2014

STATE OF CHARGE BASED DROOP SURFACE FOR OPTIMAL CONTROL OF DC MICROGRIDS

Arthur John Jones
Michigan Technological University

STATE OF CHARGE BASED DROOP SURFACE FOR OPTIMAL CONTROL OF DC MICROGRIDS

By

Arthur John Jones

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2014

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical Engineering.

Department of Electrical and Computer Engineering

Thesis Advisor: *Dr. Wayne W. Weaver*

Committee Member: *Dr. Duane J. Bucheger*

Committee Member: *Dr. Lucia Gauchia*

Committee Member: *Dr. Gordon G. Parker*

Department Chair: *Dr. Daniel R. Fuhrmann*

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Dr. Wayne Weaver for his time, patience and continued support during my time in graduate school. He has been a role model to me as a teacher and mentor.

I would also like to thank my committee members Dr. Gordon Parker, Dr. Lucia Gauchia, and Dr. Duane Bucheger for their time and assistance.

I am grateful for all of the help and support that my fellow graduate school colleagues provided over the years. I would especially like to acknowledge and thank Trever Hassell, Kaitlyn Bunker, Rob Smith, Greg Vosters, Bibaswan Banerjee, Ramchandra Kotecha, and Nishantha Ekneligoda.

Thank you to my family and friends who have been supportive as I continued my education.

Finally, I would like to thank my wife for continually being there with me throughout the graduate school experience.

Abstract

For a microgrid with a high penetration level of renewable energy, energy storage use becomes more integral to the system performance due to the stochastic nature of most renewable energy sources. This thesis examines the use of droop control of an energy storage source in dc microgrids in order to optimize a global cost function. The approach involves using a multidimensional surface to determine the optimal droop parameters based on load and state of charge. The optimal surface is determined using knowledge of the system architecture and can be implemented with fully decentralized source controllers. The optimal surface control of the system is presented. Derivation of a cost function along with the implementation of the optimal control are included. Results were verified using a hardware-in-the-loop system.

Introduction

One goal of the use of microgrids as a method of electricity distribution is to be able to more effectively take advantage of the benefits of different renewable distributed energy resources (DERS) [1]. The power available from many of these renewable DERS can be stochastic in nature. The use of this variable generation at high penetrations could lead to a decreased overall reliability if either curtailment or energy storage is not used [2]. This thesis focuses on a multidimensional surface control that optimizes a cost function while still meeting load requirements.

The system being studied is a dc microgrid with three sources, a resistive load, and bus capacitance. The three sources are a renewable energy input, a conventional source, and an energy storage source that is representative of a supercapacitor bank. The renewable energy input is simplified to be a current source. The conventional source and the energy storage source are fed into the grid through dc/dc converters.

The cost function is a combined sum of the input power from the traditional source and

the energy storage source. For this research, it is assumed that the two controlled sources are running dc voltage droop control. The optimization chosen takes advantage of known system architecture to develop an optimal control surface for the energy storage source controller. The surface controls the droop resistance parameter of the energy storage source to minimize the cost function developed. The optimization makes some assumptions about the method of control of the traditional source and is able to implement this optimal surface completely decentralized, which maintains the robustness of droop control.

Results of running simulations of this system with the optimal surface are presented. The optimal results are then compared to the results of the system operating with a fixed droop resistance. The subsystem was then implemented in a hardware-in-the-loop (HIL) system. A comparison of the simulated results to the HIL results is made.

This document is organized in the following way:

Chapter 1: Background - This chapter provides an overview of the topics this research explores. Included is an overview of dc microgrids, energy storage and supercapacitors, droop control, and optimal control.

Chapter 2: Microgrid Under Study - This chapter describes the microgrid model being studied. It includes system schematics along with mathematical models governing the system.

Chapter 3: Optimization of the Microgrid - This chapter gives a derivation of the cost function along with the rationale behind it. A simplified algebraic model of the system is presented. System operation using the algebraic model is presented. Chapter 3 also explains how the optimal surface is derived using the algebraic simplification.

Chapter 4: Optimal Surface Control of Microgrid - This chapter includes results of using the optimal surface to control the system. A comparison of the results of using the optimal droop surface to that of linear droop is made. Chapter 4 also compares simulated results to the microgrid implementation using a HIL system.

Chapter 5: Summary - This chapter provides a summary of the work. The benefits associated with the system and drawbacks are discussed. Recommendations for future related work are included.

Chapter 1

Background

This research focuses on optimization of a dc microgrid through control of energy storage sources, specifically supercapacitor devices. Background material on dc microgrids, energy storage, supercapacitors, droop control, and optimal control are presented.

1.1 Dc Microgrids

A microgrid is a system of interconnected sources and loads operating to provide power to a localized area [3]. The use of microgrids offers some unique benefits to system operation. Among these benefits are opportunities for increased reliability, potential for lower costs, and potential for lower environmental impacts due to a higher penetration of renewable

energy [1]. In a microgrid, each source and load has an effect on the system performance, giving the ability to perform system optimization using fewer elements.

This research focuses on control of microgrid sources based on an optimal solution to a known system architecture. The use of dc microgrids has been pursued due to a number of reasons. Many loads use dc power as their end electrical load, and it has been shown that dc distribution can lead to a more efficient distribution [4]. Dc distribution has been shown to improve overall system efficiency in data centers with high step down dc/dc converters [5]. The implementation of droop control is simplified using dc distribution as ac droop requires two variables (voltage and frequency) along with a phase locked loop, while dc distribution only requires bus voltage and the line current to implement voltage droop control [6].

1.2 Energy Storage and Supercapacitors

Energy storage in microgrids has been studied extensively. It has been shown that energy storage in a microgrid with renewables can help support power quality [7]. Previous work has been done investigating a generalized approach to sizing of energy storage for systems with high renewables [8].

For this thesis, a model of an energy storage source is used to develop an optimal control solution. An electric double layer capacitor (also known as a supercapacitor or

ultracapacitor) is a type of capacitor characterized by a high capacitance and low parallel resistance [9]. The energy storage source used in this analysis utilizes a mathematical model of a supercapacitor bank; however, the optimization performed could be generalized to multiple energy storage models. The choice of energy storage model chosen for an interface to a converter makes a difference in optimizing the design of the interface [10]. For this a supercapacitor was chosen primarily due to its high ability be cycled (charged and discharged) compared to a battery [11][12].

The model used for the supercapacitor is the standard model for a capacitor, [13] and the state of charge can be determined by (1.1). Supercapacitors have been shown to be able to support renewable energy shortages well due to their high power capability and higher round-trip efficiency [14].

$$SOC = \frac{v_{bank}^2}{V_{bankMax}^2} \tag{1.1}$$

1.3 Droop Control

Droop control is a method of distributed control of sources that is used in microgrids [15]. Droop control is a method of control that allows for multiple sources to feed into a common bus without any form of communication between the sources [16]. The name droop control

relates to the fact that it allows for a system parameter to drop, or droop, within a tolerance during system operation. In ac systems usually two parameters, frequency and voltage, are allowed to droop [17]. In dc systems there is only one parameter that drops, the dc link voltage [18]. Although droop control provides high reliability and robustness because of this independent control, some drawbacks exist [6]. Droop control does not always take into account more specific goals of the system. Previous work has used droop control for control of load sharing based on the SOC of energy storage elements in microgrids [19]. In this research, a cost function is developed based on desired goals of the system, and some of the droop control parameters are varied in order to minimize the cost function.

Dc voltage droop control uses an error function that is input into a feedback loop that then controls the power electronic devices interfacing the sources to the grid. The error function takes into account the voltage of the bus and the line current of the source. The error function for dc voltage droop is

$$e(t) = \left(\frac{1}{R_{droop}}\right)(V_{nom} - v_{bus}) - i_b \tag{1.2}$$

In steady-state, with the feedback loop being controlled by a proportional-integral controller, this system will follow the relationship demonstrated visually in Fig. 1.1. In Fig. 1.1 the voltage decreases under load from the nominal bus value of V_{nom}. Also it shows how the two sources will share the current when following the control given in (1.2). The parameter, R_{droop}, is known as the droop resistance as the units of this control parameter

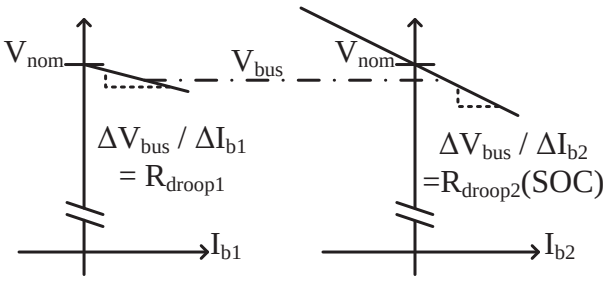

Figure 1.1: Dc voltage droop for the traditional source and the energy storage source.

are ohms (Ω). The droop resistance is inversely proportional to the percent of the load that the source will share (a higher droop resistance for a source results in the source supplying less current, and a lower resistance value yields more current).

In this research, the droop parameter R_{droop} for an energy storage source is controlled optimally, and the droop control parameter for a conventional source is left to follow linear percentage droop. Linear percentage droop is defined here as holding the R_{droop} value as a constant, with the value of R_{droop} designed such that the droop parameter will drop a designed percentage from the nominal at full load. For this research the optimal R_{droop} for the energy storage source is not constant and varies based on system operation. The optimal R_{droop} function is programmed into the controller for the energy storage source prior to system operation and is then implemented based on local information only, which maintains the reliability of the system by not requiring a communication network [20, 21].

1.4 Optimal Control

Optimal control is a branch of control theory that deals with implementing system control that yields the optimal output. The optimal output is defined quantitatively by a cost function that mathematically describes what is considered optimal [22]. Optimal control for microgrids has been performed in many different ways. An open loop optimal control has been designed to control energy storage for wind farms [23]. In [23], Teleke uses optimal control to control battery sources to help determine a dispatch strategy to mitigate wind farm transients. The optimal design of microgrids has been performed using optimal control in [24]. In [24], Patra uses a dynamic programming method to optimally determine the most reliable microgrid architecture. Optimal control has also been used for market bidding control of microgrids; Tsikalakis used a centralized control for optimal control of microgrids based on market policy considerations [25].

The optimization performed in this thesis minimizes a cost function based on the input power to the microgrid system. In [26] and [27] Bunker used a similar approach to minimizing a cost function. In [26] the cost function minimizes a cost function to better support variability created by wind energy. In [27] Bunker uses a non-linear droop reference to implement an optimal solution. The work in this thesis also uses an optimization through control of the droop resistance, but the SOC of the energy storage element is used as an input to the calculation.

Chapter 2

Microgrid Under Study

The microgrid being investigated for this study is shown in Fig. 2.1. The microgrid is made up of two sources, a purely resistive load, one energy storage source, and a capacitance on the dc bus. The resistive load in Fig. 2.1 is only one load but can be representative of the aggregate of all of the loads of a system.

The first source in Fig. 2.1 is the renewable energy source, represented by a dependent current source, i_{RE}. This current source is designed to be representative of a renewable energy source. The second source in Fig. 2.1 is the traditional power source, source 2. This

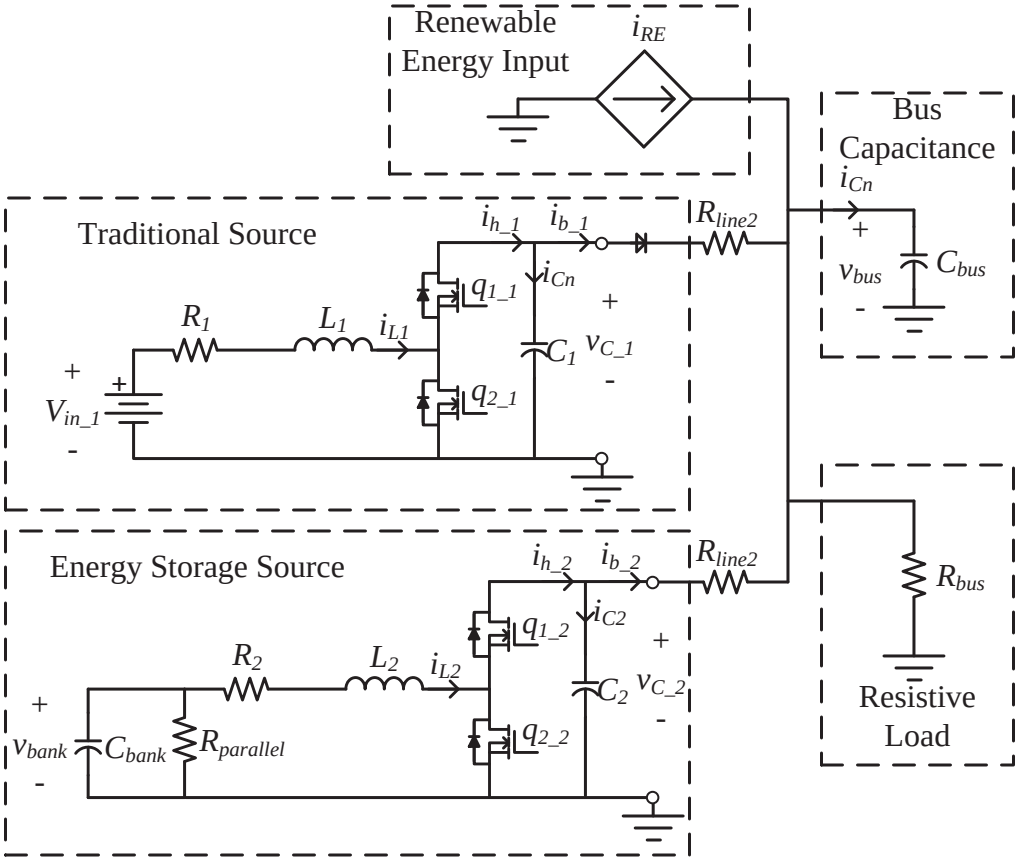

Figure 2.1: System diagram of a dc microgrid with sources and loads. Sources include a renewable energy source, represented by i_{RE}, a traditional source, and an active energy storage source, both connected to the bus through dc/dc converters. The load, R_{bus}, is a purely resistive load.

source can be defined by

$$\frac{di_{L1}(t)}{dt} = \frac{V_{in1} - R_1 i_{L1}(t) - q_{11}(t) v_{C1}(t)}{L_1} \tag{2.1}$$

$$\frac{dv_{C1}(t)}{dt} = \frac{i_{L1}(t) q_{11}(t) - i_{b1}(t)}{C_1} \tag{2.2}$$

$$i_{b1}(t) = \frac{v_{C1}(t) - v_{bus}(t)}{R_{line1}}. \tag{2.3}$$

The traditional source is represented by a constant voltage source on the input feeding the bus through a dc/dc converter operating in boost configuration. The traditional source can be representative of a few different scenarios. It could be a generator, a grid-connected, full-bridge rectifier with a capacitive coupling on the output, or the ouput of another dc/dc converter. For this analysis, the traditional source is considered undesirable to use due to fuel costs, electricity costs, or environmental costs.

The energy storage source in Fig. 2.1 is represented as a model of an energy storage source on the input of a bidirectional dc/dc converter. This analysis is performed with a supercapacitor energy storage bank; however, the theory could be adapted for many other types of energy storage with the model being changed appropriately. The dc/dc converter is similar to that of the traditional source with the only difference being that the input voltage is dependent upon the voltage from the supercapacitor model output. The supercapacitor model is based on the standard model of a capacitor [13]. It is however assumed that the parallel resistance is high enough to be negligible. The supercapacitor and the dc/dc converter are modeled as

$$\frac{di_{L2}(t)}{dt} = \frac{-R_2 i_{L2}(t) + v_{bank}(t) - q_{12}(t)v_{C2}(t)}{L_2} \tag{2.4}$$

$$\frac{dv_{C2}(t)}{dt} = \frac{i_{L2}(t)q_{12}(t) - i_{b2}(t)}{C_2} \tag{2.5}$$

$$\frac{v_{bank}(t)}{dt} = \frac{-i_{L2}(t) - \frac{v_{bank}(t)}{R_{parallel}}}{C_{bank}} \tag{2.6}$$

$$i_{b2}(t) = \frac{v_{C2}(t) - v_{bus}(t)}{R_{line2}}. \tag{2.7}$$

The load for this microgrid is a purely resistive load. The resistive load is a representation of the aggregate of all of the loads across the system. The tolerance on the bus voltage is within five percent. When the bus voltage is within tolerance, the difference between constant power loads and purely resistive loads is negligible.

The capacitor on the dc bus represents the sum of all of the capacitance on the dc bus. The bus voltage is determined by the sum of the currents of the bus, as described by

$$\frac{dv_{bus}(t)}{dt} = \frac{i_{RE} + i_{b1}(t) + i_{b2}(t) - \frac{v_{bus}(t)}{R_{bus}}}{C_{bus}}. \tag{2.8}$$

An over-voltage protection circuit is enabled on the grid to prevent the voltage on the bus from going too high. This over-voltage protection works by switching in a resistive element across the bus in order to draw current and bring down the voltage. This grid has a high penetration of variable generation, and there may be times during which i_{RE} is greater than the sum of load currents. The line impedances are resistive as the system is a low voltage dc system, and the inductive portion of the line impedance is neglected. With this, the physical system in Fig. 2.1 is fully represented by (2.1) - (2.8).

Chapter 3

Optimization of the Microgrid

The system presented in Chapter 2 can be controlled to meet desired objectives. Through the proper determination of the cost function, the optimal solution can help in making a system meet desired goals. This chapter first explains the development of a cost function. Next, an algebraic simplification of the system is used to determine the optimal control. The procedure for determining the optimal control that is presented in this chapter can be performed at many different voltage and power levels. In this chapter, a nominal 100 V, 500 W system is used as an example.

3.1 Cost Function Development

In optimal control, a quantitative definition needs to be given as to what is optimal. This is referred to as the cost function [22]. Developing a cost function involves analyzing the goals of the system.

The goals of the design are

- Maintaining the stability of the bus voltage
- Minimizing the energy from the traditional source
- Utilizing as much renewable energy as possible
- Maintaining the SOC of the energy storage source
- Maintaining a robust system.

Developing a cost function to meet the desired goals requires an understanding of the system and the control methods. Although some of the goals listed need to be directly addressed by the cost function, some goals may already be addressed due to the nature of the control method used. Therefore, design of the cost function should focus on the goals not already achieved through normal operation.

If the constraint is made that the sources are to be following dc voltage droop control, some simplifications can be made about the cost function. Using droop control keeps the robustness of the system because it allows the sources to share the load independent of a communication network. Another assumption for the system is that the power rating of the traditional source is high enough such that it can source all of the load independently. Also it will be assumed that the traditional source will implement linear droop control with the droop resistance based on drooping a specified percent at full power. This is similar to traditional droop control for ac systems, as discussed in Chapter 1. Having this constraint on the control of the traditional source also ensures the stability of the dc voltage on the bus. Defining the control for the traditional source allows the optimization to be performed on the control of the energy storage source converter, simplifying the analysis.

With the constraints defined, there are now two goals that it will have to account for: minimizing the energy from the traditional source, and maintaining the SOC of the energy storage source. Both of these goals relate to the power used from both sources, leading to the cost function being defined as

$$J = P_{Source1} + P_{Source2} = V_{in1} i_{L1}(t) + V_{bank1}(t) i_{L2}(t). \tag{3.1}$$

This cost function is a direct sum of the input power from the conventional source and the input power from the energy storage source. In designing a cost function, often a weighting factor is applied to the different parts of the cost function [28]. For this cost function, a

weighting factor could be used to force the optimal solution to favor using one source or another. For instance, a weighting factor could be used if it were more desirable to maintain the SOC of the energy storage than to use less power overall. Because no weighting factor is used, the cost function looks at the input power from both sources equally, making the cost equal to the total combined input power from the two sources. Having the traditional source's control set as linear droop and the cost function defined leaves the optimization to rely on using the droop settings for the energy storage source.

