1-1-2011

Power Supply on Chip DC-DC converter identification using black-box modeling techniques

Charles Craig Bilberry

POWER SUPPLY ON CHIP DC-DC CONVERTER IDENTIFICATION USING BLACK-BOX MODELING TECHNIQUES

By

Charles Craig Bilberry

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2011

POWER SUPPLY ON CHIP DC-DC CONVERTER IDENTIFICATION USING

BLACK-BOX MODELING TECHNIQUES

By

Charles Craig Bilberry

Approved:

Michael S. Mazzola Professor of Electrical and Computer Engineering (Director of Thesis)	Yaroslav Koshka Associate Professor of Electrical and Computer Engineering (Committee Member)
Raymond S. Winton Professor of Electrical and Computer Engineering (Committee Member)	James E. Fowler Professor of Electrical and Computer Engineering (Interim Graduate Coordinator)
Sarah A. Rajala Dean of The Bagley College of Engineering and Professor	

Name: Charles Craig Bilberry

Date of Degree: December 9, 2011

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Michael S. Mazzola

Title of Study: POWER SUPPLY ON CHIP DC-DC CONVERTER IDENTIFICATION USING BLACK-BOX MODELING TECHNIQUES

Pages in Study: 95

Candidate for Degree of Master of Science

With recent developments in power conversion technologies and market trends that are driving those technologies toward further miniaturization and greater integration, the need for verifying an empirically based modeling methodology for proprietary power converters such as Power Supply on Chip (PwrSoC) products has risen significantly. This need motivates the investigation of black-box models which require little or no knowledge of the internal workings of a system, for those areas of industry adopting PwrSoC technology as a point-of-load solution. This thesis reports a black-box modeling method tailored to accommodate but not limited to the requirements of a specific commercially available PwrSoC technology.

DEDICATION

The author would like to dedicate this thesis to God, his late great-grandmother, Myrtle Day Cox, and his loving wife all of whom have been great inspirations.

ACKNOWLEDGEMENTS

The author would first like to thank Dr. Michael Mazzola, his major advisor and mentor, who has been an endless source of knowledge, guidance, opportunity, and financial support. The author would also like to thank the graduate committee members, Dr. Yaroslav Koshka and Dr. Raymond Winton, for their feedback and time spent reviewing this thesis. In addition, the author would like to extend gratitude to Andy Lemmon, Chris Parker, and Jim Gafford for their advice, equipment expertise, and sharp criticism.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

CCM	Continuous Conduction Mode
DCM	Discontinuous Conduction Mode
PwrSoC	Power Supply on Chip
Y	Input Admittance
H	Reverse Current Gain
G	Forward Voltage Gain
Z	Output Impedance
Ω	Ohm
V	Volt
A	Ampere
i	Current
v	Voltage
PEM	Prediction Error Method
LEM	Least-square Error Method
MLM	Maximum-likelihood Method
dc	Direct Current
ARMAX	Regressive Moving Average with eXogenous Input
GTHOM	Oga Ota Ellha Leoa Issma
ARX	Auto-Regressive with eXogenous Input
OE	Output Error

BJ	Box-Jenkins (BJ)
s	Complex Frequency
t	Time
n	Index
EMI	Electromagnetic Interference
BNC	Bayonet Neill-Concelman
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
LTI	Linear Time Invariant

CHAPTER I

INTRODUCTION

1.1 Motivation and Problem Statement

With recent developments in power conversion technologies and market trends that are driving those technologies toward further miniaturization and greater integration, the need for developing an empirically based modeling methodology for proprietary power converters such as Power Supply on Chip (PwrSoC) products is compelling. To those in industry who may be considering adopting PwrSoC technology as a point of load solution, such models are essential. This need motivates the investigation into black-box models and underscores the contributions of this thesis.

The main goal of this thesis is to develop and apply a method for obtaining an accurate averaged model representation of a PwrSoC component across a wide range of operating conditions. Model complexity is a major concern due to the anticipation of this model being implemented in a simulation environment for man-portable power systems where execution time is an important factor and temporal results on the order of the controller bandwidth the primary objective. Furthermore, the model must also accommodate a range of input and output variations due to the many (and unknown a priori) operating conditions available to the user within the ratings of commercial PwrSoC product.

1.2 Literature Review

Throughout the past two decades with the advent of high performance computing tools, great strides have been made in the arenas of modeling and system identification. During that time, numerous methods of modeling power converters have been suggested in the literature. The most common approaches can be categorized as "white-box," "grey-box," and "black-box" modeling.

White-box modeling is a modeling approach where the physical design of a system is known prior to the modeling process. The black-box modeling approach, which does not require a priori information about the system, is completely derived from experimental data. Therefore, the resulting model has no physical connection to the internal state variables of the system [1]. The grey-box approach to modeling is a mixture of both the white-box and black-box modeling approaches. In grey-box modeling, some of the internal design information about the system to be modeled is known while other critical pieces of information are missing. Any missing information must be sought through experimental data collection and analysis. The model resulting from this type of approach contains both physical and non-physical parameters and it is occasionally referred to as a semi-physical model. The modeling approach chosen, as previously implied, is directly influenced by the availability of technical information about the internal workings of the device to be modeled. In the following review, a mixture of these approaches was taken by the authors of the work reviewed.

1.2.1 Circuit Equivalent Two-port Networks

In 1988, P. Maranesi, V. Tavazzi, and V. Varoli formally suggested the use of two-port hybrid g- parameter networks as a suitable equivalent circuit structure to model PWM voltage regulators [2]. Their exploration focused on the open loop characterization

of switched dc-dc converters in steady state. Those systems analyzed included the Boost, Buck, and Buck-Boost converters in both continuous conduction mode (CCM) and discontinuous conduction mode (DCM). Specific physical insight regarding each of the converters in the analysis was known. With that information, a physical connection between the two-port network parameters and the topologies physical component values was realized. The result was an analytical description for each of the two-port network parameters. This implies that a white-box approach was taken to model the systems. The closed loop system characterization was then derived from the open loop parameters.

It is concluded in Ref. [2] that a closed form model could be obtained independently of the switch locations within the circuit. It is also important to note that this type of two-port equivalent circuit network is not limited to a specific converter and may be used as a basis to model any of the basic switched dc-dc converters. It is also independent of the modeling approach taken whether it is a white-box, grey-box or a black-box approach.

1.2.2 Arrival of System Identification Theory

A key development in launching the black-box approach to a level of economic viability was the maturation of system identification theory. System identification theory is a broad and statistically rich topic that was first formalized by Dr. Lennart Ljung at Linköpings University in Ref. [3]. The text draws out a general procedure for the system identification process and details model structures and parameter estimation methods. A few of the parametric methods detailed include prediction error (PEM), least-square (LEM), and maximum-likelihood (MLM). It goes further considering experimental design and addresses some of the obscurities of system identification in practice. The

discussion also includes time-varying nonlinear systems as well as nonparametric methods, which do not require a finite set of parameters in the estimation process [3].

In a later paper by Dr. Ljung entitled "Black-box Models from Input-output Measurement", Ref. [4], a relationship between many of the different types of model structures in system identification theory as applied to the black-box approach is defined. In the paper, there is specific mention of linear input-output black-box models that directly apply to the system identification problem addressed in this thesis. The paper proposes four different types of models suitable for the black-box approach namely Auto-Regressive Moving Average with eXogenous input (ARMAX), Auto-Regressive with eXogenous input (ARX), output error (OE), and Box-Jenkins (BJ). Each of these models is similar in that they are input-output type models but differ in how disturbances enter the system. A more direct treatment of system identification theory as it is applied to this thesis is explained in CHAPTER III.

System identification has been termed both an art and science due to the inherently iterative processes of minimization that it requires. The text in Ref. [3] as well as earlier and later works, paved the way for many applications in the fields of black-box modeling and control systems. Specifically, Dr. Lennart Ljung's writing of MATLAB's "System Identification Toolbox" in 1986 allowed for a direct application of system identification algorithms to real world modeling and control problems. Many of the more recent works in the subject of switched dc-dc converter modeling employs MATLAB's "System Identification Toolbox".

1.2.3 Averaged Model Concept

Another key development in making black-box modeling and modeling of power electronic systems in general computationally efficient is the average model concept. The average dynamic model concept was first conceived and formally applied to switched dc-dc power converters by G.W. Wester and R.D. Middlebrook in Ref. [5]. In their work, an averaging technique is developed and applied to buck, boost and buck-boost converter power stages. This work was later extended in Ref. [6] by R.D. Middlebrook and Slobodan Ćuk where a bridge was formed between the average model techniques defined in Ref [5] and state-space model descriptions.

This averaged model concept is widely used today in modeling and controller design and is merely an average of two or more state-space model descriptions over a single switching cycle. Most importantly, it results in a single continuous state-space description eliminating the problem associated with full-bandwidth state-space models where multiple state-space descriptions are required e.g. a buck converter in continuous conduction mode requires two, one for the on state of the switch and one for the off state of the switch. A third description would be required if operation in DCM is considered. Obviously, dealing with multiple state-space descriptions is not ideal and should be avoided in time-constrained simulations. In [6], it is also pointed out that as long as the switching frequency is at least one order of magnitude above the corner frequency of the power converters output filter a sufficient averaged representation of the full-bandwidth system can be obtained. This statement is important in that it excludes the application of this technique to resonant type converters.

1.2.4 Modeling with Frequency Domain Data

The concept of modeling switched dc-dc converters as circuit equivalent two-port networks was revisited in [7] by a group at Virginia Polytechnic Institute in 2007. Armed with significant advances in system identification methods and computational tools capable of running complex identification algorithms e.g. MATLAB's "System Identification Toolbox", the team took a system level black-box approach to develop two-port network models for both a closed- and open-loop buck dc-dc converter. In this work, a complete small and large signal model was constructed using frequency response data obtained directly from the terminals of the hardware under test. In the experiment, using a linear amplifier, unity turns transformer, and a network analyzer, a broadband sinusoidal disturbance generated by the network analyzer was injected into the terminals of the hardware under test while the response of the system due to the disturbance was measured. This was repeated at both the input terminals and the output terminals. The frequency response data was then processed using the System Identification Toolbox in MATLAB. This resulted in the identification of the parameters to be used in the two-port network model.

In validating their model in the time domain, it was found that by changing the operating point and re-analyzing the data significant discrepancies existed between the models steady state response and the steady state response of the hardware. This was mostly attributed to non-linearity within the system. Being that the non-linearity was mostly found at dc, a simple look up table was utilized to accommodate the error. As a result, a good match between the dynamic and steady state response of the hardware and that of the models was shown. A similar validation approach is taken in this work.

1.2.5 Modeling with Time Domain Transient Data

Similar work was reported as recently as 2010 by V. Valdivia, A. Barrado, A. Lázaro, C. Fernández, and P. Zumel as described in [8] and [9] which details a modeling methodology. The methodology relies strictly on transient data acquired in the time domain. Simple disturbances such as input voltage steps and output current steps are applied to the terminals of the system under test and the resulting transient responses are measured.

The team showed that, in spite of the fact that closed-loop power converters are generally complex systems, the summation of first and second order dynamical systems is a suitable method for identifying the parameters of a two-port network. It was also pointed out that this method usually results in the identification of a relatively low order system description. The drawback to this method is that the transient response data must be meticulously analyzed resulting in a large number of iterations to obtain a reasonable model.

As a solution to eliminating tedious and highly iterative analysis during the identification stage, an automated method was suggested in Ref. [10]. This work uses MATLABS's "System Identification Toolbox" as a fitting and estimation tool for identifying the parameters of a two-port network utilized as a basis for dc-dc converter models.

1.2.6 Hybrid Wiener-Hammerstein Structures

Taking a grey-box approach to modeling dc-dc converters, J.A. Oliver, R, Prieto, J.A. Cobos, O. García and P. Alou at the Universidad Politécnica de Madrid proposed a hybrid Wiener-Hammerstein structure to model dc-dc converters based on experimental data and information available from the manufacturer [11]. The Wiener-Hammerstein

structure is shown to address static non-linear issues associated with inefficiencies in dc-dc power converters as well as dependencies of the output voltage on both load current and input voltage. The proposed dynamic input and output network models are assumed linear. Their mathematical descriptions or transfer functions are derived from information based on circuit configurations generally found in dc-dc converters. The work also considers effects of event driven behaviors such as over-voltage protection and over-current protection on overall stability within a large distributed dc power system. The implications of these effects are also considered in the work presented in this document.

1.3 Thesis Scope

The scope of this thesis is restricted to the development of an average model that simulates both the large and small signal behavior, using a black-box approach, of a particular PwrSoC product. The average model accurately replicates the dynamic response of the PwrSoC converter up to the Nyquist frequency. The model parameters are identified, using system identification techniques, from transient response data that is obtained in the time domain. The purpose of obtaining an average model is to eliminate computational efficiency problems associated with full-bandwidth models. The two-port network model structure as first suggest in Ref. [2] was selected as the basic average model structure, and the specific methodology as defined in Ref. [10]is used as a basis for the identification of the two-port network parameters.

The original work as defined in Ref. [10] applies to more traditional switched dc-dc converters while the model presented in this thesis extends that work to the highly integrated PwrSoC line of dc-dc converters. In addition, the constructed model considers

the effects of variation in output voltage set point to some arbitrary voltage within the normal operating space of the PwrSoC on the dynamic response of the system. This additional functionality is not considered in the approach as defined by Ref. [10] and is a significant contribution in terms of the model's flexibility for use in a wide range of operating scenarios. Furthermore, the model replicates event driven behaviors such as over current protection and quasi-current control operation typically found in commercially available PwrSoC products. The model, however, does not consider operation in discontinuous conduction mode (DCM) and is limited to operation in continuous conduction mode (CCM) only.

The final model is constructed in the MATLAB/Simulink simulation environment. It is packaged as a function block within the Simulink Library Browser. Deploying the model in a "wrapper" adds the convenience of a drag and drop system block where critical information is hidden behind a mask. The mask helps to prevent the user from unknowingly changing the structure of the model while providing an interface through which the settings of the system can be modified.

1.4 Thesis Organization

The origination of this thesis is geared toward presenting the material in a scientific form such that the reader will develop a clear understanding of the model and the methods used to construct it. The beginning chapter, CHAPTER II, presents the generic two-port black box model used as the base for the modeling of the PwrSoC module. It also presents the methods used in acquiring transient response data in the laboratory that is suitable for the parameter identification algorithm that is presented in CHAPTER III. CHAPTER III not only presents the parameter identification algorithm

but also details the techniques used to preprocess the data along with an analysis of the estimated parameters. In CHAPTER IV, the analysis of how the characteristics of the system change with operating point is presented along with the solution selected to address those issues arising due to changes in operating point. CHAPTER V presents the final model and large signal wrapper developed in Simulink while CHAPTER VI describes the validation process where independent data is used to measure the performance of the overall model. Finally, in CHAPTER VII, conclusions on the model are stated along with the known limitations of the model. Suggestions for future work and places were the model could be improved are also mentioned.

CHAPTER II

MODEL AND TESTING METHODOLOGY

2.1 Introduction

Modeling power converters using a black-box approach has been suggested by experts in the field of modeling, simulation, and control for a number of years. It is a broad concept that varies vastly in terms of complexity and implementation. The approach used in this work is based on obtaining transient response data from which the frequency domain parameters of a two-port network can be estimated. The approach is considered a black-box approach due to the lack of physical insight available for the PwrSoC module being modeled. This chapter focuses on introducing general concepts as well as the explanation of the two-port network model and the hardware testing procedure used to develop the averaged small signal model.

2.2 Concept Overview

To address the general model concept before probing into the more technical details drawn out in this document an overview of the general concept is presented. Figure 2.1 introduces the key components associated with the model, namely, the small signal model, the large signal model, event driven behaviors and the software wrapper.

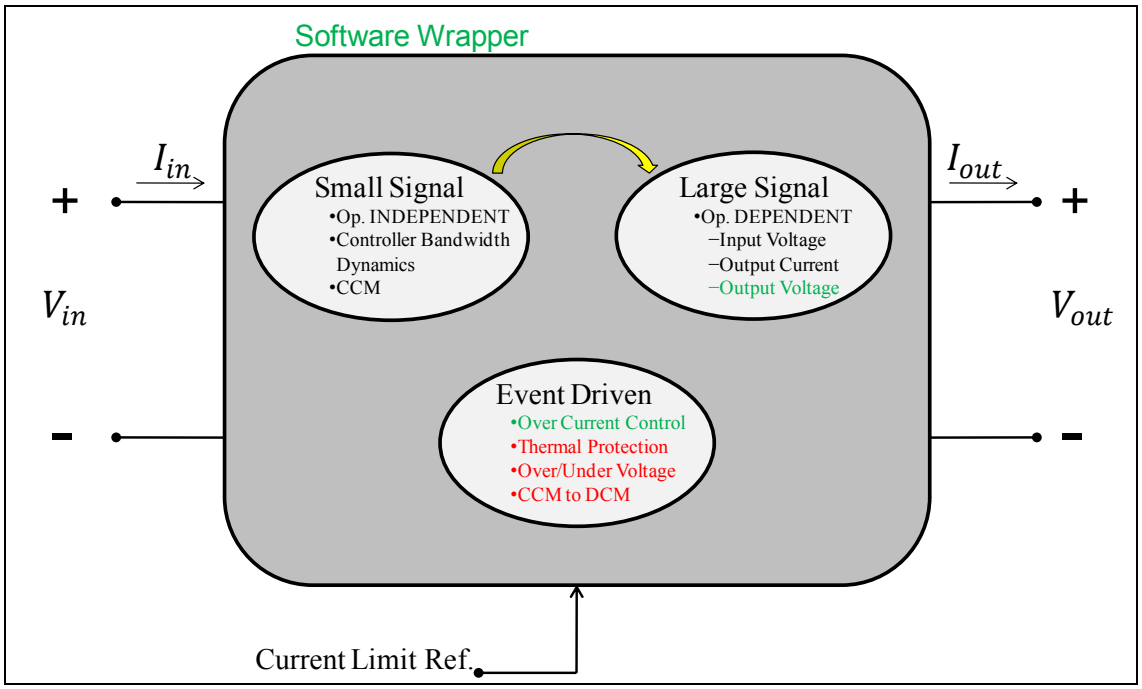

Figure 2.1 Graphic Overview

The overall black-box modeling concept contains the essential components of the PwrSOC model. The various components of the model are packaged in a software wrapper designed in the MATLAB/Simulink simulation environment which is represented by the dark grey rectangle in Figure 2.1. The wrapper allows the model to be quickly integrated into larger system simulations, and the interaction of the small signal model with the large signal features to be seamlessly controlled transparently to the larger simulation and to the user. The general inputs and outputs that are required for proper operation of the model are also indicated in Figure 2.1which includes the input voltage, output voltage and current limit reference. In the modeling process small and large signal models are separated. This separation transforms the operating point dependence of the PwrSoC into tractable analytical blocks that when combined reproduce a reasonable facsimile of the original physical behaviors. The small signal model is first obtained at a

single operating point. Then through analysis of the system response at a series of operating points, a mapping relationship between the small signal model and the large signal model is developed. As shown in CHAPTER IV, the transformation mapping is simplistic in an effort to preserve computational efficiency. The last of the major model components is the event driven behavior component. This component takes into consideration the high level of non-linearity associated with mode changes found in the PwrSoC device, or any similar power supply. Several types of operational mode changes exist in the PwrSoC but only the mode change from normal operation to over current control was considered in this work. The graphic also illustrates by text color what is commonly reported in the literature (black text) along with what is included in this work (green text). Figure 2.1 also presents in red text a few modeling tasks that are suggested for future work.

2.3 Black-Box Two-Port Model

A two-port g-parameter network was selected as a foundation to model the PwrSoC module using a black-box approach. The network is comprised of four dynamic transfer function models also referred to as parameters, namely input admittance $Y_i(s)$, reverse current gain $H_i(s)$, forward voltage gain $G_o(s)$, and output impedance $Z_o(s)$. A network diagram for the two-port g-parameter network commonly used in literature is also used in this work to construct the averaged small signal model and is shown in Figure 2.2.

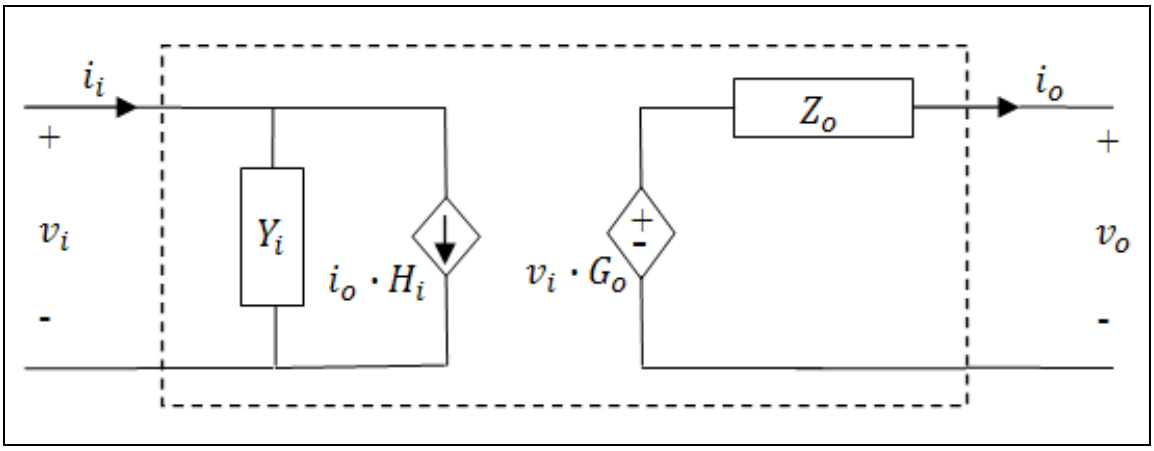

Figure 2.2 Black-Box Two-Port Network

The two-port network model utilizes dependent voltage and current sources to model dynamics that are reflected across the converter along with equivalent admittances and impedances to emulate the dynamic behavior found in the input and output networks. Therefore, standard circuit techniques are applied and by Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL) a mathematical relationship between each of the four parameters can be obtained and is expressed in 2.1 and 2.2.

$$i_i(t) = Y_i(t) \cdot v_i(t) + H_i(t) \cdot i_o(t) \quad 2.1$$

$$v_o(t) = G_o(t) \cdot v_i(t) - Z_o(t) \cdot i_o(t) \quad 2.2$$

It is important to observe that 2.1 and 2.2 form a linear set consisting of only time varying variables meaning that the small signal model does not consider dc quantities. More specifically, the input voltage $v_i(t)$ and the output current $i_o(t)$ are the two independent variables while the input current $i_i(t)$ and output voltage $v_o(t)$ are dependent quantities. It can be shown that holding the independent variable $v_i(t)$

constant, the following expressions for the reverse current gain parameter and output impedance parameter result in 2.3 and 2.4.

$$H_i(t) = \frac{i_i(t)}{i_o(t)}\Big|_{v_i(t)=0} \quad 2.3$$

$$Z_o(t) = -\frac{v_o(t)}{i_o(t)}\Big|_{v_i(t)=0} \quad 2.4$$

The same conclusions can be made about the other two parameters in the case that the output current is held constant. Under that assumption, the following equations for the input admittance parameter and forward voltage gain parameter are also valid.

$$Y_i(t) = \frac{i_i(t)}{v_i(t)}\Big|_{i_o(t)=0} \quad 2.5$$

$$G_o(t) = \frac{v_o(t)}{v_i(t)}\Big|_{i_o(t)=0} \quad 2.6$$

Now, by applying a Laplace transform with zero initial conditions to 2.3 and 2.4 the following frequency domain expressions shown in 2.7 and 2.8 result.

$$H_i(s) = \frac{i_i(s)}{i_o(s)}\Big|_{v_i(s)=0} \quad 2.7$$

$$Z_o(s) = -\frac{v_o(s)}{i_o(s)}\Big|_{v_i(s)=0} \quad 2.8$$

The same transformation is also applied to the input admittance and the forward voltage gain parameters shown in 2.5 and 2.6. Taking the Laplace transform yields the following frequency domain expressions in 2.9 and 2.10.

$$Y_i(s) = \frac{i_i(s)}{v_i(s)}\Big|_{i_o(s)=0} \quad 2.9$$

$$G_o(s) = \frac{v_o(s)}{v_i(s)}\Big|_{i_o(s)=0} \qquad 2.10$$

As a result of the previously derived expressions, each of the four dynamic parameters can be estimated independently of one another from measurements taken at the input and output terminals of the device under test.

