tick);
91
92         atWork = (startHour*timeScale < tick) && (tick <=
lastHour*timeScale);
93         motorON = (water < daily_targetFlow) && atWork;
94         Q = 0;
95
96         charging = mat_charging(day,tick);
97         vbat = batteryCurve(charging,qbat/batNum);
98
99         % if water delivered so far < day target and after 9am
and before 6pm
100        if motorON % motor ON
101            if G > 0
102                % find current into the motor
103                S = motorState(v_conv);
104                [Q,eff] = pumpCurve_v1(S,daily_H);
105                Imotor = Q*daily_H / (eff*v_conv);
106                i = eff_conv * P_MPPT_max / v_conv;
107                v = v_conv;
108                ibat = i - Imotor;
109                % deal with batt Min and Max cases
110                if qbat == batCap*hystMin && i < Imotor
111                    % Q drops propotional to i
112                    Q = Q*i/Imotor;
113                    ibat = 0;
114                elseif qbat == batCap*hystMax && i > Imotor
115                    % battery charge does not increase
116                    ibat = 0;
117                end
118                % update variables
119                v_prev = v;
120                charging = (i > Imotor);
121                % record variables
122                mat_v(day,tick) = v;
123                mat_i(day,tick) = i;
124                mat_ibat(day,tick) = ibat;
125                mat_state(day,tick) = S;
126                mat_P_MPPT(day,tick) = P_MPPT_max;
127                mat_flow(day,tick) = Q;
128            else % G = 0

```

```

129         % power motor with only battery, like DLL
130         if qbat == batCap*hystMin
131             v = v_prev;
132             S = 0; Q = 0; Imotor = 0;
133         else % we use findOP_DLL3
134             [v,S,Q,Imotor] = findOP_DLL3(vbat,rbat,
v_prev,daily_H);
135         end
136         % update variables
137         i = 0; v_prev = v; charging = 0;
138         % record variables
139         mat_v(day,tick) = v;
140         mat_i(day,tick) = i;
141         mat_ibat(day,tick) = i-Imotor;
142         mat_state(day,tick) = S;
143         mat_P_MPPT(day,tick) = 0;
144         mat_flow(day,tick) = Q;
145     end
146
147     % update water delivered because motor is ON
148     water = water + Q*3600/timeScale;
149     % update qbat and record charging state
150     qbat_new = qbat + (i-Imotor)*dt;
151     if qbat_new > batCap*hystMax
152         qbat_new = batCap*hystMax;
153     elseif qbat_new < batCap*hystMin
154         qbat_new = batCap*hystMin;
155     end
156
157     if tick < daySpan
158         mat_qbat(day,tick+1) = qbat_new;
159         mat_charging(day,tick+1) = charging;
160     else % tick == daySpan
161         if day < yearSpan
162             mat_qbat(day+1,1) = qbat_new;
163             mat_charging(day+1:1) = charging;
164         end
165     end
166
167     else % motor OFF
168         if G > 50 % should have been zero but just in case
169             % charge battery right from v_conv
170             ibat = (v_conv - vbat)/rbat;
171             i = ibat; v = v_conv;
172             charging = 1;
173             if i*v > eff_conv*P_MPPT_max
174                 vec_roots = roots([1 -vbat -1*eff_conv*
P_MPPT_max*rbat]);
175                 v = vec_roots(1);
176                 i = (v - vbat)/rbat;
177                 P_MPPT = P_MPPT_max;
178             else % i*v <= eff_MPPT*P_MPPT_max
179                 P_MPPT = i*v;

```

```

180         end
181         % update variable
182         v_prev = v;
183         % record variables
184         mat_v(day,tick) = v;
185         mat_i(day,tick) = i;
186         mat_ibat(day,tick) = i;
187         mat_state(day,tick) = 0; % motor is OFF
188         mat_P_MPPT(day,tick) = P_MPPT;
189         mat_flow(day,tick) = 0;
190     else % G = 0
191         % nothing happens
192         i = 0; v = vbat; charging = 0;
193         v_prev = v;
194         % record variables
195         mat_v(day,tick) = v;
196         mat_i(day,tick) = 0;
197         mat_ibat(day,tick) = 0;
198         mat_state(day,tick) = 0; % motor is OFF
199         mat_P_MPPT(day,tick) = 0;
200         mat_flow(day,tick) = 0;
201     end
202
203     % no change in water delivered because motor is OFF
204     % update charge and record charging state
205     qbat_new = qbat + i*dt; % increase or unchanged
206     if qbat_new > batCap*hystMax
207         qbat_new = batCap*hystMax;
208     end
209
210     if tick < daySpan
211         mat_qbat(day,tick+1) = qbat_new;
212         mat_charging(day,tick+1) = charging;
213     else % tick == daySpan
214         if day < yearSpan
215             mat_qbat(day+1,1) = qbat_new;
216             mat_charging(day+1:1) = charging;
217         end
218     end
219
220     end
221 end
222
223 % update and record at the end of the day
224 vec_water(day) = water;
225 end
226
227 mat_qbat_percent = mat_qbat ./ batCap;
228
229 % Reformating and plotting
230 vec_P_MPPT = reshape(transpose(mat_P_MPPT), [yearSpan*daySpan, 1])
;
231 vec_qbat_percent = reshape(transpose(mat_qbat_percent), [yearSpan

```

```

232     *daySpan,1]);
233     vec_flow = reshape(transpose(mat_flow),[yearSpan*daySpan,1]);
234     % calculate daily energy harvest - to something with vec_Ppv
235     vec_E_MPPT = sum(mat_P_MPPT,2)/timeScale; % [Wh]
236 end

```

Listing A.4: Simulator for an irrigation system with the maximum power point tracking approach and 260-W photovoltaic panel

A.3 Helper Functions

A.3.1 DLL Operating Point 1

```

1 % findOP_DLL1 - a function to calculate system op 1
2 % Condition: motor is ON and G > 0
3
4 function [i,v,S,Q,Imotor] = findOP_DLL1(PV_rating,G,T,vbat,r,v_prev,
5     H)
6     % initialize
7     i1 = 0; i2 = 0; i3 = 0;
8     v1 = v_prev; v2 = v_prev; v3 = v_prev;
9     ite = 10;
10    S_prev = motorState(v_prev);
11
12    S = zeros(1,3);
13    valid = zeros(1,3);
14    v_array = zeros(1,3); i_array = zeros(1,3);
15    Q_array = zeros(1,3); Imotor_array = zeros(1,3);
16
17    S(1) = 1; % for S = 1
18    [Q1,eff1] = pumpCurve_v1(S(1),H);
19    % iteration method
20    for n = 1:ite
21        inew1 = pvCurve(PV_rating,G,T,v1);
22        vnew1 = vbat + (inew1 - (Q1*H)/(eff1*v1))*r;
23        if abs(vnew1 - v1) < 0.005
24            i1 = inew1; v1 = vnew1;
25            break
26        end
27        v1 = vnew1;
28    end
29    v_array(1) = v1; i_array(1) = i1;
30    Q_array(1) = Q1; Imotor_array(1) = Q1*H/(eff1*v1);
31    if S(1) == motorState(v1)
32        valid(1) = 1;
33    end
34
35    S(2) = 2; % for S = 2
36    [Q2,eff2] = pumpCurve_v1(S(2),H);

```