3.2 Dc Voltage Droop Control

Droop control is the method of control that is to be used on both controlled sources. The traditional source will follow linear percentage droop. The implementation of the droop control for the conventional source is

$$e_1(t) = \left(\frac{1}{R_{droop1}}\right)(V_{nom} - v_{bus}) - i_{b1}. \tag{3.2}$$

The error function $e_1(t)$ is then corrected using a PI loop

$$d_1(t) = x_{i1} + k_{p1}e_1(t) \tag{3.3}$$

$$x_{i1} = k_{i1}\int e_1(t)dt \tag{3.4}$$

where x_{i1} is the state associated with the integral portion of the control feedback and d_1 is the duty cycle of the converter for source 1. For the system it is assumed that the energy storage source will also follow droop control implemented similarly; however, the droop resistance gain R_{droop2} will be a function of the SOC of the energy storage source. The droop control and error feedback for source 2 are implemented as

$$e_2(t) = \left(\frac{1}{R_{droop2}(SOC)}\right)(V_{nom} - v_{bus}) - i_{b2} \tag{3.5}$$

$$d_2(t) = x_{i2} + k_{p2}e_2(t) \tag{3.6}$$

$$x_{i2} = k_{i2}\int e_2(t)dt, \tag{3.7}$$

where x_{i2} is the state associated with the integral portion of the control feedback and d_2 is the duty cycle of the converter for the supercapacitor bank. It is assumed that both sources have access to the bus voltage values v_{bus} either through estimation or through direct measurement.

The droop control for the two sources is shown in Fig. 3.1. Note that the traditional source does not allow for negative current. Given the system type and the fact that the error is controlled by a PI loop, this system will approach the relationship shown in Fig. 3.1 with zero steady-state error. The droop control will ensure that the two sources will share the load inversely proportional to their respective droop resistance values.

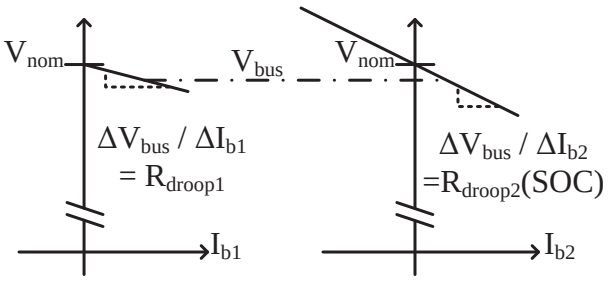

Figure 3.1: Dc voltage droop for the traditional source and the energy storage source.

3.3 Optimization

3.3.1 Short-Term Steady-State Approximation

The full differential-equation-based model presented in Chapter 3 contains six states. The control of the system, (3.2) - (3.7), contains two additional states. Combining the physical system model with the control leads to a nonlinear eight-state system. To simplify the optimization, some reasonable assumptions about the time constants are used. Generally when designing dc/dc converters with power electronics, the inductors and capacitors are designed to be as low of a value as possible due to economic reasons. This also aids in making the time constants of these devices small in comparison to other time constants of the system. Generally, the energy storage in a system will be operating at a time scale that is much greater than the elements used for energy conversion. This leads to the system being able to be broken into two categories for time constants associated with the states. The

first category of states includes the source inductors L_1, and L_2, and the source capacitors C_1, and C_2. The bus capacitance, C_{bus}, also falls into the same time constant category. The second state category is associated with relatively large time constants. The only state in this category is the voltage associated with the energy storage capacitor. Category 1 contains the physical system states: i_{L1}, i_{L2}, v_{C1}, v_{C2}, and v_{Cbus}. Category 2 contains the signal v_{bank}.

Dividing the system into two discrete categories allows for an algebraic simplification to be made. If the time frame of interest is slower than the time constants in category 1 but faster than the time constants in category 2, all of these states can be approximated by algebraic equations. This will be referred to as short-term steady-state (STSS). The system is represented schematically at STSS as shown in Fig. 3.2.

The algebraic simplification assumes that the states in category 1 have already reached steady-state, which makes the derivatives with respect to time equal to zero. The state, v_{bank}, in category 2 is assumed to not change during the time frame that is of interest. This makes the value of that state equal to a constant. At STSS, the model of the physical system

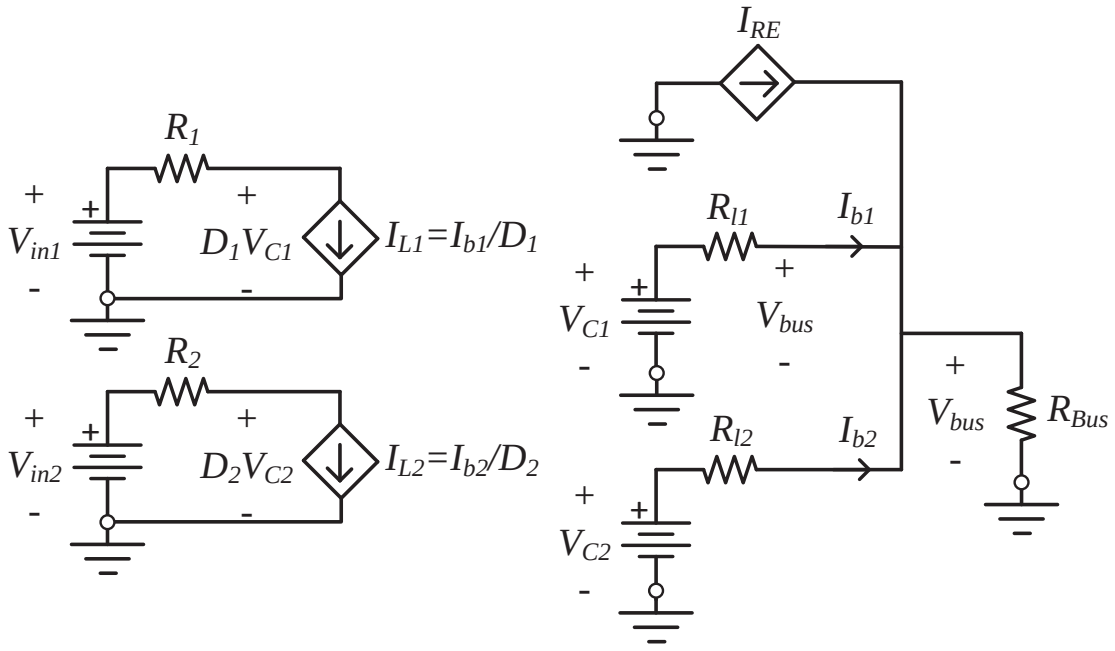

Figure 3.2: Diagram of the microgrid system simplified to short-term steady-state.

is represented by

$$0 = \frac{V_{in1} - R_1 i_{L1}(t) - q_{11}(t) v_{C1}(t)}{L_1} \tag{3.8}$$

$$0 = \frac{i_{L1}(t) q_{11}(t) - i_{b1}(t)}{C_1} \tag{3.9}$$

$$i_{b1}(t) = \frac{v_{C1}(t) - v_{bus}(t)}{R_{line1}} \tag{3.10}$$

$$0 = \frac{-R_2 i_{L2}(t) + v_{bank}(t) - q_{12}(t) v_{C2}(t)}{L_2} \tag{3.11}$$

$$0 = \frac{i_{L2}(t) q_{12}(t) - i_{b2}(t)}{C_2} \tag{3.12}$$

$$v_{bank} = V_{bank} \tag{3.13}$$

$$i_{b2}(t) = \frac{v_{C2}(t) - v_{bus}(t)}{R_{line2}}. \tag{3.14}$$

The control of the two sources for the system is already assumed to be droop control as discussed in section 3.2. The equations for the control can be simplified to be at STSS as well. The states for the control, x_{i1} and x_{i2}, fall into category 1, the faster time constants. This is assuming that the control gains are tuned appropriately to allow for a relatively fast response. At steady-state, the error terms, $e_1(t)$ and $e_2(t)$, are zero due to the PI feedback. At STSS, the error terms become

$$0 = \left(\frac{1}{R_{droop1}}\right)(V_{nom} - v_{bus}) - i_{b1} \tag{3.15}$$

$$0 = \left(\frac{1}{R_{droop2}(SOC)}\right)(V_{nom} - v_{bus}) - i_{b2}. \tag{3.16}$$

With this, the system is approximated to a series of algebraic relationships.

3.3.2 Short-Term Steady-State System Operation

Now that the system has been algebraically approximated at short-term steady-state, the STSS equations can be re-written into a form that makes analysis more accessible. Because the optimization has been chosen to be performed using the control on source 2, one analysis that is of interest is what effect the value of the droop parameter R_{droop2} has on the system operation. Given a fixed load resistance, R_{bus}, and the assumption that the other parameters of the system are held constant, R_{droop2} is used as an input to the system. (3.8)

- (3.16) can be rearranged to

$$I_{L1} = \frac{b_1 - \sqrt{b_1^2 + 4a_1c_1}}{2a_1} \tag{3.17}$$

$$a_1 = R_1(R_{bus} + R_{droop1}) \tag{3.18}$$

$$b_1 = V_1(R_{bus} + R_{droop1}) \tag{3.19}$$

$$c_1 = \frac{((I_{b2} + I_{RE})R_{bus} - V_{nom})\left(R_{bus}(I_{b2} + I_{RE})(R_{droop1} - R_{l1}) + V_{nom}(R_{bus} + R_{l1})\right)}{R_{bus} + R_{droop1}} \tag{3.20}$$

$$I_{L2} = \frac{b_2 - \sqrt{b_2^2 + 4a_2c_2}}{2a_2} \tag{3.21}$$

$$a_2 = R_2(R_{bus} + R_{droop2}) \tag{3.22}$$

$$b_2 = V_2(R_{bus} + R_{droop2}) \tag{3.23}$$

$$c_2 = \frac{((I_{b1} + I_{RE})R_{bus} - V_{nom})\left(R_{bus}(I_{b1} + I_{RE})(R_{droop2} - R_{l2}) + V_{nom}(R_{bus} + R_{l2})\right)}{R_{bus} + R_{droop2}} \tag{3.24}$$

$$I_{b1} = \frac{V_{nom} - R_{bus}(I_{b2} + I_{RE})}{R_{bus} + R_{droop1}} \tag{3.25}$$

$$I_{b2} = \frac{V_{nom} - R_{bus}(I_{b1} + I_{RE})}{R_{bus} + R_{droop2}} \tag{3.26}$$

$$V_{bus} = \frac{R_{bus}(R_{droop1}(I_{b2} + I_{RE}) + V_{nom})}{R_{bus} + R_{droop1}}. \tag{3.27}$$

The equations in the form (3.17) - (3.27) allow the bus voltage (v_{bus}), the line currents (i_{b1} and i_{b2}), and the input currents (i_{L1}, i_{L2}) to be determined for any given R_{droop2} value. To extend this analysis further, plots have been generated showing the system outputs based

on varying the R_{droop2} value.[1] A Matlab script was used to calculate (3.17) - (3.27) for an array of values of R_{droop2}. The script used for this is shown in Appendix A. Fig. 3.3 shows the bus voltage V_{bus} for varying values of R_{droop2}. It is seen that the bus voltage decreases for an increasing R_{droop2} value, but the voltage stays within the limits of 95 V and 105 V. Fig. 3.4 and Fig. 3.5 show the input current and line currents respectively for varying R_{droop2}.

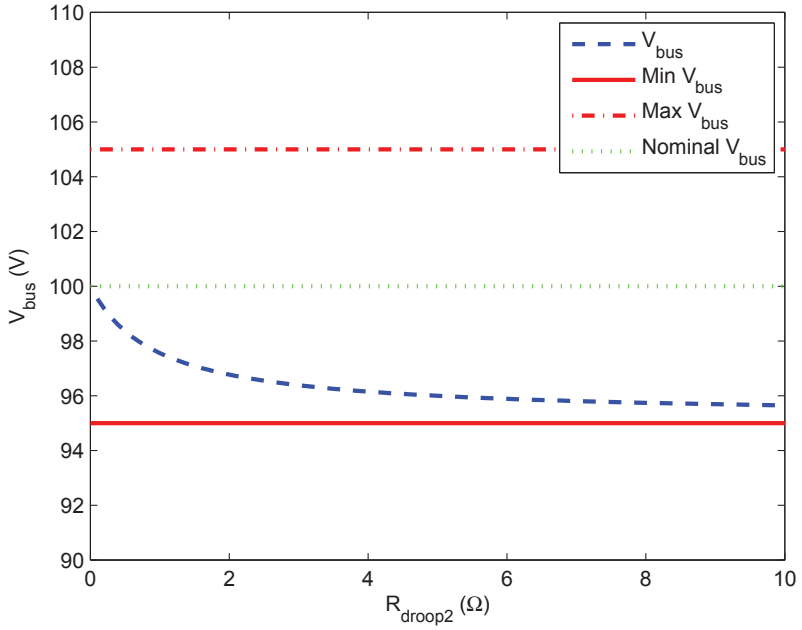

Figure 3.3: Effect of the source 2 droop resistance, R_{droop2} on the bus voltage V_{bus}.

A calculation that is of interest is the resulting input power and output power as a function of R_{droop2}. Fig. 3.6 shows both the combined input power (the numerical sum of $V_1 I_{L1}$ and $V_{bank} I_{L2}$) and output power vs. R_{droop2}. From the plot in Fig. 3.6, the minimum combined input power, which is also the minimum of the cost function from (3.1), can be found. For

[1]These plots are based on system values for the low voltage system example in Chapter 4

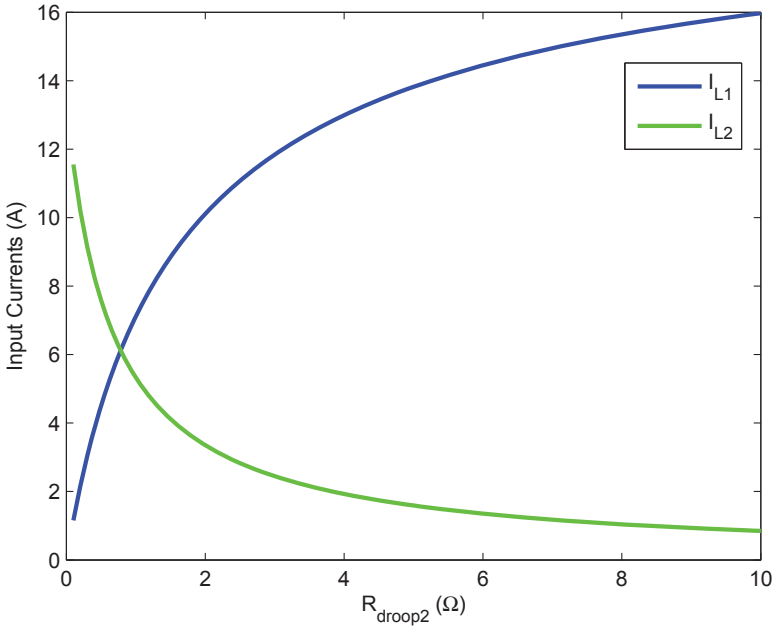

Figure 3.4: Effect of the source 2 droop resistance, R_{droop2} on the input currents I_{L1} and I_{L2}.

this example, the minimum combined input power is shown in Fig. 3.6 as a marked point. The R_{droop2} value that this occurs at is the R_{droop2} value that results in the optimal control.

Fig. 3.6 shows the optimal control value R_{droop2} for a specific load resistance R_{bus} and source 2 input voltage V_{bank}. However, using the STSS model, the analysis can be repeated for many different values of R_{bus} and V_{bank}. Fig. 3.7 shows the optimal value for five different values of V_{bank}. Using the STSS model developed, the optimal R_{droop2} value for any combination of load resistance, R_{bus}, and source 2 input voltage, V_{bank}, can be determined.

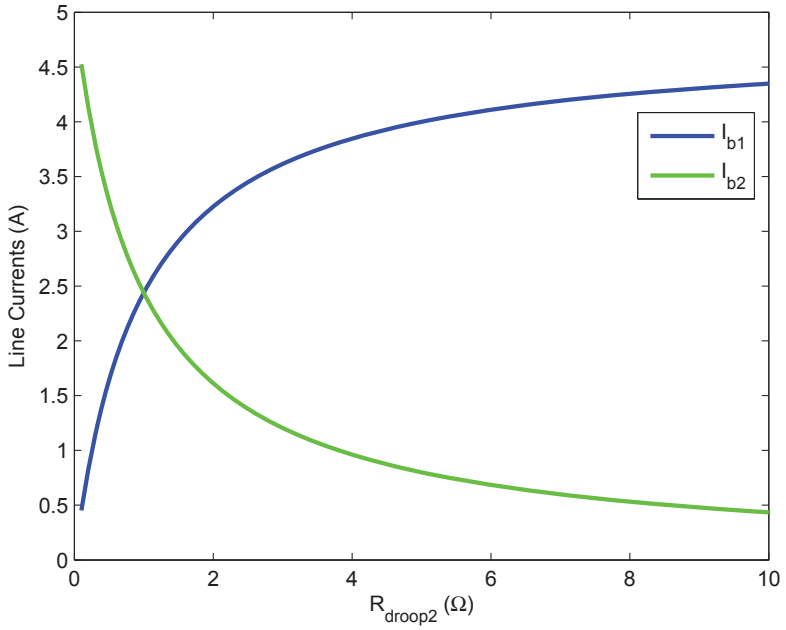

Figure 3.5: Effect of the source 2 droop resistance, R_{droop2}, on the line currents I_{b1} and I_{b2}.

3.3.3 Optimal Surface

Now that it has been shown that the optimal R_{droop2} value can be determined for any combination of R_{bus} and V_{bank}, the optimal R_{droop2} value for all reasonable combinations is calculated. This results in a two-dimensional array of values that can be represented as a surface as shown in Fig. 3.8.[2]

Fig. 3.8 is significant because it is the surface that represents the most optimal value of R_{droop2} for every SOC and power level. This surface could be programmed into the

[2]It should be noted that the source 2 input voltage has been converted to SOC and the load resistance has been converted to effective load, based on the nominal bus voltage.

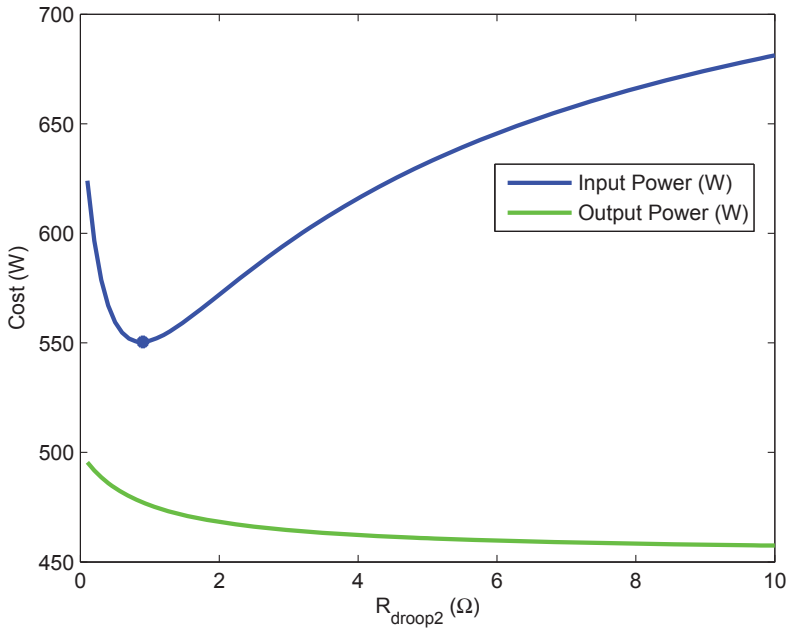

Figure 3.6: Effect of the source 2 droop resistance, R_{droop2} on the input power and output power of the system.

controller for source 2, and as long as the controller was able to obtain the state of charge of the supercapacitor bank and able to determine the output power level, the system would operate optimally given the constraints discussed previously.