2.4 Hardware Testing Methodology

The device under test is the Linear Technology LTM4612EV, Ultra-low noise, Buck DC/DC µModule Regulator. It has a wide operating voltage range capable of input voltages as high as 36 V and as low as 5 V with a corresponding output voltage range between 15 V and 3.3 V. It also boasts a continuous output power rating of nearly 60 W. The module is considered a PwrSoC technology due to its high-level of passive component integration and power density. The module is packaged in a small 15 mm x 15 mm x 2.8 mm Land Grid Array (LGA) package that encapsulates the analog controller, power-switching components, inductor, and a major portion of the required bulk capacitance [12]. Although, the module contains a limited amount of internal decoupling capacitance, the system requires additional external capacitance at the input and output power terminals in order to minimize output voltage ripple and ensure stable operation of the device. In the experiment, the recommended amount of externally supplied bulk capacitance is considered part of the device, i.e. inside the black-box, and is included in the complete PwrSoC model.

To characterize the PwrSoC device in accordance with the black-box approach, a minimum of two small signal laboratory experiments must be conducted. Both of the experiments are executed independently of one another based on the assumptions defined in the previous section. The first of the two laboratory test is the input perturbation test

where the excitation is a voltage step applied to the input terminals of the PwrSoC module. The second is an output perturbation test where the excitation is a load current step applied to the output terminals of the module. The input perturbation experiment pertains to the acquisition of the input admittance $Y_i(s)$, and reverse current gain $H_i(s)$, while the output perturbation experiment applies to the acquisition of the forward voltage gain $G_o(s)$, and the output impedance $Z_o(s)$. It is important to note that the amplitude of the applied voltage or load current step change, i.e. the excursion from the dc operating point, should be limited such that the small signal dynamics are not convolved with large signal behaviors.

Each of the aforementioned tests is conducted, as suggested in (4) and (5), with the constraint of holding the output current constant in the case of the voltage step experiment, and holding the input voltage constant as in the case of the output current step experiment. In practice, constant current and/or constant voltage sources do not exist, however, it is assumed and later shown in the model results that a nearly ideal source is sufficient. A detailed discussion of the test procedure along with measurement techniques is in the following subsections.

2.4.1 Input Terminal Perturbation

The voltage step is implemented at the input terminals of the PwrSoC device by connecting a MOSFET, acting as a switch, in parallel with a diode. Next, the combination is placed in series with a bench top dc power supply and the device under test. The schematic representation for the circuit designed to execute the input voltage step is shown in Figure 2.3 below.

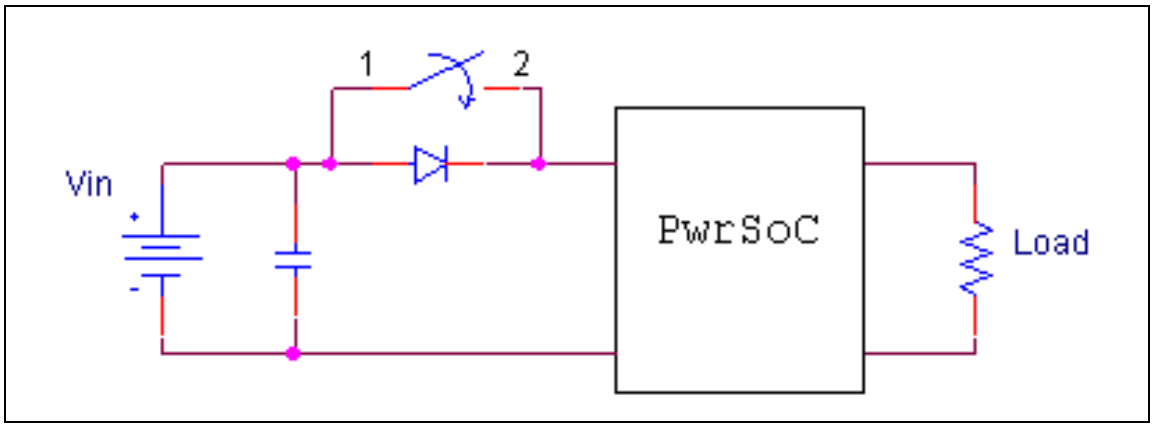

Figure 2.3 Input Terminal Excitation Test Circuit

At the time the MOSFET switch is closed, the inherent forward voltage drop of the diode is removed resulting in a nearly instantaneous rise in voltage at the terminals of the device. The magnitude of the voltage step is then approximately equal to the forward voltage of the diode, approximately 0.7 V for Silicon. If a larger amplitude voltage step is desired multiple diodes should be placed in series to generate a $n \cdot V_f$ step as suggested by [10].

Once the voltage step is applied, the response to that excitation is measured in the output voltage as well as in the input and output currents. The data from each measurement is saved as a MATLAB (*.dat) data file for later analysis and processing. In addition, care must be taken to ensure that the output impedance of the laboratory power supply is small so that it does not greatly influence dynamics measured at the terminals of the device under test.

2.4.2 Output Terminal Perturbation

The current step experiment was implemented with an N-channel MOSFET, operating as a constant current source, in parallel with a resistive load. The dc operating

point of the system is set by the resistive load while the load step is sourced through the MOSFET by applying a pulse signal between its gate and source terminals. The pulse or step change applied to the gate of the MOSFET is generated using a standard laboratory signal generator in pulse mode. As with the voltage step test, the response due to the current step excitation is measured in the input current as well as in the input and output voltages. The observed transient response data is then saved as a MATLAB (*.dat) data file for analysis and processing. A simplified circuit diagram of the input current step experiment is illustrated in Figure 2.4 below.

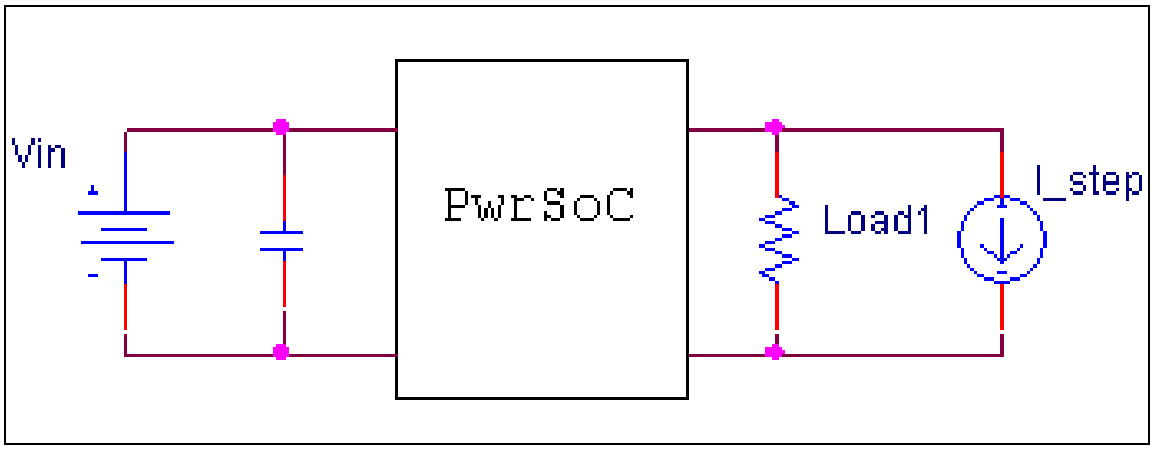

Figure 2.4 Output Terminal Excitation Test Circuit

2.4.3 Equipment and Metrology Considerations

Measurement and data acquisition is an integral yet challenging part of the laboratory testing process, especially, when capturing dynamics of small amplitudes. Therefore, special attention must be given to the metrology. In the test conducted in this work, a combination of input/output currents and input/output voltages were measured simultaneously with a digital oscilloscope. Care was taken to ensure that the sample rate setting on the oscilloscope was high enough, at least twice as high as the switching

frequency in order to meet the requirements of Shannon's sampling theorem, so that any aliasing effects would be avoided.

The input current was transformed to a voltage using a 40 mΩ shunt resistor allowing the input current dynamic response to be captured by the oscilloscope. The dynamics found in the output current were measured with a non-invasive but bandwidth limited current probe in the input voltage step test. A current probe was selected to measure the output current in this case because the dynamic response at the output terminals due to a perturbation at the input is assumed constant. In the output current step case, the dc output current was measured with a current probe while the step change riding on the dc value was measured using a 10 mΩ shunt so that the nearly instantaneous dynamics could be captured. The two measurements were summed together to form the complete signal.

BNC cables were used in place of generic probes so that the measurements were less susceptible to EMI and other noise sources. A clean noise free measurement is important for the estimation algorithm to produce a good model from the input/output measurements. In addition to BNC cables to further reduces noise in the input and output current measurements, the BNC cables on the current measurement channels were terminated into 50 Ω termination loads.

2.5 Small-Signal Data Analysis

With the hardware test plan finalized and the proper measurement equipment in place, the previously proposed input/output experiments were executed in the laboratory. The PwrSoC module was first configured to operate with 10 V at the input while bucking down the output to 5 V. A static load current of 2 A was set by connecting a 2.5 Ω load

resistor to the output terminals of the PwrSoC. These operating conditions for the small signal model were selected such that the PwrSoC module is operating in the middle of its normal operating envelope. This is to ensure that practical implementation issues such as control signal saturation, minimum and maximum duty cycle limitations and DCM is avoided. Both the voltage step test and the current step test were performed on the LTM4612 at the previously suggested values for the output current, input voltage and output voltage. The experimental test data obtained at this particular operating point is presented and analyzed below.

The experimental data shown in Figure 2.5 is the input current transient response due to the voltage step shown in Figure 2.7, with amplitude of approximately 0.75 V, applied at the input terminals of the PwrSoC module. The transient response in Figure 2.5 is as expected of a buck converters input network, which is dominated by bulk capacitance and the associated parasitic resistances.

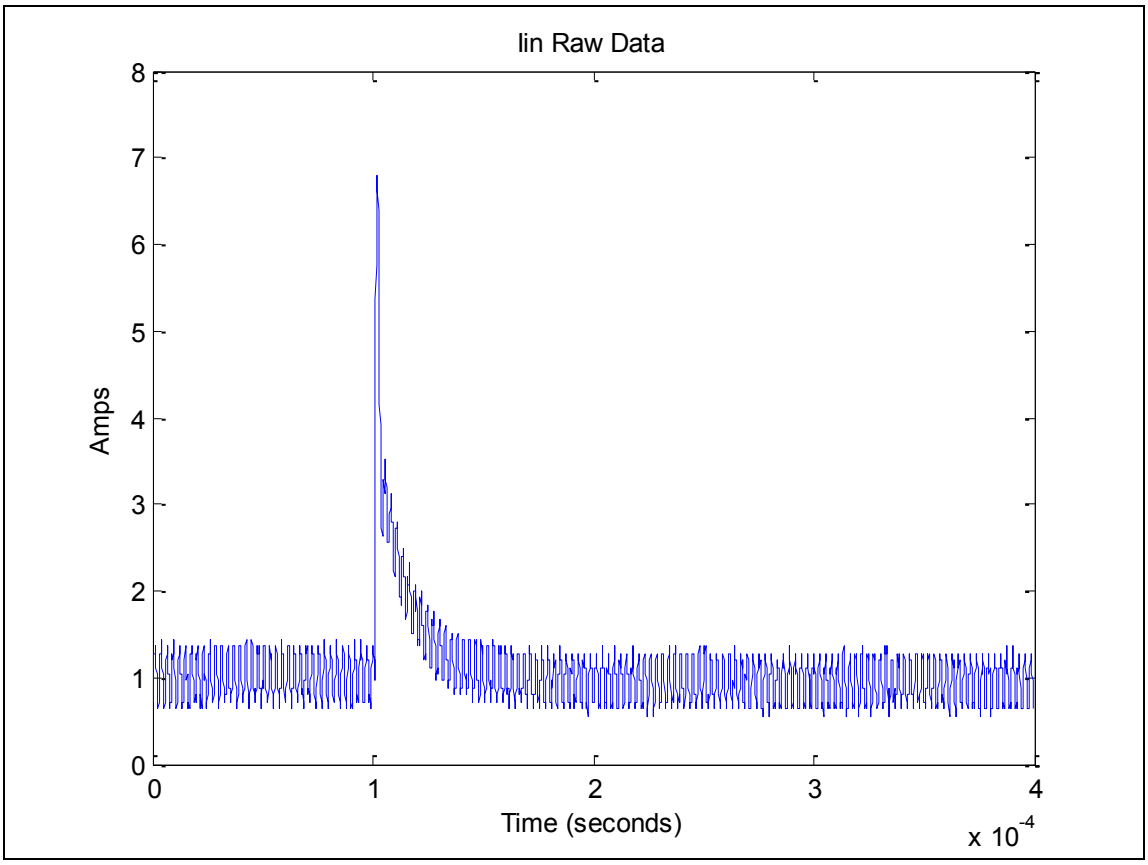

Figure 2.5 Input Current Response

Figure 2.6 is the PwrSoC modules output voltage response due to the voltage step excitation shown in Figure 2.7 that is applied to the input terminals of the module. The response is that of a well-designed buck converter in that any voltage disturbances at the input terminals of the converter are significantly attenuated at the output. This results in a minimum disturbance effect on the regulated output voltage. It will become apparent in analyzing the frequency response of the estimated forward voltage gain parameter $G_o(s)$ that this is a valid statement.

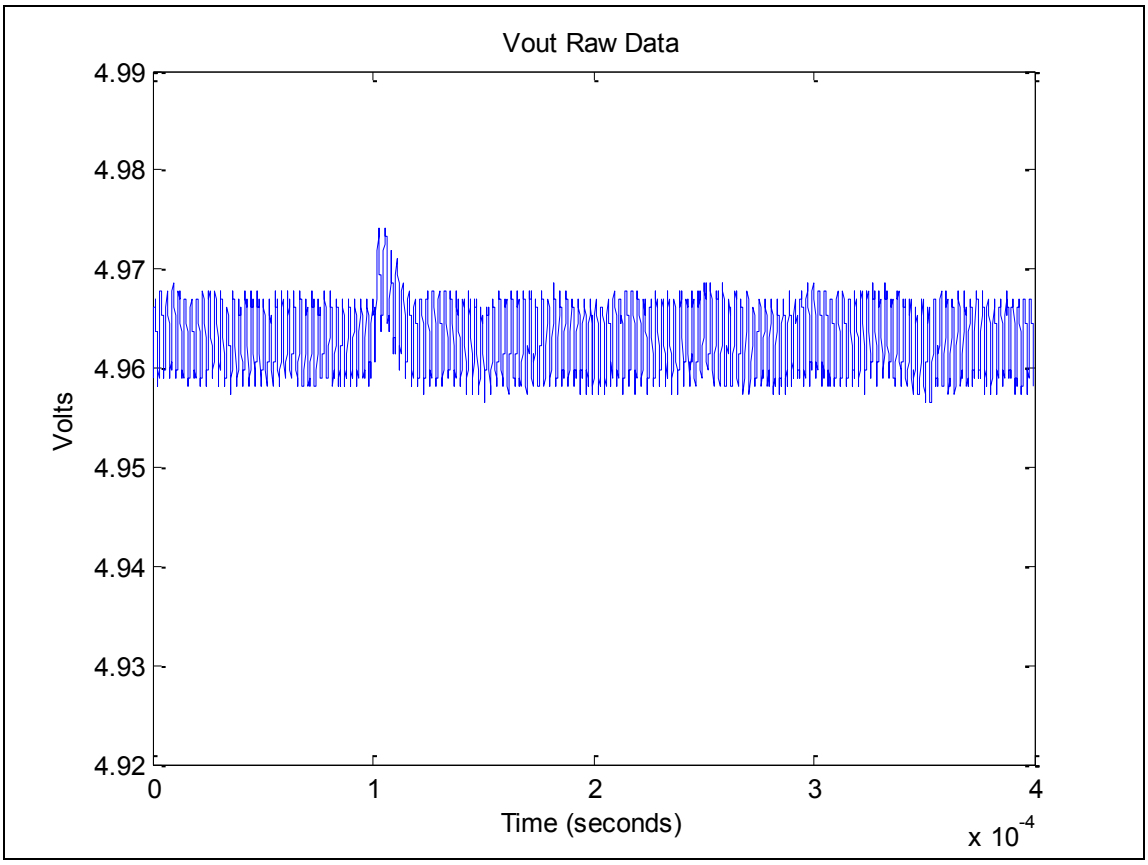

Figure 2.6 Output Voltage Response

The voltage step shown in Figure 2.7 was used to generate the previously discussed transient responses in Figure 2.5 and Figure 2.6. The voltage steps amplitude is kept small to around 0.75 V to ensure that large signal behaviors are minimized. It is important that the slew rate of the voltage step function is sufficiently high so that all of the natural modes of the system are excited.

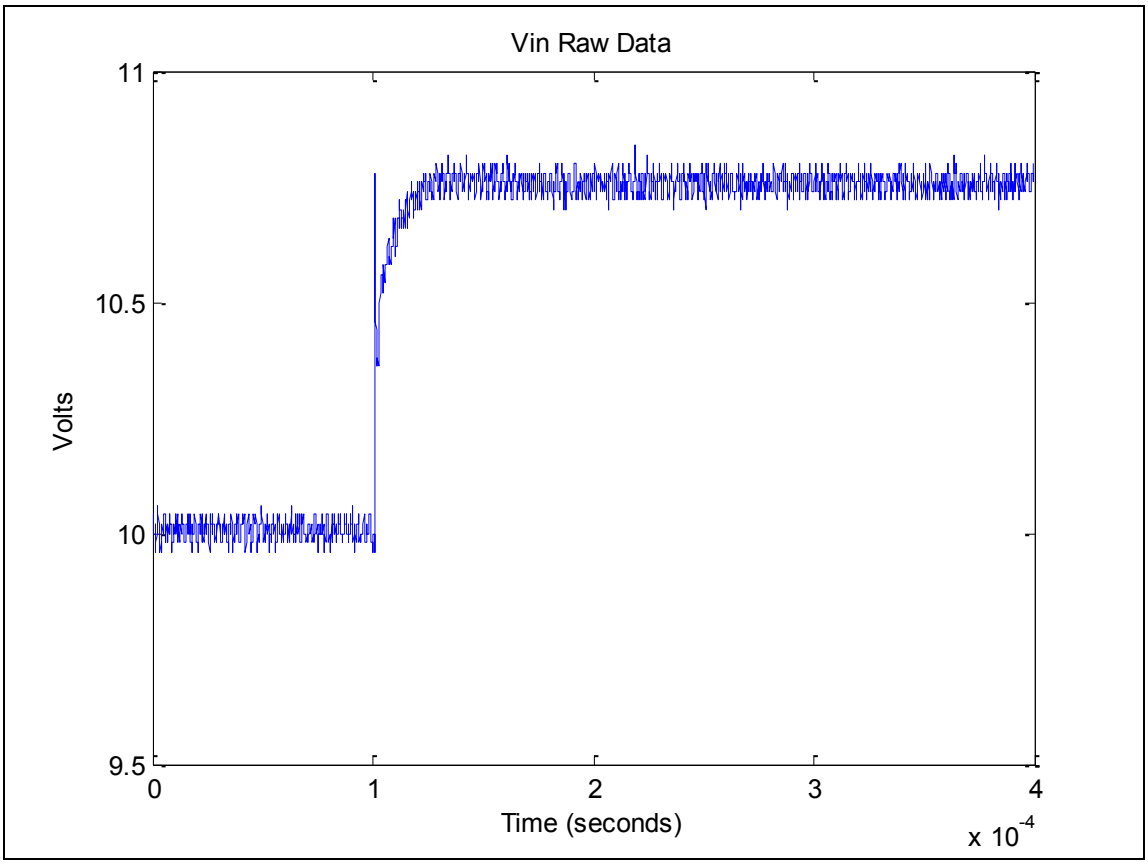

Figure 2.7 Input Terminal Excitation

Shown in Figure 2.8 is the output load current that is recorded during the input voltage step test. It should be noted that the output current is held virtually constant during the voltage step excitation. This makes it possible to estimate the input admittance and forward voltage gain parameters independently as was suggested earlier in Section 2.3.