```

36 % iteration method
37 for n = 1:ite
38     inew2 = pvCurve(PV_rating,G,T,v2);
39     vnew2 = vbat + (inew2 - (Q2*H)/(eff2*v2))*r;
40     if abs(vnew2 - v2) < 0.005
41         i2 = inew2; v2 = vnew2;
42         break
43     end
44     v2 = vnew2;
45 end
46 v_array(2) = v2; i_array(2) = i2;
47 Q_array(2) = Q2; Imotor_array(2) = Q2*H/(eff2*v2);
48 if S(2) == motorState(v2)
49     valid(2) = 1;
50 end
51
52 S(3) = 3; % for S = 3
53 [Q3,eff3] = pumpCurve_v1(S(3),H);
54 % iteration method
55 for n = 1:ite
56     inew3 = pvCurve(PV_rating,G,T,v3);
57     vnew3 = vbat + (inew3 - (Q3*H)/(eff3*v3))*r;
58     if abs(vnew3 - v3) < 0.005
59         i3 = inew3; v3 = vnew3;
60         break
61     end
62     v3 = vnew3;
63 end
64 v_array(3) = v3; i_array(3) = i3;
65 Q_array(3) = Q3; Imotor_array(3) = Q3*H/(eff3*v3);
66 if S(3) == motorState(v3)
67     valid(3) = 1;
68 end
69
70 if valid(S_prev) == 1
71     S = S_prev;
72     i = i_array(S_prev); v = v_array(S_prev);
73     Q = Q_array(S_prev); Imotor = Imotor_array(S_prev);
74 else % when S_prev does not give valid EQB point
75     for x = 1:3
76         if valid(x) == 1
77             S = x;
78             i = i_array(x); v = v_array(x);
79             Q = Q_array(x); Imotor = Imotor_array(x);
80             break
81         end
82         if x == 3 % if no state is valid, default to S1
83             disp('No EQB found!');
84             S = 1;
85             i = i_array(1); v = v_array(1);
86             Q = Q_array(1); Imotor = Imotor_array(1);
87         end
88     end

```

```

89     end
90 end

```

Listing A.5: findOP_DLL1.m

A.3.2 DLL Operating Point 2

```

1 % findOP_DLL2 - a function to calculate system op 2
2 % Condition: motor is OFF and G > 0
3
4 function [i,v] = findOP_DLL2(PV_rating,G,T,vbat,r,v_prev)
5     % initialize
6     i = 0; v = v_prev;
7     ite = 20;
8
9     % iteration method
10    for n = 1:ite
11        inew = pvCurve(PV_rating,G,T,v);
12        vnew = vbat + inew*r;
13        if abs(vnew - v) < 0.005
14            i = inew; v = vnew;
15            break
16        end
17        v = vnew;
18    end
19 end

```

Listing A.6: findOP_DLL2.m

A.3.3 DLL Operating Point 3

```

1 % findOP_DLL3 - a function to calculate system op 3
2 % Condition: motor is ON and G = 0
3
4 function [v,S,Q,Imotor] = findOP_DLL3(PV_rating,vbat,r,v_prev,H)
5     % guess the motor state based on previous OP voltage
6     S = motorState(v_prev);
7     [Q,eff] = pumpCurve_v1(S,H);
8
9     ite = 10;
10    v = v_prev;
11    for n = 1:ite
12        vnew = vbat - (Q*H/(eff*v))*r;
13        if abs(vnew - v) < 0.005
14            v = vnew;
15            break
16        end
17        v = vnew;
18    end
19    Imotor = (Q*H)/(eff*v);
20

```

```
21 end
```

Listing A.7: findOP_DLL3.m

A.3.4 Water Demand

```
1 % waterDemand - a function to calculate target water delivered per
  day
2
3 function targetFlow = waterDemand(span)
4     targetFlow = zeros(span,1);
5     targetFlow(1:round(span*1/4)) = 13;
6     targetFlow(round(span*1/4):round(span*1/2)-19) = 20;
7     targetFlow(round(span*3/4):end) = 10; %
8 end
```

Listing A.8: waterDemand.m

A.3.5 Photovoltaic Panel

This function hosts the $i - v$ characteristic data of all photovoltaic panel models.

```
1 % pvCurve - a function for the PV panel
2
3 function i = pvCurve(PV_rating,G,T_c,v)
4 % if statement by the following order, PV_rating -> G -> T_c
5     if PV_rating == 260 % Tata 260W
6         if T_c < 13 % approximate T_c = 0 C
7             if G < 50 % approximate G = 0
8                 vec_v = [0 50];
9                 vec_i = [0 0];
10            elseif 50 <= G && G < 150 % approximate G = 100
11                vec_v = [36.9531300000000,36.8200080438698,
12                    36.6793868891343,36.5302493942185,36.3713563275446,
13                    36.2011765996205,36.0177877459554,35.8187300075713,
14                    35.6007853503580,35.3596297201472,35.0892598280289,
15                    34.7809925356885,34.4215862863366,33.9893593567465,
16                    33.4450277738626,32.7052929732603,31.5349381873925,
17                    28.4054664679134,0];
18                vec_i = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,
19                    0.5,0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.863076900000000];
20            elseif 150 <= G && G < 250 % approximate G = 200
21                vec_v = [37.4531300000000,37.3050080438698,
22                    37.1493868891343,36.9852493942185,36.8113563275446,
23                    36.6261765996205,36.4277877459554,36.2137300075713,
24                    35.9807853503580,35.7246297201472,35.4392598280289,
25                    35.1159925356885,34.7415862863366,34.2943593567465,
26                    33.7350277738626,32.9802929732603,31.7949381873925,
27                    28.6504664679134,0];
28                vec_i = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,
29                    1.2,1.3,1.4,1.5,1.6,1.7,1.726153800000000];
30            elseif 250 <= G && G < 350 % approximate G = 300
```