3.4 Optimal Surface Inputs

The optimal R_{droop2} surface developed is only useful if it is able to be implemented. As discussed in Chapter 1, the benefit to using droop control is the ability to control the sources using only locally available information. The optimal surface developed allows for determination of the optimal R_{droop2} only if the SOC of the energy storage is known

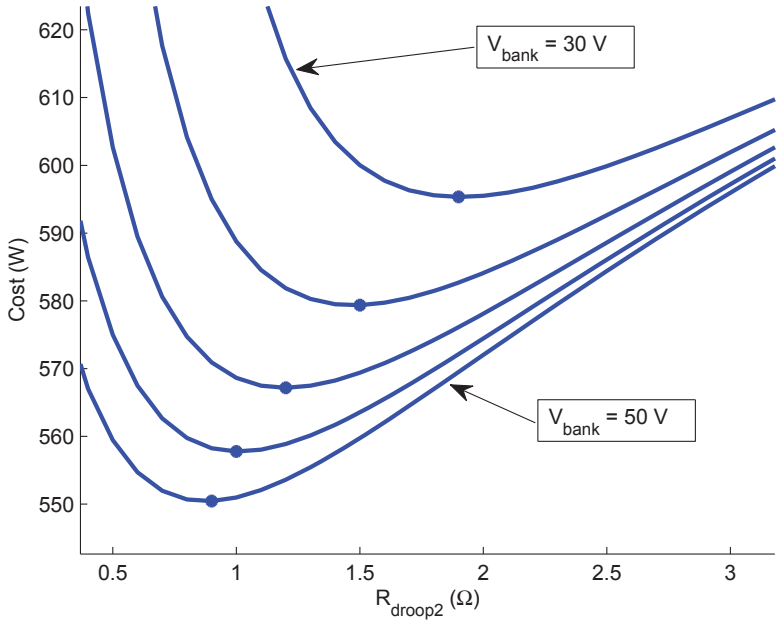

Figure 3.7: Effect of R_{droop2} on cost for different values of SOC. Shown are the optimal points for V_{bank} values of 30, 35, 40, 45, and 50 V.

and the output resistive load of the system is known. Both of these values are obtainable via local measurements.

3.4.1 Obtaining the SOC of the Energy Storage Source

The SOC of the energy storage can be calculated as described in (1.1). The value that needs to be measured in order to calculate SOC is the supercapacitor bank voltage. The assumption that the controller for source 2 is able to obtain this voltage is not an unreasonable assumption. This is because there are multiple methods of obtaining the supercapacitor bank voltage. The first method is for controller source 2 to use a direct

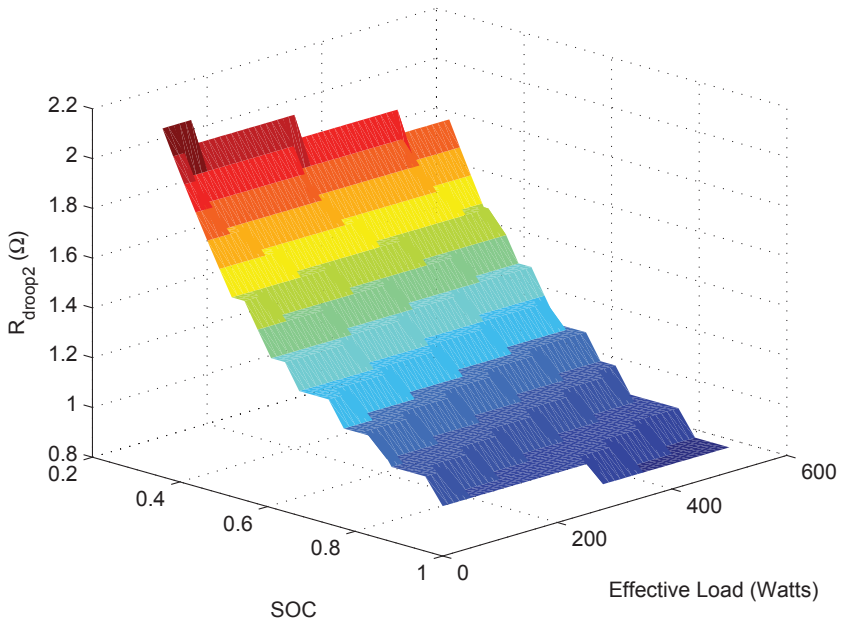

Figure 3.8: The optimal R_{droop2} surface based on the SOC of source 2 and the effective load on the bus.

measurement of the supercapacitor bank voltage. This is reasonable because very often the source converter would be placed in close proximity to the energy storage source. The second method of obtaining the supercapacitor bank voltage is to calculate it by using the input current. If controller 2 is able to read the voltage at the input of the IGBT leg and is able to read the input current, the supercapacitor bank voltage can be obtained using

$$v_{bank} = v_q + i_{L2}R_2 \tag{3.28}$$

where the voltage at the input of the IGBT leg is labeled as v_q. This method requires source 2 to have an accurate estimate of R_2.

3.4.2 Obtaining the Effective Output Load

The next value to be obtained is the effective output resistive load of the system, R_{bus}. For all of the examples used to determine the optimal control surface, it was assumed that I_{RE} was zero. However, this is usually not the case. The output load is an input used in the optimal controller. The contribution of I_{RE} changes what this output load is. This seems to imply that the renewable energy input I_{RE} would need to be measured in order to be able to implement the optimal controller. However, the actual load is not what is needed by the controller. Rather, the controller needs the effective load, R_{eff}, as defined by

$$P_{effective} = P_{load} - P_{RE} = \frac{V_{bus}^2}{R_{load}} - V_{bus} I_{RE} = \frac{V_{bus}^2}{R_{eff}}. \tag{3.29}$$

Following the assumption that the system is at STSS and using Ohm's law, the relationship

$$I_{b1} + I_{b2} + I_{RE} - \frac{V_{bus}}{R_{load}} = 0 \tag{3.30}$$

can be obtained. Combining (3.29) with (3.30) yields

$$\frac{V_{bus}^2}{R_{eff}} = (I_{b1} + I_{b2}) V_{bus}. \tag{3.31}$$

Equation (3.31) is significant because it shows that the effective load, R_{eff}, can be obtained provided that the bus voltage, v_{bus}, and the two line currents, i_{b1} and i_{b2}, are measurable.

It is already assumed that the controller for source 2 can read its own line current, I_{b2}, and the bus voltage, V_{bus}. This means that the controller can implement the optimal controller provided it has I_{b1}. It is assumed that the controller for source 2 does not have a direct measurement of I_{b1}. However, an estimate of I_{b1} can be made by controller 2 assuming it knows some information about the control method of source 1 prior to operation. Controller 1 follows linear droop control defined by (3.15). Combining (3.31) and (3.15) yields

$$R_{eff} = \frac{V_{bus}}{\left(\frac{V_{nom}-V_{bus}}{R_{droop1}} - I_{b2}\right)} \tag{3.32}$$

which is an estimate of R_{eff} using only locally available measurements.

3.5 Summary of Optimization

This chapter presented the optimal control of the microgrid. First a cost function was developed based on stated goals for the system. Then the STSS model of the system was presented. This model was then used to develop an optimal surface. Finally, it was shown that the input parameters for this surface could be obtained using only local measurements. In Chapter 4 this optimal surface will be used to control source 2. The benefits of using this surface will be discussed.

Chapter 4

Optimal Surface Control of the Microgrid

In Chapter 3, the optimal R_{droop2} surface was developed. Chapter 4 focuses on demonstrating quantitatively the usefulness of using the optimal R_{droop2} surface for control. The cost function is calculated over the entire range of operation using the optimal surface, and the results are then compared to the results of using linear droop (a constant R_{droop} value). The analysis is performed for a low voltage system. Finally, the practical implementation of this system and its control are demonstrated using a hardware-in-the-loop (HIL) system.

4.1 Optimal Surface Implementation

The microgrid system presented in Chapter 2 could represent a number of different voltages and power levels. For this analysis, the system in Fig. 2.1 is used with the parameters given in Table 4.1.

Table 4.1
Low Voltage System Parameters.

Parameter	Value	Units
V_{in1}	40	V
R_1	0.8	Ω
L_1	500	μH
C_1	500	μF
R_{line1}	1.0	Ω
C_{bank}	150	F
$R_{parallel}$	10	$k\Omega$
R_2	0.8	Ω
L_2	500	μH
C_2	500	μF
R_{line2}	1.0	Ω
C_{bus}	500	μH
Nominal V_{bus}	100	volts
Max V_{bus}	105	volts
Min V_{bus}	95	volts
P_{b1}	500	W
R_{droop1}	1.0	Ω
Max v_{bank}	50	V
Min v_{bank}	30	V
Max SOC	100	%
Min SOC	36	%

With the parameters given in Table 4.1, the optimal droop surface is calculated by following the method described in Chapter 3. The surface for the system is shown in Fig. 4.1.

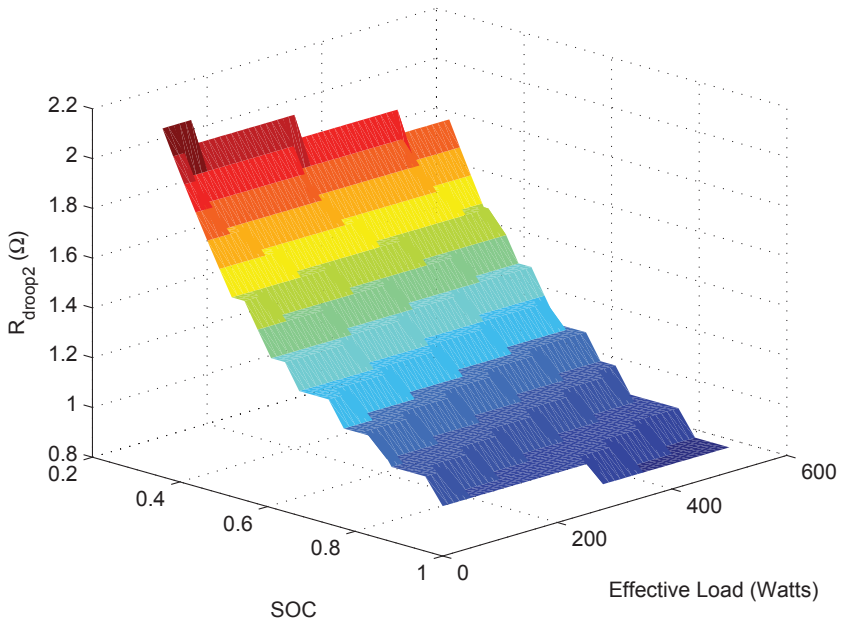

Figure 4.1: Optimal R_{droop2} surface based on the SOC of source 2 and the effective load on the bus for the low voltage system.

Implementing this surface for the control of source 2 yields the minimal combined input power. This is due to the cost function (3.1) accounting for power and the surface being an optimal solution to the cost function.

To show the benefit of this optimal surface, the results of using the optimal surface will be compared to linear droop. Linear droop implements the control for source 2 as a constant R_{droop2} value. A constant R_{droop2} would yield a surface similar to Fig. 4.1 except it would be flat, as opposed to the optimal surface generated. The cost function (3.1) addresses the combined input power from the two controlled sources. For the low voltage system, with the parameters in Table 4.1, the cost was calculated using the optimal surface. The results of this are shown in Fig. 4.2.

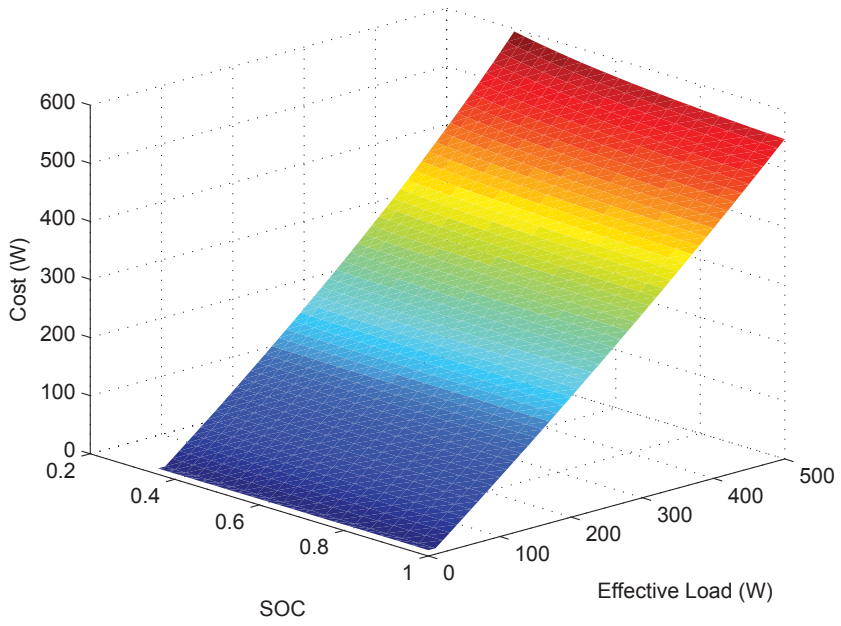

Figure 4.2: Cost calculated using the optimal R_{droop2} surface for the low voltage system.

The surface in Fig. 4.2 represents the minimal combined input power that can be achieved given the constraints of this system. A similar surface was generated using the system and a constant R_{droop2} value of 2.0 Ω. The results of this are shown in Fig. 4.3. The difference in cost between using the arbitrary droop value and using the optimal surface was calculated, and this is shown in Fig. 4.4. The percent power savings was also calculated and is shown in Fig. 4.5. Fig. 4.5 is significant because it shows not only that the optimal control minimizes the cost function but also that there is significant savings to using the optimal droop control surface rather than using an arbitrarily picked value of 2.0 Ω for R_{droop2}. Using the optimal R_{droop2} surface will always result in equal to or better than performance than a constant value for R_{droop2}.

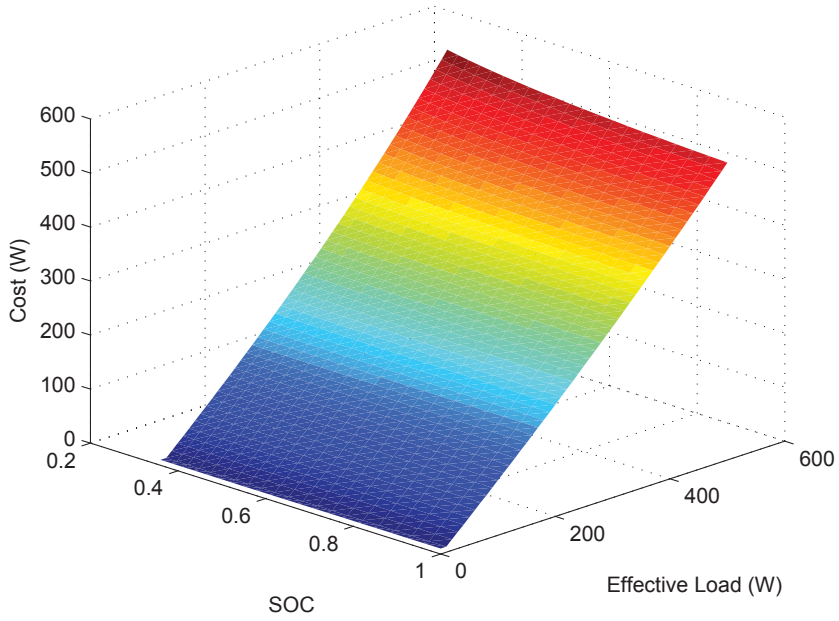

Figure 4.3: Cost calculated using an arbitrary R_{droop2} of 2.0 Ω for the low voltage system.

4.2 Hardware-in-the-Loop Implementation of System

The microgrid system presented in Chapter 2 was implemented in a Hardware-in-the-loop (HIL) system. First, initial testing was performed to verify that the model and controller implemented in the HIL system matched the model that was used in simulation. After that, verification was performed to show that the results of varying the R_{droop2} value matched what was calculated with the STSS model from Chapter 3. Finally, the optimal surface was implemented. The results of the optimal surface along with the results from using a constant R_{droop2} value are presented.

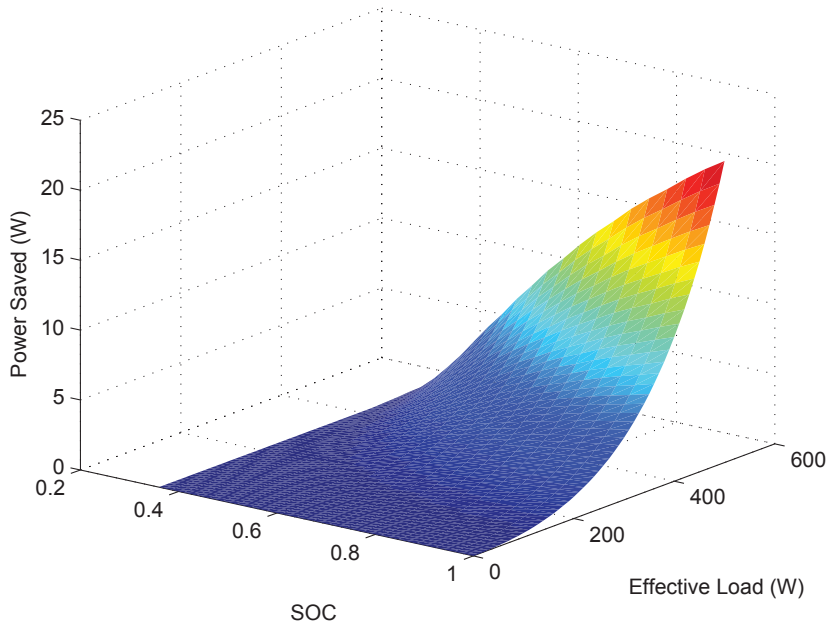

Figure 4.4: Power saved using the optimal R_{droop2} surface when compared to a constant R_{droop2} of 2.0 Ω.

4.2.1 The Hardware-in-the-loop System

The HIL system that was used for this implementation was the TyphoonHIL400 [29]. The HIL system emulates the physical microgrid system presented in Chapter 2. An image of the TyphoonHIL400 system setup is shown in Fig. 4.6. The TyphoonHIL400 allows for a circuit-based physical system model to be created in software and for the inputs and outputs of the model to be interfaced with actual hardware. The circuit-based model that was implemented in the HIL is shown in Fig. B.1 in Appendix B.1.