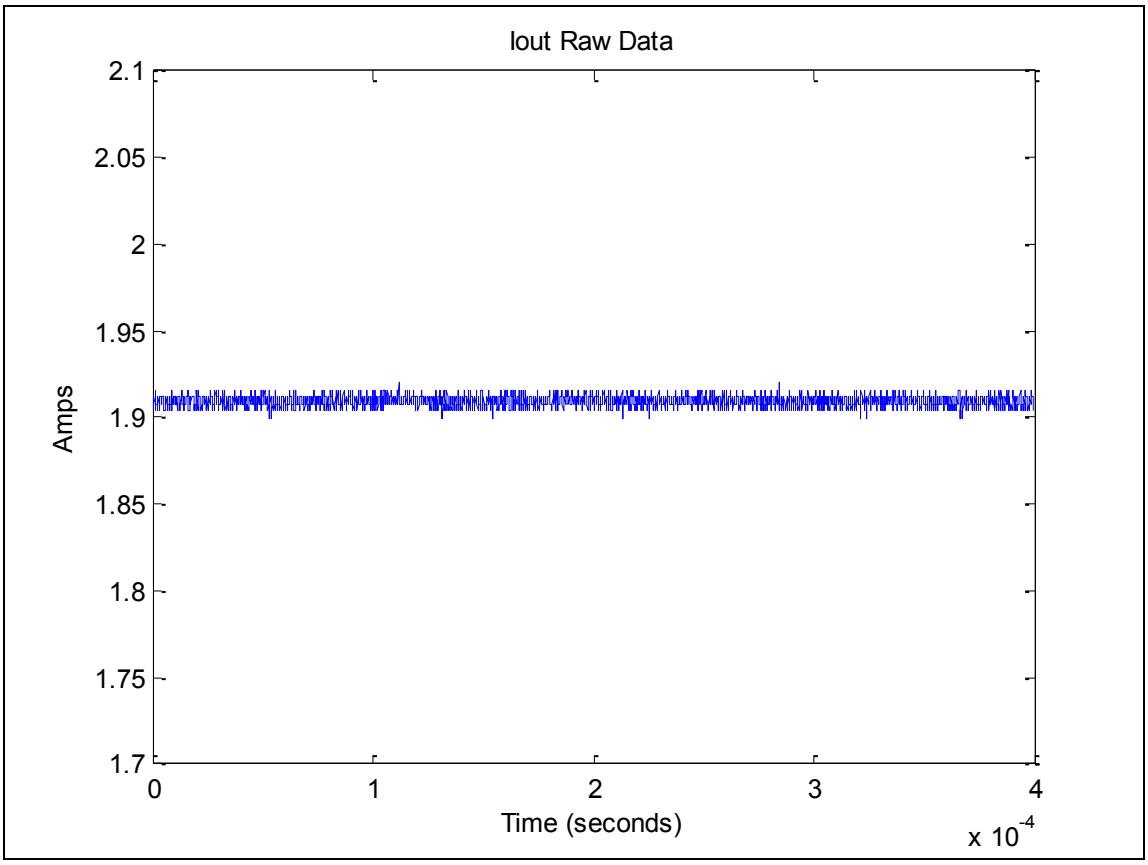

Figure 2.8 Output Load Current Response

The transient response data presented in Figure 2.9 and Figure 2.10 is the measured transient response due to the current step excitation as shown in Figure 2.12 applied to the output terminals of the PwrSoC device. The experimental test was executed at the same operating point as the previous voltage step test, 10 V at the input, 5 V at the output and a 2.5 Ω resistive load corresponding to approximately 2 A of static load current. An additional load of 0.5 A was switched in resulting in the output current step excitation.

The dynamic response seen in the input current is as expected from a voltage regulated converter. This is due to the conservation of energy law and results in the input

current being directly proportional to the output current, i.e. increasing one results in the increase of the other.

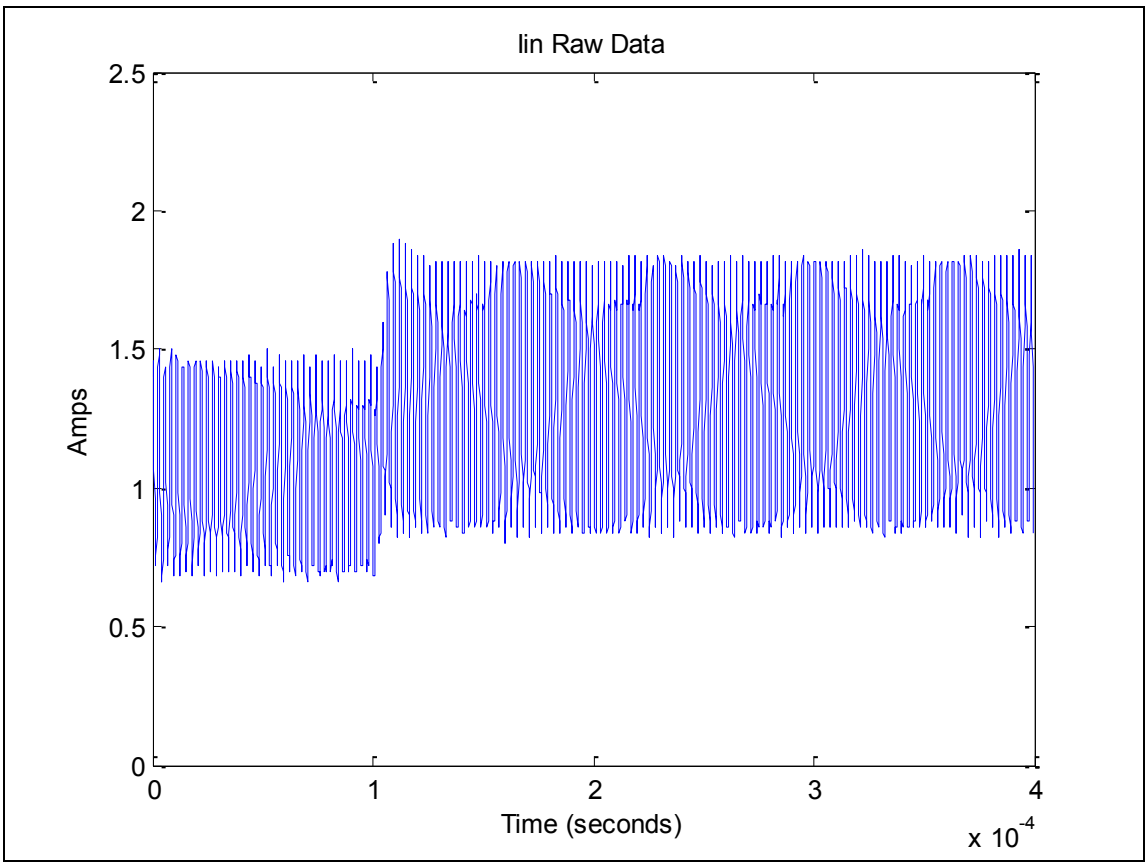

Figure 2.9 Input Current Response Due to a Load Step

The raw transient response data shown in Figure 2.10 is the transient response seen in the voltage at the output terminals of the device. The response here is regulated by feedback control where the design goal is to minimize overshoot while returning to the desired output voltage as fast as possible. The response is approximately second order which is expected of the buck converter topology that is predominately made up of an inductor in parallel with a large capacitance.

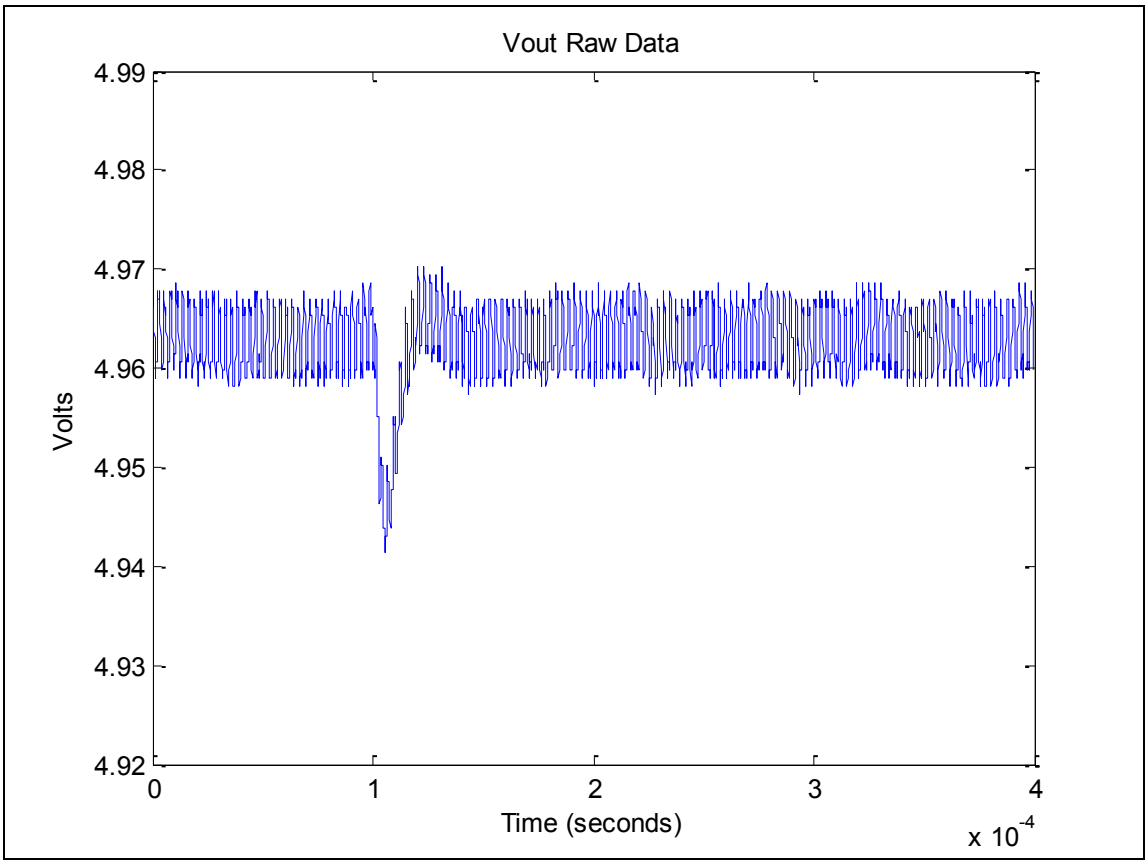

Figure 2.10 Output Voltage Response

The input voltage was measured at the time of the load step and is shown in Figure 2.11 below. It should also be noted that, similarly to the previous voltage step test, the input voltage is relatively constant allowing for the output impedance parameter and the reverse current gain parameter to be estimated independently. The nearly perfect input voltage source was achieved by minimizing the output impedance of the dc laboratory supply. This was done by making the supply leads a twisted pair and by minimizing their length. Also, bulk capacitance was added at the output of the laboratory supply to further reduce the influence of stray inductances.

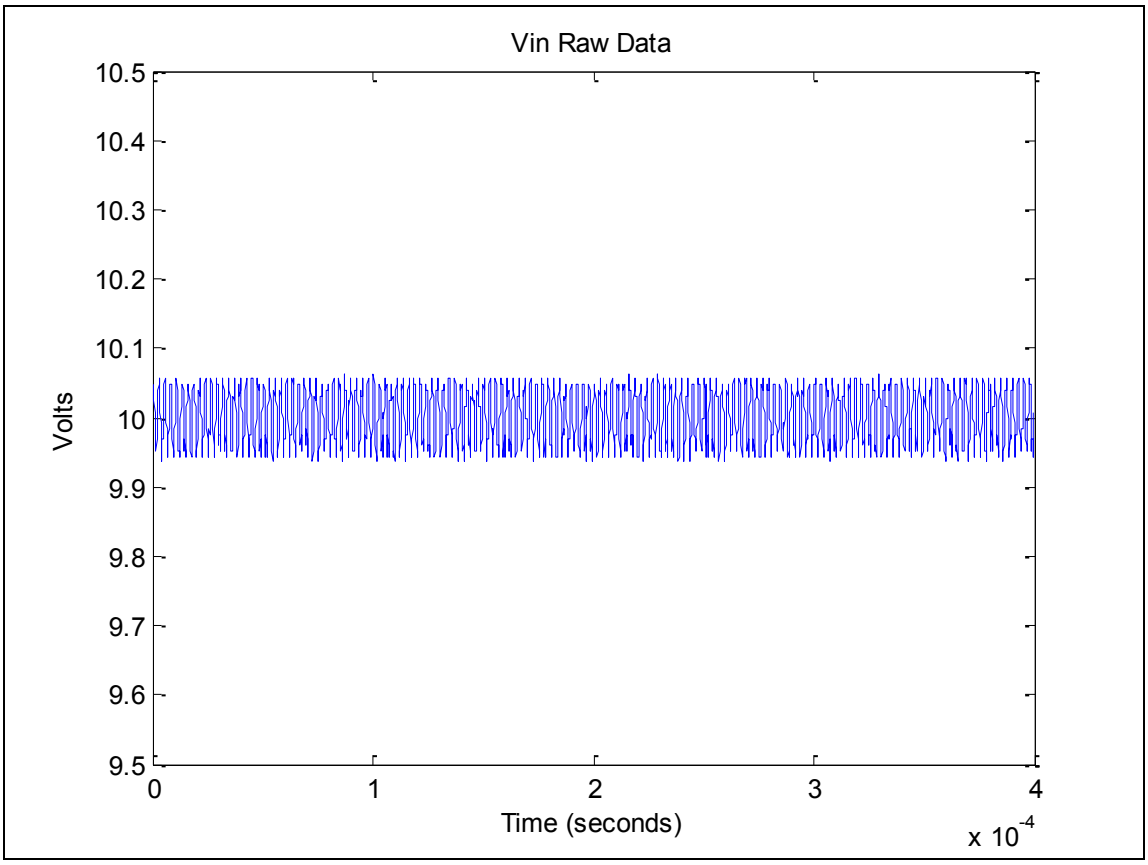

Figure 2.11 Input Voltage Response Due to a Load Step

The last of the transient responses to be analyzed is the output current step. As described before, the step function is generated by switching on an N-channel MOSFET driving it quickly into the saturation region. Again, the main issue to be addressed with the step function is that the slew rate be high enough to excite all of the dynamic modes within the system. The step function used as the disturbance in this test is shown in Figure 2.12 below.

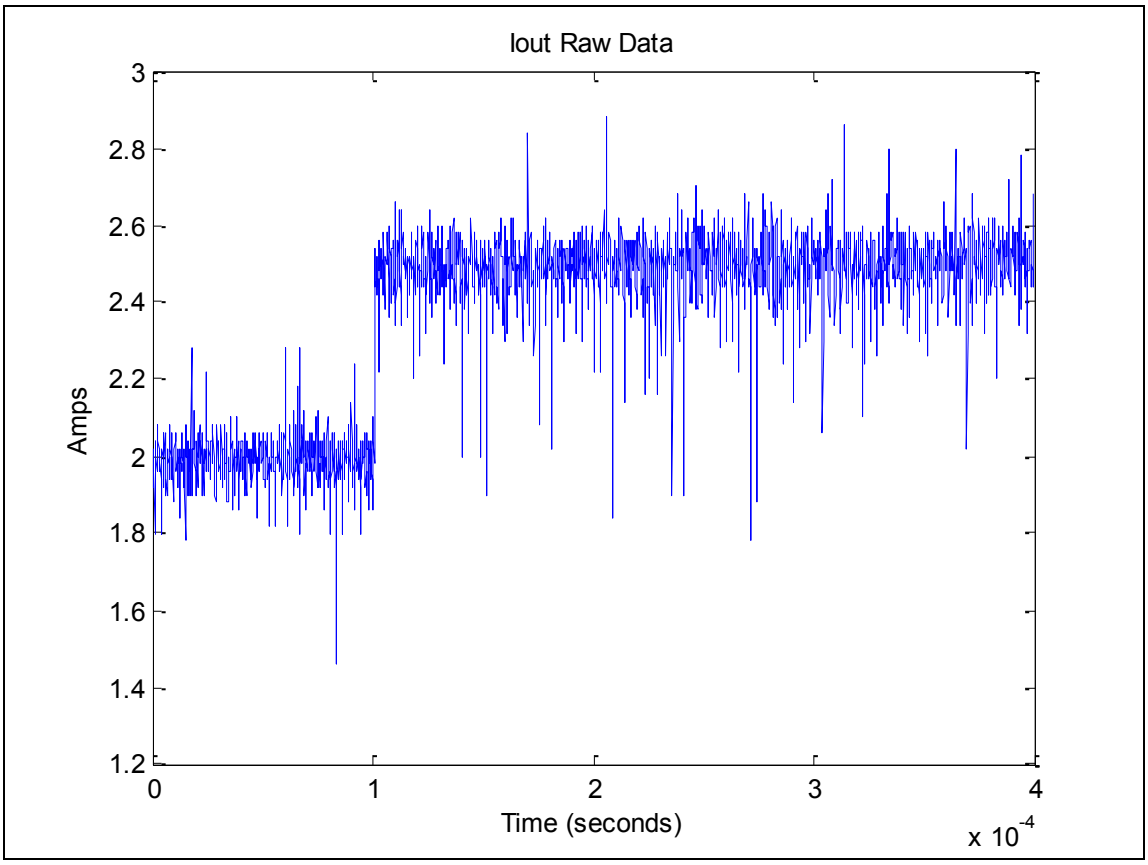

Figure 2.12 Output Current Step Excitation

CHAPTER III

PARAMETER ESTIMATION

3.1 Introduction

One of the toughest challenges of the black-box approach to modeling is in the estimation of the two-port network parameters. The difficulty is in the iterative nature of the estimation processes where the desire is to obtain a transfer function for each network parameter that accurately reproduces transient responses observed in laboratory tests. The process used to identify two-port network parameters and an analysis of the estimated parameters resulting from the process is presented in this chapter.

3.2 Parameter Estimation Algorithm

Once the time domain transient data was acquired from the experimental excitation tests described in CHAPTER II, the experimental transient response data was then imported into MATLAB where each of the four dynamic models in the form of transfer functions, namely: input admittance $Y_i(s)$, reverse current gain $H_i(s)$, forward voltage gain $G_o(s)$, and output impedance $Z_o(s)$ of the two-port network model could be estimated. The system identification toolbox, available in MATLAB, provides software tools that are designed for estimating transfer functions from input/output transient data. Many of these tools have been utilized extensively within a software algorithm developed specifically for identifying two-port network parameters. A flow chart of the algorithm is provided in Figure 3.1.

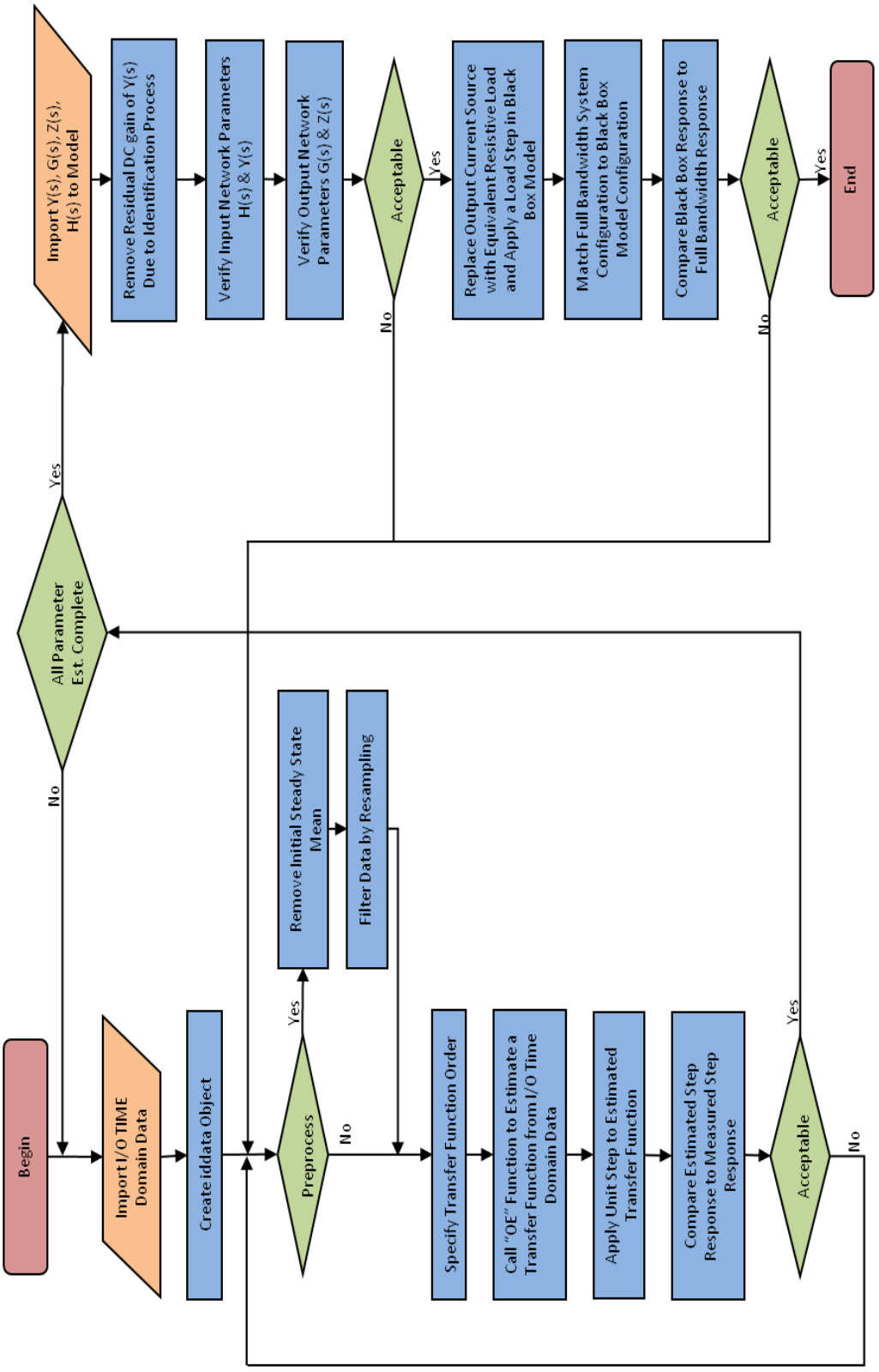

Figure 3.1 System Identification Algorithm

The flowchart in Figure 3.1 documents the process through which the transient response data is transformed from the time domain to the frequency domain. The flowchart indicates the iterative nature of the parameter identification process. A few key data preprocessing steps must be taken first in order to get the experimental data into the proper form.

3.2.1 Raw Data Processing

The system identification algorithm is, firstly, responsible for preprocessing the raw experimental transient data collected at the test bench. Preprocessing includes the removal of any initial mean value in the data as well as filtering. By removal of initial mean, it is meant that the signal begins at zero, e.g., the average steady state value of the signal before any excitation is applied is set to zero. The initial mean value must be removed so that the estimation algorithm fits to the small signal dynamics and not the large signal characteristics, which are discussed later in CHAPTER IV. In addition to removal of the average initial mean, the experimental data must be filtered in order to remove any unwanted high frequencies. The high frequencies to be filtered include any frequency that is above half the switching rate of the converter. The process of filtering can be accomplished using a number of different methods but it was found that the most effective means of filtering in this case is by decimation, also known as re-sampling. Now, the selected re-sampling rate is at approximately the frequency of the ripple in the output voltage data or at the switching frequency of the converter if it is known. Re-sampling the data at approximately the switching frequency results in a nearly averaged representation of the original waveform and since the objective is to obtain an averaged

model and not a full-bandwidth model the pre-averaged data results in the correct parameter estimates [13].