```

17         vec_v = [37.9531300000000, 36.6870035005449,
36.5283668458987, 36.3607853503580, 36.1829214556568,
35.9931131678165, 35.7892598280289, 35.5686527158835,
35.3277150609675, 35.0615862863366, 34.7634230531603,
34.4231485677471, 34.0250277738626, 33.5424330419242,
32.9246792629024, 32.0549381873925, 30.5370020127612, 0];
18         vec_i = [0, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2,
2.1, 2.2, 2.3, 2.4, 2.5, 2.58923070000000];
19         elseif 350 <= G && G < 450 % approximate G = 400
20         vec_v = [38.4531300000000, 37.4761765996205,
36.1392598280289, 35.9680527440843, 35.7859925356885,
35.5912709218839, 35.3815862863366, 35.1539461056681,
34.9043593567465, 34.6273360615506, 34.3150277738626,
33.9556472617418, 33.5302929732603, 33.0057448268414,
32.3149381873925, 31.2858835991001, 29.1404664679134, 0];
21         vec_i = [0, 1, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3
, 3.1, 3.2, 3.3, 3.4, 3.45230760000000];
22         elseif 450 <= G && G < 550 % approximate G = 500
23         vec_v = [38.9531300000000, 38.1313563275446,
37.1207853503580, 35.7015862863366, 35.5150845005350,
35.3151376642221, 35.0992166788031, 34.8640040466408,
34.6050277738626, 34.3160545247094, 33.9880228660892,
33.6070223412852, 33.1500663197282, 32.5749381873925,
31.7900900865881, 30.5246891528539, 26.5065419666912, 0];
24         vec_i = [0, 1, 2, 3, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9
, 4, 4.1, 4.2, 4.3, 4.31538450000000];
25         elseif 550 <= G && G < 650 % approximate G = 600
26         vec_v = [39.4531300000000, 38.7284377925225,
37.8870035005449, 36.8392598280289, 35.3231485677471,
35.1176339201177, 34.8950277738626, 34.6516387207105,
34.3824330419242, 34.0802929732603, 33.7346792629024,
33.3290120196658, 32.8349381873925, 32.1976147925256,
31.2870020127612, 29.6304664679134, 0];
27         vec_i = [0, 1, 2, 3, 4, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8,
4.9, 5, 5.1, 5.17846140000000];
28         elseif 650 <= G && G < 750 % approximate G = 700
29         vec_v = [39.9531300000000, 39.2949774867558,
38.5573781955017, 37.6947332713872, 36.6056586243061,
34.9735678673216, 34.7437774488818, 34.4915326672121,
34.2111395775710, 33.8944196405703, 33.5290140830385,
33.0949381873925, 32.5566394809677, 31.8409407515935,
30.7530788705188, 28.2965248368714, 0];
30         vec_i = [0, 1, 2, 3, 4, 5, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8
, 5.9, 6, 6.04153830000000];
31         elseif 750 <= G && G < 850 % approximate G = 800
32         vec_v = [40.4531300000000, 39.8434542289016,
39.1761765996205, 38.4248575959575, 37.5392598280289,
36.4039461056681, 36.2671963064996, 36.1243593567465,
35.9746988424977, 35.8173360615506, 35.6512108277004,
35.4750277738626, 35.2871811979628, 35.0856472617418,
34.8678248909859, 34.6302929732603, 34.3684247290000,
34.0757448268414, 33.7427911363865, 33.3549381873925,
32.8877866185497, 32.2958835991001, 31.4782768464694,

```

```

30.1204664679134,0];
33         vec_i = [0,1,2,3,4,5,5.1,5.2,5.3,5.4,5.5,5.6,5.7,5.8
,5.9,6,6.1,6.2,6.3,6.4,6.5,6.6,6.7,6.8,6.90461520000000];
34         elseif 850 <= G && G < 950 % approximate G = 900
35         vec_v = [40.9531300000000,40.3803542823764,
39.7638848284268,39.0870035005449,38.3210725537514,
37.4105541297709,36.2231485677471,36.0778847451158,
35.9254913116081,35.7650277738626,35.5953543189010,
35.4150711811257,35.2224330419242,35.0152250084637,
34.7905774189139,34.5446792629024,34.2723153318405,
33.9660785056663,33.6149381873925,33.2014076281204,
32.6952543777178,32.0370020127612,31.0812592708895,
29.2557007729029,0];
36         vec_i = [0,1,2,3,4,5,6,6.1,6.2,6.3,6.4,6.5,6.6,6.7,
6.8,6.9,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7,7.76769210000000];
37         else % approximate G = 1000
38         vec_v = [41.4531300000000,40.9093868891343,
40.3313563275446,39.7077877459554,39.0207853503580,
38.2392598280289,37.3015862863366,36.0550277738626,
35.8997736267975,35.7360545247094,35.5626428401237,
35.3780228660892,35.1802929732603,34.9670223412852,
34.7350343421340,34.4800663197282,34.1962100975473,
33.8749381873925,33.5032824992167,33.0600900865881,
32.5072497327712,31.7646891528539,30.6104664679134,
27.7165419666912,0];
39         vec_i = [0,1,2,3,4,5,6,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7
,7.8,7.9,8,8.1,8.2,8.3,8.4,8.5,8.6,8.63076900000000];
40         end
41         elseif 13 <= T_c && T_c < 38 % approximate T_c = 25 C
42         if G < 50 % approximate G = 0
43         vec_v = [0 50];
44         vec_i = [0 0];
45         elseif 50 <= G && G < 150 % approximate G = 100
46         vec_v = [33.4000000000000,33.2721558816937,
33.1372984335693,32.9944977790684,32.8426258938252,
32.6802959737796,32.5057766768009,32.3168676042001,
32.1107128922044,31.8835118840105,31.6300501330222,
31.3428976134722,31.0109431644613,30.6164724094296,
30.1286105056766,29.4858432377232,28.5342576963583,
26.6271553098305,0];
47         vec_i = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,
0.5,0.55,0.60,0.65,0.7,0.75,0.8,0.85,0.88];
48         elseif 150 <= G && G < 250 % approximate G = 200
49         vec_v = [33.9000000000000,33.8294029423405,
33.7571558816937,33.6831578775326,33.6072984335693,
33.5294562521865,33.4494977790684,33.3672754941396,
33.2826258938252,33.1953670952135,33.1052959737796,
33.0121847212677,32.9157766768009,32.8157812389266,
32.7118676042001,32.6036569917647,32.4907128922044,
32.3725287058262,32.2485118840105,32.1179633149949,
31.9800501330222,31.8337692604116,31.6778976134722,
31.5109226529569,31.3309431644613,31.1355235072973,
30.9214724094296,30.6844939633001,30.4186105056766,

```