The actual hardware that is interfaced with the HIL system is a DSP controller. The

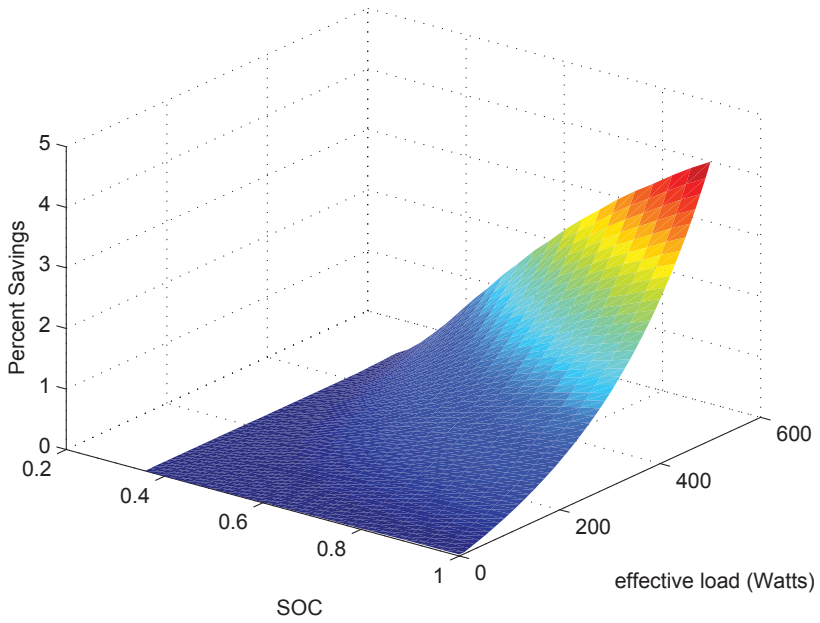

Figure 4.5: Percent power saved using the optimal R_{droop2} surface when compared to a constant R_{droop2} of 2.0 Ω.

controller for this was a dSPACE rapid prototype DSP [30]. The controller interfaces to the HIL system inputs by applying gating to the IGBTs in the sources of the HIL schematic. The HIL output analog voltages are scaled outputs of the sensors in the HIL model. These outputs are then read by the analog-to-digital converters on the DSP. Also, the scaled analog outputs from the HIL system are able to be read on an oscilloscope. The dSPACE system also has a digital-to-analog output that can be input to the HIL system to allow for control of the output load resistance and renewable energy power. This allows for the system to be able to run different load cycles.

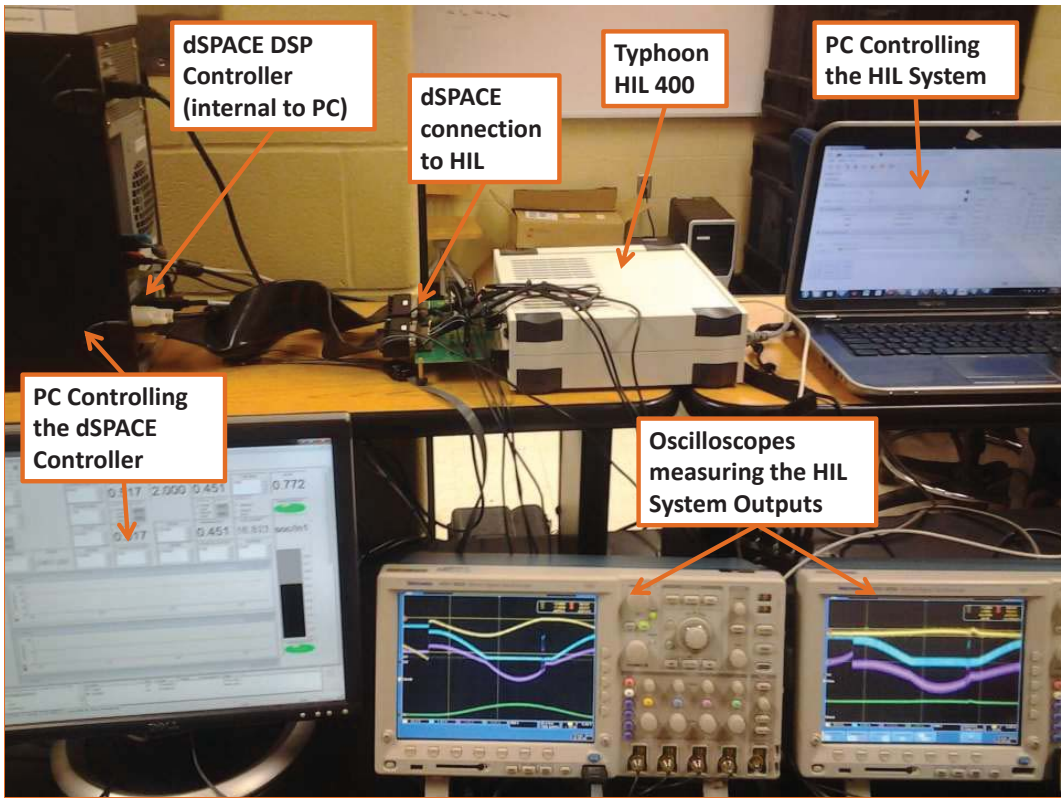

Figure 4.6: Physical setup of the TyphoonHIL400 HIL system with dSPACE controller and oscilloscopes reading analog outputs.

4.2.2 Performing the R_{droop} Sweep using the HIL system

For the development of the optimal surface presented in Chapter 2, a sweep of the droop parameter, R_{droop2} was performed, and the effect of the system outputs vs. R_{droop2} was analyzed. This same analysis was performed using the HIL system. For this analysis, the voltage of the supercapacitor bank was held constant. For all of the plots, the raw oscilloscope data has been filtered with a first-order, low-pass filter with a cutoff frequency of 10 Hz.

With the SOC at 100% and the load resistance, R_{bus}, at a constant of 20 Ω, the control parameter, R_{droop2}, was swept across a spectrum from 0.1 Ω to 4.5 Ω over a time of approximately 6 minutes. The voltage of the bus is shown in Fig. 4.9. The line currents, I_{b1} and I_{b2}, are shown in Fig. 4.10. The input currents, I_{L1} and I_{L2}, are shown in Fig. 4.11.

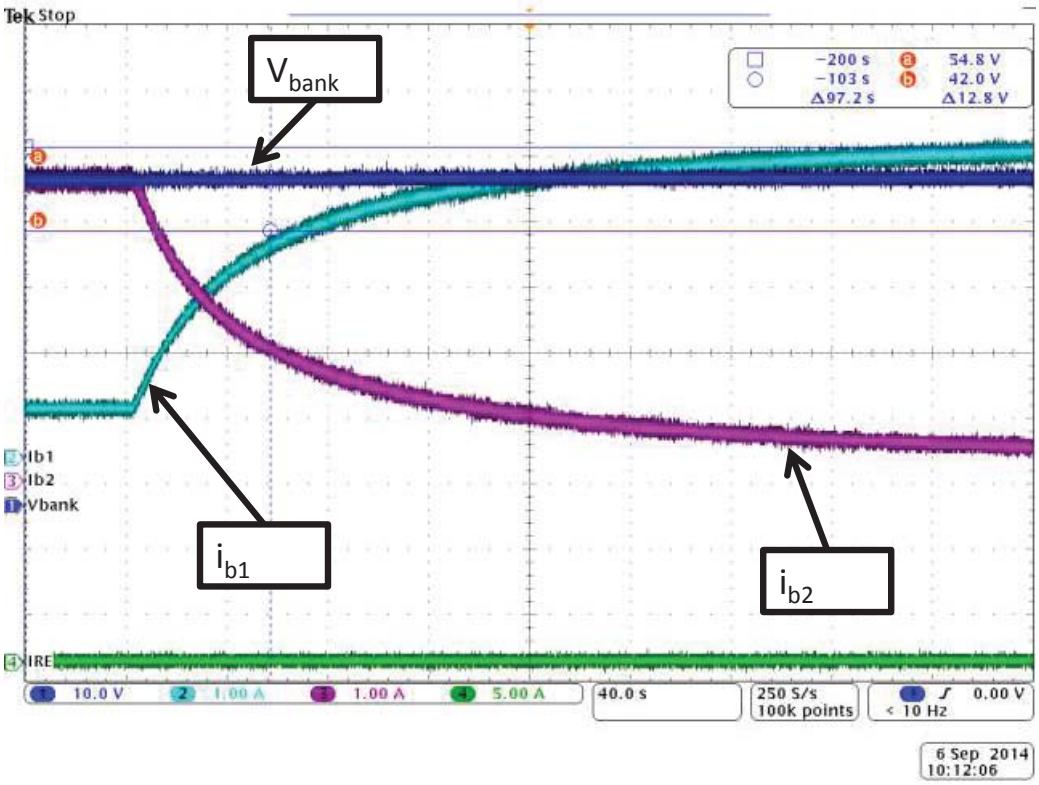

Figure 4.7: Oscilloscope results of the R_{droop2} sweep. Shown is the supercapacitor bank voltage, v_{bank}, the renewable energy input, i_{RE}, and the line currents, i_{b1} and i_{b2}.

As can be seen in Fig. 4.9 - Fig. 4.11, the results follow the same pattern as the simulated plots from Chapter 3. Using the input currents read from the oscilloscope and the input voltages, the input powers as a function of R_{droop2} can be obtained. The input powers and the combined input power, which is the cost function, is shown in Fig. 4.12.

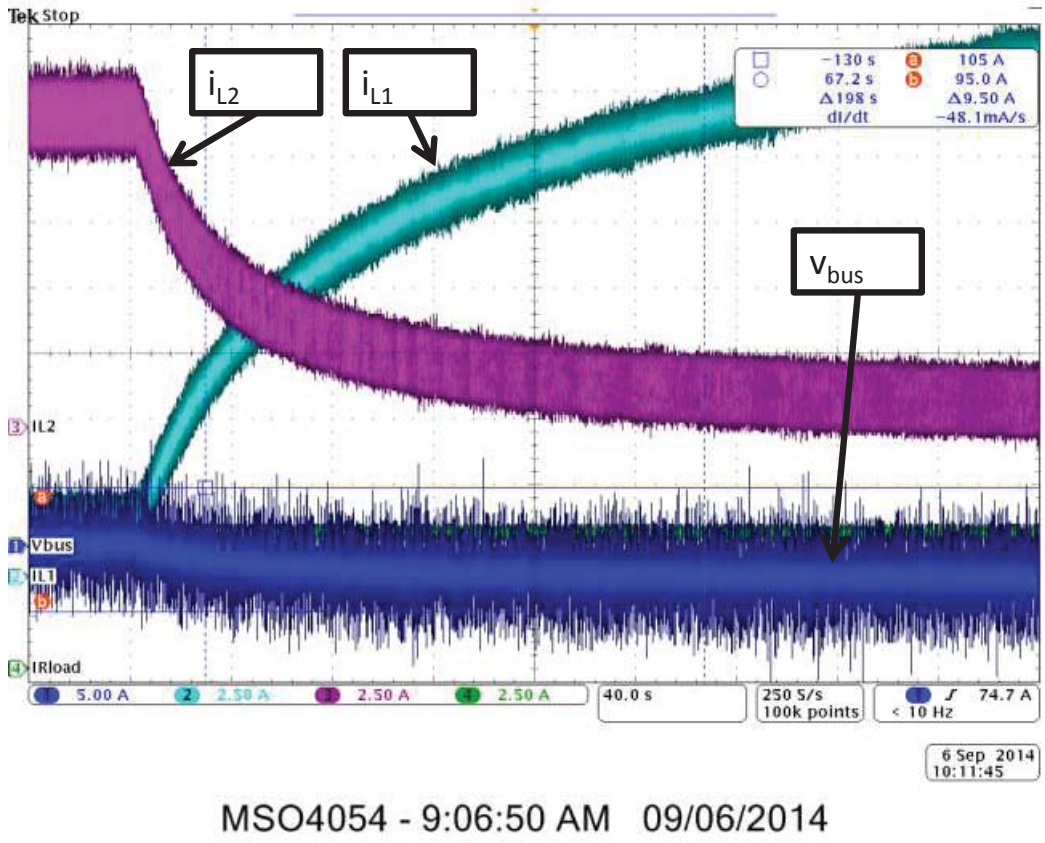

Figure 4.8: Oscilloscope results of the R_{droop2} sweep. Shown is the bus voltage, v_{bus}, the load current, i_{Rload}, and the input currents, i_{L1} and i_{L2}.

As can be seen in Fig. 4.12, the trend of the input powers obtained from the HIL implementation of the R_{droop2} sweep matches that of the simulation data from Chapter 3. The combined input power starts by decreasing as R_{droop2} increases until it reaches a minimum; after that, it only increases. The reason for this is due to the influence that the parameter R_{droop2} has on the power sharing of the load between the sources. When only one source has to share a large percentage of the load, the losses associated with the currents increase. The minimum R_{droop2} from this plot can be found to be approximately 0.7 Ω. This is fairly close to the R_{droop2} value for 100% and 500 W of power from the optimal surface. This indicates that the theoretical optimal surface should yield an optimal

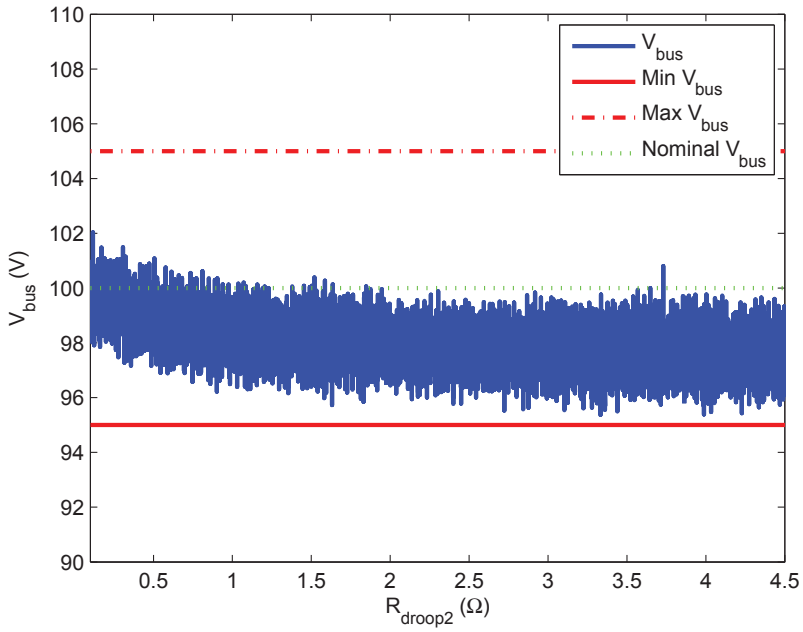

Figure 4.9: Bus voltage, V_{bus}, as a function of R_{droop2} as implemented in the HIL system.

cost when implemented in the HIL system.

4.2.3 Implementation of Decentralized Control on the HIL system

As explained in Chapter 3, the STSS model makes it possible to implement the optimal control surface with completely decentralized controllers. In order to implement the optimal surface in the controller for source 2, a measurement of the supercapacitor bank's voltage and an estimate of the output resistive load are needed. In Chapter 3 it was shown that both of these parameters could be obtained using sensor readings local to source 2. This was done by making the assumption that the system is operating at or near STSS and

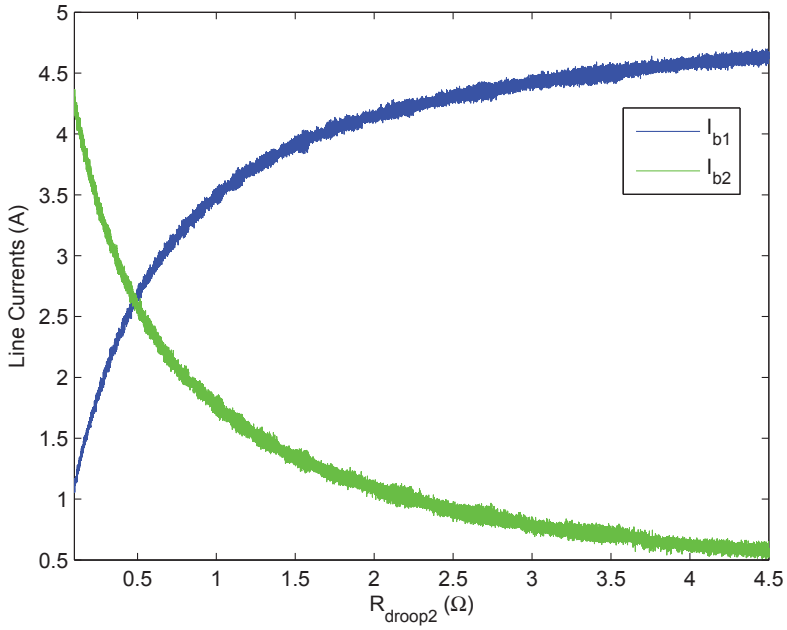

Figure 4.10: Line currents, I_{b1} and I_{b2}, as a function of R_{droop2} as implemented in the HIL system.

that the controller for source 2 knows the value of the droop resistance R_{droop1} for source 1. This section shows that the assumptions are reasonable and that this implementation is possible.

The microgrid system presented in Chapter 2 was implemented in the HIL system. Then the controller was implemented in the dSPACE rapid prototyping system. The dSPACE system allows for the DSP to be programmed using block diagrams via Matlab. The block diagram shown in Fig. 4.13 was compiled into code that was implemented on the DSP that controlled the dc/dc converter for source 2. The block diagram is based off of (3.32).

In order to make sure that the controller for source 2 could accurately estimate the resistive load of the system, a test was performed. The implementation of the HIL system allows

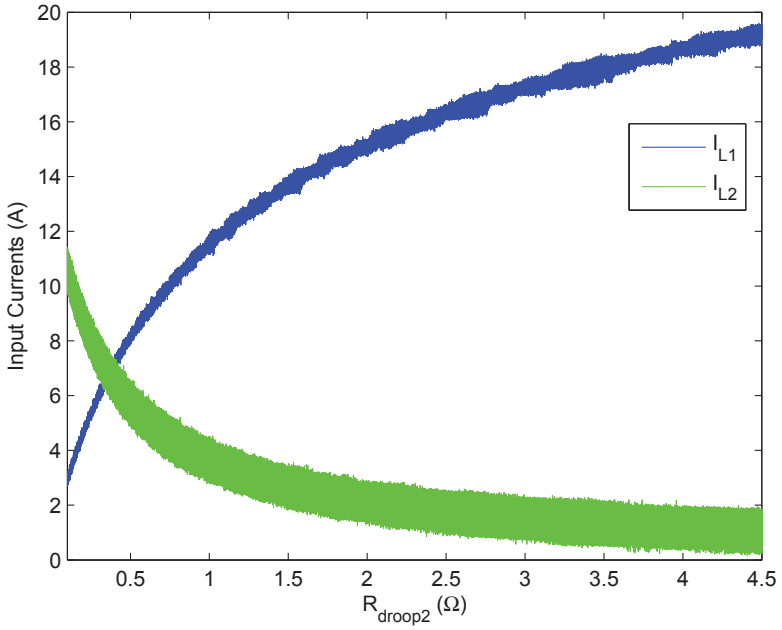

Figure 4.11: Input currents, I_{L1} and I_{L2}, as a function of R_{droop2} as implemented in the HIL system.

for the dSPACE system to control the output resistive load implemented in the HIL via a digital-to-analog converter. It also is able to read the R_{eff} value that is calculated in controller 2 due to the fact that both the controller and the digital-to-analog output were both implemented in the same DSP. This allows for the two values to be measured directly over time and a comparison to be made between the two. The HIL was run using linear droop, and the actual resistive load, R_{bus}, and the effective load, R_{eff}, were both measured. The actual load R_{bus} was varied over a cycle, and R_{eff} was measured to determine how well it tracked to the load. A plot of this run is shown in Fig. 4.14. As shown in Fig. 4.14, the controller for source 2 is able to estimate the effective resistive load of the system accurately. It should be noted that the implementation of the load estimation included the use of three low-pass filters that were experimentally determined.