3.3 Identification Results and Analysis

The estimated two-port network parameters for the small signal model were validated using two different approaches, i.e. by examining the frequency response and time domain response of each of the estimated parameters transfer functions. The frequency responses were first examined to see if they were reasonable from a practical perspective. Although, there is very little knowledge of the physical characteristics of a system when using a black-box approach to modeling it is of good measure to put to use any practical insight of the system that was gained from previous experiences. With that said an examination and analysis of frequency and transient responses for the estimated two port network parameters is presented in the following sections.

The first of the estimated parameters to be examined is that of the input admittance parameter $Y_i(s)$, which is based on the transient response data acquired from the input voltage excitation test. The estimated transfer function generated by the system identification algorithm for the input admittance is in 3.1 below. In analyzing the transfer function, it should be noted that it is of relatively low order.

$$Y_i(s) = \frac{6.622\times10^{6}\cdot s^{2}+5.005\times10^{12}\cdot s-8.125\times10^{15}}{s^{3}+1.094\times10^{6}\cdot s^{2}+1.061\times10^{12}\cdot s+1.048\times10^{17}} \qquad 3.1$$

In order to provide a lucid picture of the input admittance parameter's frequency response a Bode plot is generated using MATLAB. The Bode plot of the estimated input admittance parameter is shown in Figure 3.2. It is clear that the frequency response is dominated by an RC time constant associated with the input network. This RC type of

frequency response is a reasonable result in that the input impedance of the PwrSoC module is expected to be predominately that of resistances and capacitances.

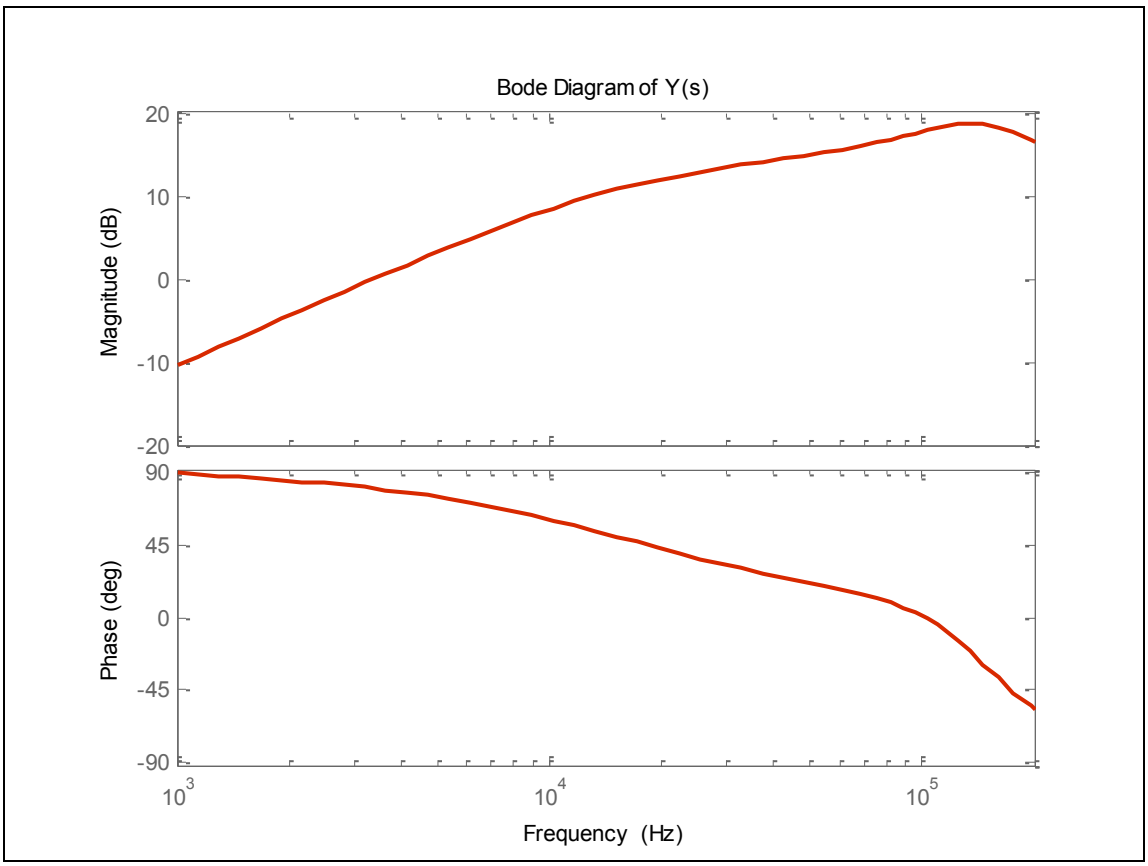

Figure 3.2 Input Admittance Estimated Bode Diagram

To verify that the estimated transfer function predicts the transient response of the PwrSoC converter a comparison of the time domain transient response of the estimated transfer function and the experimental data is done. The responses are plotted on a single plot for comparison as shown in Figure 3.3 below.

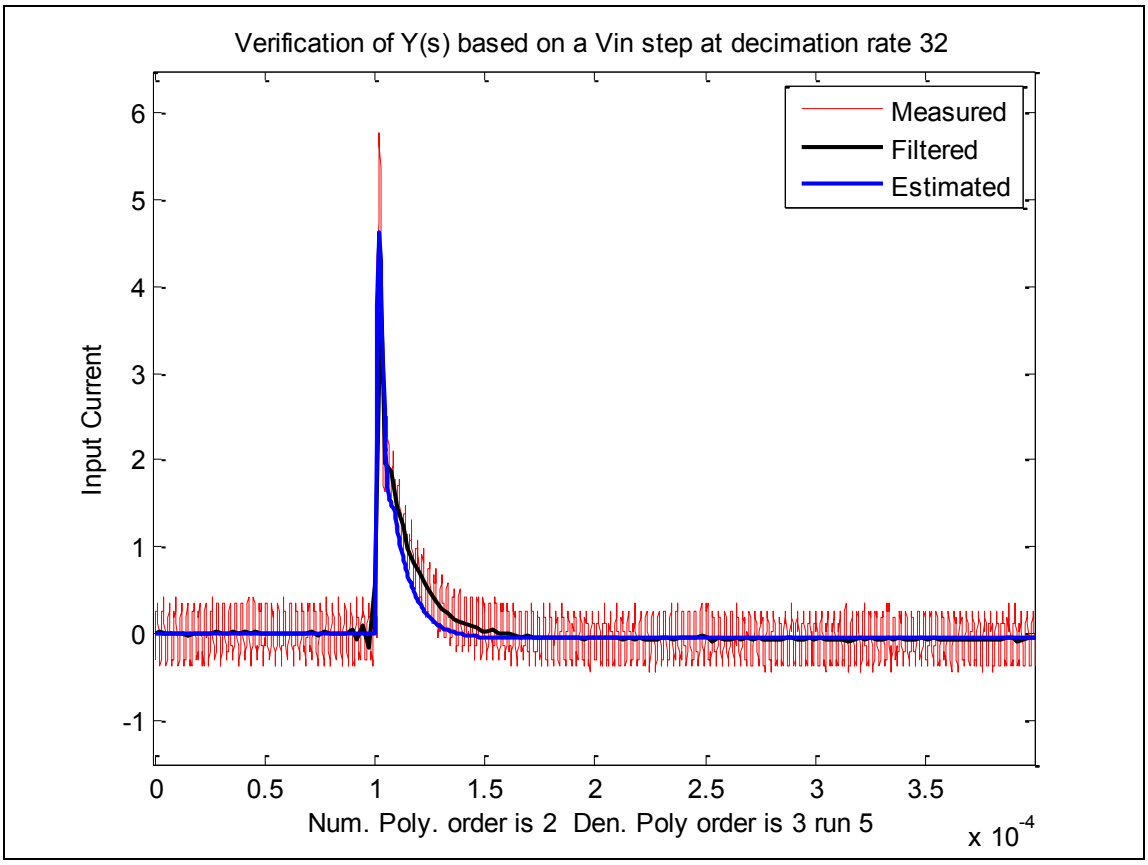

Figure 3.3 Validation of Input Admittance Parameter

Figure 3.3 shows three different waveforms, the measured experimental transient data, the filtered version of the measured experimental transient data and the resulting transient response of the estimated input admittance parameter. The transient response of the input admittance parameter was generated using the "step" function in MATLAB. It is clear that the estimated response does a good job of replicating the average response of the experimental data.

The second parameter to be analyzed is that of the forward voltage gain parameter $G_o(s)$. The forward voltage gain parameter is based on transient responses generated by the input voltage excitation test. The input voltage step data and the output

voltage transient response data was fed into the system identification algorithm and the transfer function in 3.2 was generated for the forward voltage gain parameter.

$$G_o(s) = \frac{8.087\times10^{8}\cdot s^{2}+2.211\times10^{9}\cdot s-7.471\times10^{10}}{s^{3}+8.562\times10^{5}\cdot s^{2}+2.999\times10^{11}\cdot s+4.246\times10^{16}} \qquad 3.2$$

Once again, the system identification algorithm produced an estimated transfer function that is of low order, which is highly desired for computational purposes. The frequency response of the forward voltage gain parameter is provided in the Bode plot of Figure 3.4.

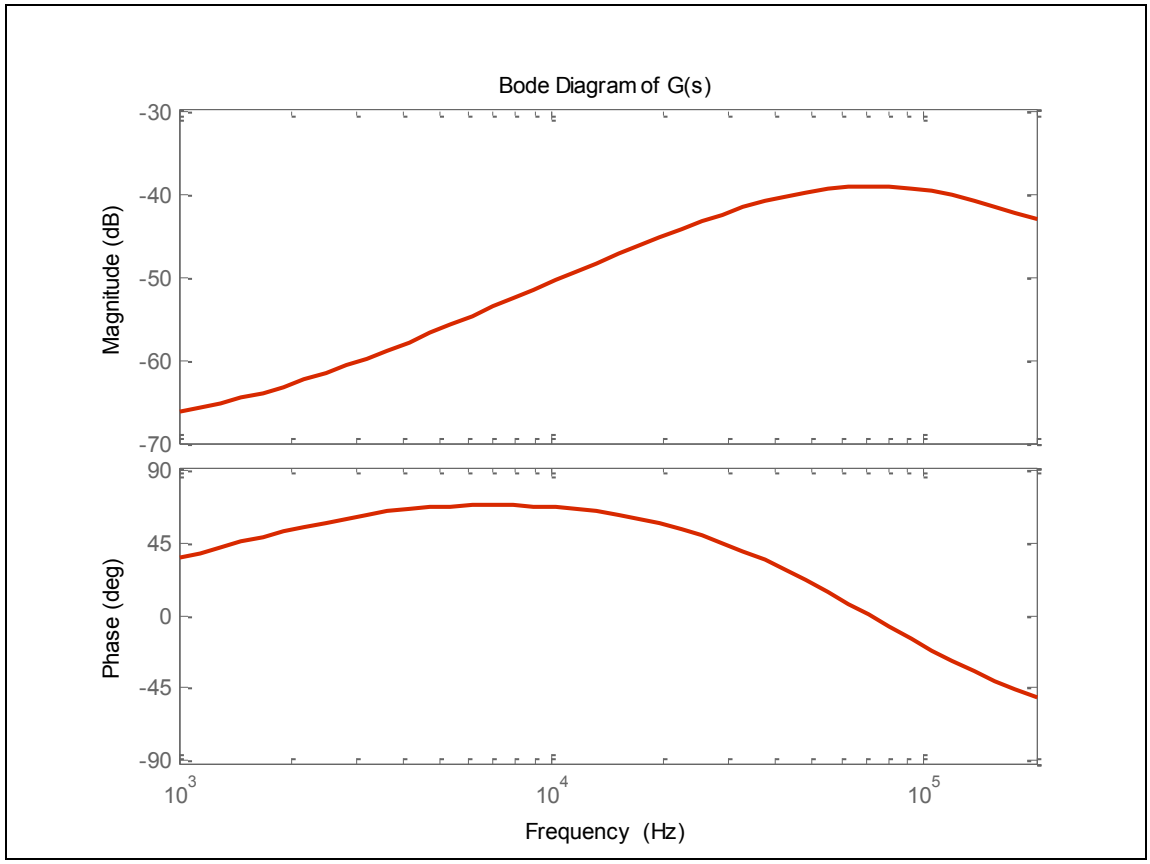

Figure 3.4 Forward Voltage Gain Bode Diagram

In analyzing the Bode plot in Figure 3.4, the magnitude plot shows that any signal or disturbance injected at the input of the system will be highly attenuated at the output. This response is reasonable in that a buck converter operating in closed loop is specifically designed to prevent disturbances in the input voltage from influencing the output voltage which is also known as line regulation. This characteristic makes measurement of the response in output voltage due to an excitation in the input voltage particularly difficult because the transient response is low in amplitude. The transient response seen at the output terminals of the PwrSoC device is shown in Figure 3.5 below where a comparison is made between the raw experimental data and the transient response of the estimated transfer function.

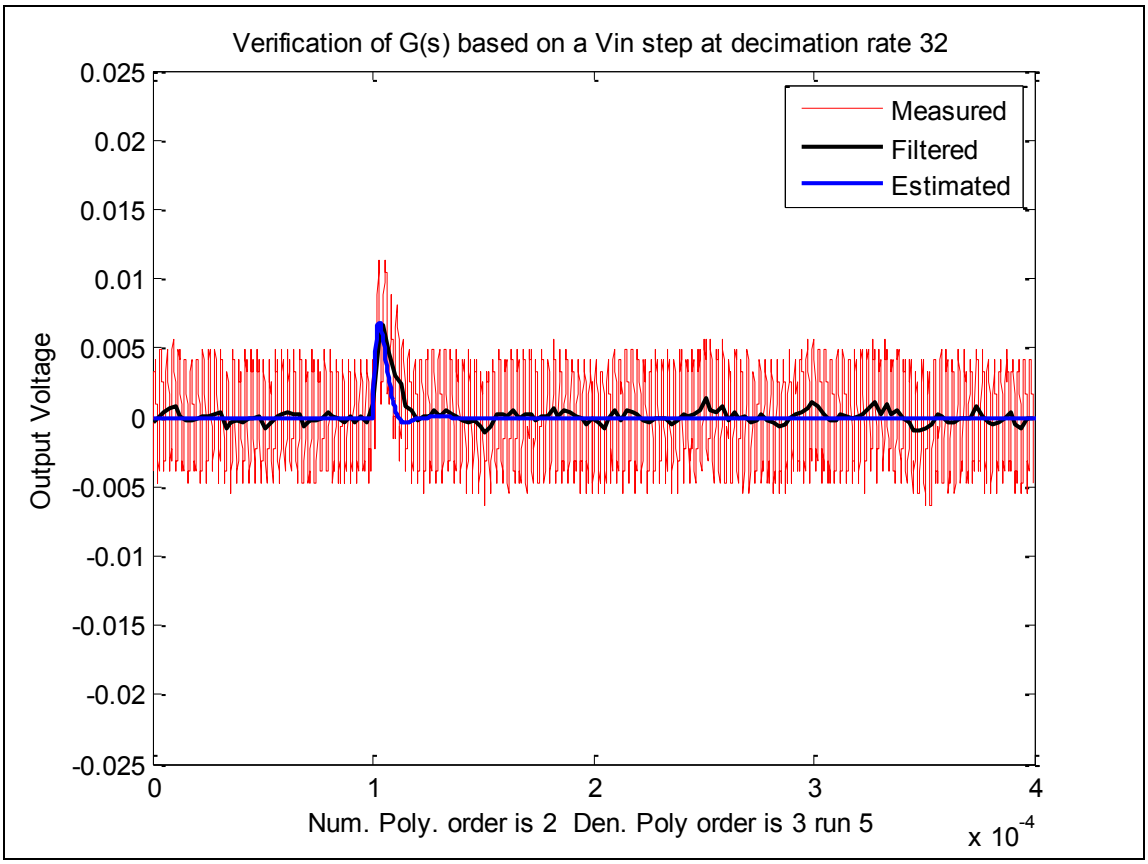

Figure 3.5 Validation of Forward Voltage Gain Parameter

As shown in Figure 3.5, the stepped transient response of the estimated forward voltage gain parameter matches the experimental data nearly perfectly.

The output impedance parameter $Z_o(s)$, is estimated using transient response data generated by the output current excitation test. The experimental data is then fed once again into the system identification algorithm were the result is a rational transfer function expression as shown in 3.3. The estimated network parameter model is also of relatively low order similar to the previously estimated parameters.

$$Z_o(s) = \frac{1.837\times10^{4}\cdot s^{2} - 5.899\times10^{9}\cdot s + 6.420\times10^{12}}{s^{3} + 6.401\times10^{5}\cdot s^{2} + 1.514\times10^{11}\cdot s + 2.293\times10^{16}} \qquad 3.3$$

The frequency response of the estimated output impedance parameter is shown in Figure 3.6 below.

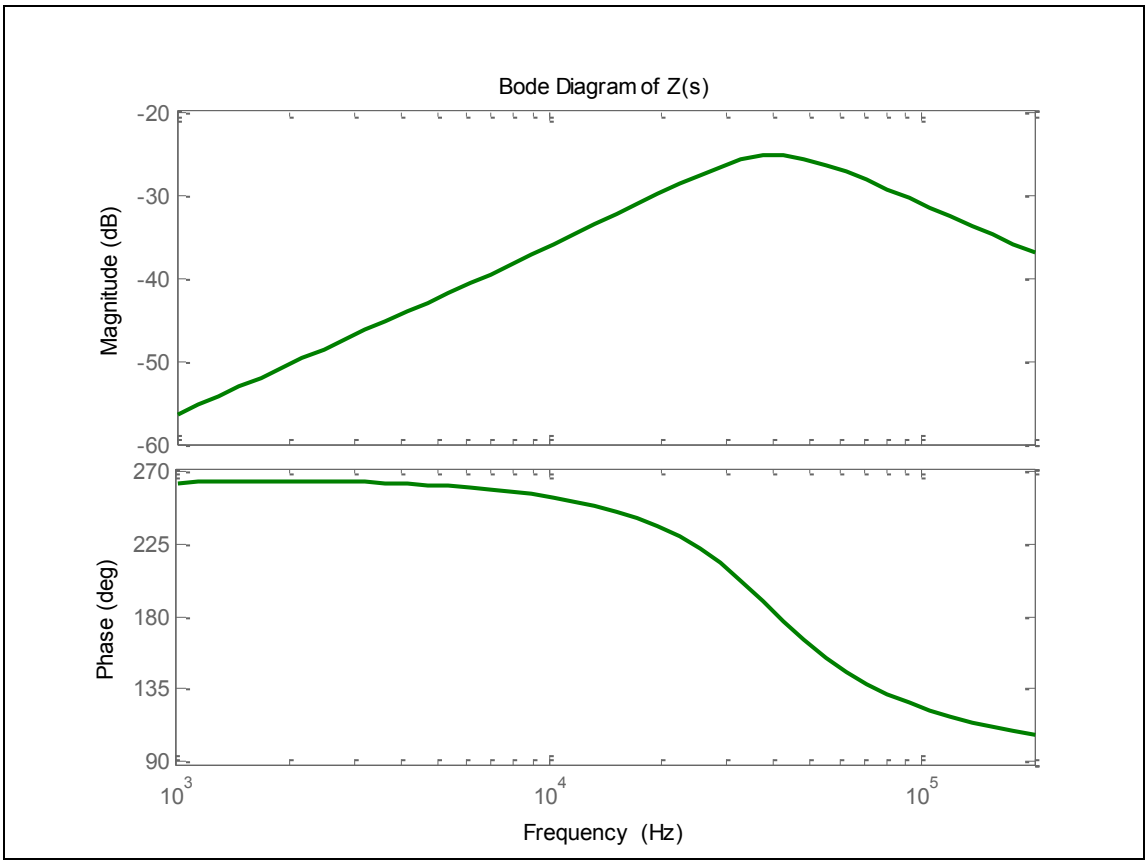

Figure 3.6 Output Impedance Bode Diagram

A comparison between the experimental transient response data and the transient response of the estimated output impedance parameter is shown in Figure 3.7. The response is typical of a voltage regulator in that the response to a load change is approximately that of a slightly under damped second order system. The comparison shows that the estimated transfer function for the output impedance parameter very closely matches the experimental data collected in the laboratory.