```

30.1151511741065,29.7608432377232,29.3338407779335,
28.7942576963583,28.0564424176260,26.8721553098305,
23.4007001139843,0];
50     vec_i = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,
0.5,0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.90,0.95,1,1.05,1.1,1.15,1.2
,1.25,1.3,1.35,1.4,1.45,1.5,1.55,1.6,1.65,1.7,1.75,1.76];
51     elseif 250 <= G && G < 350 % approximate G = 300
52     vec_v = [34.4000000000000,33.1815963498279,
33.0302302722569,32.8707128922044,32.7018699386636,
32.5222535606344,32.3300501330222,32.1229456242510,
31.8979226848213,31.6509431644613,31.3764284296083,
31.0663600828861,30.7086105056766,30.2835448041539,
29.7561706979790,29.0542576963583,27.9844879686152,
25.5378025005122,0];
53     vec_i = [0,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2,
2.1,2.2,2.3,2.4,2.5,2.6,2.64];
54     elseif 350 <= G && G < 450 % approximate G = 400
55     vec_v = [34.9000000000000,33.9552959737796,
32.6800501330222,32.5187692604116,32.3478976134722,
32.1659226529569,31.9709431644613,31.7605235072973,
31.5314724094296,31.2794939633001,30.9986105056766,
30.6801511741066,30.3108432377232,29.8688407779335,
29.3142576963583,28.5614424176260,27.3621553098305,
23.8757001139844,0];
56     vec_i = [0,1,2,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3
,3.1,3.2,3.3,3.4,3.5,3.52];
57     elseif 450 <= G && G < 550 % approximate G = 500
58     vec_v = [35.4000000000000,34.6026258938252,
33.6307128922044,32.2909431644613,32.1179826333167,
31.9335734966109,31.7357212088130,31.5218598301740,
31.2886105056766,31.0313970919518,30.7438037688678,
30.4164341010174,30.0347204345806,29.5742576963583,
28.9892947054240,28.1771183007648,26.8099789051713,0];
59     vec_i = [0,1,2,3,3.1,3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9
,4,4.1,4.2,4.3,4.4];
60     elseif 550 <= G && G < 650 % approximate G = 600
61     vec_v = [35.9000000000000,35.1949441965286,
34.3815963498279,33.3800501330222,31.9663600828861,
31.7794374808411,31.5786105056766,31.3611734367182,
31.1235448041539,30.8608432377232,30.5661706979790,
30.2293214809547,29.8342576963583,29.3535949371761,
28.7344879686152,27.8521553098305,26.2578025005122,0];
62     vec_i = [0,1,2,3,4,4.1,4.2,4.3,4.4,4.5,4.6,4.7,4.8,
4.9,5,5.1,5.2,5.28];
63     elseif 650 <= G && G < 750 % approximate G = 700
64     vec_v = [36.4000000000000,35.7582816117536,
35.0427153519941,34.2123270166397,33.1783802334197,
31.6790912086978,31.4751821360970,31.2540274241014,
31.0118264159074,30.7433646649192,30.4412121453691,
30.0942576963583,29.6847869413266,29.1819250375736,
28.5241577696201,27.5575722282553,25.6354698417275,0];
65     vec_i = [0,1,2,3,4,5,5.1,5.2,5.3,5.4,5.5,5.6,5.7,5.8
,5.9,6,6.1,6.16];

```

```

66         elseif 750 <= G && G < 850 % approximate G = 800
67             vec_v = [36.9000000000000,36.3044562521866,
35.6552959737796,34.9286569917647,34.0800501330222,
33.0105235072973,31.4108432377232,31.1858202982934,
30.9388407779335,30.6643260430804,30.3542576963583,
29.9965081191488,29.5714424176260,29.0440683114512,
28.3421553098305,27.2723855820874,24.8257001139844,0];
68             vec_i = [0,1,2,3,4,5,6,6.1,6.2,6.3,6.4,6.5,6.6,6.7,
6.8,6.9,7,7.04];
69         elseif 850 <= G && G < 950 % approximate G = 900
70             vec_v = [37.4000000000000,36.8396209950240,
36.2385308161361,35.5815963498279,34.8432758232461,
33.9751449697501,32.8663600828861,32.7331758965079,
32.5941590746922,32.4486105056766,32.2956973237040,
32.1344164510934,31.9635448041539,31.7815698436386,
31.5865903551430,31.3761706979790,31.1471196001113,
30.8951411539818,30.6142576963583,30.2957983647883,
29.9264904284049,29.4844879686152,28.9299048870400,
28.1770896083077,26.9778025005122,23.4913473046661,0];
71             vec_i = [0,1,2,3,4,5,6,6.1,6.2,6.3,6.4,6.5,6.6,6.7,
6.8,6.9,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7,7.8,7.9,7.92];
72         else % approximate G = 1000
73             vec_v = [37.9000000000000,37.3672984335693,
36.8026258938252,36.1957766768009,35.5307128922044,
34.7800501330222,33.8909431644613,32.7386105056766,
32.5983471026382,32.4513970919518,32.2968968392397,
32.1338037688678,31.9608432377232,31.7764341010174,
31.5785818132195,31.3647204345806,31.1314711100831,
30.8742576963583,30.5866643732743,30.2592947054240,
29.8775810389871,29.4171183007648,28.8321553098305,
28.0199789051713,26.6528395095778,0];
74             vec_i = [0,1,2,3,4,5,6,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7
,7.8,7.9,8,8.1,8.2,8.3,8.4,8.5,8.6,8.7,8.8];
75         end
76         elseif 38 <= T_c && T_c < 63 % approximate T_c = 50 C
77             if G < 50 % approximate G = 0
78                 vec_v = [0 50];
79                 vec_i = [0 0];
80             elseif 50 <= G && G < 150 % approximate G = 100
81                 vec_v = [30,29.9076865184313,29.8117243438753,
29.7117469781454,29.6073297693862,29.4979769183419,
29.3831046396774,29.2620190266968,29.1338864851868,
28.9976935260875,28.8521909622007,28.6958146344152,
28.5265697246478,28.3418565353689,28.1381981544535,
27.9107951850742,27.6527563311192,27.3536726427005,
26.9967225966541,26.5520074542094,25.9580744473077,
25.0518163699587,23.0067510310841,0];
82                 vec_i = [0,0.04,0.08,0.12,0.16,0.2,0.24,0.28,0.32,
0.36,0.4,0.44,0.48,0.52,0.56,0.6,0.64,0.68,0.72,0.76,0.8,0.84,
0.88,0.896923100000000];
83             elseif 150 <= G && G < 250 % approximate G = 200
84                 vec_v = [30.3640514661974,30.2939311676765,
30.2222626013365,30.1489470545587,30.0738760625865,

```