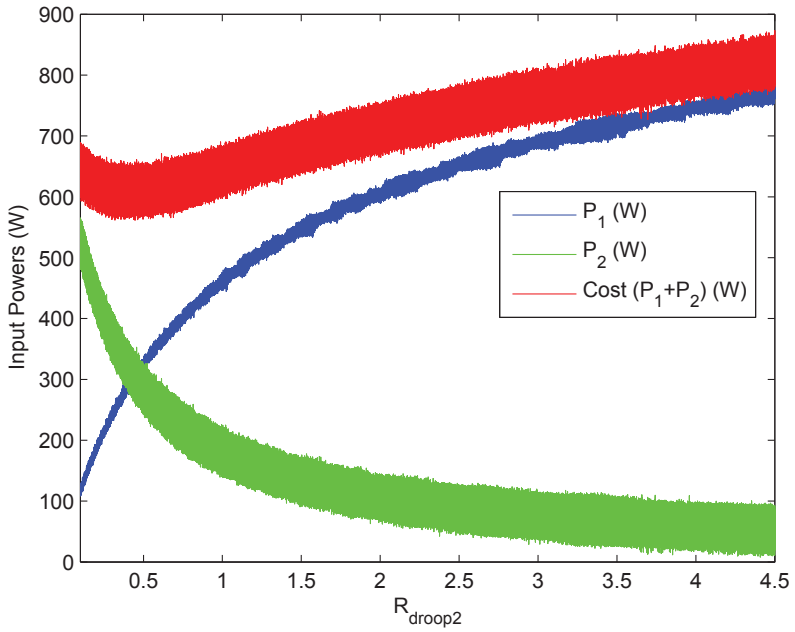

Figure 4.12: Input powers, P_1 and P_2, and the combined input power (also the cost function) as a function of R_{droop2} as implemented in the HIL system.

Fig. 4.13 is significant because it indicates that the optimal control surface developed can be implemented practically using only local measurements.

With the effective resistive load, R_{eff}, calculated and the supercapacitor bank voltage, V_{bank}, measured, it is possible to implement the optimal surface in the DSP controller for source 2. The two parameters, R_{eff} and V_{bank}, can be fed into a two-dimensional lookup table that is based on the surface from Fig. 4.1. This calculates the R_{droop2} value that is then used for the optimal control.

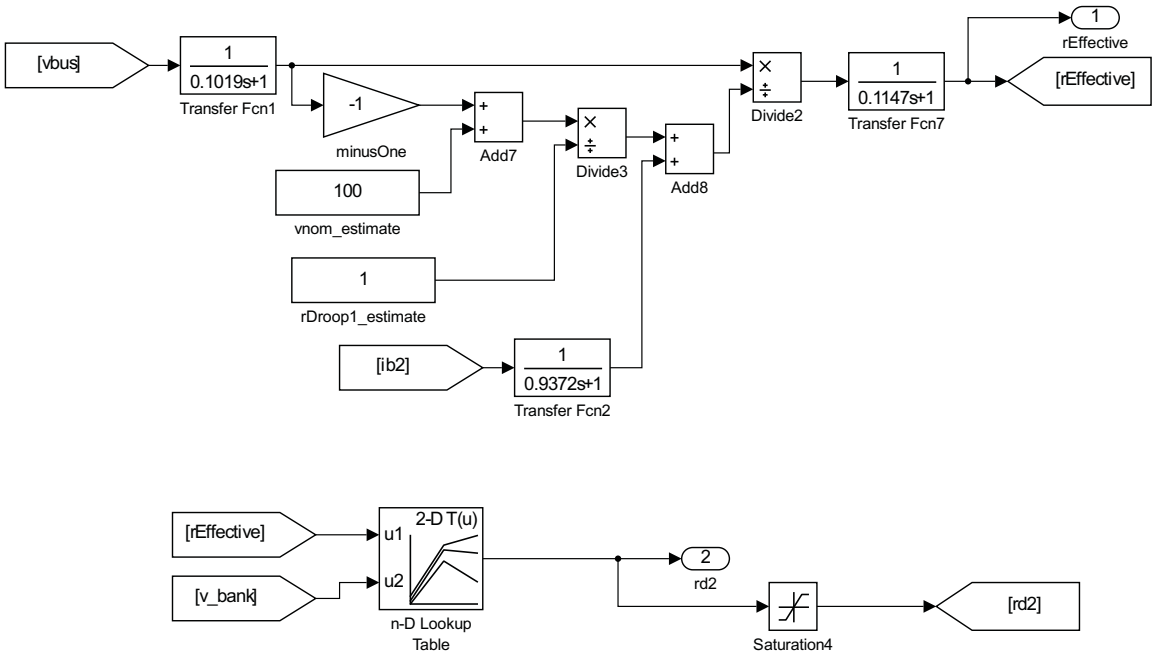

Figure 4.13: Simulink block diagram showing the calculation of the effective resistive load, R_{eff}. This block diagram was compiled and programmed into the DSP that was controlling the source converter for source 2, the energy storage source.

4.2.4 Running the System with the Optimal Surface

The physical model of the microgrid was implemented and run on the HIL system. The linear droop controller for source 1 and the optimal surface for source 2 were implemented in the dSPACE DSP. The system was run with a constant resistive load of 20 Ω, and the renewable energy input followed the cycle shown in Fig. 4.17. The resulting oscilloscope images obtained from this run are given in Fig. 4.15, and Fig. 4.16. The bus voltage

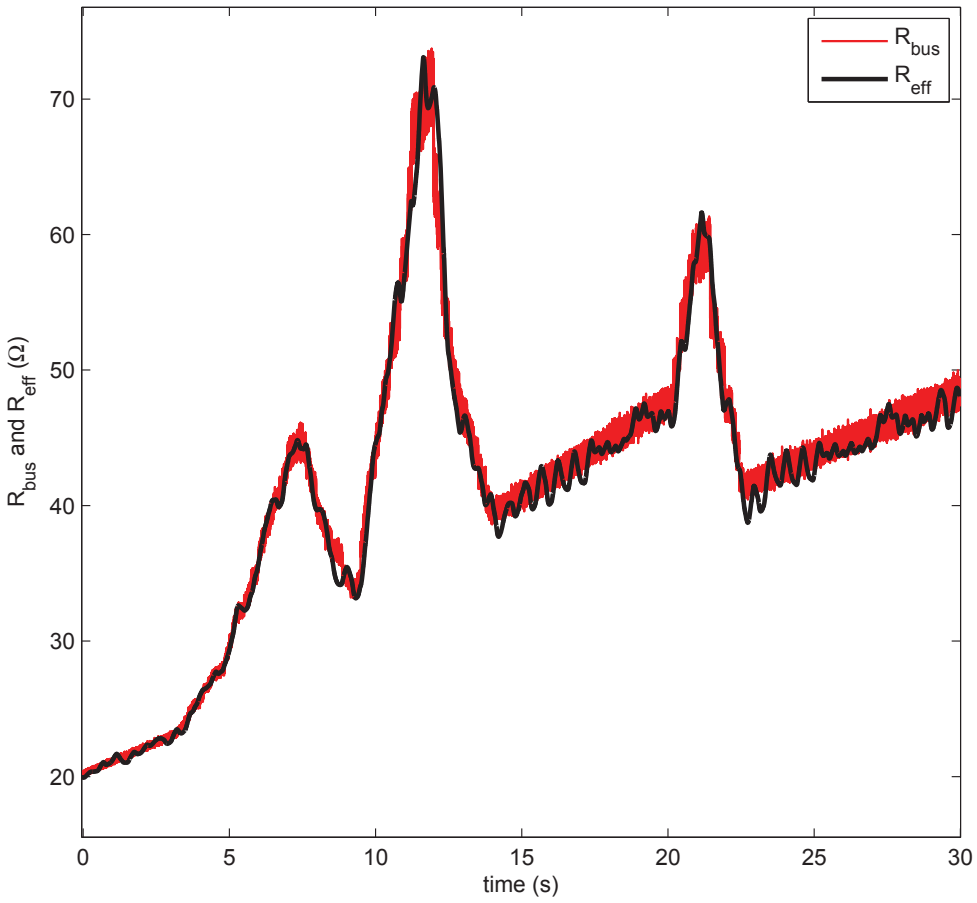

Figure 4.14: Comparison of the actual load resistance, R_{bus}, and the source 2 controller's estimate of the load, R_{eff}. The actual load resistance, R_{bus}, was implemented via a digital-to-analog channel on the dSPACE system, and the calculated resistance, R_{eff}, was calculated internal to controller 2 on the dSPACE system.

obtained from this run is shown in Fig. 4.18. The input currents, i_{L1} and i_{L2}, are shown in Fig. 4.19. The line currents, i_{b1} and i_{b2}, are shown in Fig. 4.20. The SOC of the supercapacitor bank for the cycle was calculated using the bank voltage, v_{bank}, and (1.1) and is shown in Fig. 4.21. The input powers, P_{in1} and P_{in2}, and the combined input power (which is also the cost function) were calculated and are shown in Fig. 4.22.

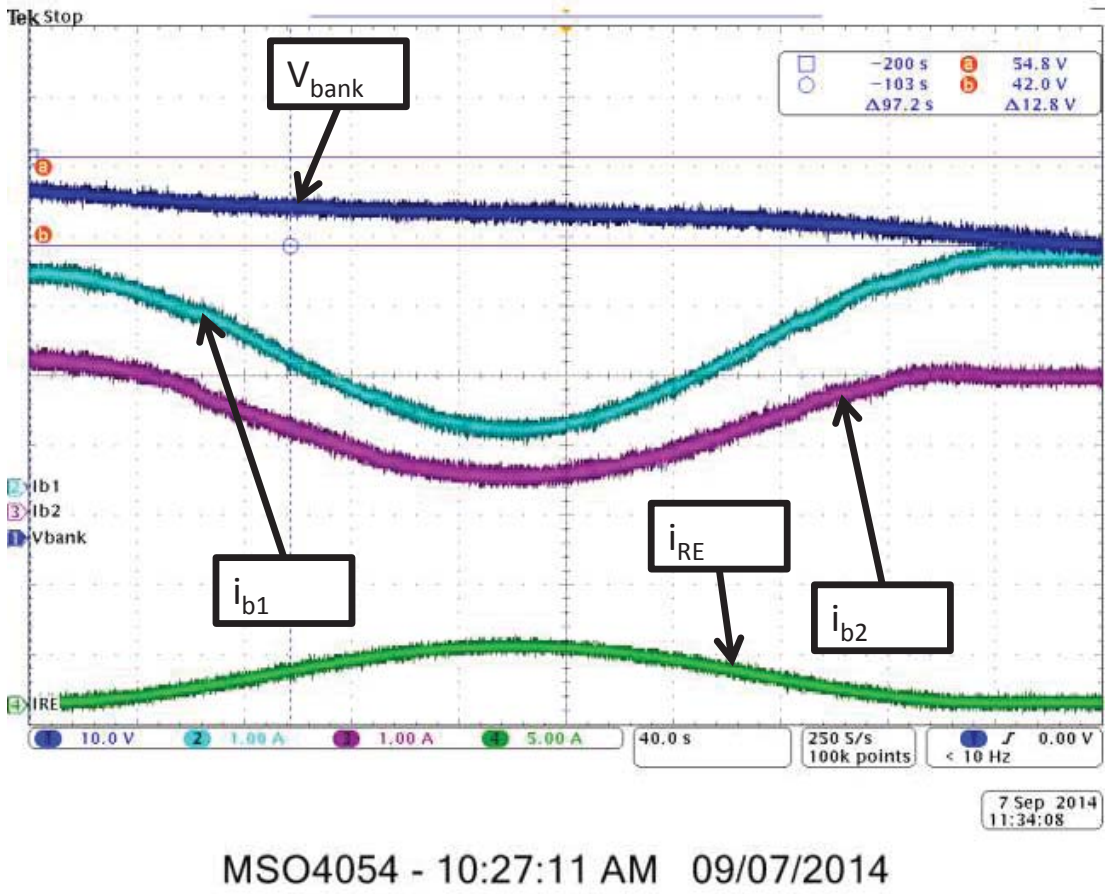

Figure 4.15: Oscilloscope results of running the microgrid system with the optimal control surface. Shown is the supercapacitor bank voltage, v_{bank}, the renewable energy input, i_{RE}, and the line currents, i_{b1} and i_{b2}.

To show the benefits of using the optimal surface for control, the same load resistance and i_{RE} cycle were used but this time the control for source 2 was linear droop, with the droop resistance, R_{droop}, equal to a constant of 2.0 Ω. The results of this are shown in Appendix B.2. The oscilloscope images for this are shown in Appendix B.3. The results of running the system with a constant R_{droop} of 2.0 Ω show that the bus voltage and the SOC of the supercapacitor bank both stay within the limits. Fig. 4.23 shows a comparison of the combined input powers using the two different control techniques.

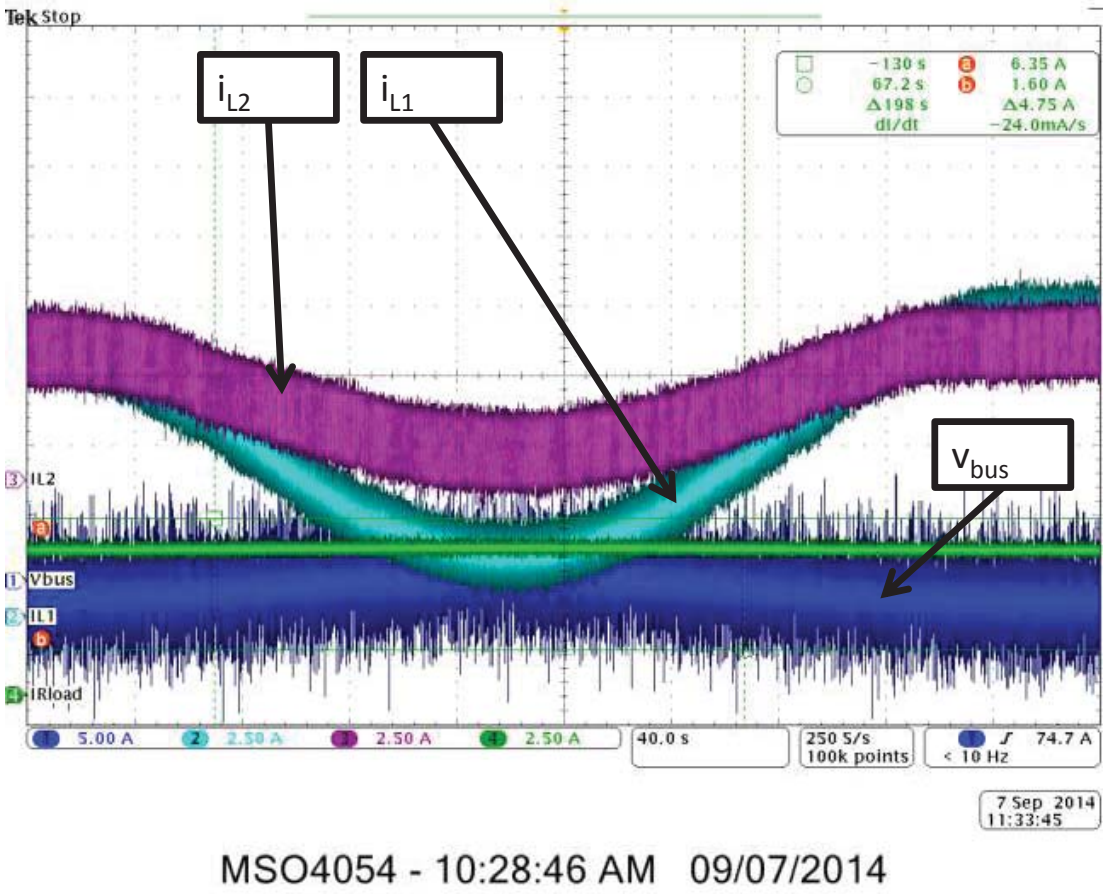

Figure 4.16: Oscilloscope results of running the microgrid system with the optimal control surface. Shown is the bus voltage, v_{bus}, the load current, i_{Rload}, and the input currents, i_{L1} and i_{L2}.

One notable difference between the results is that for the optimal surface, the power at every level is equal to or lower than the power level obtained using linear droop. This is because the control is designed to implement the optimal instantaneous power. As seen in Fig. 4.23, the power level is fairly comparable at lower power levels, and there is a significant difference at higher power levels. This is consistant with Fig. 4.5.

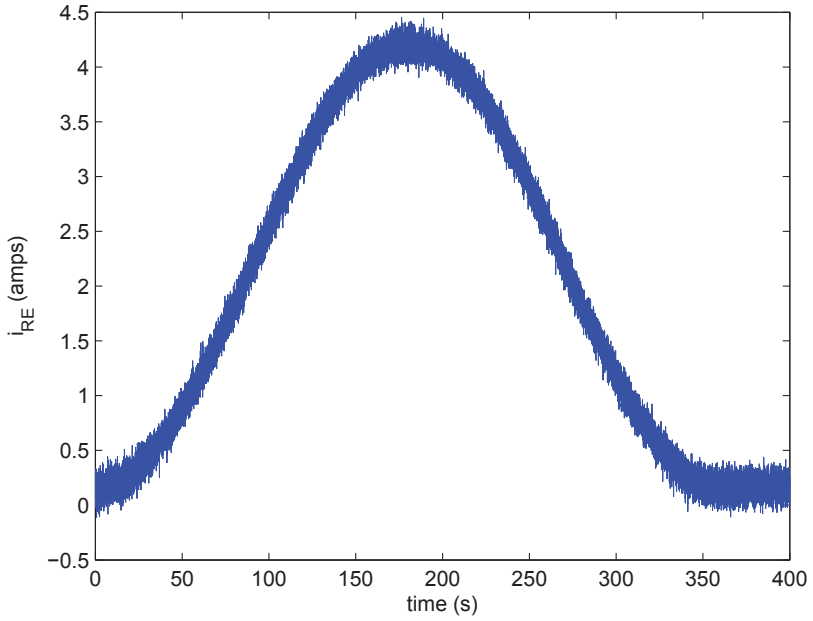

Figure 4.17: Renewable energy source current, i_{RE}, used for the cycle run.

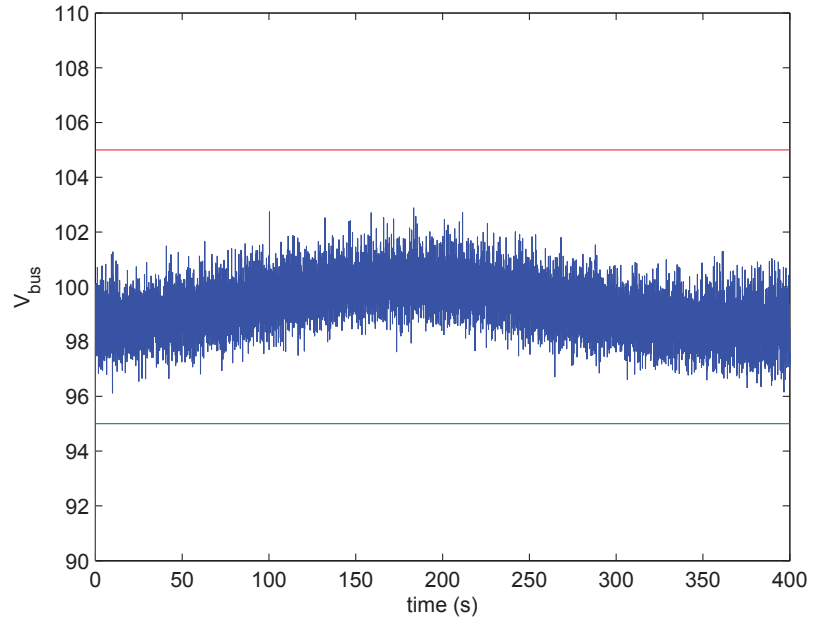

Figure 4.18: Bus voltage, v_{bus}, obtained using the optimal control surface.