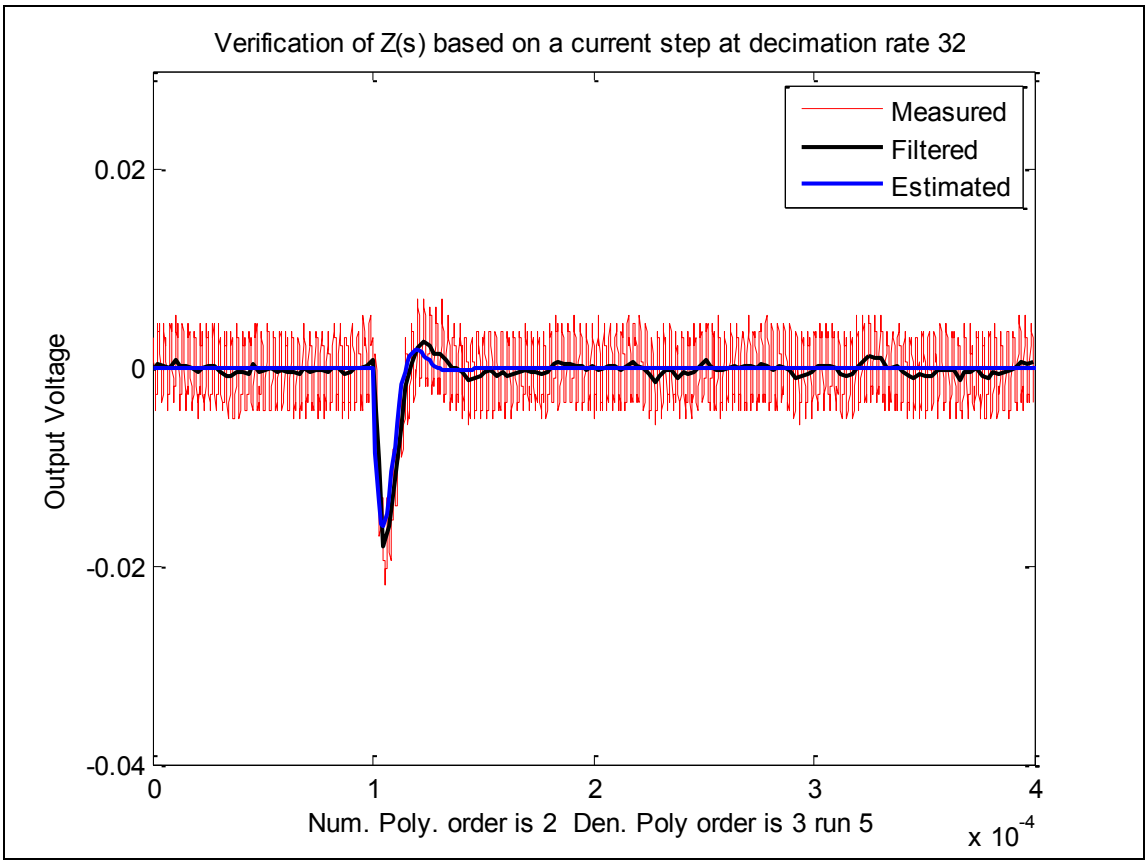

Figure 3.7 Validation of Output Impedance Parameter

The last parameter to be analyzed is the reverse current gain parameter $H_i(s)$. It too, like the output impedance parameter, is generated from transient response data collected in the output current excitation test. The following rational expression in 3.4 is the estimated transfer function model generated by the system identification algorithm.

$$H_i(s) = \frac{1.413\times10^{5}\cdot s^{2}\ 1.715\times10^{11}\cdot s+2.105\times10^{16}}{s^{3}+9.956\times10^{5}\cdot s^{2}+3.396\times10^{11}\cdot s+4.105\times10^{16}} \qquad 3.4$$

The frequency response of the estimated reverse current gain parameter in 3.4 is shown in the Bode plot of Figure 3.8. The frequency response is reasonable because the magnitude plot shows that the system tends to have a constant low frequency gain. This

is typical of regulated converters because the input power is approximately equal to the output power and since the input voltage and output voltages are stiff, the input and output current must change to comply with the law of energy conservation. As a result, if the output current is increased there is a corresponding decrease in the input current. This explains why the magnitude in dB is a constant negative value at the zero frequency.

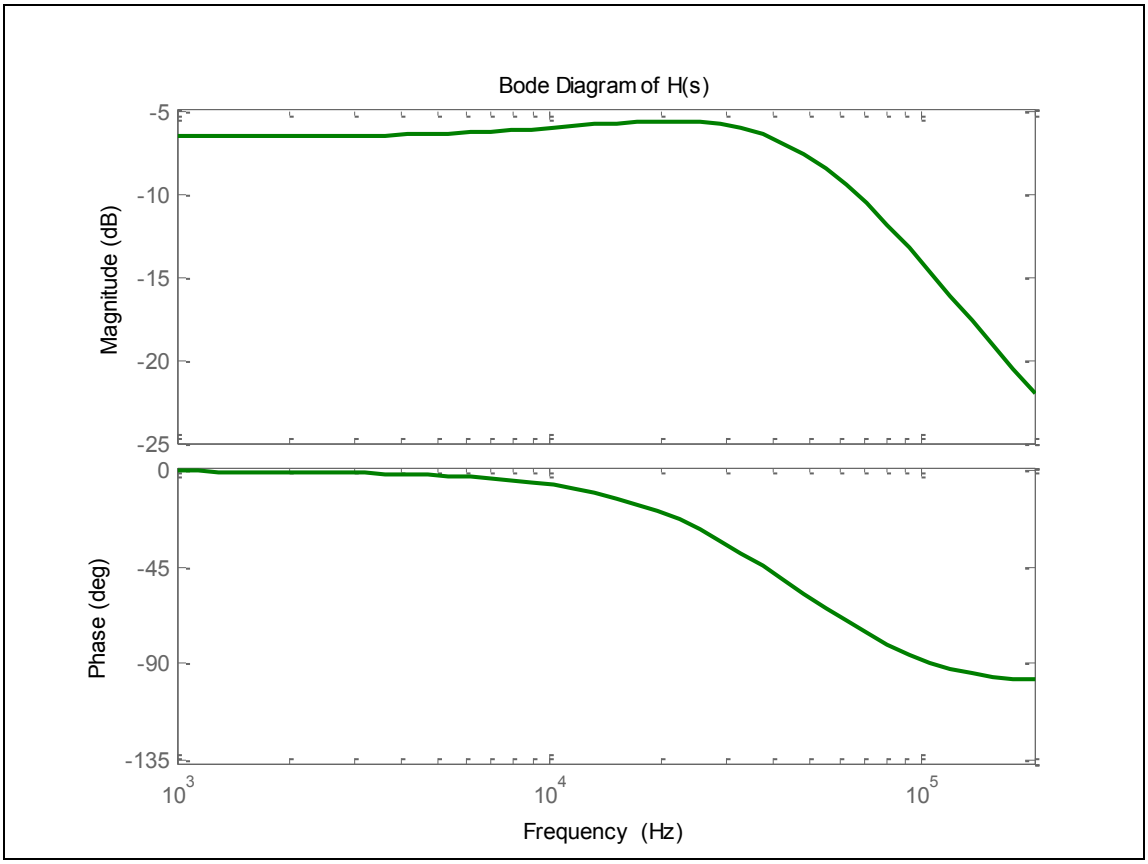

Figure 3.8 Reverse Current Gain Bode Diagram

The previous statements are further verified by the transient response comparisons shown in Figure 3.9 below. As with the other parameters, the estimated transfer function model of the reverse current gain parameter produced by the system identification algorithm fits the experimental transient response data almost perfectly.

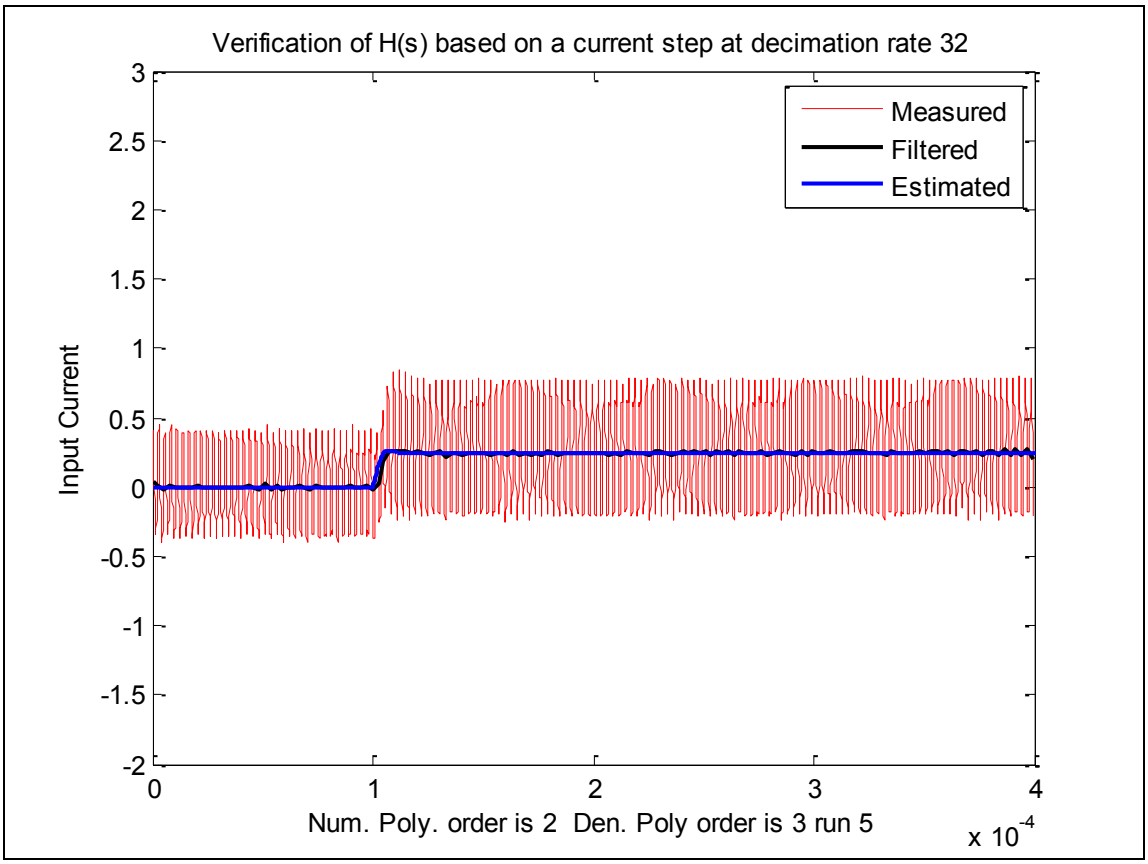

Figure 3.9 Validation of Reverse Current Gain

CHAPTER IV

LARGE SIGNAL ANALYSIS

4.1 Introduction

In order to construct a representative model of the PwrSoC module over its complete operating space the two experimental tests discussed earlier in Section 2.4 are repeated over a range of output currents, input voltages, and output voltages. In this way, an understanding can be developed of how the operating point influences both steady state and dynamic behaviors. More specifically, the purpose is to discover which of the two-port network parameters can be modeled as LTI systems and which ones may require a non-linear treatment. In the following sections a detailed analysis of the PwrSoC modules large signal characteristics and an explanation of the procedure used to obtain the large signal model is given.

4.2 Output Current Variation

To obtain the systems dependence on operating points defined by the load current, a set of input and output excitation tests were executed in the laboratory at the same 10 V input and 5 V output as before with the small signal characterization but instead at a discrete set of output currents. The tested output currents include 0.667 A, 1 A, 1.33 A, 2 A, 3 A, and 5 A. Each transient response corresponding to each of the six load current settings was then processed through the previously discussed system identification algorithm. The frequency response of each estimated network parameter, six total sets

with four parameters in each set, was plotted on a three-axis surface plot for analysis. The data analysis is discussed in the following sections.

The first of the estimated parameters to be examined, in a large signal sense, is the input admittance parameter $Y_i(s)$. As one can visualize in Figure 4.1 below, the value of the output current has little effect on the frequency characteristics of the input admittance parameter, especially, at the higher frequencies. There is, however, a trend in the lower end of the frequency spectrum indicating that the input admittance parameter is dependent on the load current but only at frequencies near dc. This static dependency is accounted for by simply subtracting the dc offset generated by the low frequency gain allowing the input admittance parameter to be modeled with a single LTI in regard to the output current.

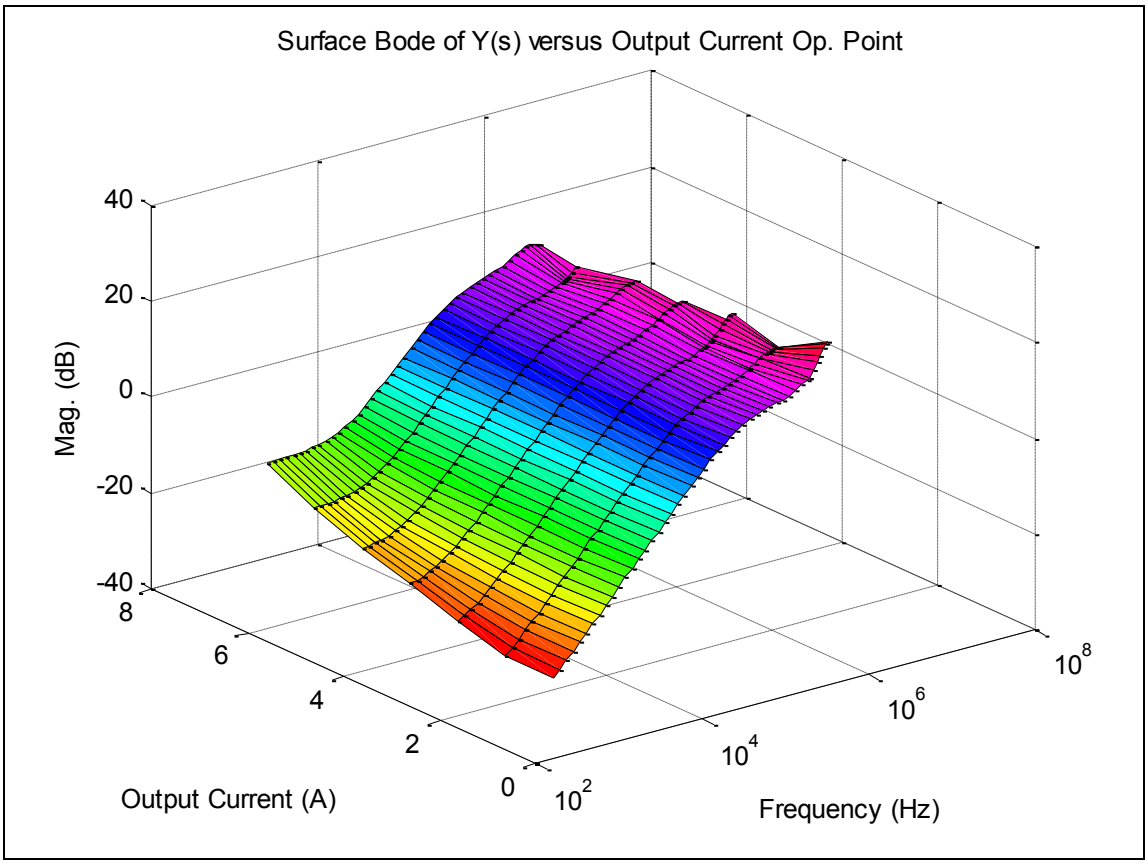

Figure 4.1 Variation in Y(s) with Output Current

The analysis of the forward voltage gain parameter $G_o(s)$ is not as straight forward as would be expected. The problem as can be seen in Figure 4.2 is that there is not an immediately identifiable trend in the frequency response with respect to output current levels. If examined a bit closer, considering the logarithmic scale of the magnitude axis, it is soon realized that even though the frequency response appears to be heavily dependent on the output current the deviation in amplitude of the magnitude response is small. This "hunting" type of behavior is due to the highly attenuated transient responses data, typical of a closed loop system, used to estimate the forward voltage gain parameter. It is later shown that this is not much of an issue and it is

concluded that the output current does not greatly influence the dynamics associated with the forward voltage gain parameter.

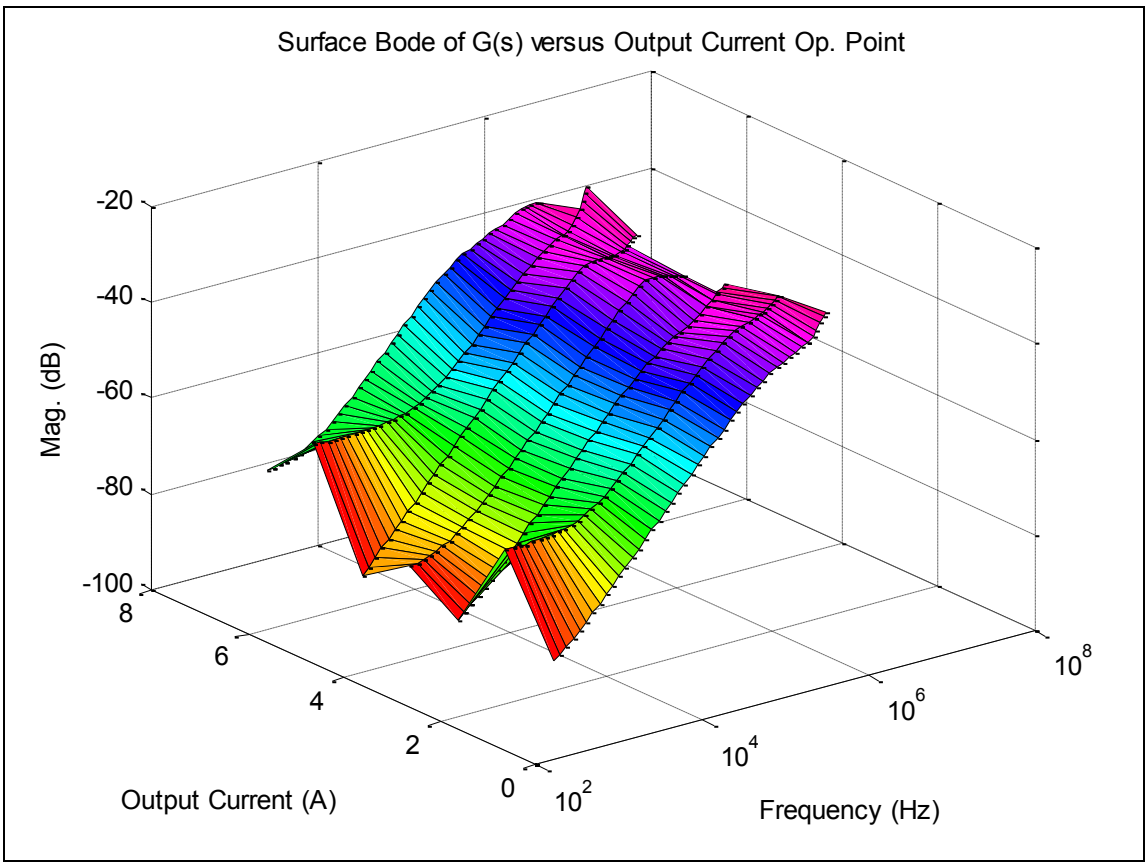

Figure 4.2 Variation in G(s) with Output Current

The next model parameter to be examined as a function of the load current is the output impedance parameter. The estimated output impedance parameter's frequency response versus the output current is shown in the surface plot of Figure 4.3. Examining the surface plot shows that there is a negligible change in the amplitude across the load current operating space. This indicates that the dynamics of the output impedance is independent of the output current and is therefore considered linear with respect to the load current.

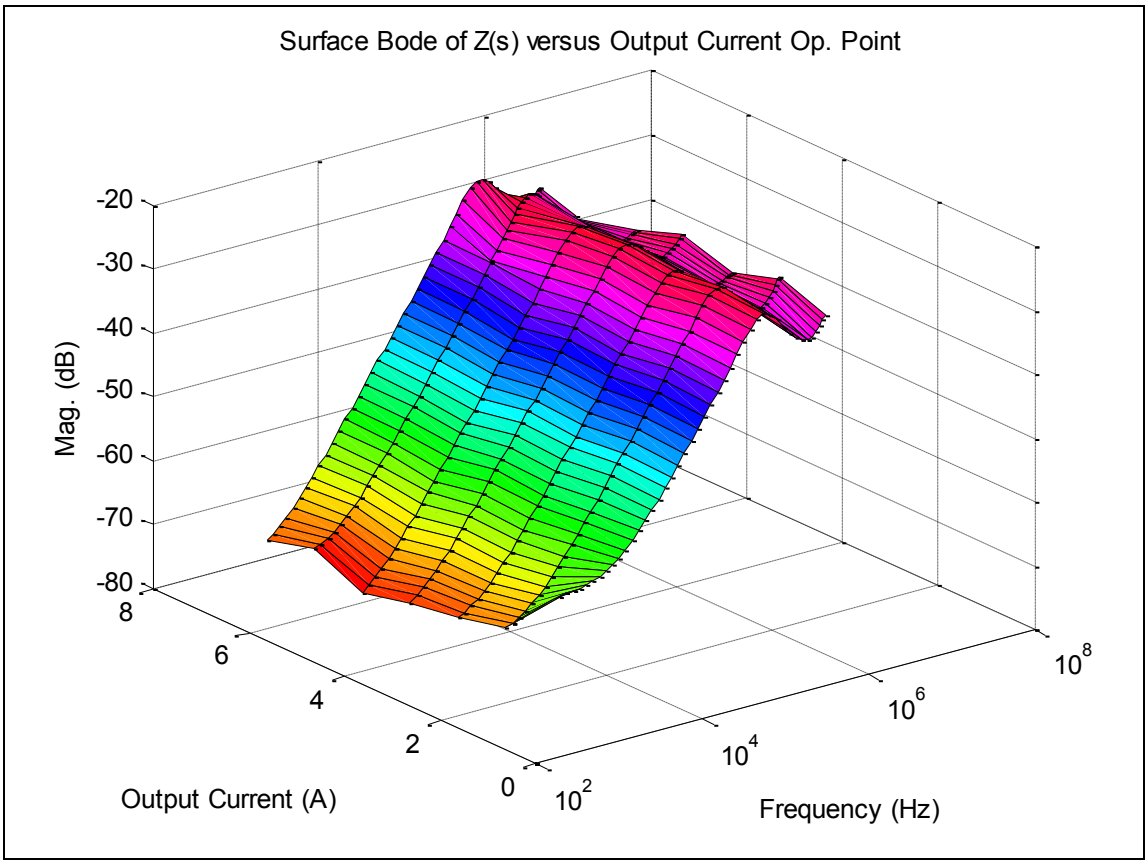

Figure 4.3 Variation in Z(S) with Output Current

The final of the two-port network parameters to be examined with regard to the load current is the reverse current gain parameter. This parameter's frequency response is plotted in Figure 4.4 versus load current. In examining the response, it is clear that each of the response curves is approximately the same indicating that the dynamic characteristics of the parameter are independent of the output current.