```

29.9969300800600,29.9179769183419,29.8368698972681,
29.7534456463143,29.6675214722201,29.5788921862673,
29.4873262523937,29.3925610738473,29.2942971763162,
29.1921909622007,29.0858455929866,28.9747993876334,
28.8585108778686,28.7363392933037,28.6075186893818,
28.4711230588060,28.3260183712921,28.1707951850742,
28.0036715467170,27.8223489257911,27.6237909774640,
27.4038695081474,27.1567687760399,26.8739184087991,
26.5419216465124,26.1380744473077,25.6190631203591,
24.8846585424975,23.5917577496418,0];
85     vec_i = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,
0.5,0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,1,1.05,1.1,1.15,1.2,
1.25,1.3,1.35,1.4,1.45,1.5,1.55,1.6,1.65,1.7,1.75,
1.79384620000000];
86     elseif 250 <= G && G < 350 % approximate G = 300
87     vec_v = [31,29.8227739651943,29.7525610738473,
29.6807940996680,29.6073737644867,29.5321909622007,
29.4551254171705,29.3760441055767,29.2947993876334,
29.2112267846811,29.1251423169244,29.0363392933037,
28.9445844123784,28.8496129887788,28.7511230588060,
28.6487680337349,28.5421474490719,28.4307951850742,
28.3141642808233,28.1916070867836,28.0623489257911,
27.9254525351699,27.7797691245595,27.6238695081474,
27.4559447068770,27.2736581853892,27.0739184087991,
26.8525138567776,26.6034967742019,26.3180744473077,
25.9824442457703,25.5730741206532,25.0446585424975,
24.2906597725863,22.9318154354833,0];
88     vec_i = [0,1,1.05,1.10,1.15,1.2,1.25,1.3,1.35,1.4,
1.45,1.5,1.55,1.6,1.65,1.7,1.75,1.8,1.85,1.9,1.95,2,2.05,2.1,2.15
,2.2,2.2,2.3,2.35,2.4,2.45,2.5,2.55,2.6,2.65,2.69076930000000];
89     elseif 350 <= G && G < 450 % approximate G = 400
90     vec_v = [31.5000000000000,30.5534456463143,
29.3363392933037,29.1875186893818,29.0311230588060,
28.8660183712921,28.6907951850742,28.5036715467170,
28.3023489257911,28.0837909774640,27.8438695081474,
27.5767687760399,27.2739184087991,26.9219216465124,
26.4980744473077,25.9590631203591,25.2046585424975,
23.8917577496418,0];
91     vec_i = [0,1,2,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3
,3.1,3.2,3.3,3.4,3.5,3.58769240000000];
92     elseif 450 <= G && G < 550 % approximate G = 500
93     vec_v = [32,31.1779769183419,30.2121909622007,
28.9507951850742,28.7941497121328,28.6287563311192,
28.4531957740862,28.2656726427005,28.0638695081474,
27.8447225966541,27.6040623296039,27.3360074542094,
27.0318773810940,26.6780744473077,26.2514855974026,
25.7078163699587,24.9437492955024,23.5987510310841,0];
94     vec_i = [0,1,2,3,3.1,3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9
,4,4.1,4.2,4.3,4.4,4.48461550000000];
95     elseif 550 <= G && G < 650 % approximate G = 600
96     vec_v = [32.5000000000000,31.7561282872290,
30.9227739651943,29.9363392933037,28.6254525351699,
28.4597691245595,28.2838695081474,28.0959447068770,

```

```

27.8936581853892,27.6739184087991,27.4325138567776,
27.1634967742019,26.8580744473077,26.5024442457703,
26.0730741206532,25.5246585424976,24.7506597725863,
23.3718154354833,0];
97     vec_i = [0,1,2,3,4,4.1,4.2,4.3,4.4,4.5,4.6,4.7,4.8,
4.9,5,5.1,5.2,5.3,5.38153860000000];
98     elseif 650 <= G && G < 750 % approximate G = 700
99     vec_v = [33,32.3097965067864,31.5584530085460,
30.7131384844974,29.7044735951595,28.3378947225418,
28.1616543454707,27.9733256796066,27.7705528720067,
27.5502162894469,27.3080619037812,27.0380744473077,
26.7313471328756,26.3738681944459,25.9416763003954,
25.3884233580878,24.6042121024226,23.1896186664029,0];
100    vec_i = [0,1,2,3,4,5,5.1,5.2,5.3,5.4,5.5,5.6,5.7,5.8
,5.9,6,6.1,6.2,6.27846170000000];
101    elseif 750 <= G && G < 850 % approximate G = 800
102    vec_v = [33.5000000000000,32.8489470545587,
32.1534456463143,31.3942971763162,30.5363392933037,
29.5036715467170,28.0739184087991,27.8851836655834,
27.6819216465124,27.4609842776472,27.2180744473077,
26.9471083456389,26.6390631203591,26.2797135941388,
25.8446585424975,25.2864745585968,24.4917577496418,
23.0393381446719,0];
103    vec_i = [0,1,2,3,4,5,6,6.1,6.2,6.3,6.4,6.5,6.6,6.7,
6.8,6.9,7,7.1,7.17538480000000];
104    elseif 850 <= G && G < 950 % approximate G = 900
105    vec_v = [34,33.3787763416361,32.7237708529177,
32.0227739651943,31.2554644514985,30.3841194054493,
29.3254525351699,27.8233708052049,27.6196166228061,
27.3980744473077,27.1544034979691,26.8824503734308,
26.5730741206532,26.2118317648548,25.7738712453396,
25.2106597725863,24.4051313275098,22.9126114695590,0];
106    vec_i = [0,1,2,3,4,5,6,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7
,7.8,7.9,8,8.07230790000000];
107    else % approximate G = 1000
108    vec_v = [34.5000000000000,33.9022626013365,
33.2779769183419,32.6188921862673,31.9121909622007,
31.1363392933037,30.2507951850742,29.1638695081474,
29.0366931943240,28.9047225966541,28.7673995438603,
28.6240623296039,28.4739184087991,28.3160074542094,
28.1491503769111,27.9718773810940,27.7823237717103,
27.5780744473077,27.3559234043181,27.1114855974026,
26.8385369641086,26.5278163699587,26.1646585424976,
25.7237492955024,25.1554110056542,24.3387510310841,
22.8036327637397,0];
109    vec_i = [0,1,2,3,4,5,6,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7
,7.8,7.9,8,8.1,8.2,8.3,8.4,8.5,8.6,8.7,8.8,8.9,8.96923100000000];
110    end
111    else % approximate T_c = 75 C
112    if G < 50 % approximate G = 0
113    vec_v = [0 50];
114    vec_i = [0 0];
115    elseif 50 <= G && G < 150 % approximate G = 100

```