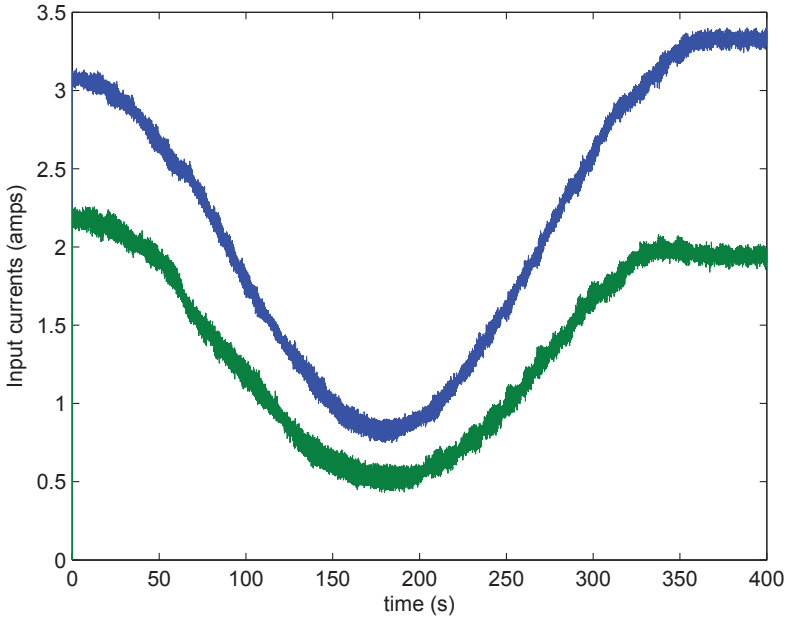

Figure 4.19: Input currents, i_{L1} and i_{L2}, obtained using the optimal control surface.

4.3 Optimal Surface Control of the Microgrid: Conclusion

The data in Fig. 4.23 was filtered with a first-order low-pass filter with a cutoff frequency of 1 Hz for easier visual representation and is shown in Fig. 4.24. Fig. 4.24 shows more clearly that there is significant savings to using the optimal surface as compared to the linear droop. When the renewable energy input, i_{RE} is low (at the beginning and end of the cycle), the difference between the two control methods is approximately 60 W. When the renewable energy input is at its highest (halfway through the cycle), there is

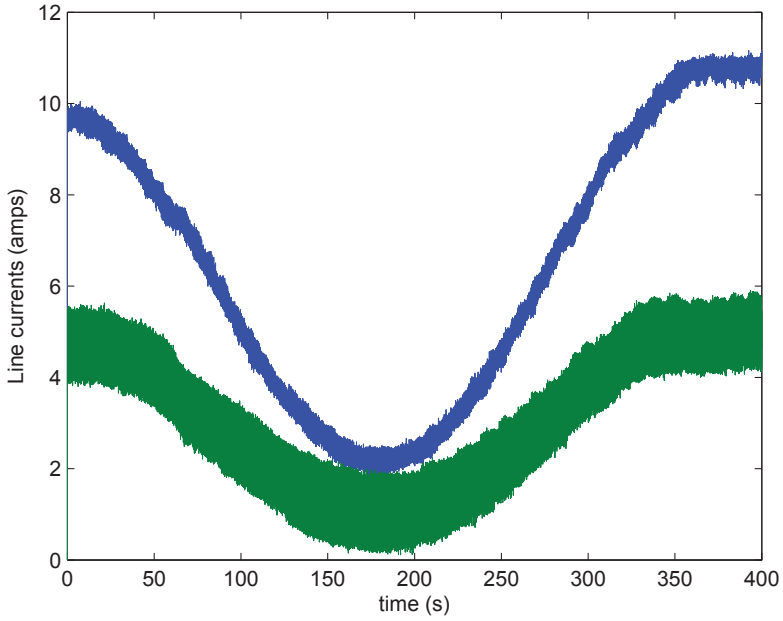

Figure 4.20: Line currents, i_{b1} and i_{b2}, obtained using the optimal control surface.

very negligible savings. This is also consistent with the trend present in Fig. 4.5. This is because the linear droop resistance value that was chosen is actually very close to the optimally determined droop resistance when the load is low. The combined energy of the system over the 400-second cycle for the optimal solution turned out to be 44.3 W hrs. The combined energy of the linear droop control was 46.8 W hrs. This yields a percent energy savings of approximately 5%. It should be noted that the system was optimized for combined input power and not combined input energy. Due to the non-linear nature of the system, the optimal solution for power yields a sub-optimal solution for energy.

In Chapter 3, the cost function (3.1) was developed. Then the algebraic simplification of the model was used to develop an optimal droop surface. In Chapter 4, the optimal surface

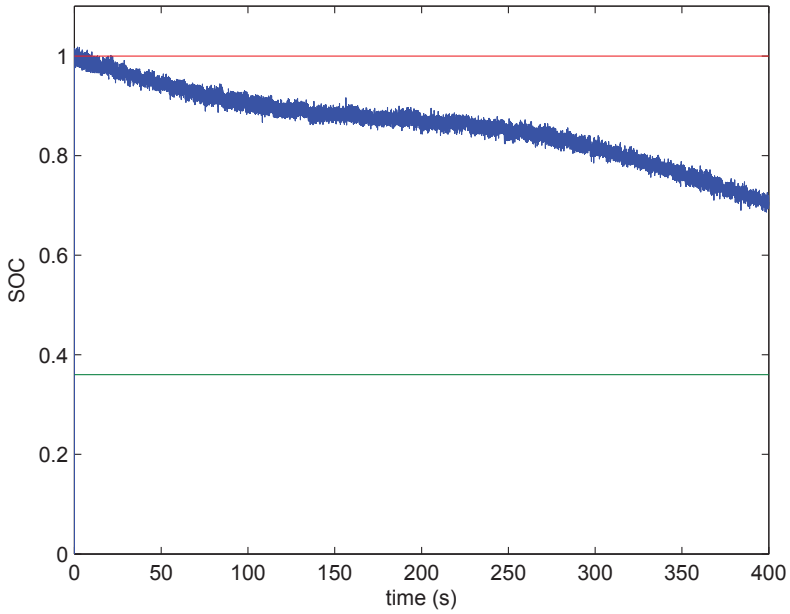

Figure 4.21: Supercapacitor bank SOC obtained using the optimal control surface.

controlled the supercapacitor source in the microgrid. It was shown that using the optimal surface gave a significant improvement to linear droop. Finally, the HIL system was used to verify that the control of the system could be implemented with local sensor measurements.

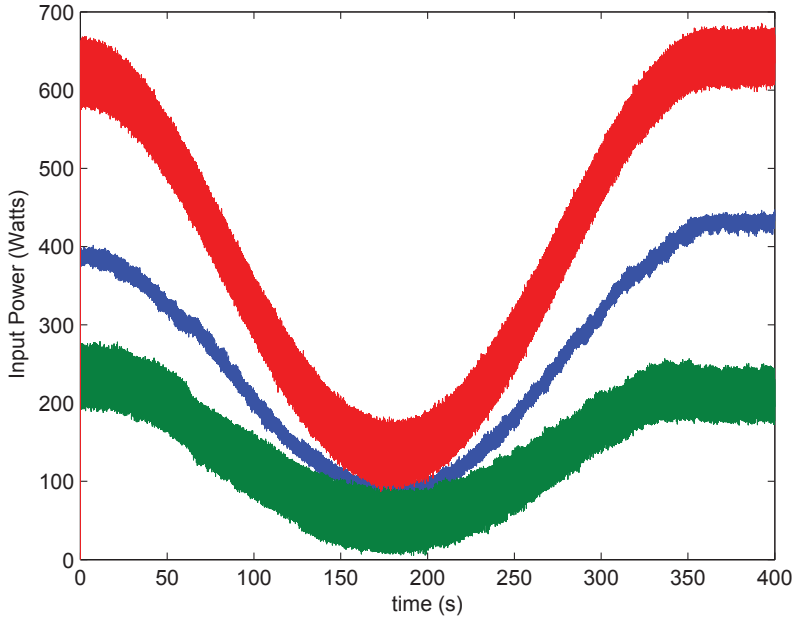

Figure 4.22: Input powers P_{in1} and P_{in2}, along with the combined input power (cost) obtained using the optimal control surface.

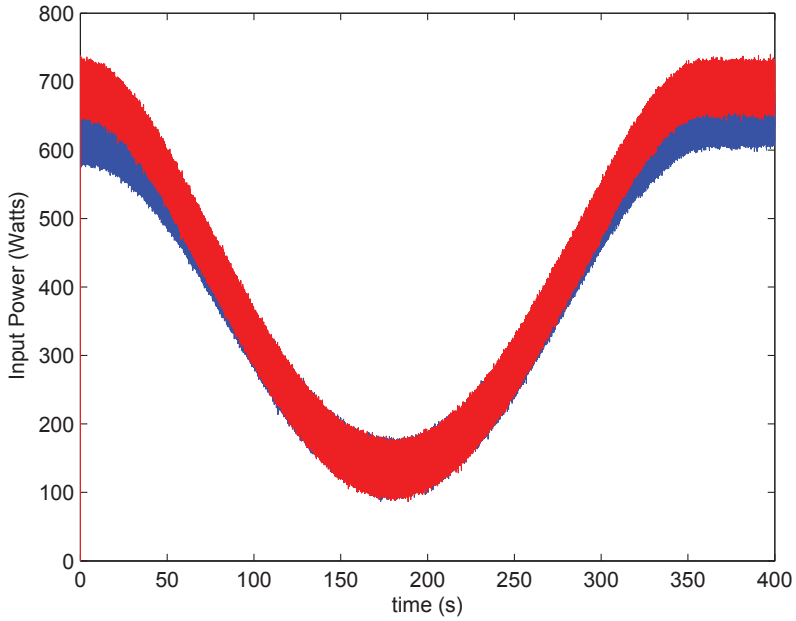

Figure 4.23: Cost of the the optimal surface when compared to a constant R_{droop2} of 2.0 Ω.

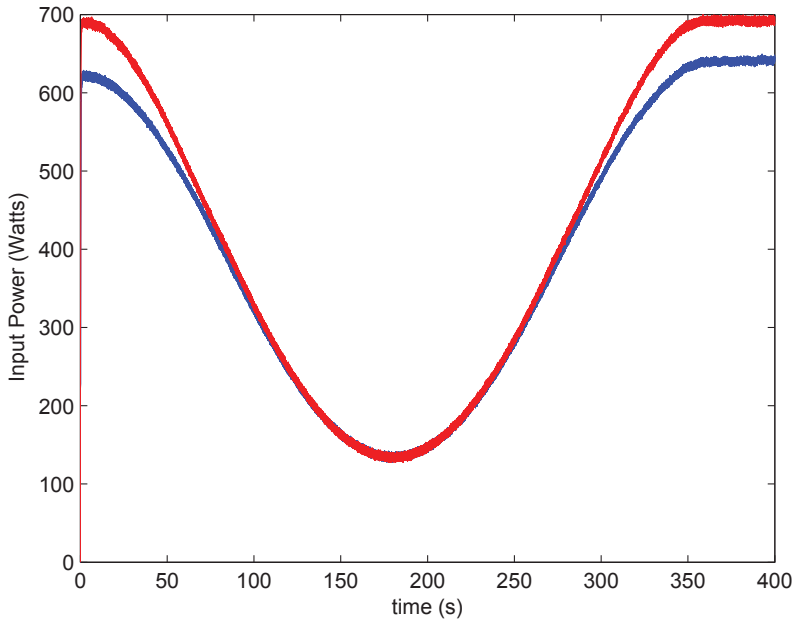

Figure 4.24: Filtered cost of the the optimal surface when compared to a constant R_{droop2} of 2.0 Ω.

Chapter 5

Summary

5.1 Conclusions

This thesis has demonstrated that there is a significant system benefit to the optimal surface control of the energy storage source. In Chapter 2, the model of the microgrid system was presented. In Chapter 3, the cost function was developed based on stated goals of the system. Then the STSS simplification of the system was used to develop an optimal control surface. It was shown how the parameters needed to implement the optimal control can be obtained via local measurements only. In Chapter 4, the optimal control surface was used to control a microgrid system; results were compared to that of linear droop. Also the validity of the STSS model and the decentralized control technique were demonstrated

using an HIL system.

5.2 Future Work

The system was optimized for the instantaneous input power from the two sources. The system was not optimized for the combined input energy. The cost function for minimum input energy would be

$$Cost_{Energy} = E_{Source1} + E_{Source2} = \int_0^t \left(V_{in1} i_{L1}(t) + v_{bank1}(t) i_{L2}(t)\right) dt \tag{5.1}$$

The control that would yield the minimum combined input energy would depend highly on the run cycle (the renewable energy input and load power over time) applied to the system. Because the run cycle would be both highly variable, this would be difficult to implement without some prediction of the cycle. The optimal solution found for (3.1) is a sub-optimal solution for (5.1). Further work could optimize for (5.1) instead.

The cost function (3.1) was optimized by the surface in Fig. 3.8. Although this cost function does apply a weight to the power drawn from the energy storage, it does not itself guarantee that the SOC of the energy storage source will be maintained. The equal weighting of the input power from source 1 and 2 in (3.1) helps to ensure that any power savings from source 1 is not at the direct expense of the energy stored in source 2. The

optimal surface obtained through the minimization process does cause the energy storage source to supply less power when it is further discharged and supply more power when it is more charged. This does add some regulation to the SOC of source 2, but does not guarantee maintaining the SOC within some acceptable bounds. Further research could be done to ensure that the SOC is maintained at an acceptable level.

The energy storage source used was a supercapacitor bank. Other energy storage sources could also be used. The results from using a battery would likely be similar due to the fact that the open circuit voltage of a battery would also drop at a lower SOC [31]. The downside of using a battery for this type of analysis is that the usable life of a battery decreases as the battery is cycled (discharged and charged) [11]. This thesis did not take into account the cycling of the energy storage device due to the high number of cycles permitted by supercapacitors [12].

References

[1] S. Chowdhury, C. S. P., and P. Crossley, *Microgrids and Active Distribution Networks*. IET renewable energy series, Institution of Engineering and Technology, 2009.

[2] P. Denholm, E. Ela, B. Kirgy, and M. Milligan, "The role of energy storage with renewable electricity generation," tech. rep., University of Nevada, Las Vegas, 2010.

[3] R. Lasseter, "Microgrids," in *2002. IEEE Power Engineering Society Winter Meeting*, vol. 1, pp. 305–308 vol.1, 2002.

[4] A. Sannino, G. Postiglione, and M. H. J. Bollen, "Feasibility of a dc network for commercial facilities," *IEEE Transactions on Industry Applications*, vol. 39, pp. 1499–1507, Sept 2003.

[5] D. Costinett, H. Nguyen, R. Zane, and D. Maksimovic, "Gan-fet based dual active bridge dc-dc converter," in *2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 1425–1432, March 2011.

[6] J. Guerrero, J. Vasquez, J. Matas, L. de Vicuña, and M. Castilla, "Hierarchical control of droop-controlled ac and dc microgrids; a general approach toward standardization," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 158–172, Jan 2011.

[7] A. Esmaili and A. Nasiri, "Energy storage for short-term and long-term wind energy support," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 3281–3286, Nov 2010.

[8] Q. Fu, L. Montoya, A. Solanki, A. Nasiri, V. Bhavaraju, T. Abdallah, and D. Yu, "Microgrid generation capacity design with renewables and energy storage addressing power quality and surety," *IEEE Transactions on Smart Grid*, vol. 3, pp. 2019–2027, Dec 2012.

[9] A. Yoshida, K. Imoto, A. Nishino, and H. Yoneda, "An electric double-layer capacitor with high capacitance and low resistance," in *Electronic Components and Technology Conference, 1991. Proceedings., 41st*, pp. 531–536, May 1991.

[10] Y.-H. Kim and H.-D. Ha, "Design of interface circuits with electrical battery models," *IEEE Transactions on Industrial Electronics*, vol. 44, pp. 81–86, Feb 1997.

[11] T. B. Reddy and D. Linden, *Linden's Handbook of Batteries*. McGraw-Hill, 4th ed., 2011.

[12] Maxwell Technologies Inc. *Maxwell Technologies BOOSTCAP Ultracapacitors Product Guide* [electronic] available:

http://www.maxwell.com/images/documents/PG_boostcap_product_guide.pdf Accessed: September 24, 2014.

[13] P. T. Krein, *Elements of Power Electronics*. New York, NY: Oxford University Press, 1st ed., 1998.

[14] C. Abbey and G. Joos, "Supercapacitor energy storage for wind energy applications," *IEEE Transactions on Industry Applications*, vol. 43, pp. 769–776, May 2007.

[15] T. Vandoorn, B. Meersman, J. D. M. De Kooning, and L. Vandevelde, "Analogy between conventional grid control and islanded microgrid control based on a global dc-link voltage droop," *IEEE Transactions on Power Delivery*, vol. 27, pp. 1405–1414, July 2012.

[16] F. Luo, Y. Lai, K. Loo, C. Tse, and X. Ruan, "A generalized droop-control scheme for decentralized control of inverter-interfaced microgrids," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1320–1323, May 2013.

[17] P. Kundur, *Power System Stability and Control*. McGraw-Hill Professional, 1994.

[18] P. Karlsson and J. Svensson, "Dc bus voltage control for a distributed power system," *IEEE Transactions on Power Electronics*, vol. 18, pp. 1405–1412, Nov 2003.

[19] X. Lu, K. Sun, J. Guerrero, J. Vasquez, L. Huang, and R. Teodorescu, "Soc-based droop method for distributed energy storage in dc microgrid applications," in *2012*

IEEE International Symposium on Industrial Electronics (ISIE), pp. 1640–1645, May 2012.

[20] H. Kakigano, Y. Miura, T. Ise, and R. Uchida, "Dc voltage control of the dc micro-grid for super high quality distribution," in *Power Conversion Conference - Nagoya, 2007. PCC '07*, pp. 518–525, April 2007.

[21] B. Banerjee and W. Weaver, "Geometric manifold control of power electronics in dc microgrids," in *2012 IEEE 13th Workshop on Control and Modeling for Power Electronics (COMPEL)*, pp. 1–8, June 2012.

[22] A. P. Sage and C. C. I. White, *Optimum Systems Control*. Prentice-Hall Inc., 2nd edition ed., 1977.

[23] S. Teleke, M. Baran, S. Bhattacharya, and A. Huang, "Optimal control of battery energy storage for wind farm dispatching," *IEEE Transactions on Energy Conversion*, vol. 25, pp. 787–794, Sept 2010.

[24] S. Patra, J. Mitra, and S. Ranade, "Microgrid architecture: a reliability constrained approach," in *2005 IEEE Power Engineering Society General Meeting*, pp. 2372–2377 Vol. 3, June 2005.

[25] A. Tsikalakis and N. Hatziargyriou, "Centralized control for optimizing microgrids operation," in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–8, July 2011.

[26] K. Bunker and W. Weaver, "Optimization of grid-connected wind and battery energy storage system," in *2014 Power and Energy Conference at Illinois (PECI)*, pp. 1–6, Feb 2014.