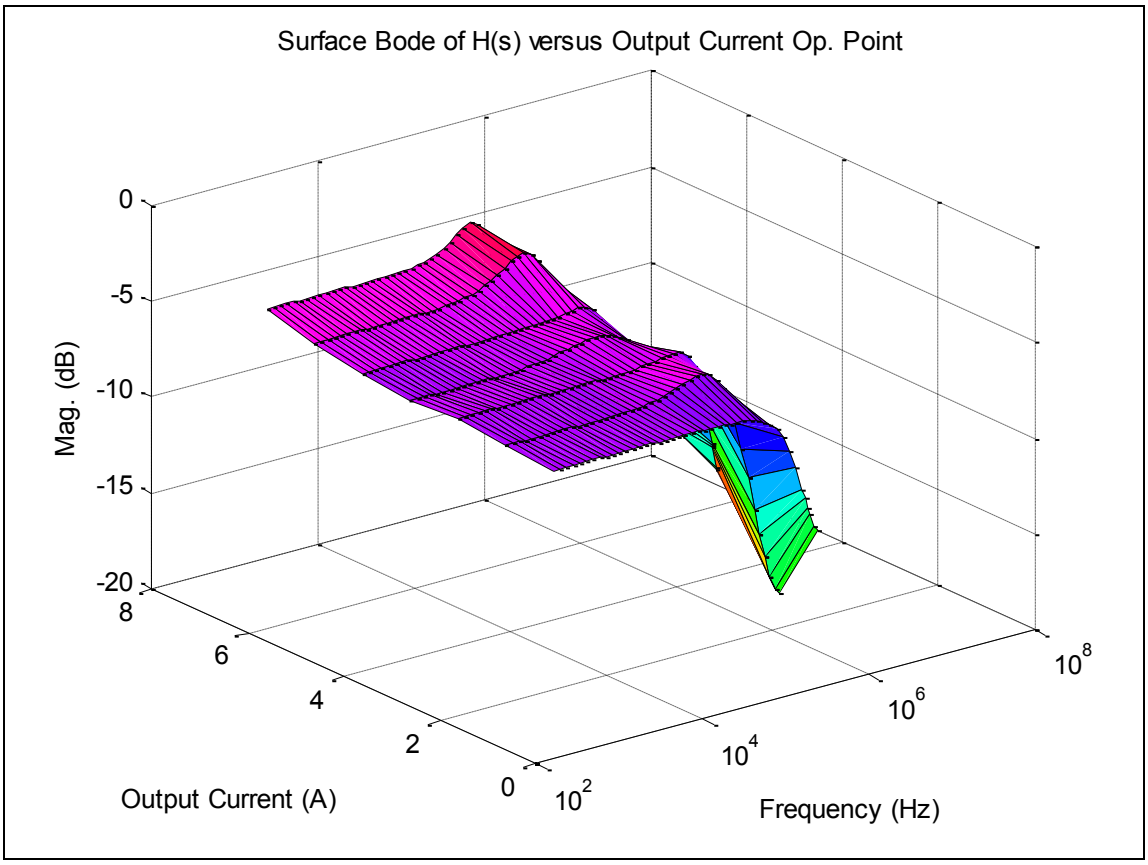

Figure 4.4 Reverse Current Gain Variation with Output Current

The estimated frequency responses previously examined illustrate that the natural dynamics of the PwrSoC are not heavily dependent in any regard to changes in the output load current as long as the system is operating under normal conditions and in CCM. This same procedure and analysis was also conducted on the LTM4612 PwrSoC module at input voltage settings of 8 V and 12 V. The results indicated the same conclusion at both 8 V and 12 V, i.e. the output load current does not have a significant effect on the systems dynamic modes. The system, however, depends significantly on the input voltage, which is discussed in Section 4.3.

4.3 Input Voltage Variation

Once the influences of load current on the systems frequency characteristics was understood, changes in operating point with respect to the input voltage could be considered. Consideration of a variable input voltage is particularly important for applications where the converter is sourcing from an unregulated voltage bus, which is often the case. Similar to the load current dependency analysis, an input and output excitation test was executed at a discrete set of input voltages. The tested input voltages include 8 V, 10 V, 12 V and 30 V. Tests were also conducted, as previously mentioned, at different load currents including 0.667 A, 1 A, 1.33 A, 2 A, 3 A, and 5 A for each input voltage setting. By adding in a second dimension to the operating point data set, a sort of "sanity check" assessment could be made.

In analyzing the resulting transient responses as well as the frequency response of the estimated two-port network parameters, a few conclusions about the influences of the input voltage on the system could be drawn. In terms of input admittance, it was found that a change in input voltage has no significant influence on the parameter resulting in the input admittance parameter being modeled with a single LTI transfer function. A similar result was found of the forward voltage gain parameter in that its dependence on the input voltage is negligible. Also, as expected, the output impedance parameter was found to be independent of the any change made in the input voltage. It was found, however, that the static value of the input current is heavily influenced by changes made in the input voltage. The static dependence was found to be due to the constant power characteristics of the PwrSoC module and this type of behavior is commonly reported in the literature, and indeed it is an expected property of any voltage regulator. The issue was addressed in this work by adding in a multiplier block in series with the reverse

current gain parameter. A model that addresses static non-linearity in this way is often referred to as a Weiner model and its use in modeling non-linearity is suggested in [11]. The gain used in the multiplier block is listed below and is scaled to the original operating point of the small signal model.

$$P_{in} = P_{out} \tag{4.1}$$

$$V_{in} \cdot I_{in} \cdot \eta = V_{out} \cdot I_{out} \tag{4.2}$$

$$I_{in} = \frac{V_{out} \cdot I_{out}}{V_{in} \cdot \eta} \tag{4.3}$$

$$I_{in} = \frac{10 \cdot V_{out} \cdot I_{out} \cdot 0.9}{V_{in} \cdot 5 \cdot \eta} \tag{4.4}$$

4.4 Output Voltage Variation

Another variable that must be considered in the large signal model is the output voltage. The consideration of the output voltage as a system operating point is largely overlooked in related literature but it is mandatory that its influence is understood if a true large-signal model is desired. Again, similarly to the previous two operating point variables, excitation tests were conducted at different output voltages in order to assess how the PwrSoC modules frequency characteristics change with operating point as defined by the output voltage.

In analyzing the experimental data, it was found that a change in output voltage does not command a change in the frequency response characteristics of the input admittance parameter. Therefore, a single LTI model was used to model the input admittance parameter. A similar conclusion was drawn for the model of the forward voltage gain parameter in which the output voltage set point was found to not significantly influence its characterization. Hence, a single LTI was used to model the

two-port network parameter. It was found, however, that the output voltage does affect both the reverse current gain parameter and the output impedance parameter. The influence of the output voltage on the reverse current gain parameter is controlled by the gain block previously described in Section 4.3. As for the output impedance parameter, a Weiner model, similar to that used for the reverse current gain parameter, was used to scale the model with respect to the output voltage that the small signal model was originally based. A simple scale factor was selected because the experimental data shows that the output impedance response tends to exhibit increased overshoot as the output voltage is increased. The more elaborate and complex Polytopic Model solution as defined by [14] could be applied to this type of situation but is avoided for the sake of simplicity and computational efficiency. The resulting scale factor was found to be approximately proportional to the output voltage and is shown in 4.5 below.

$$K = \frac{-V_{out}}{5} \tag{4.5}$$

In conclusion, a satisfactory large signal model is obtained for the PwrSoC module, with good accuracy, using relatively simple methods of dealing with non-linearity arising from changes in operating point. The results show that the more elaborate modeling techniques can be avoided with the tradeoff being in the accuracy of the model.

4.5 Event Driven Modes of Operation

In addition to modeling the dynamic response of the PwrSoC module under normal operating conditions, event driven behaviors such as current limiting and other protection schemes were considered. These behaviors are found in many of today's power converters including the PwrSoC line of converters considered in this work. The

methods of implementation vary between manufactures but as a rule, event driven behaviors are encountered under abnormal operating conditions where they provide a self-induced mechanism for preventing catastrophic failure of the device. The mechanisms are generally triggered by thermal, voltage, and/or current thresholds being exceeded which in turn causes the device to react in a predetermined manner. Now, depending on the nature of the event, the system may simply respond by going into a shutdown lockout mode until the problem is resolved by some external method or may respond by limiting the problematic value with some type of automatic feedback control. Usually, the first type of response is due to an overvoltage or thermal trigger while the latter is typically due to a current overload at the output. The specific device analyzed in this work incorporates a current mode controller to remedy problems associated with over current and fault conditions. This large signal characteristic can be a useful operational mode for some applications where current control is required and is included in the final large signal model. The details of this mode of operation are discussed further in the next few paragraphs.

Some advantage can be taken of the PwrSoC modules current mode controller, which limits the inductor current and hence the output current on a cycle-by-cycle basis. The controller further limits the inductor current in the case that the output voltage is reduced by more than approximately 50% indicating that a fault condition has occurred. Under this condition, the output current is progressively reduced to approximately one sixth of the full current limit value [12]. This current control behavior is a significant feature of the PwrSoC module in that it opens a wide field of practical applications, especially, in regards to battery charging and power management applications. Since battery charging applications, at least in some specific man-portable systems applications,

do not require a tightly regulated charge rate but instead are able to charge at a quasi-constant current level as long as the level does not exceed a maximum specified value. The maximum specified value, which is identifiably the current control limit, is set by providing a voltage to an external pin available on the PwrSoC device.

The experimental test used to capture the current limiting character of the device is straightforward. The test was conducted using a fine adjustment dc power supply and an electronic load. The dc power supply was connected to the external current limit pin on the PwrSoC module for adjustment of the current limit setting while the electronic load was connected to the output terminals of the PwrSoC device. By using the electronic load as a variable resistor, the load was gradually increased while the output voltage and output current was recorded. The test was then repeated at a discrete set of voltages applied to the current limit pin on the PwrSoC module.

In an effort to model the current limiting characteristics of the device, a mapping between the voltage provided to the external pin and the corresponding current limit setting or curve was generated in the laboratory. The family of curves corresponding to each set point is shown in Figure 4.5. In analyzing the mapping, it is clear that the current limiting characteristic of the current mode controller in this device is of the fold-back current limiting form as described in the devices datasheet. This type of current limiting scheme is often used in modern dc-dc power supply designs. The basic concept is that once the current limit threshold is reached the current mode controller begins to reduce the output voltage, thereby, limiting the output current. Once the critical voltage indicating a fault condition is reached, the current is gradually reduced.

The trend is clear in Figure 3.24 where it is obvious that once the current limit is exceeded the controller drops the voltage at a constant rate until approximately a third of

the output voltage is reached. At this point, the current mode controller begins to ease back the output current at a rate that is unclear in the experimental data. This does not prove to be much of a problem because the objective of this model is to capture the current limiting behavior that can be applied to battery charging applications for man-portable power systems. As a result, in this work the fault protection behavior is not included in the model.

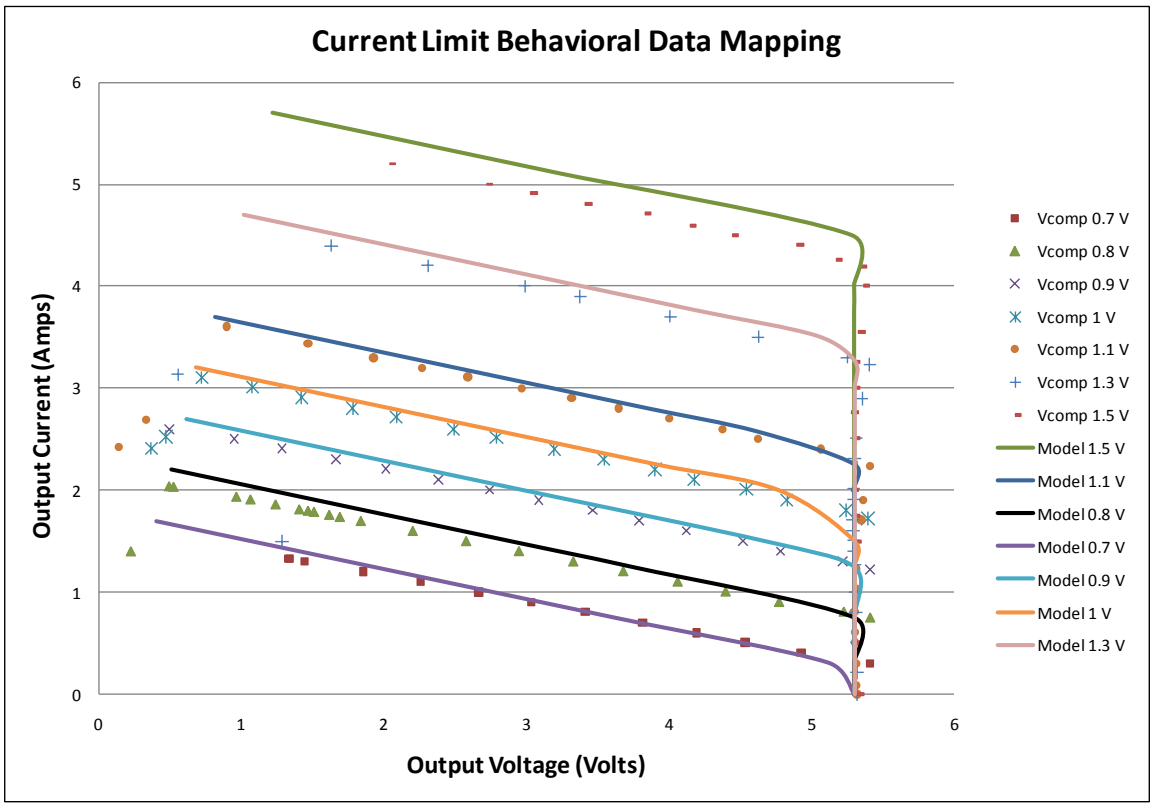

Figure 4.5 Current Limit Mapping

To model the current limiting behavior, a linear curve fit procedure was applied to each set of experimental data as shown in Figure 4.5. Since the rate at which the voltage decreases is relatively constant across all of the current limit settings, a linear trend line

was fit to the experimental data. The resulting trend line in 4.6 was then used to model the output behaviors of the system once a set current limit is exceeded. The following equation, where z is the external voltage applied to the current limit pin and x is the output current, was used in the model to mimic the current limiting behavior of the PwrSoC module.

$$V_{out} = -3.3877x + 17.951z - 6.3978 \quad (4.6)$$

CHAPTER V

SIMULINK MODEL

5.1 Introduction

Simulink is a multi-domain tool that provides a highly customizable simulation environment for the modeling of dynamical systems. It is extensively used in systems engineering as well as a multitude of other engineering disciplines and is an ideal fit for the system based dynamic modeling done in this work. With that said, the complete PwrSoC black-box model was implemented in the Simulink simulation environment. The model was constructed in a series of steps with increasing levels of model complexity. At each step, the model was verified to ensure that proper operation was being achieved. The complete model including both the large and small signal features as well as the current control behaviors were built and tested in the environment one at a time. A detailed description of how the model was constructed is given in this chapter.

5.2 Small Signal Model

The first step taken towards a complete model was to implement the small signal two-port network model in Simulink. The model was constructed in accordance with the general black-box model shown in Figure 2.2. The system was then simulated at the same operating point at which the parameters were estimated: 10 V input voltage, 5 V output voltage, and 2 A output current. This simulation was conducted in order to verify the basic operation of the model. The small signal Simulink model is shown in Figure 5.1 below.

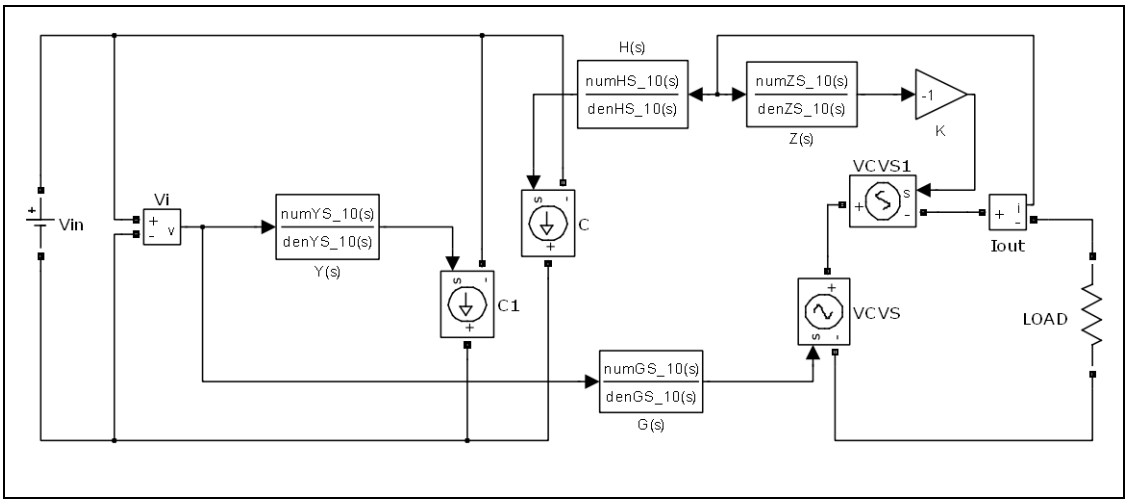

Figure 5.1 Small Signal Model

5.3 Large Signal Model

The next advancement toward a complete model was to add in large signal behaviors to the small signal model. A few of the key additions include: an ideal dc voltage source in the output network, removal of the input admittance dc gain, a gain block for the reverse current gain parameter, and a gain block for the output impedance parameter. Since the output voltage is tightly regulated, at least in voltage regulation mode, a simple dc source is used to set the output voltage to the desired value. This is a common solution as documented in Ref. [7]-[9] and [15]-[16]. The previously discussed Weiner model gains in CHAPTER IV that are required to address the response to variation in input/output voltages are also included in the large signal model where they are placed at the output nodes of the output impedance and reverse current gain LTI blocks. The large signal model is shown in Figure 5.2

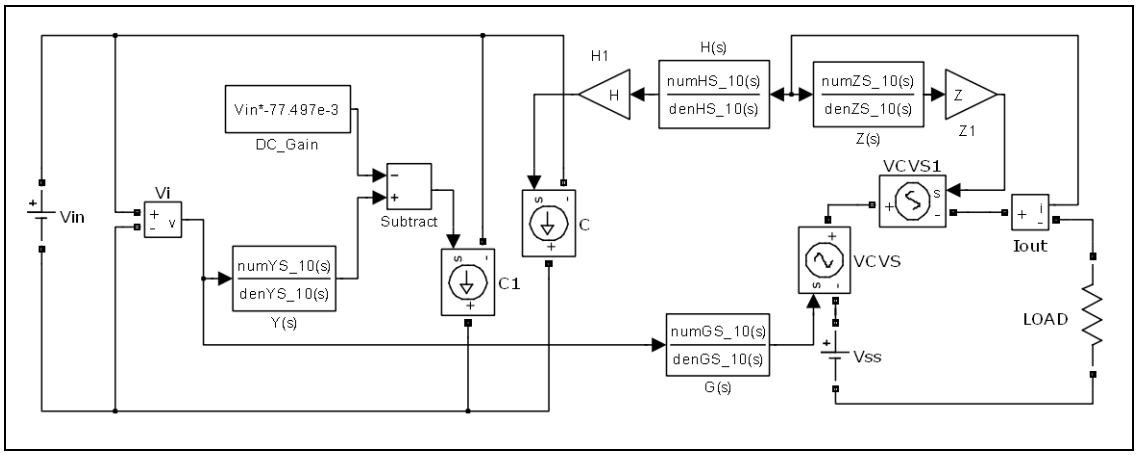

Figure 5.2 Large Signal Model

After the large signal model was constructed, it was then verified that the system operated properly by comparing the response of the system to the data used to estimate the parameters of the large signal model.

5.4 Final Model

In the final model, the current limit behavior of the system is incorporated into the large signal model by replacing the ideal dc source in the output network with a voltage controlled voltage source. In addition, the output current is measured and compared to the current limit set point generated by the reference current control voltage. This allows the output voltage to be modified in accordance with the equations developed in CHAPTER IV and therefore modeling the current control behavior seen in the LTM 4612 PwrSoC module. The final model is shown in Figure 5.3 below.

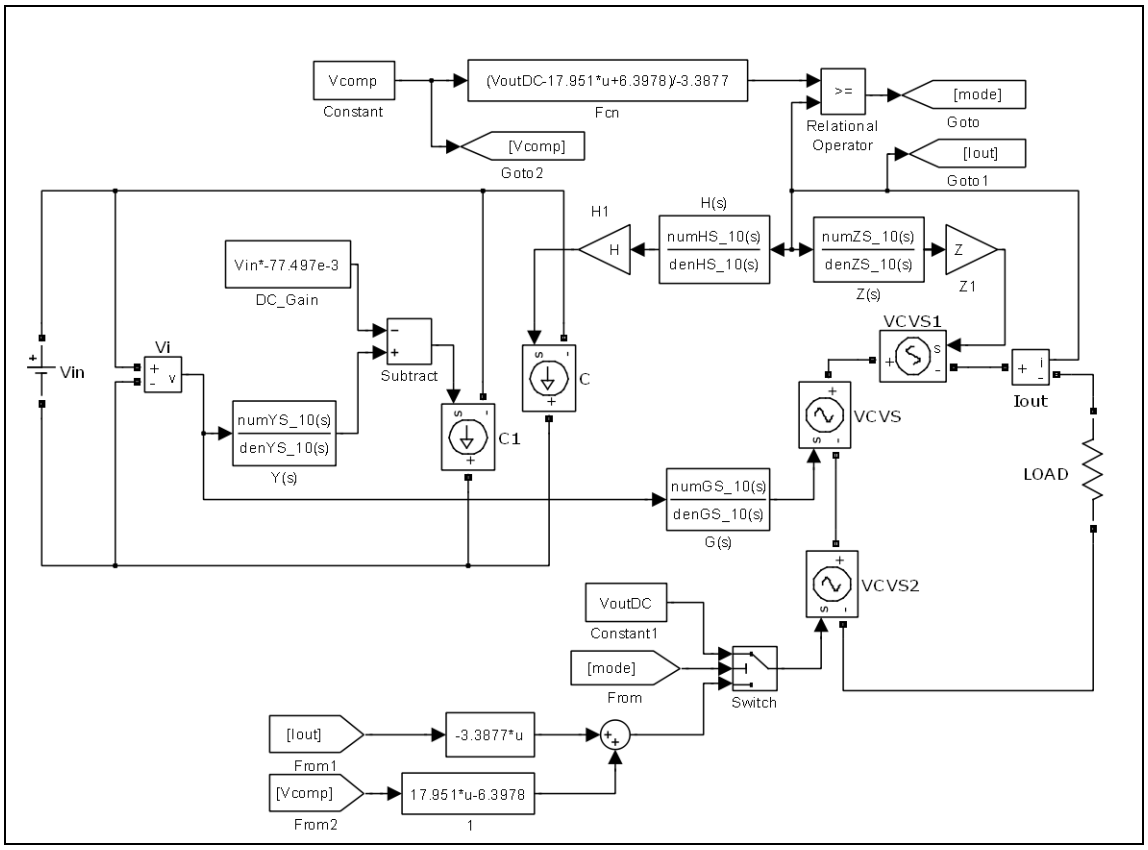

Figure 5.3 Final Model

5.5 Model Wrapper and Mask

Once the final model was constructed, there was a need for it to be packaged in a user-friendly modular wrapper. The selected solution was to package the model in a functional block and place it within the general Simulink Library Browser. This allows the user to drag and drop the model into larger system models that are also constructed in the Simulink simulation environment.

The developed Simulink library in the Simulink Library Browser is illustrated in Figure 5.4 below.