```

116         vec_v = [30, 29.9065799022108, 29.8092287048564,
29.7075515430173, 29.6010908970477, 29.4893125931361,
29.3715876613243, 29.2471684869675, 29.1151569561765,
28.9744611364658, 28.8237351543215, 28.6612937856857,
28.4849878147908, 28.2920163291281, 28.0786333057269,
27.8396678774174, 27.5676953660355, 27.2515011754172,
26.8729618069214, 26.3998643779098, 25.7659997141271,
24.7956362128348, 22.5983224731185, 0];
117         vec_i = [0, 0.04, 0.08, 0.12, 0.16, 0.2, 0.24, 0.28, 0.32,
0.36, 0.40, 0.44, 0.48, 0.52, 0.56, 0.6, 0.64, 0.68, 0.72, 0.76, 0.8, 0.84,
0.88, 0.896923100000000];
118         elseif 150 <= G && G < 250 % approximate G = 200
119         vec_v = [30.5000000000000, 28.7507927989773,
28.6464163291281, 28.5369586068801, 28.4218333057269,
28.3003463674562, 28.1716678774174, 28.0347941170966,
27.8884953660355, 27.7312425058266, 27.5611011754172,
27.3755745784374, 27.1713618069214, 26.9439705594494,
26.6870643779098, 26.3912908006833, 26.0419997141271,
25.6142891266134, 25.0604362128348, 24.2691267951543,
22.8519224731185, 0];
120         vec_i = [0, 1, 1.04, 1.08, 1.12, 1.16, 1.2, 1.24, 1.28, 1.32,
1.36, 1.4, 1.44, 1.48, 1.52, 1.56, 1.6, 1.64, 1.68, 1.72, 1.76,
1.793846200000000];
121         elseif 250 <= G && G < 350 % approximate G = 300
122         vec_v = [31, 29.8826171513548, 29.8145178133092,
29.7447441073822, 29.6731890713355, 29.5997351543215,
29.5242527714406, 29.4465986029123, 29.3666135817226,
29.2841204986657, 29.1989211340238, 29.1107927989773,
29.0194841346987, 28.9247099693344, 28.8261449673823,
28.7234157143646, 28.6160907500837, 28.5036678774174,
28.3855578010108, 28.2610627436554, 28.1293480686542,
27.9894039697806, 27.8399927410115, 27.6795745784374,
27.5062004891430, 27.3173530913791, 27.1097015673749,
26.8787084277356, 26.6179655647748, 26.3179997141271,
25.9639399304274, 25.5304327987223, 24.9686661696258,
24.1638563494119, 22.7073862896229, 0];
123         vec_i = [0, 1, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4,
1.45, 1.5, 1.55, 1.6, 1.65, 1.7, 1.75, 1.8, 1.85, 1.9, 1.95, 2, 2.05, 2.1, 2.15
, 2.2, 2.25, 2.3, 2.35, 2.4, 2.45, 2.5, 2.55, 2.6, 2.65, 2.690769300000000];
124         elseif 350 <= G && G < 450 % approximate G = 400
125         vec_v = [31.5000000000000, 30.6311426777627,
29.4707927989773, 29.3991244192659, 29.3255495476931,
29.2499378061330, 29.1721449673823, 29.0920109214895,
29.0093572545190, 28.9239843475909, 28.8356678774174,
28.7441545637594, 28.6491569605496, 28.5503470203978,
28.4473480686542, 28.3397246907245, 28.2269698456888,
28.1084882400467, 27.9835745784374, 27.8513846719520,
27.7108963902845, 27.5608558473908, 27.3997015673749,
27.2254548471301, 27.0355564488741, 26.8266146396942,
26.5939997141271, 26.3311569792023, 26.0283652194582,
25.6702982376480, 25.2306661696258, 24.6583750662282,
23.8312436435769, 22.2955036357455, 0];

```

```

126         vec_i = [0,1,2,2.05,2.1,2.15,2.2,2.25,2.3,2.35,2.4,
127         2.45,2.5,2.55,2.6,2.65,2.7,2.75,2.8,2.85,2.9,2.95,3,3.05,3.1,3.15
128         ,3.2,3.25,3.3,3.35,3.4,3.45,3.50,3.55,3.58769240000000];
127         elseif 450 <= G && G < 550 % approximate G = 500
128         vec_v = [32,31.2653125931361,30.3757351543215,
29.1676678774174,29.0139941170966,28.8508953660355,
28.6768425058266,28.4899011754172,28.2875745784374,
28.0665618069214,27.8223705594494,27.5486643779098,
27.2360908006834,26.8699997141271,26.4254891266134,
25.8548362128348,25.0467267951543,23.6127224731185,0];
129         vec_i = [0,1,2,3,3.1,3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9
,4,4.1,4.2,4.3,4.4,4.48461550000000];
130         elseif 550 <= G && G < 650 % approximate G = 600
131         vec_v = [32.5000000000000,31.8495128295841,
31.1026171513548,30.1907927989773,28.9294039697806,
28.7659927410115,28.5915745784374,28.4042004891430,
28.2013530913791,27.9797015673749,27.7347084277356,
27.4599655647748,27.1459997141271,26.7779399304274,
26.3304327987223,25.7546661696258,24.9358563494119,
23.4653862896229,0];
132         vec_i = [0,1,2,3,4,4.1,4.2,4.3,4.4,4.5,4.6,4.7,4.8,
4.9,5,5.1,5.2,5.3,5.38153860000000];
133         elseif 650 <= G && G < 750 % approximate G = 700
134         vec_v = [33,32.4073349361042,31.7487976686502,
30.9890160761768,30.0532409012864,28.7318494224290,
28.5570641245319,28.3692549117743,28.1658835896543,
27.9435890672679,27.6977880573246,27.4219997141271,
27.1066278707143,26.7365762590314,26.2860289552247,
25.7050505529862,24.8752378069444,23.3662516901183,0];
135         vec_i = [0,1,2,3,4,5,5.1,5.2,5.3,5.4,5.5,5.6,5.7,5.8
,5.9,6,6.1,6.2,6.27846170000000];
136         elseif 750 <= G && G < 850 % approximate G = 800
137         vec_v = [33.5000000000000,32.9495157120323,
32.3511426777627,31.6841963385697,30.9107927989773,
29.9491569605496,28.5597015673749,28.3714548471301,
28.1675564488741,27.9446146396942,27.6979997141271,
27.4211569792023,27.1043652194582,26.7322982376480,
26.2786661696258,25.6923750662282,24.8512436435769,
23.3015036357455,0];
138         vec_i = [0,1,2,3,4,5,6,6.1,6.2,6.3,6.4,6.5,6.6,6.7,
6.8,6.9,7,7.1,7.17538480000000];
139         elseif 850 <= G && G < 950 % approximate G = 900
140         vec_v = [34,33.4816537674594,32.9269110118122,
32.3226171513548,31.6468781087353,30.8590512479763,
29.8694039697806,28.4020217963198,28.1975931415992,
27.9739997141271,27.7265647594218,27.4486586066685,
27.1304327987223,26.7563264834969,26.2995640661863,
25.7078563494119,24.8550764766978,23.2621324811348,0];
141         vec_i = [0,1,2,3,4,5,6,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7
,7.8,7.9,8,8.07230790000000];
142         else % approximate G = 1000
143         vec_v = [34.5000000000000,34.0069578491180,
33.4853125931361,32.9261748632229,32.3157351543215,

```