[27] K. Bunker and W. Weaver, "Optimal geometric control of dc microgrids," in *2014 IEEE 15th Workshop on Control and Modeling for Power Electronics (COMPEL)*, pp. 1–6, June 2014.

[28] P. Zanchetta, "Heuristic multi-objective optimization for cost function weights selection in finite states model predictive control," in *2011 Workshop on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, pp. 70–75, Oct 2011.

[29] Typhoon HIL, Inc. typhoon-hil.com [electronic] available: http://www.typhoon-hil.com/ Accessed: Sept. 29, 2014.

[30] dSPACE GmbH, dspace.com, available: http://www.dspace.com/ Accessed: Sept. 29, 2014.

[31] T. Blank, J. Badeda, J. Kowal, and D. Sauer, "Deep discharge behavior of lead-acid batteries and modeling of stationary battery energy storage systems," in *2012 IEEE 34th International Telecommunications Energy Conference (INTELEC)*, pp. 1–4, Sept 2012.

Appendix A

Matlab Code

```
clc
clear
close all

params_ss_power
ire=0;

max_rd2 = rd1.*10;
rd2_array = (max_rd2./100):(max_rd2./100):max_rd2;
efficiency = zeros(1,length(rd2_array));
powerOut = zeros(1,length(rd2_array));
powerIn_1 = zeros(1,length(rd2_array));
powerIn_2 = zeros(1,length(rd2_array));
combinedPowerIn = zeros(1,length(rd2_array));
vbus = zeros(1,length(rd2_array));
il1_array = zeros(1,length(rd2_array));
il2_array = zeros(1,length(rd2_array));
ib1_array = zeros(1,length(rd2_array));
ib2_array = zeros(1,length(rd2_array));


effectiveLoad = 0:(pnom./50):pnom;
effectiveResistance = vnom.^2./effectiveLoad;
effectiveResistance(1) = effectiveResistance(2);
v2_array=V_bank_min:(V_bank_max./50):V_bank_max;
optimalDroop = zeros(length(effectiveLoad),length(v2_array));
vbusOptimal = zeros(length(effectiveLoad),length(v2_array));
```

```
%%
v2=V_bank_max;
load = pnom;
rbus = vnom.^2./load;
for k_rd2=1:length(rd2_array);
    rd2=rd2_array(k_rd2);


    il1 = (1/2).*r1.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.*rd1.
    *v1+ ...
        rbus.*rd2.*v1+rd1.*rd2.*v1+(-1).*((rd1.*rd2+rbus.*(rd1+rd2)).^2.
        * ...
        v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd2.*(rd1+
        ( ...
        -1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*vnom)).^(1/2));

    il2 = (1/2).*r2.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.*rd1.
    *v2+ ...
        rbus.*rd2.*v2+rd1.*rd2.*v2+(-1).*((rd1.*rd2+rbus.*(rd1+rd2)).^2.
        * ...
        v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd1.*(rd2+
        ( ...
        -1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*vnom)).^(1/2));

    ib1 = rd2.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+vnom);

    ib2 = rd1.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+vnom);

    d1 = (1/2).*(ire.*rbus.*rd2.*(rd1+(-1).*rl1)+(rbus.*(rd1+rd2)+rd2.*r
    l1) ...
        .*vnom).^(-1).*(rbus.*rd1.*v1+rbus.*rd2.*v1+rd1.*rd2.*v1+((rd1.*
         ...
        rd2+rbus.*(rd1+rd2)).^2.*v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom
        ) ...
        .*(ire.*rbus.*rd2.*(rd1+(-1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*
         ...
        vnom)).^(1/2));

    d2 = (1/2).*(ire.*rbus.*rd1.*(rd2+(-1).*rl2)+(rbus.*(rd1+rd2)+rd1.*r
    l2) ...
        .*vnom).^(-1).*(rbus.*rd1.*v2+rbus.*rd2.*v2+rd1.*rd2.*v2+((rd1.*
         ...
        rd2+rbus.*(rd1+rd2)).^2.*v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom
        ) ...
        .*(ire.*rbus.*rd1.*(rd2+(-1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*
        ...
        vnom)).^(1/2));

    vbus(k_rd2) = rbus.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(ire.*rd1.*rd2
    +(rd1+rd2).* ...
        vnom);

    powerOut(k_rd2) = vbus(k_rd2).^2./rbus;
```

```
    powerIn_1(k_rd2) = v1*il1;

    powerIn_2(k_rd2) = v2*il2;

    combinedPowerIn(k_rd2) = powerIn_1(k_rd2)+powerIn_2(k_rd2);

    efficiency(k_rd2) = powerOut(k_rd2)./combinedPowerIn(k_rd2);

    il1_array(k_rd2) = il1;
    il2_array(k_rd2) = il2;
    ib1_array(k_rd2) = ib1;
    ib2_array(k_rd2) = ib2;

end

eff_index = find(efficiency==max(efficiency));
maximumEff = efficiency(eff_index);

minPow_index = find(combinedPowerIn==min(combinedPowerIn));
minPow = combinedPowerIn(minPow_index);
figure
plot(rd2_array,efficiency,'b-',rd2_array(eff_index),maximumEff,'b*',rd2
_array(minPow_index),efficiency(minPow_index),'g*','lineWidth',2)
%plot(rd2_array,efficiency,'k-',rd2_array(eff_index),maximumEff,'k*',r
d2_array(minPow_index),efficiency(minPow_index),'k*','lineWidth',2)
xlabel('R_{droop2} (\Omega)')
ylabel('efficiency')
figure
plot(rd2_array,combinedPowerIn,'b-',rd2_array,powerOut,'g-',rd2_array(mi
nPow_index),minPow,'b*','lineWidth',2)
%plot(rd2_array,combinedPowerIn,'k-',rd2_array,powerOut,'k--',rd2_array(
minPow_index),minPow,'k*','lineWidth',2)
xlabel('R_{droop2} (\Omega)')
ylabel('Cost (W)')
legend('Input Power (W)','Output Power (W)')
figure
plot(rd2_array,vbus,'b--',[0 max_rd2],[0.95.*vnom 0.95.*vnom],'r-',[0 ma
x_rd2],[1.05.*vnom 1.05.*vnom],'r-.',[0 max_rd2],[vnom vnom],'g:','lineW
idth',2)
%plot(rd2_array,vbus,'k--',[0 max_rd2],[0.95.*vnom 0.95.*vnom],'k-',[0 m
ax_rd2],[1.05.*vnom 1.05.*vnom],'k-',[0 max_rd2],[vnom vnom],'k:','lineW
idth',2)
axis([0 max_rd2 0.9.*vnom 1.1.*vnom])
xlabel('R_{droop2} (\Omega)')
ylabel('V_{bus} (V)')
legend('V_{bus}','Min V_{bus}','Max V_{bus}','Nominal V_{bus}')
figure
plot(rd2_array,il1_array,'b-',rd2_array,il2_array,'g-','lineWidth',2)
%plot(rd2_array,il1_array,'k-',rd2_array,il2_array,'k--','lineWidth',2)
xlabel('R_{droop2} (\Omega)')
ylabel('Input Currents (A)')
legend('I_{L1}','I_{L2}')
figure
```

```
plot(rd2_array,ib1_array,'b-',rd2_array,ib2_array,'g-','lineWidth',2)
%plot(rd2_array,ib1_array,'k-',rd2_array,ib2_array,'k--','lineWidth',2)
xlabel('R_{droop2} (\Omega)')
ylabel('Line Currents (A)')
legend('I_{b1}','I_{b2}')
%%
figure
varySOCPlot = [30,35,40,45,50];
hold on
for k_varySOCPlot = 1:length(varySOCPlot)

    v2=varySOCPlot(k_varySOCPlot);
    load = pnom;
    rbus = vnom.^2./load;
    for k_rd2=1:length(rd2_array);
        rd2=rd2_array(k_rd2);


        il1 = (1/2).*r1.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.*
        rd1.*v1+ ...
            rbus.*rd2.*v1+rd1.*rd2.*v1+(-1).*((rd1.*rd2+rbus.*(rd1+rd2))
            .^2.* ...
            v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd2.*(
            rd1+( ...
            -1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*vnom)).^(1/2));

        il2 = (1/2).*r2.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.*
        rd1.*v2+ ...
            rbus.*rd2.*v2+rd1.*rd2.*v2+(-1).*((rd1.*rd2+rbus.*(rd1+rd2))
            .^2.* ...
            v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd1.*(
            rd2+( ...
            -1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*vnom)).^(1/2));

        ib1 = rd2.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+vn
        om);

        ib2 = rd1.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+vn
        om);

        d1 = (1/2).*(ire.*rbus.*rd2.*(rd1+(-1).*rl1)+(rbus.*(rd1+rd2)+rd
        2.*rl1) ...
            .*vnom).^(-1).*(rbus.*rd1.*v1+rbus.*rd2.*v1+rd1.*rd2.*v1+((r
            d1.* ...
            rd2+rbus.*(rd1+rd2)).^2.*v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).
            *vnom) ...
            .*(ire.*rbus.*rd2.*(rd1+(-1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl
            1).* ...
            vnom)).^(1/2));

        d2 = (1/2).*(ire.*rbus.*rd1.*(rd2+(-1).*rl2)+(rbus.*(rd1+rd2)+r
        d1.*rl2) ...
            .*vnom).^(-1).*(rbus.*rd1.*v2+rbus.*rd2.*v2+rd1.*rd2.*v2+(
            (rd1.* ...
```

```
            rd2+rbus.*(rd1+rd2)).^2.*v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).
            *vnom) ...
            .*(ire.*rbus.*rd1.*(rd2+(-1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl
            2).* ...
            vnom)).^(1/2));

        vbus(k_rd2) = rbus.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(ire.*rd1
        .*rd2+(rd1+rd2).* ...
            vnom);

        powerOut(k_rd2) = vbus(k_rd2).^2./rbus;

        powerIn_1(k_rd2) = v1*il1;

        powerIn_2(k_rd2) = v2*il2;

        combinedPowerIn(k_rd2) = powerIn_1(k_rd2)+powerIn_2(k_rd2);

        efficiency(k_rd2) = powerOut(k_rd2)./combinedPowerIn(k_rd2);

        il1_array(k_rd2) = il1;
        il2_array(k_rd2) = il2;
        ib1_array(k_rd2) = ib1;
        ib2_array(k_rd2) = ib2;

    end
    minPow_index = find(combinedPowerIn==min(combinedPowerIn));
    minPow = combinedPowerIn(minPow_index);
    plot(rd2_array,combinedPowerIn,'b-',rd2_array(minPow_index),minPow,
    'b*','lineWidth',2)
    %plot(rd2_array,combinedPowerIn,'k-',rd2_array(minPow_index),minPow,
    'k*','lineWidth',2)
end
hold off
xlabel('R_{droop2} (\Omega)')
ylabel('combinedPower')



%%

for k_v2 = 1:length(v2_array)
    v2 = v2_array(k_v2);
    for k_load = 1:length(effectiveLoad)
        for k_rd2=1:length(rd2_array);
            rd2=rd2_array(k_rd2);
            rbus = effectiveResistance(k_load);

            il1 = (1/2).*r1.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(r
            bus.*rd1.*v1+ ...
                rbus.*rd2.*v1+rd1.*rd2.*v1+(-1).*((rd1.*rd2+rbus.*(rd1+
                rd2)).^2.* ...
                v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*r
                d2.*(rd1+( ...
```

```
            -1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*vnom)).^(1/2));

        il2 = (1/2).*r2.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(r
        bus.*rd1.*v2+ ...
            rbus.*rd2.*v2+rd1.*rd2.*v2+(-1).*((rd1.*rd2+rbus.*(rd1+
            rd2)).^2.* ...
            v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*r
            d1.*(rd2+( ...
            -1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*vnom)).^(1/2));

        ib1 = rd2.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rb
        us+vnom);

        ib2 = rd1.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*r
        bus+vnom);

        vbus(k_rd2) = rbus.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(ire.
        *rd1.*rd2+(rd1+rd2).* ...
            vnom);

        powerOut(k_rd2) = vbus(k_rd2).^2./rbus;

        powerIn_1(k_rd2) = v1*il1;

        powerIn_2(k_rd2) = v2*il2;

        combinedPowerIn(k_rd2) = powerIn_1(k_rd2)+powerIn_2(k_rd2);

        efficiency(k_rd2) = powerOut(k_rd2)/combinedPowerIn(k_rd2);

    end
    if 1==2
        % account for any situations where the system goes unstable.
        asdf=find(combinedPowerIn==max(combinedPowerIn));
        maxPowerForDeleting = combinedPowerIn(asdf);
        for k_rd2=1:length(rd2_array);
            if abs(imag(combinedPowerIn(k_rd2)))>1e-6
                combinedPowerIn(k_rd2)=maxPowerForDeleting;
            end
        end
        clear asdf
        clear maxPowerForDeleting
    end
    minPowerIndex = find(combinedPowerIn==min(combinedPowerIn));

    optimalDroop(k_load,k_v2) = rd2_array(minPowerIndex(1));
    vbusOptimal(k_load,k_v2) = vbus(minPowerIndex(1));
  end
end
soc_array = v2_array.^2./V_bank_max.^2;
figure
surf(v2_array,effectiveResistance,optimalDroop,'edgecolor','none')
%%colormap(gray)
xlabel('V_{bank}')
```

```
ylabel('R_{bus}')
zlabel('R_{droop2} (\Omega)')
figure
surf(soc_array,effectiveLoad,optimalDroop,'edgecolor','none')
%colormap(gray)
xlabel('SOC')
ylabel('Effective Load (Watts)')
zlabel('R_{droop2} (\Omega)')
figure
surf(v2_array,effectiveLoad,vbusOptimal,'edgecolor','none')
%colormap(gray)
%%
%Determine the power in and out using the optimal droop surface

vbus_opt = zeros(length(effectiveLoad),length(v2_array));
powerOut_opt = zeros(length(effectiveLoad),length(v2_array));
powerIn_1_opt = zeros(length(effectiveLoad),length(v2_array));
powerIn_2_opt = zeros(length(effectiveLoad),length(v2_array));
combinedPowerIn_opt = zeros(length(effectiveLoad),length(v2_array));
efficiency_opt = zeros(length(effectiveLoad),length(v2_array));


for k_v2 = 1:length(v2_array)
    v2 = v2_array(k_v2);
    for k_load = 1:length(effectiveLoad)
        rd2 = optimalDroop(k_load,k_v2);
        rbus = effectiveResistance(k_load);

        il1 = (1/2).*r1.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.
        *rd1.*v1+ ...
            rbus.*rd2.*v1+rd1.*rd2.*v1+(-1).*((rd1.*rd2+rbus.*(rd1+rd2)
            ).^2.* ...
            v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd2.*
            (rd1+( ...
            -1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*vnom)).^(1/2));

        il2 = (1/2).*r2.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.
        *rd1.*v2+ ...
            rbus.*rd2.*v2+rd1.*rd2.*v2+(-1).*((rd1.*rd2+rbus.*(rd1+rd2
            )).^2.* ...
            v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd1.*
            (rd2+( ...
            -1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*vnom)).^(1/2));

        ib1 = rd2.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+v
        nom);

        ib2 = rd1.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+v
        nom);

        vbus_opt(k_load,k_v2) = rbus.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).
        *(ire.*rd1.*rd2+(rd1+rd2).* ...
            vnom);
```

```
        powerOut_opt(k_load,k_v2) = vbus_opt(k_load,k_v2).^2./rbus;

        powerIn_1_opt(k_load,k_v2) = v1*il1;

        powerIn_2_opt(k_load,k_v2) = v2*il2;

        combinedPowerIn_opt(k_load,k_v2)=powerIn_1_opt(k_load,
        k_v2+powerIn_2_opt(k_load,k_v2);

        efficiency_opt(k_load,k_v2) = powerOut_opt(k_load,k_v2)/
        combinedPowerIn_opt(k_load,k_v2);

    end
end

figure
surf(soc_array,effectiveLoad,combinedPowerIn_opt,'edgecolor','none')
xlabel('SOC')
ylabel('Effective Load (W)')
zlabel('Cost (W)')

%%
%Determine the power in and out using an arbitrary Rd2

arbitraryDroop = 2.0;% ohms

vbus_arb = zeros(length(effectiveLoad),length(v2_array));
powerOut_arb = zeros(length(effectiveLoad),length(v2_array));
powerIn_1_arb = zeros(length(effectiveLoad),length(v2_array));
powerIn_2_arb = zeros(length(effectiveLoad),length(v2_array));
combinedPowerIn_arb = zeros(length(effectiveLoad),length(v2_array));
efficiency_arb = zeros(length(effectiveLoad),length(v2_array));


for k_v2 = 1:length(v2_array)
    v2 = v2_array(k_v2);
    for k_load = 1:length(effectiveLoad)
        rd2 = arbitraryDroop;
        rbus = effectiveResistance(k_load);

        il1 = (1/2).*r1.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.
        *rd1.*v1+ ...
            rbus.*rd2.*v1+rd1.*rd2.*v1+(-1).*((rd1.*rd2+rbus.*(rd1+rd2)
            ).^2.* ...
            v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd2.*
            (rd1+( ...
            -1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*vnom)).^(1/2));

        il2 = (1/2).*r2.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rbus.
        *rd1.*v2+ ...
            rbus.*rd2.*v2+rd1.*rd2.*v2+(-1).*((rd1.*rd2+rbus.*(rd1+rd2)
            ).^2.* ...
            v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*rd1.*
            (rd2+( ...
```

```
            -1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*vnom)).^(1/2));

        ib1 = rd2.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+
        vnom);

        ib2 = rd1.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rbus+
        vnom);

        vbus_arb(k_load,k_v2) = rbus.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1)
        .*(ire.*rd1.*rd2+(rd1+rd2).* ...
            vnom);

        powerOut_arb(k_load,k_v2) = vbus_arb(k_load,k_v2).^2./rbus;

        powerIn_1_arb(k_load,k_v2) = v1*il1;

        powerIn_2_arb(k_load,k_v2) = v2*il2;

        combinedPowerIn_arb(k_load,k_v2) = powerIn_1_arb(k_load,k_v2)
        +powerIn_2_arb(k_load,k_v2);

        efficiency_arb(k_load,k_v2) = powerOut_arb(k_load,k_v2)/
        combinedPowerIn_arb(k_load,k_v2);

    end
end
figure
surf(soc_array,effectiveLoad,combinedPowerIn_opt,'edgecolor','none')
xlabel('SOC')
ylabel('Effective Load (W)')
zlabel('Cost (W)')
%%
% compare the optimal droop with the arbitrary droop

combinedPowerIn_diff = zeros(length(effectiveLoad),length(v2_array));
percentPowerSaved = zeros(length(effectiveLoad),length(v2_array));

combinedPowerIn_diff = combinedPowerIn_arb-combinedPowerIn_opt;
figure
surf(soc_array,effectiveLoad,combinedPowerIn_diff,'edgecolor','none')
xlabel('SOC')
ylabel('Effective Load (W)')
zlabel('Power Saved (W)')

percentPowerSaved = 100*(combinedPowerIn_diff./(powerOut_arb));
figure
surf(soc_array,effectiveLoad,percentPowerSaved,'edgecolor','none')
xlabel('SOC')
ylabel('effective load (Watts)')
zlabel('Percent Savings')


%%
%Determine the percent savings using every arbitrary Rd2
```

```
maxSavings = zeros(1,length(rd2_array));
for k_arbitraryDroop = 1:length(rd2_array)

    arbitraryDroop = rd2_array(k_arbitraryDroop);% ohms

    vbus_arb = zeros(length(effectiveLoad),length(v2_array));
    powerOut_arb = zeros(length(effectiveLoad),length(v2_array));
    powerIn_1_arb = zeros(length(effectiveLoad),length(v2_array));
    powerIn_2_arb = zeros(length(effectiveLoad),length(v2_array));
    combinedPowerIn_arb = zeros(length(effectiveLoad),length(v2_array));
    efficiency_arb = zeros(length(effectiveLoad),length(v2_array));


    for k_v2 = 1:length(v2_array)
        v2 = v2_array(k_v2);
        for k_load = 1:length(effectiveLoad)
            rd2 = arbitraryDroop;
            rbus = effectiveResistance(k_load);

            il1 = (1/2).*r1.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*
            (rbus.*rd1.*v1+ ...
                rbus.*rd2.*v1+rd1.*rd2.*v1+(-1).*((rd1.*rd2+rbus.*(rd1
                +rd2)).^2.* ...
                v1.^2+4.*r1.*rd2.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*r
                d2.*(rd1+( ...
                -1).*rl1)+(rbus.*(rd1+rd2)+rd2.*rl1).*vnom)).^(1/2));

            il2 = (1/2).*r2.^(-1).*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*(rb
            us.*rd1.*v2+ ...
                rbus.*rd2.*v2+rd1.*rd2.*v2+(-1).*((rd1.*rd2+rbus.*(rd1+r
                d2)).^2.* ...
                v2.^2+4.*r2.*rd1.*(ire.*rbus+(-1).*vnom).*(ire.*rbus.*r
                d1.*(rd2+( ...
                -1).*rl2)+(rbus.*(rd1+rd2)+rd1.*rl2).*vnom)).^(1/2));

            ib1 = rd2.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*r
            bus+vnom);

            ib2 = rd1.*(rd1.*rd2+rbus.*(rd1+rd2)).^(-1).*((-1).*ire.*rb
            us+vnom);

            vbus_arb(k_load,k_v2) = rbus.*(rd1.*rd2+rbus.*(rd1+rd2)).^
            (-1).*(ire.*rd1.*rd2+(rd1+rd2).* ...
                vnom);

            powerOut_arb(k_load,k_v2) = vbus_arb(k_load,k_v2).^2./rbus;

            powerIn_1_arb(k_load,k_v2) = v1*il1;

            powerIn_2_arb(k_load,k_v2) = v2*il2;

            combinedPowerIn_arb(k_load,k_v2) = powerIn_1_arb(k_load,k_
            v2)+powerIn_2_arb(k_load,k_v2);
```

```
            efficiency_arb(k_load,k_v2) = powerOut_arb(k_load,k_v2)/
            combinedPowerIn_arb(k_load,k_v2);

        end
    end

    if 1==1
        % account for any situations where the system goes unstable.
        for k_rd2=1:length(rd2_array);
            if abs(imag(combinedPowerIn_arb(k_rd2)))>1e-9
                combinedPowerIn_arb(k_rd2)=combinedPowerIn_opt(k_rd2);
            end
        end
    end

    combinedPowerIn_diff = combinedPowerIn_arb-combinedPowerIn_opt;
    percentPowerSaved = 100*(combinedPowerIn_diff./(powerOut_arb));
    maxSavings(k_arbitraryDroop) = max(max(percentPowerSaved));
end

r1=0.8;
r2=0.8;
rl1=1.0;
rl2=1.0;
vnom=100;% volts
pnom = 500;% watts
rd1=(0.05.*vnom)/(pnom./vnom);
v1=40;% volts

% energy storage
C_bank = 150;
V_bank_max = 50;% volts
V_bank_min = 30;% volts
```