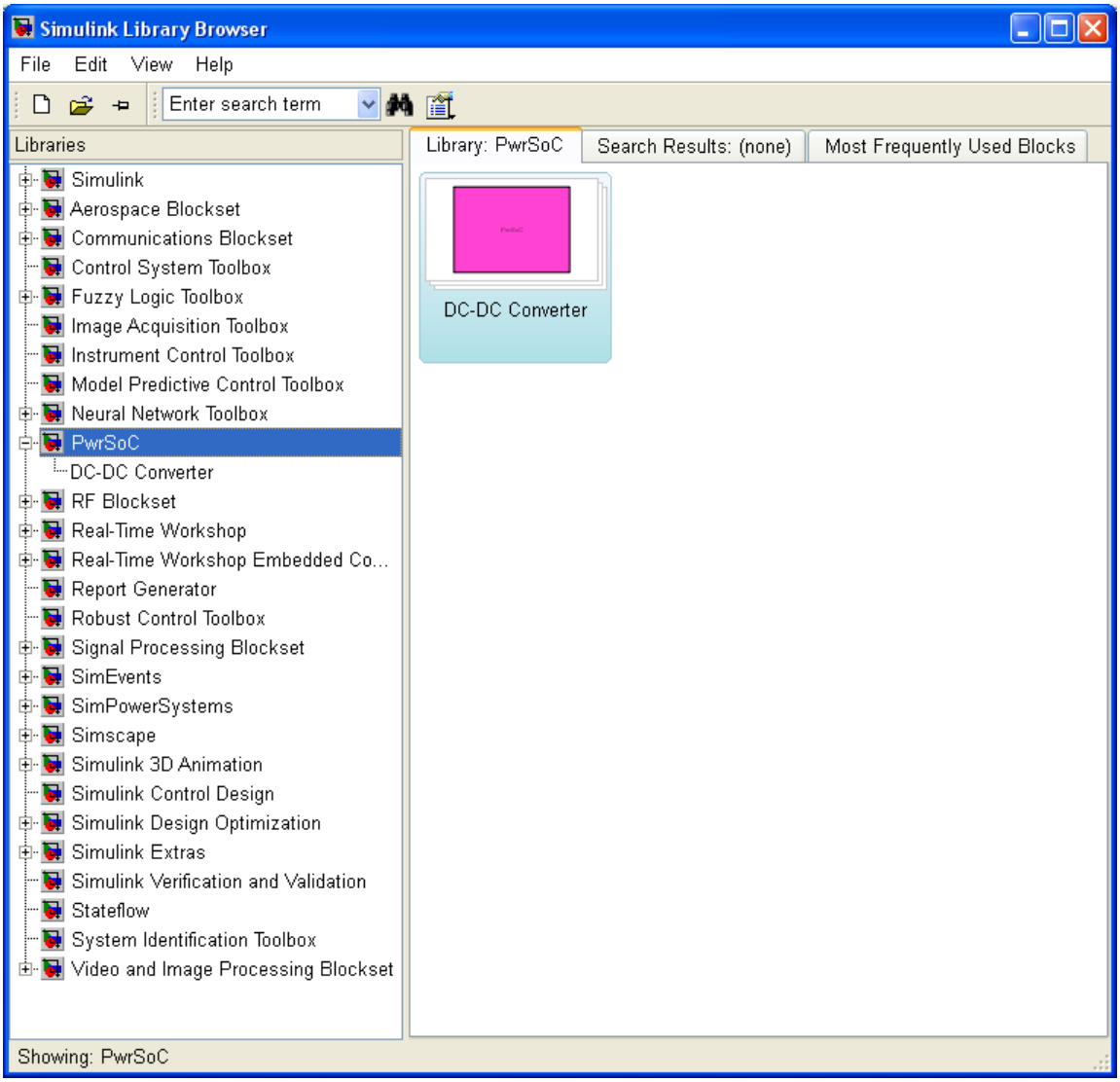

Figure 5.4 PwrSoC Library

The functional block is located within the PwrSoC library and is accessed through the Simulink Library Browser interface shown previously in Figure 5.4. The PwrSoC functional block that can be placed within a Simulink simulation window is shown in Figure 5.5 below.

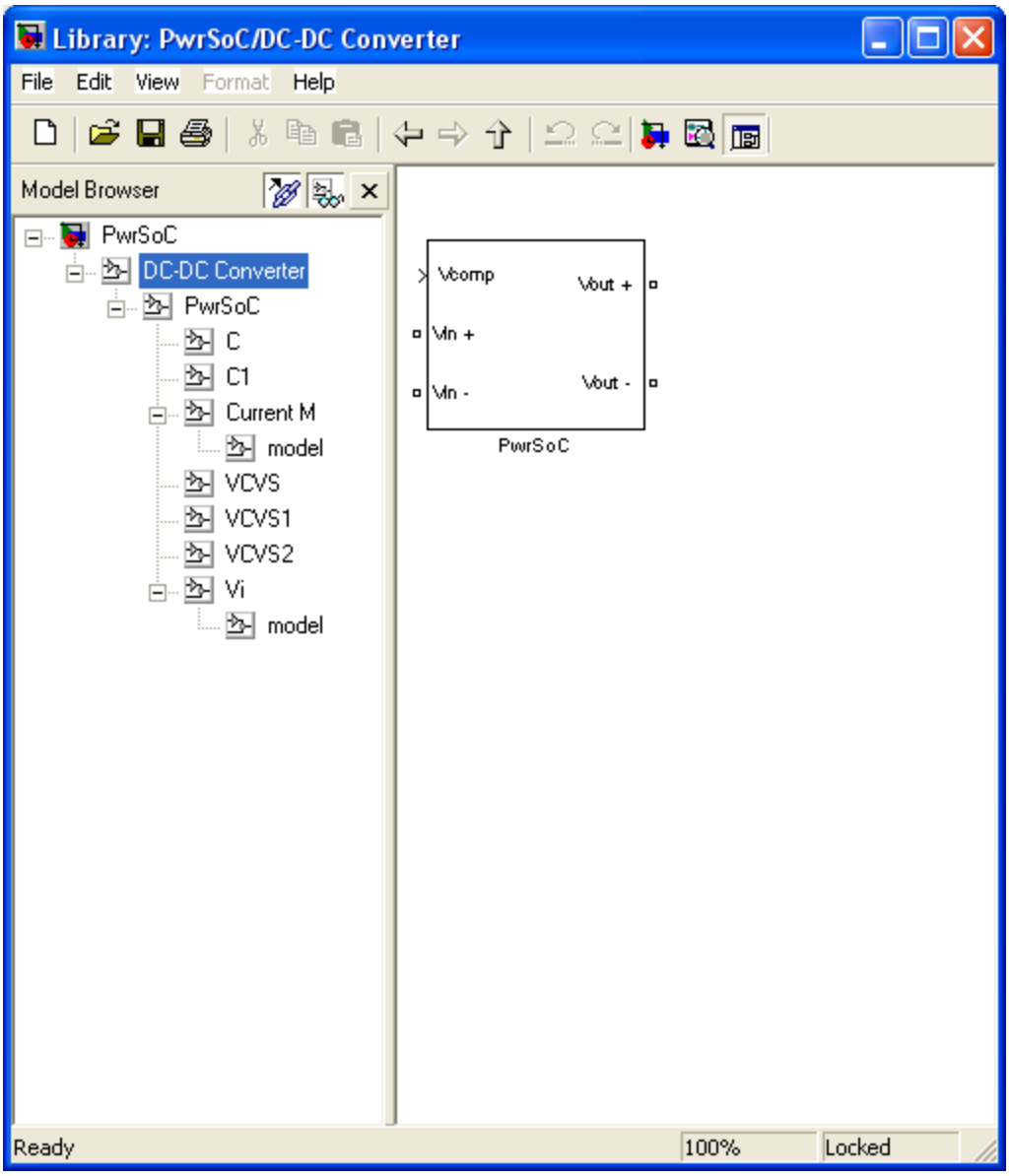

Figure 5.5 PwrSoC Function Block

The final model is hidden behind the functional block mask, which is activated by double clicking on function block. The mask serves two major functions. The first function of the mask is to provide a user interface through which parameters settings such as power conversion efficiency and output voltage settings can be manipulated.

Secondly, the mask provides a fail-safe way of preventing the user from unknowingly making modifications to the original model. The mask is shown below in Figure 5.6

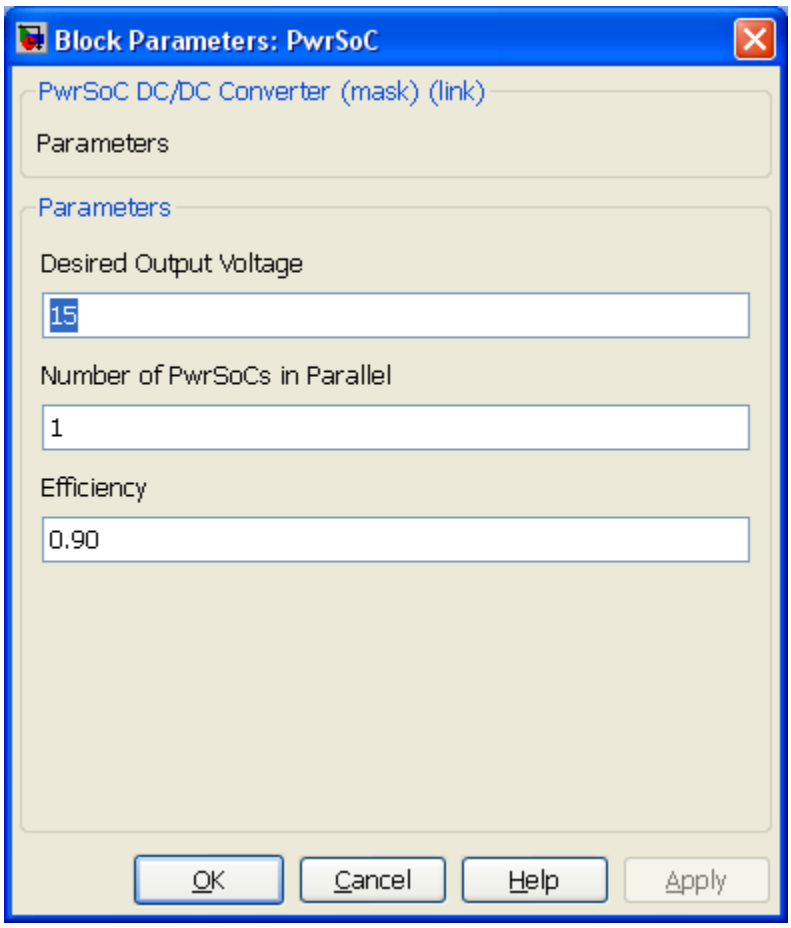

Figure 5.6 Model Mask

Under the model mask is the two-port Simulink model in the form of a subsystem as shown in Figure 5.7. A few modifications were required to make the model into a portable subsystem that could deal with computational issues associated with algebraic loops. Algebraic loops are common when modeling in the Simulink simulation environment. The problem occurs when the value of the input port directly controls the

output of a feed through block, such as a gain block [17]. The problem is easily dealt with by simply adding in a memory block to the node causing the issue.

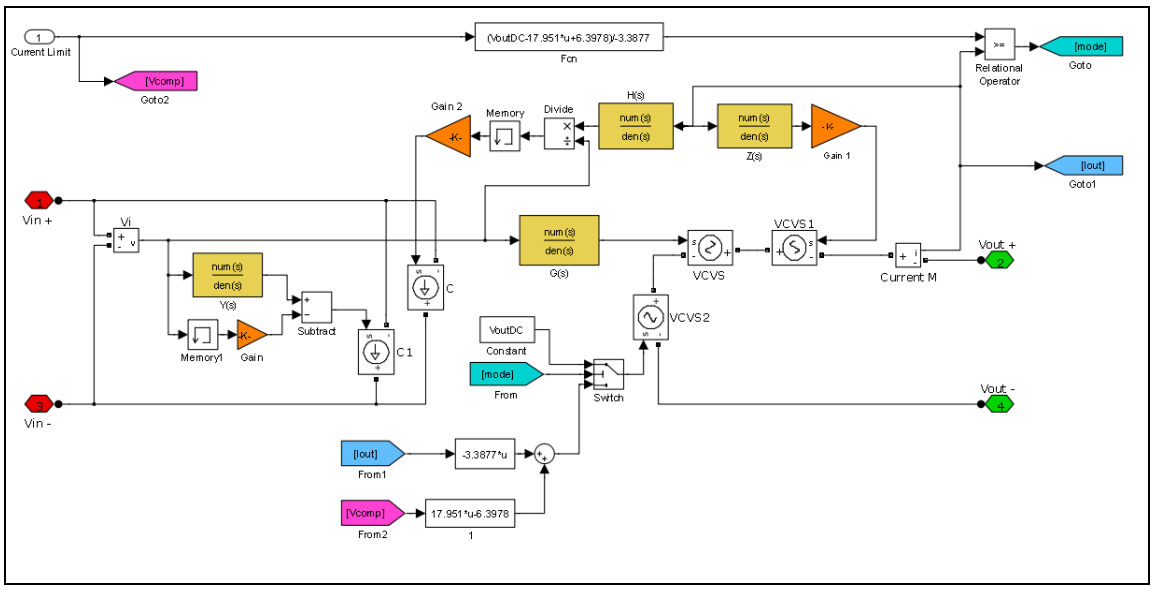

Figure 5.7 Model Under the Mask

In the following chapter, the complete model shown in Figure 5.7 will be validated against an independent set of experimental data.

CHAPTER VI

MODEL VALADATION

6.1 Introduction

Model validation is a vital part of the modeling process and requires that the model be tested and then compared to experimental transient response data that is independent of the operating point at which the two-port network model parameters were estimated. This allows an assessment of how well the model performs at replicating the physical characteristics and behaviors found in the PwrSoC module. In this chapter, the validation results of the final Simulink model including the large signal dynamic response and the current limiting behavior are presented.

6.2 Large Signal Validation

In order to check the validity of the final model, a model test bed was constructed using the PwrSoC functional block from the Simulink library developed in the previous chapter. The test bed was then used to replicate the validation tests, in software, that was conducted on LTM4612 hardware. The model test bed is shown in a general configuration in Figure 6.1below.

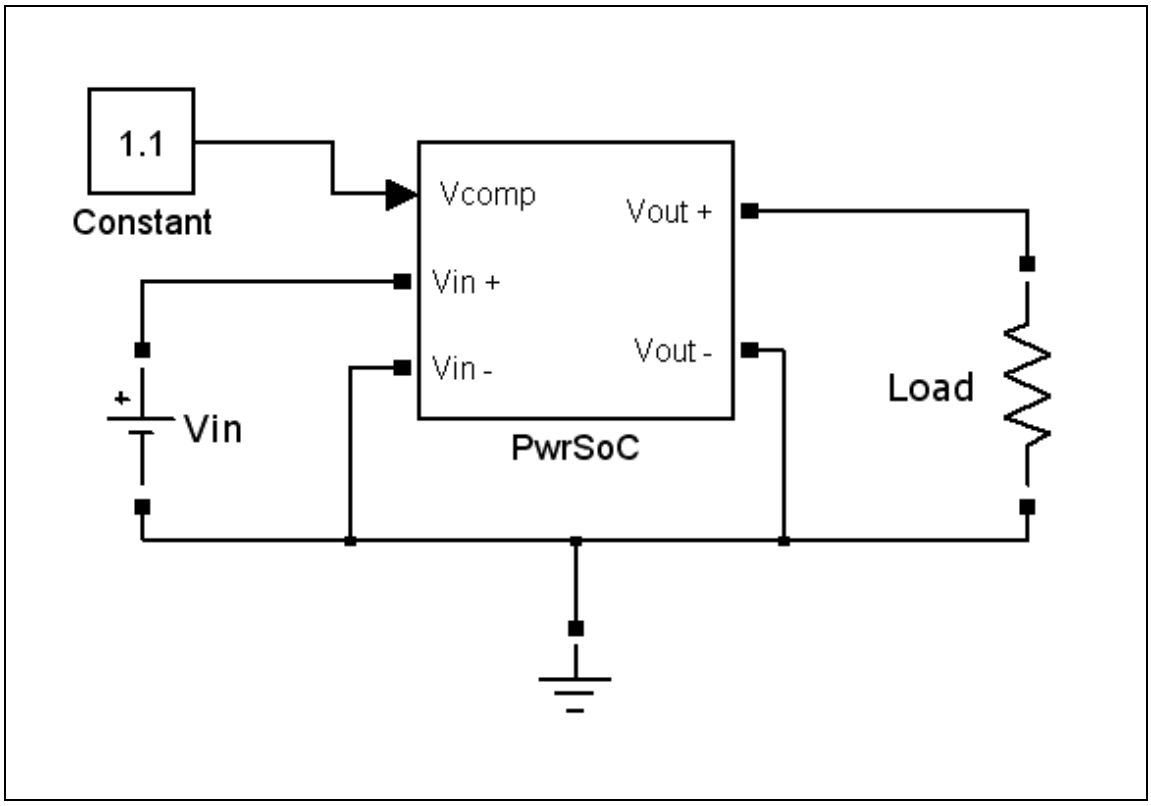

Figure 6.1 Model Test Bed

To accommodate the requirements of the validation process, the operating point of the PwrSoC module as well as the operating point of the Simulink model was changed to the validation configuration defined in Table 6.1. Recall that the two-port network parameters of the Simulink model are based on transient response data that was generated at the base configuration also shown in Table 6.1.

Table 6.1 Base and Validation Operating Configuration

Operating Parameter	Validation Configuration	Base Configuration
Input Voltage	30 V	10 V
Output Voltage	12 V	5 V
Output Current	1 A	2 A

When considering the operating window of the LTM4612 and the conditions that govern the non-linearity found in the module, the change in operating point from that of the validation configuration to that of the base configuration as shown in Table 6.1 is substantial. The output voltage is 2.4 times larger and the input voltage is 3 times larger than the base configuration while the output current was cut by 50%. The large change in operating point suggests that the model will perform well over a wide range of operating conditions

In acquiring the validation data, a load step excitation was applied to both the Simulink model and the LTM4612. The transient response of the LTM4612 and the transient response of the final Simulink model are compared on the same plots shown in Figure 6.2. The figure shows that the Simulink model predicts the response of the LTM4612 due to a load transient at an operating point that is independent of the operating point the model was identified at, confirming the successful integration of the small-signal model into the large-signal "wrapper."

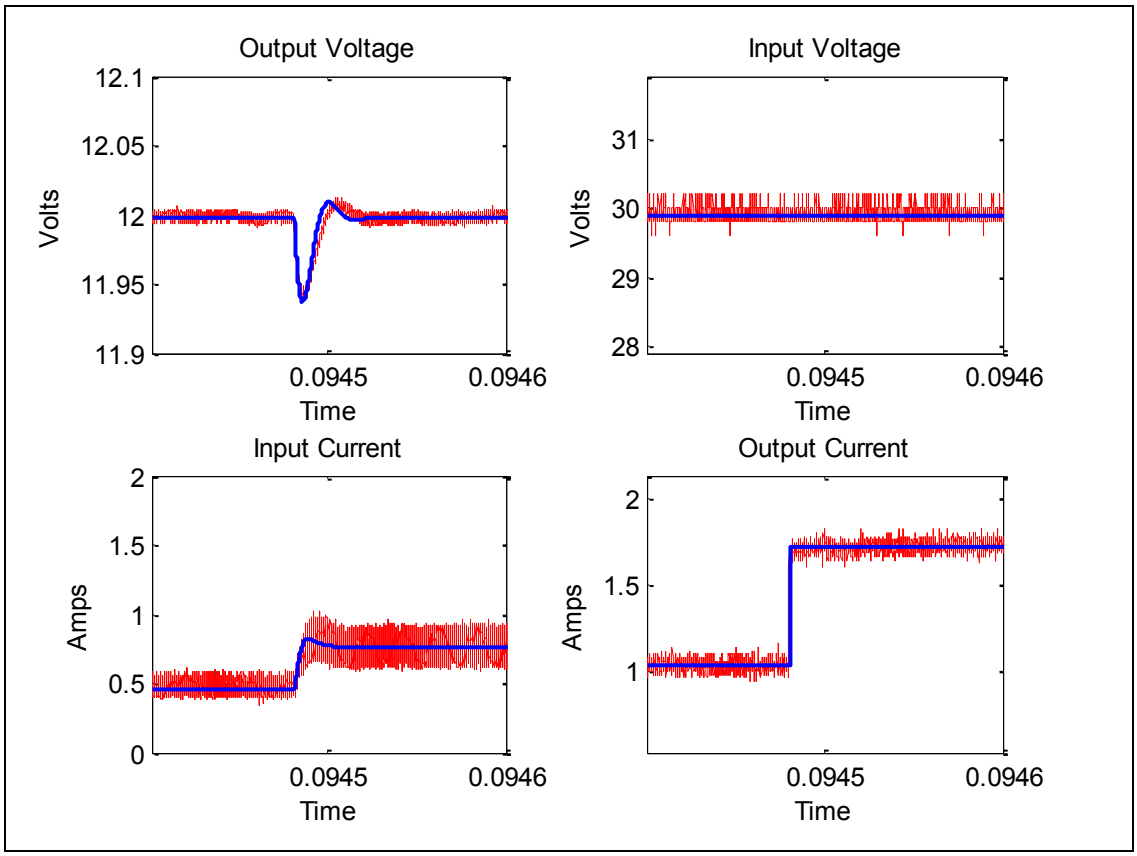

Figure 6.2 Validation Data

6.3 Current Limit Behavior

To validate the current limit behavior designed into the Simulink model against current limit behaviors found in the LTM4612, a similar approach to that of the previous section was taken. The test conducted on the LTM4612 hardware in laboratory was replicated in the Simulink simulation environment. The model test bed is shown in Figure 6.3.