```

31.6307927989773,30.8276678774174,29.8075745784374,
29.6856508719033,29.5585618069214,29.4257060101034,
29.2863705594494,29.1397015673749,28.9846643779098,
28.8199886413779,28.6440908006834,28.4549618345361,
28.2499997141271,28.0257502931972,27.7774891266134,
27.4985104133430,27.1788362128348,26.8026661696258,
26.3427267951543,25.7454954488163,24.8807224731185,
23.2418829776515,0];
144     vec_i = [0,1,2,3,4,5,6,7,7.1,7.2,7.3,7.4,7.5,7.6,7.7
,7.8,7.9,8,8.1,8.2,8.3,8.4,8.5,8.6,8.7,8.8,8.9,8.96923100000000];
145     end
146     end
147     i = interp1(vec_v,vec_i,v);
148
149     elseif PV_rating == 180 % Sharp 180W
150         % roughly estimate that i is INdependent of T_c
151         if G < 100 % approximate G = 0
152             vec_v = [0 40];
153             vec_i = [0 0];
154         elseif 100 <= G && G < 300 % approximate G = 200
155             vec_v = [0 15 16.42857143 17.85714286 19.28571429
20.71428571 22.14285714 23.57142857...
156                 25 25.57142857 26.14285714 26.71428571 27.28571429];
157             vec_i = [1.620689655 1.620689655 1.620689655
1.586206897 1.517241379 1.448275862 1.310344828...
158                 1.103448276 0.75862069 0.586206897 0.413793103
0.206896552 0];
159         elseif 300 <= G && G < 500 % approximate G = 400
160             vec_v = [0 15 16.42857143 17.85714286 19.28571429
20.71428571 22.14285714 23.57142857...
161                 25 25.57142857 26.14285714 26.71428571 27.28571429
27.85714286];
162             vec_i = [3.310344828 3.310344828 3.310344828 3.310344828
3.275862069 3.24137931...
163                 3.137931034 2.896551724 2.344827586 2.068965517
1.655172414 1.172413793 0.689655172 0.137931034];
164         elseif 500 <= G && G < 700 % approximate G = 600
165             vec_v = [0 15 16.42857143 17.85714286 19.28571429
20.71428571 22.14285714 23.57142857...
166                 25 25.57142857 26.14285714 26.71428571 27.28571429
27.85714286 28.42857143 29];
167             vec_i = [5.137931034 5.137931034 5.103448276 5.068965517
4.965517241 4.827586207 4.586206897...
168                 4.24137931 3.586206897 3.310344828 2.896551724
2.448275862 1.896551724 1.172413793 0.689655172 0];
169         elseif 700 <= G && G < 900 % approximate G = 800
170             vec_v = [0 15 16.42857143 17.85714286 19.28571429
20.71428571 22.14285714 23.57142857...
171                 25 25.57142857 26.14285714 26.71428571 27.28571429
27.85714286 28.42857143 29 29.57142857];
172             vec_i = [6.793103448 6.793103448 6.793103448 6.793103448
6.75862069 6.689655172 6.517241379...

```

```

173         6.275862069 5.655172414 5.310344828 4.896551724
4.310344828 3.586206897 2.620689655 1.862068966 0.965517241 0];
174     else % approximate G = 1000
175         vec_v = [0 15 16.42857143 17.85714286 19.28571429
20.71428571 22.14285714 23.57142857...
176             25 25.57142857 26.14285714 26.71428571 27.28571429
27.85714286 28.42857143 29 29.57142857 30.14285714];
177         vec_i = [8.310344828 8.310344828 8.310344828 8.275862069
8.206896552 8.103448276 7.931034483 7.689655172...
178             7.172413793 6.827586207 6.344827586 5.931034483
5.206896552 4.482758621 3.655172414 2.344827586 1.310344828 0];
179     end
180     i = interp1(vec_v,vec_i,v);
181
182     elseif PV_rating == 320 % JAP 320W
183         % roughly estimate that V_oc and I_sc is INdependent of G
184         GO = 1000;
185         if T_c < 18 % approximate T_c = 10 C
186             vec_v = [0 35.06849315 36.43835616 37.80821918
39.17808219 40.54794521 41.91780822 43.28767123...
187                 44.65753425 46.02739726 47.39726027 48.63013699];
188             vec_i = [8.823 8.823 8.829268293 8.780487805 8.731707317
8.536585366 8.146341463 7.512195122...
189                 6.341463415 4.634146341 2.487804878 0];
190         elseif 18 <= T_c && T_c < 33 % approximate T_c = 25 C
191             vec_v = [0 33.69863014 35.06849315 36.43835616
37.80821918 39.17808219 40.54794521 41.91780822 43.28767123...
192                 44.65753425 46.02739726];
193             vec_i = [8.9 8.9 8.87804878 8.780487805 8.585365854
8.195121951 7.658536585 6.634146341 5.219512195...
194                 3.12195122 0.634146341];
195         elseif 33 <= T_c && T_c < 48 % approximate T_c = 40 C
196             vec_v = [0 30.95890411 32.32876712 33.69863014
35.06849315 36.43835616 37.80821918 39.17808219...
197                 40.54794521 41.91780822 43.28767123];
198             vec_i = [8.977 8.977 8.926829268 8.829268293 8.634146341
8.243902439 7.609756098 6.634146341...
199                 5.317073171 3.317073171 0.975609756];
200         elseif 48 <= T_c && T_c < 63 % approximate T_c = 55 C
201             vec_v = [0 26.84931507 28.21917808 29.5890411
30.95890411 32.32876712...
202                 33.69863014 35.06849315 36.43835616 37.80821918
39.17808219 40.54794521];
203             vec_i = [9.055 9.055 9.073170732 9.024390244 8.926829268
8.731707317...
204                 8.390243902 7.804878049 6.829268293 5.463414634
3.609756098 1.463414634];
205         else % approximate T_c = 70 C
206             vec_v = [0 22.73972603 25.47945205 26.84931507
28.21917808 29.5890411 30.95890411 32.32876712...
207                 33.69863014 35.06849315 36.43835616 37.80821918];
208             vec_i = [9.132 9.132 9.073170732 9.024390244 8.926829268
8.731707317 8.341463415...

```