Appendix B

HIL Implementation Files

B.1 Typhoon HIL Schematic

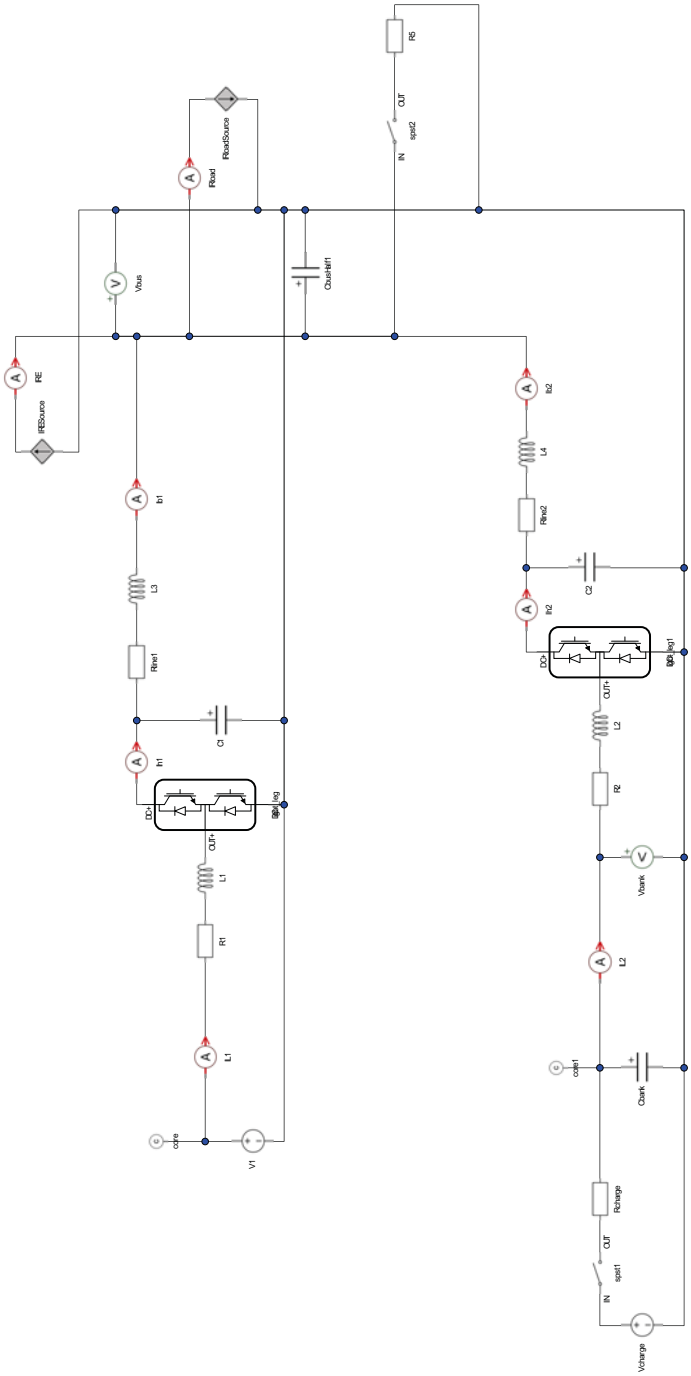

Figure B.1: Circuit-based model that was built and run in the TyphoonHIL400 HIL system.

B.2 System Plots from HIL Cycle Run with Constant R_{droop2} of 2.0 Ω

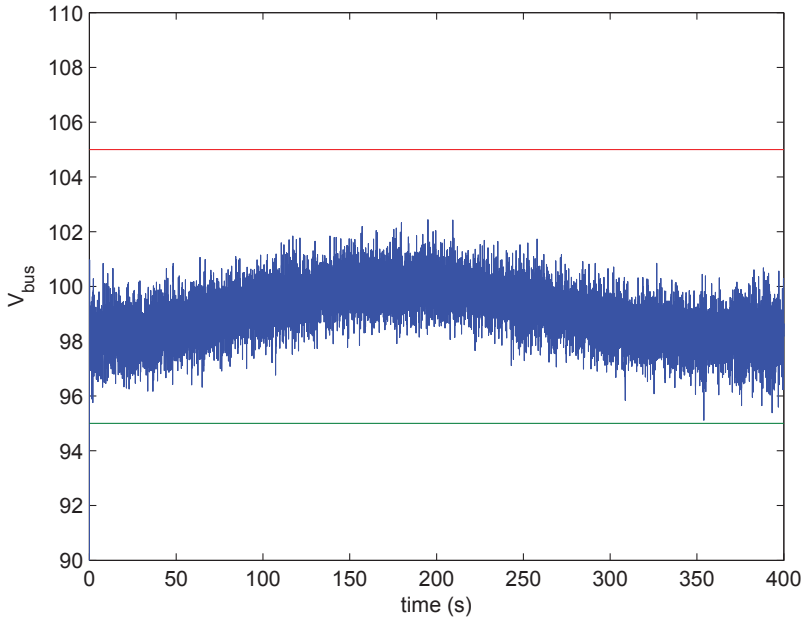

Figure B.2: Bus voltage, v_{bus}, obtained using a constant R_{droop2} of 2.0 Ω.

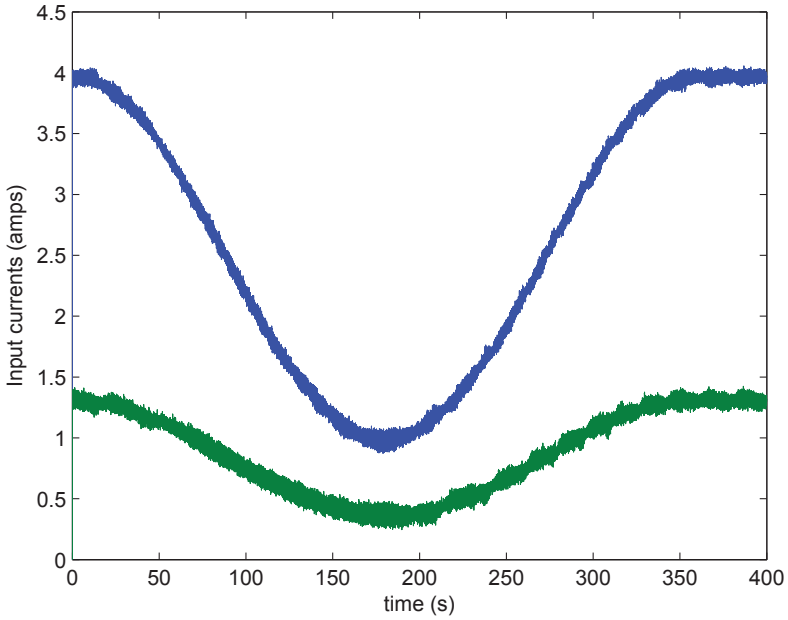

Figure B.3: Input currents, i_{L1} and i_{L2}, obtained using a constant R_{droop2} of 2.0 Ω.

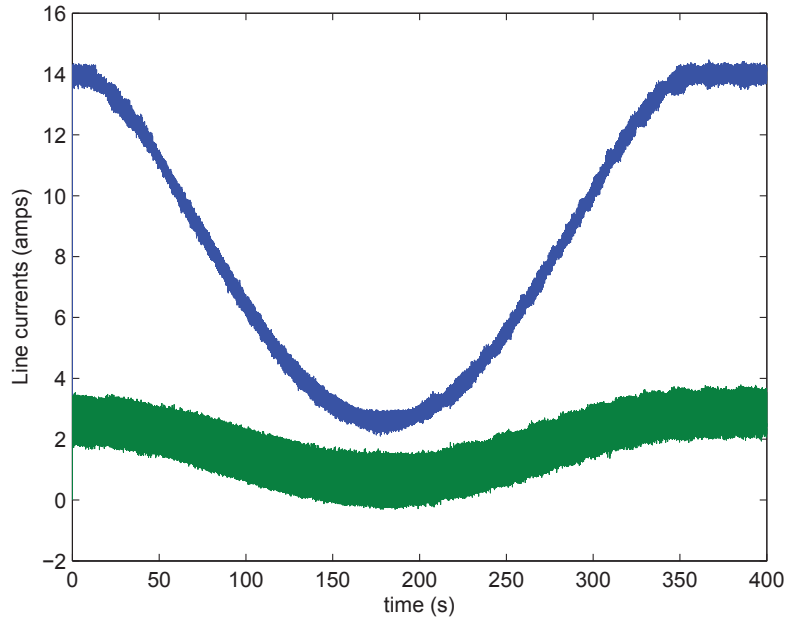

Figure B.4: Line currents, i_{b1} and i_{b2}, obtained using a constant R_{droop2} of 2.0 Ω.

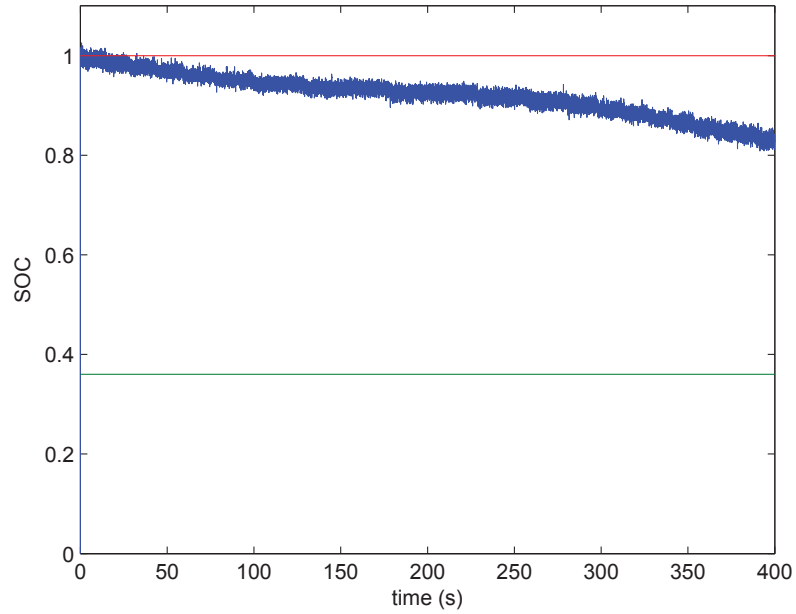

Figure B.5: Supercapacitor bank SOC obtained using a constant R_{droop2} of 2.0 Ω.

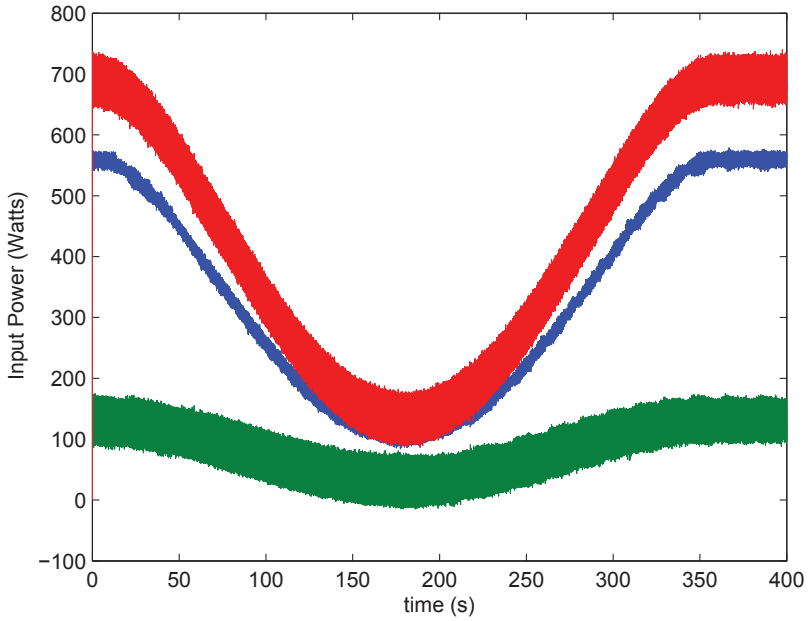

Figure B.6: Input powers P_{in1} and P_{in2}, along with the combined input power (cost) obtained using a constant R_{droop2} of 2.0 Ω.

B.3 Oscilloscope Images from HIL Cycle Run Using a Constant R_{droop2} of 2.0 Ω

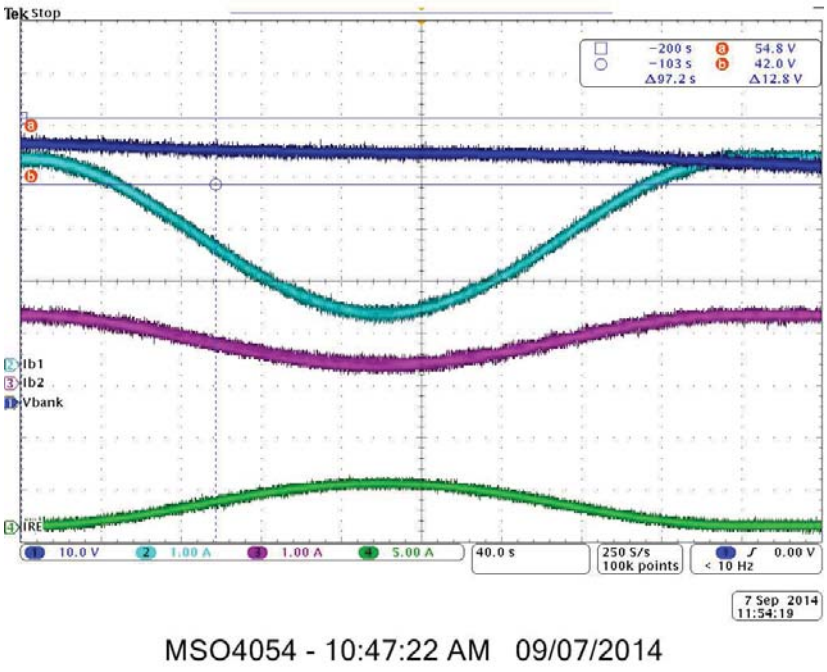

Figure B.7: Oscilloscope image "A" obtained using a constant R_{droop2} of 2.0 Ω.

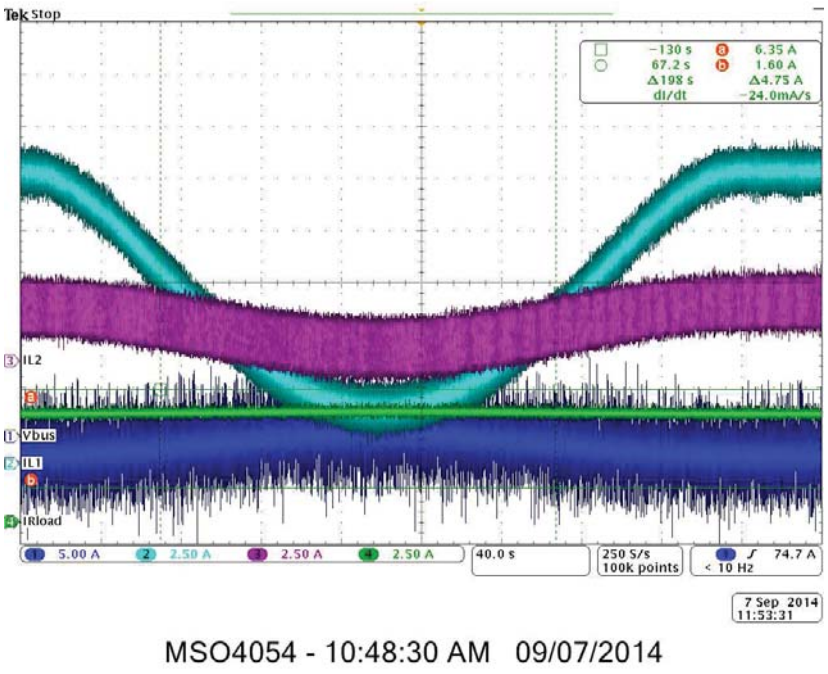

Figure B.8: Oscilloscope image "B" obtained using a constant R_{droop2} of 2.0 Ω.