```

209         7.756097561 7.024390244 5.365853659 3.707317073
1.463414634];
210     end
211     if v < 0
212         disp(v); disp(PV_rating); disp(G); disp(T_c);
213     end
214     i0 = interp1(vec_v,vec_i,v);
215     i = i0*G/G0;
216
217     else % Something is wrong
218         i = 0;
219         disp('PV Rating is INCORRECT');
220     end
221 end

```

Listing A.9: pvCurve.m

A.3.6 Pump System

```

1 % pumpCurve - a function for the pump system at Dial 32
2
3 function [Q,I] = pumpCurve(v,H)
4     if v < 26.91
5         % State S1 with v ~ 25.83
6         Vappx = 25.83;
7         vec_P = [63,70,80,90,100,110,120,130,140]*10^3;
8         vec_Q = [1.292,1.135,1.014,0.902,0.779,0.649,0.503,0.356,
0.220]*10^-3;
9         vec_eff = [0.3266,0.3347,0.3579,0.3756,0.3821,0.3748,0.3461,
0.2912,0.2110];
10        elseif 26.91 <= v && v < 29.07
11            % State S2 with v ~ 27.99
12            Vappx = 27.99;
13            vec_P = [70,80,90,100,110,120,130,140,150,160,170]*10^3;
14            vec_Q = [1.379,1.200,1.094,0.989,0.890,0.775,0.658,0.524,
0.391,0.259,0.055]*10^-3;
15            vec_eff = [0.3179,0.3311,0.3532,0.3698,0.3809,0.3827,0.3732,
0.3460,0.3004,0.2298,0.585];
16        else % 29.07 < v
17            % State S3 with v ~ 30.15
18            Vappx = 30.15;
19            vec_P = [78,90,100,110,120,130,140,150,160,170,180]*10^3;
20            vec_Q = [1.495,1.310,1.191,1.103,0.985,0.874,0.784,0.671,
0.538,0.416,0.284]*10^-3;
21            vec_eff = [0.3131,0.3342,0.3503,0.3692,0.3772,0.3786,0.3775,
0.3643,0.3376,0.2968,0.2306];
22        end
23
24        Q = interp1(vec_P,vec_Q,H,'linear');
25        eff = interp1(vec_P,vec_eff,H,'linear');
26

```

```
27     I = (Q*H)/(eff*v);  
28 end
```

Listing A.10: pumpCurve.m

Bibliography

- [1] K. P. Simon, “Applications of design for value to distributed solar generation in indian food processing and irrigation,” Master thesis, Massachusetts Institute of Technology, 2015. [Online]. Available: <http://hdl.handle.net/1721.1/100880> (visited on 04/16/2017).
- [2] Agriculture Census Division, Indian Ministry of Agriculture. (2015). Agricultural census 2010-11 phase-ii, [Online]. Available: <http://agcensus.nic.in/document/agcensus2010/allindia201011H.pdf> (visited on 05/03/2017).
- [3] A. Mukherji, “Major insights from india’s minor irrigation censuses: 1986-87 to 2006-07,” *Economic and Political Weekly*, vol. 48, no. 26-27, Feb. 2013.
- [4] Central Ground Water Board, Ministry of Water Resources, Government of India. (2016). Ground water scenario in india premonsoon, 2016, [Online]. Available: <http://www.cgwb.gov.in/Ground-Water/GW%5C%20Monitoring%5C%20Report-PREMONSOON%5C%202016.pdf> (visited on 04/23/2017).
- [5] Solar Energy Centre, Indian Ministry of New and Renewable Energy. (). Indian solar resource maps, [Online]. Available: <http://mnre.gov.in/sec/solar-assmnt.htm> (visited on 05/03/2017).
- [6] E. Gorbaty, “Development of an efficient off-grid pumping system and evaporation reduction strategies to increase access to irrigation for smallholder farmers in india,” Master thesis, Massachusetts Institute of Technology, 2013. [Online]. Available: <http://hdl.handle.net/1721.1/85225> (visited on 05/03/2017).
- [7] S. Phansalkar and S. Verma, *Mainstreaming the margins: Water-centric livelihood strategies for revitalizing tribal agriculture in central india*. Angus and Grapher, 2005.
- [8] Rotomag. (2017). Data sheet for solar submersible dc pump set: RC1200 series, [Online]. Available: http://rotosol.solar/pdf/rs_1200.pdf (visited on 04/23/2017).
- [9] ———, (2017). Data sheet for solar dc surface pumps, [Online]. Available: http://rotosol.solar/pdf/surface_pump.pdf (visited on 04/23/2017).
- [10] Honda Power Equipment. (2017). Wx10 pump, [Online]. Available: <http://powerequipment.honda.com/pumps/models/wx10> (visited on 04/23/2017).
- [11] M. Tomas, *Solar electricity*, 2nd ed. Wiley, 2000.

- [12] D. Evans, "Simplified method for predicting photovoltaic array output," *Solar Energy*, vol. 27, no. 6, pp. 555–560, 1981. DOI: [http://dx.doi.org/10.1016/0038-092X\(81\)90051-7](http://dx.doi.org/10.1016/0038-092X(81)90051-7).
- [13] E. Skoplaki, A. Boudouvis, and J. Palyvos, "A simple correlation for the operating temperature of photovoltaic modules of arbitrary mounting," *Solar Energy Materials and Solar Cells*, vol. 92, no. 11, pp. 1393–1402, 2008. DOI: <https://doi.org/10.1016/j.solmat.2008.05.016>.
- [14] M. A. Esparza, "Design for low power irrigation systems for small plot farming in eastern india," Bachelor thesis, Massachusetts Institute of Technology, 2015. [Online]. Available: <http://hdl.handle.net/1721.1/100882> (visited on 04/16/2017).
- [15] Shandong GoldenCell Electronics Technology. (2016). Lithium-ion rechargeable battery.
- [16] A. M. Harrington and C. Kroninger. (2013). Characterization of small dc brushed and brushless motors, [Online]. Available: <http://www.dtic.mil/get-tr-doc/pdf?AD=ADA577582> (visited on 05/03/2017).
- [17] Texas Instruments. (2011). Characteristics of rechargeable batteries, [Online]. Available: <http://www.ti.com/lit/an/snva533/snva533.pdf>.