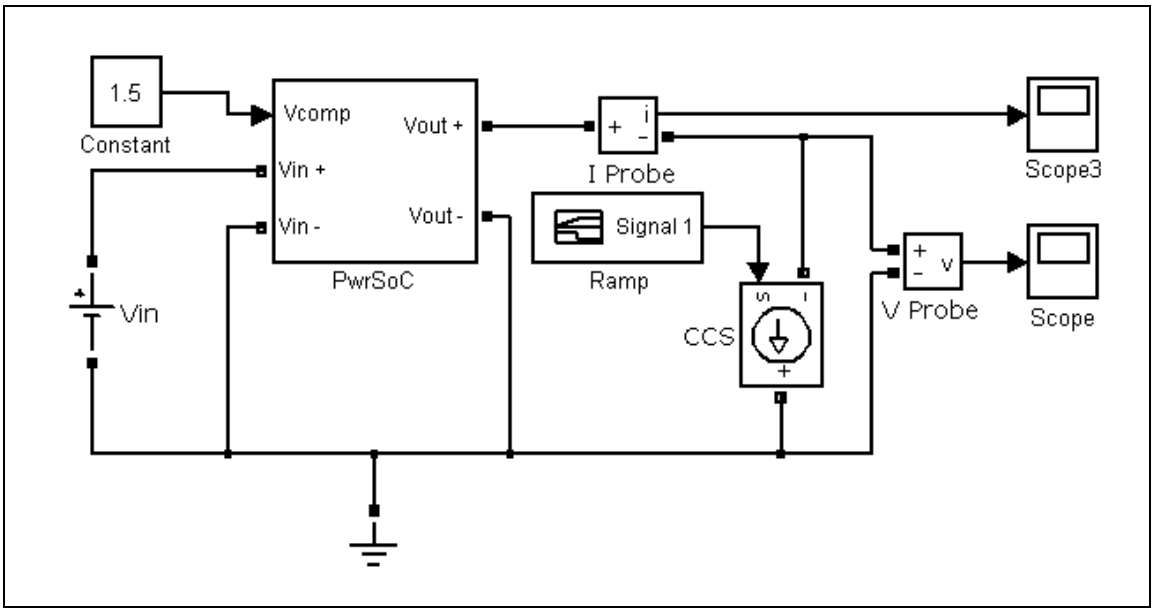

Figure 6.3 Current Limit Test Configuration

The controlled current source and arbitrary ramp signal are used to replicate the action of the load current being increased progressively from below to above the current limit setting as was done in laboratory test performed on the hardware. The same test was repeated at several different settings of the current reference control voltage and the data for each setting was recorded and imported into MATLAB. The resulting experimental and simulation data was compared by overlaying the data on the same plot. A comparison between the experimental data and the simulated data is shown in Figure 6.4 where the solid curves represent model data and the symbols represent experimental data. The figure shows that the model predicts the voltage regulation behavior up to the current limit value, which is set by the current reference control voltage, and then changes the mode of operation and predicts the current limit behavior up to approximately 25% of the regulated output voltage setting. Below 25% of the regulated output voltage, the current limit behavior is not well understood and is therefore not included in the model.

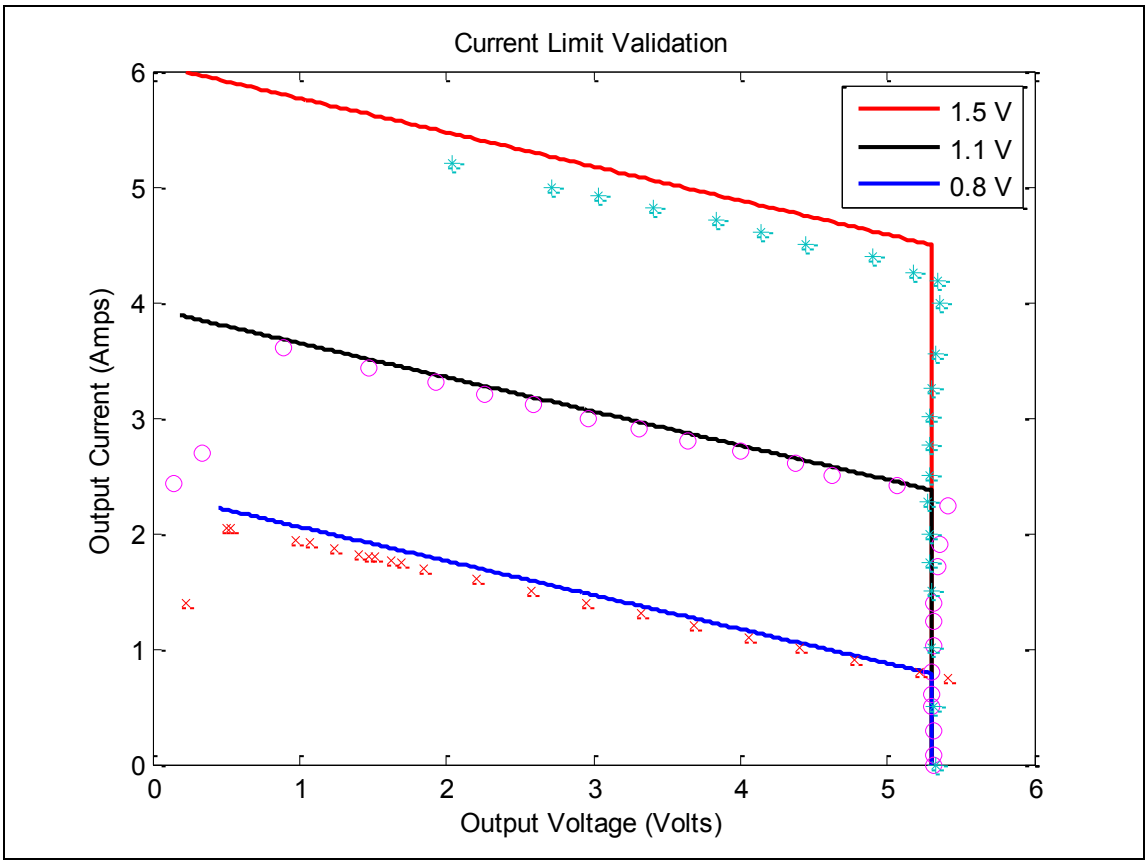

Figure 6.4 Current Limit Behavior Validation

CHAPTER VII

CONCLUSION

In this thesis, a complete averaged model of a highly integrated PwrSoC product has been developed using a two-port black-box model approach. The estimated two-port model parameters are based on transient responses generated by the application of step excitations to the input and output terminals of the PwrSoC module. The time domain responses are then used to generate transfer function descriptions of the two-port network model parameters by processing the time domain data through a parameter estimation algorithm developed in MATLAB. An analysis of both the estimated parameter responses and the transient data was conducted to identify any type of non-linear behavior in the PwrSoC module over a range of operating conditions. Although, mild non-linear behavior was found in the module, the results show that by applying scaling techniques that are simple in nature, the non-linearity is easily addressed allowing the model to maintain a balance between the advantage of computational efficiency well known to average models and the accuracy of the model. Finally, the validation results show that the final Simulink model replicates the PwrSoC module behavior over a wide range of operating conditions including both the voltage regulation mode and the current limit mode of operation.

In addition to the development of the model, a software wrapper was constructed to enhance the portability of the model into larger system type simulations. The wrapper

also provides a user-friendly interface through which system variables can be changed rapidly with ease.

The model does posses a few limitations in what it can accurately model. One of the most obvious drawbacks is that it does not consider the operation of the PwrSoC module in DCM. This is not a major problem in most applications but could become an issue in applications where light load operation of the PwrSoC module is required. Modeling the system in DCM is suggested as future work. Another drawback is that the current limiting behavior of the system after it reaches approximately 25% of the regulated output voltage is also not considered in the model. Although the current set of data in this region suggests that the current limit behavior is non-linear, a more thorough set of data collected below the threshold would be required to fully understand the behavior of the system in this region. Modeling the system in DCM as well as at the lower bounds of current limit mode would likely add to the flexibility of the model. Thus, the addition of these behaviors in the model is recommended for future work.

In conclusion, a complete black-box model of a highly integrated PwrSoC product has been developed and packaged in user-friendly wrapper in Simulink. The techniques used in developing the model are simple in nature and the validation testing shows that the model performs well over the operating space of the PwrSoC converter. Obviously, a more complex model is likely to result in a higher degree of model accuracy but the small gain in accuracy would be at the expense of computational efficiency. Lastly, the model development method presented permits the rapid development of PwrSoC models for use in system design where the technical intellectual property of the manufacturer is difficult to obtain which is a problem inherent to the PwrSoC concept and strengthens the benefits of the black box approach presented in this work.

REFERENCES

[1] V. Valdivia, A. Barrado, A. Lazaro, P. Zumel, and C. Raga, "New Nonlinear Dynamic 'Grey Box' Behavioral Modeling and Identification of Voltage Mode Controlled Buck Derived DC-DC Converters," *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, Washington, DC: IEEE, 2009, pp. 312-317.

[2] P.G. Maranesi, V. Tavazzi, and V. Varoli, "Two-part characterization of PWM voltage regulators at low frequencies," *IEEE Transactions on Industrial Electronics*, vol. 35, 1988, pp. 444-450.

[3] L. Ljung, System Identification – Theory for the User, Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.

[4] L. Ljung, "Black-box models from input-output measurements," *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188)*, Budapest , Hungary: IEEE, 2001, pp. 138-146.

[5] G.W. Wester and R.D. Middlebrook, "Low-Frequency Characterization of Switched dc-dc Converters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, May. 1973, pp. 376-385.

[6] R.D.Middlebrook and Slobodan Ćuk, "A General Unified Approach to Modelling Switching-Converter Power Stages," *International Journal of Electronics,* vol. 42, 1977, pp. 521-550.

[7] L. Amedo, R. Burgos, F. Wang, and D. Boroyevich, "Black-Box Terminal Characterization Modeling of DC-to-DC Converters," *APEC 07 - Twenty-Second Annual IEEE Applied Power Electronics Conference and Exposition*, Anaheim, CA: IEEE, 2007, pp. 457-463.

[8] V. Valdivia, A. Barrado, A. Lazaro, P. Zumel, and C. Raga, "Easy Modeling and Identification Procedure for 'Black Box' Behavioral Models of Power Electronics Converters with Reduced Order Based on Transient Response Analysis," *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, Washington, DC: IEEE, 2009, pp. 318-324.

[9] V. Valdivia, A. Barrado, A. LÁzaro, P. Zumel, C. Raga, and C. FernÁndez, "Simple Modeling and Identification Procedures for 'Black-Box' Behavioral Modeling of Power Converters Based on Transient Response Analysis," *IEEE Transactions on Power Electronics*, vol. 24, Dec. 2009, pp. 2776-2790.

[10] V. Valdivia, A. Barrado, A. Lazaro, C. Fernandez, and P. Zumel, "Black-box modeling of DC-DC converters based on transient response analysis and parametric identification methods," *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, Palm Springs, CA: IEEE, 2010, pp. 1131-1138.

[11] J.A. Oliver, R. Prieto, J.A. Cobos, O. Garcia, and P. Alou, "Hybrid Wiener-Hammerstein Structure for Grey-Box Modeling of DC-DC Converters," *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, Washington, DC: IEEE, 2009, pp. 280-285.

[12] "*LTM4612*"[Online], Available: http://cds.linear.com/docs/Datasheet/4612fb.pdf. May 5, 2011 [June13,2011].

[13] S. Buso, P. Mattavelli, "Digital current mode control,"*Digital Control in Power Electronics*, 1st ed., J. Hudgins, United States: Morgan & Claypool, p. 39, 2006.

[14] Arnedo L, Boroyevich D, Burgos R, Wang F. "Polytopic black-box modeling of dc-dc converters". In: *2008 IEEE Power Electronics Specialists Conference*. Rhodes: IEEE; 2008:1015-1021. Available at: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4592063.

[15] L. Arnedo, D. Boroyevich, R. Burgos, and F. Wang, "Black-Box Terminal Characterization Models for the Analysis and Simulation of Distributed Power Electronic Systems," *2007 IEEE Power Electronics Specialists Conference*, Orlando, FL: IEEE, 2007, pp. 1968-1973.

[16] L. Arnedo, D. Boroyevich, R. Burgos, and F. Wang, "Un-terminated frequency response measurements and model order reduction for black-box terminal characterization models," *2008 Twenty-Third Annual IEEE Applied Power Electronics Conference and Exposition*, Austin, TX: IEEE, 2008, pp. 1054-1060.

[17] R2011a Documentation, http://www.mathworks.com/help/toolbox/simulink/ug/f7-8243.html#f7-19688

APPENDIX A

MATLAB CODE

A.1 G(s) and Y(s) Parameter Estimation Program

```
%% G(s) and Y(s) Parameter Estimation
%
% WRITTEN BY: CRAIG BILBERRY
% MISSISSIPPI STATE UNIVERSITY
% 04/03/2010
%
% IN THE EXECUTION OF THIS CODE, TRANSIENT DATA DUE TO AN INPUT
% VOLTAGE STEP APPLIED TO THE INPUT TERMINALS OF A POWER
% CONVERTER IS IMPORTED IN FROM A .DAT FILE.  THE DATA IS
% PREPROCESSED BY REMOVING THE MEAN VALUE AS WELL AS FILTERING.
% DATA IS THEN CAST INTO AN IDDATA OBJECT AND PLOTTED TO THE
% SCREEN.  THE IDDATA OBJECTS ARE THEN PASSED TO A SYSTEM
% IDENTIFICATION FUNCTION WHERE A TRANSFER FUNCTION IS ESTIMATED
% FOR EACH OF THE PARAMETERS FROM THE TRANSIENT DATA.  MEANING
% THAT THERE IS A TIME TO FREQUENCY DOMAIN CONVERSION.

% Clear the workspace
clc
close all
clear all
format long e

% Select the type of model to be estimated
type = 1; % Type = 1 for analog or type = 0 for discrete

disp('Running Simulation for G(s) and Y(s) Estimates')

% Jogs through the operating points at which the models will be
% estimated
for x=1:7

    % Select the operating point
     switch x
        case 1
            file = '25_Ohm_0.2A'
        case 2
            file = '7.5_Ohm_0.667A'
        case 3
            file = '5_Ohm_1A'
        case 4
            file = '3.75_Ohm_1.33A'
        case 5
            file = '2.5_Ohm_2A'
        case 6
            file = '1.67_Ohm_3A'
        case 7
            file = '1.0_Ohm_5A'
     end
```

```
for k=1:4 %Load up the data
p=['I:\ap\data1\man_portable\Bilberry\InputVoltageStepData\10_Volt\',..
.
    file,'\Ch',num2str(k),'.dat'];
load(p);
end

% Specify the sample rate
Ts=80e-9;

% Establish the data length in seconds
time = 0:Ts:length(Ch1)*Ts-Ts;

%% Preprocess the data
%Vout
dcVout=mean(Ch1(100:500));
Ch11=Ch1-dcVout;

%Vin
dcVin=mean(Ch4(100:500));
Ch14=Ch4-dcVin;

%Iin
dcIin=mean(Ch3(100:500));
Ch13=Ch3-dcIin;

% Plot all of the raw data to the screen for verification
figure

subplot(4,1,1); plot(time,Ch3)
title('Iin Raw Data')

subplot(4,1,2); plot(time,Ch1)
title('Vout Raw Data')

subplot(4,1,3); plot(time,Ch4)
title('Vin Raw Data')

subplot(4,1,4); plot(time,Ch2)
title('Iout Raw Data')

%Create the iddata Objects

%Y(s)   (Iin,Vin,Ts)
y1=iddata(Ch13,Ch14,Ts);

%G(s)   (Vout,Vin,Ts)
g1=iddata(Ch11,Ch14,Ts);

% Plot all the data with removed mean to screen for verification
figure
```

```
subplot(3,1,1); plot(time,y1.y)
title('Iin with mean removed')

subplot(3,1,2); plot(time,g1.y)
title('Vout with mean removed')

subplot(3,1,3); plot(time,y1.u)
title('Vin Step with mean removed')


% Set the rate at which the data is to be re-sampled
 decimationRateY=[32,32,32,32,32,32,32];
 decimationRateG=[32,32,32,32,32,32,32];

% Filter the data
 y=resample(y1,1,decimationRateY(x));
 g=resample(g1,1,decimationRateG(x));

% Generate the resampled time data set
 yTime=0:Ts*decimationRateY(x):length(Ch13)*Ts-Ts;
 yTime=yTime';

 gTime=0:Ts*decimationRateG(x):length(Ch11)*Ts-Ts;
 gTime=gTime';

%% Estimate the parameters transfer functions
 thetaY=[2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1;];
 thetaG=[2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1;];

 Y=oe(y,thetaY(x,1:3),'MaxIter',100,'Tolerance',0.001);

 G=oe(g,thetaG(x,1:3),'MaxIter',100,'Tolerance',0.001);

% Convert from discrete (Z) to continuous (S) transfer functions
if type == 1
  Y=d2c(Y)
  G=d2c(G)
end

% Generate numerator and denominator polynomials for Y parameter
 numY(x,1:length(Y.b))=Y.b;
 denY(x,1:length(Y.f))=Y.f;

% Generate numerator and denominator polynomials for Z parameter
 numG(x,1:length(G.b))=G.b;
 denG(x,1:length(G.f))=G.f;


%% Plot Measured, Filtered and Estimated transient responses to
%% the sceeen

% Y parameter verification
```

```
figure

[p,t]=step(Y,400e-6); %Apply step

plot(time,y1.y,'r')
hold on
plot(yTime,y.y,'color','black','LineWidth',2)
hold on
plot(t+1e-4,0.74*p,'LineWidth',2)

legend('Measured','Filtered','Estimated')
axis([0 4e-4 -2 4])

title(['Verification of Y(s) based on a Vin step at decimation rate
',num2str(decimationRateY(x))])
xlabel(['Num. Poly. order is ',num2str(thetaY(x,1)),...
    '  Den. Poly order is ',num2str(thetaY(x,2)),' run ',num2str(x)])
ylabel('Input Current')

% G parameter verification
figure

[s,t]=step(G,400e-6); %Apply step

plot(time,g1.y,'r')
hold on
plot(gTime,g.y,'color','black','LineWidth',2)
hold on
plot(t+1e-4,0.74*s,'LineWidth',2)

legend('Measured','Filtered','Estimated')
axis([0 4e-4 -0.025 0.025])

title(['Verification of G(s) based on a Vin step at decimation rate
',num2str(decimationRateG(x))])
xlabel(['Num. Poly. order is ',num2str(thetaG(x,1)),...
        '  Den. Poly order is ',num2str(thetaG(x,2)),' run
',num2str(x)])
ylabel('Output Voltage')

end

figure

%% Frequency response of estimated transfer functions at specific
%% operating points

for x=1:7

%Bode information for the Input Admittance Y(s)
if type == 1
    % Continuous time
    kY=tf(numY(x,1:4),denY(x,1:4));
```

```
else
    % Discrete time
    kY=tf(numY(x,1:3),denY(x,1:4),Ts*decimationRateY(x));
end

% Get the bode plot data
[magY1,phaseY1,wY1]=bode(kY,{2*pi*100,2*pi*200e3});
magY((1:length(magY1)),x)=magY1;
wY((1:length(wY1)),x)=wY1;

% Plot the frequency response of Y(s)
bode(kY,{2*pi*1000,2*pi*200e3})

hold on
end

% Set title and setting in the figure
legend('0.2A','0.667','1','1.33','2','3','5A')
title('Bode Diagram of Y(s)')
figure

for x=1:7

%Bode information for the audio-susceptibility G(s)
if type == 1
    % Continuous time
    kG=tf(numG(x,1:4),denG(x,1:4));
else
    % Discrete time
    kG=tf(numG(x,1:3),denG(x,1:4),Ts*decimationRateG(x));
end

% Get the bode plot data
[magG1,phaseG1,wG1]=bode(kG,{2*pi*100,2*pi*200e3});
magG((1:length(magG1)),x)=magG1;
wG((1:length(wG1)),x)=wG1;

% Plot the frequency response of G(s)
bode(kG,{2*pi*1000,2*pi*200e3})

hold on
end

% Set the title and legend for figure
legend('0.2A','0.667','1','1.33','2','3','5A')
title('Bode Diagram of G(s)')

figure

% Set the y axis length
for d=1:7
Gyaxis(1:length(magG),d)=d;
Yyaxis(1:length(magY),d)=d;
```

```
end

%% Plot surface plots of frequency response

%Surface Plot of Y(s) in Bode Form versus Output Current Op.
%Point
surf(wY,Yyaxis,20*log10(magY));
colormap hsv;

title('Surface Bode of Y(s) versus Output Current Op. Point')
xlabel('Frequency (Hz)')
ylabel('Output Current (A)')
zlabel('Mag. (dB)')

set(gca,'XScale','log')


%Surface Plot of G(s) in Bode Form versus Output Current Op.
%Point
figure
surf(wG,Gyaxis,20*log10(magG));
colormap hsv;

title('Surface Bode of G(s) versus Output Current Op. Point')
xlabel('Frequency (Hz)')
ylabel('Output Current (A)')
zlabel('Mag. (dB)')

set(gca,'XScale','log')
```

A.2 H(s) and Z(s) Parameter Estimation Program

```
%% H(s) and Z(s) Parameter Estimation
%
% WRITTEN BY: CRAIG BILBERRY
% MISSISSIPPI STATE UNIVERSITY
% 06/05/2010
%
% IN THE EXECUTION OF THIS CODE, TRANSIENT DATA DUE TO AN OUTPUT
% CURRENT STEP APPLIED TO THE OUTPUT TERMINALS OF A POWER
% CONVERTER IS IMPORTED IN FROM A .DAT FILE.  THE DATA IS
% PREPROCESSED BY REMOVING THE MEAN VALUE AS WELL AS FILTERING.
% DATA IS THEN CAST INTO AN IDDATA OBJECT AND PLOTTED TO THE
% SCREEN.  THE IDDATA OBJECTS ARE THEN PASSED TO A SYSTEM
% IDENTIFICATION FUNCTION WHERE A TRANSFER FUNCTION IS ESTIMATED
% FOR EACH OF THE PARAMETERS FROM THE TRANSIENT DATA.  MEANING
% THAT THERE IS A TIME TO FREQUENCY DOMAIN CONVERSION.

% Clear the workspace
clc
close all
```

```
%clear all

% Format the output string
format short e

disp('Running Simulation for G(s) and Y(s) Estimates')

% Select type = 1 for Continuous or type = 0 for digital
type = 1;

% Jogs through the operating points at which the models will be
% estimated
for x=1:7

    % Select the operating points
    switch x
        case 1
            file = '25_Ohm_0.2A'
        case 2
            file = '7.5_Ohm_0.667A'
        case 3
            file = '5_Ohm_1A'
        case 4
            file = '3.75_Ohm_1.33A'
        case 5
            file = '2.5_Ohm_2A'
        case 6
            file = '1.67_Ohm_3A'
        case 7
            file = '1.0_Ohm_5A'
    end

    % Load up the experimental data
for k=1:4
p=['I:\ap\data1\man_portable\Bilberry\OutputCurrentStepData\10_Volt\',f
ile,'\Ch',num2str(k),'.dat'];
load(p);
end

% Specify the sample rate
Ts=80e-9;

% Build the time set
time = 0:Ts:length(Ch1)*Ts-Ts;

% Plot out the raw data
figure
subplot(2,1,1); plot(time,Ch3)
title('Iin Raw Data')

subplot(2,1,2); plot(time,Ch1)
title('Vout Raw Data')
hold off
```

```
figure
subplot(2,1,1); plot(time,Ch4)
title('Vin Raw Data')

subplot(2,1,2); plot(time,Ch2)
title('Iout Raw Data')

%% Preprocess the data

%Vout
dcVout=mean(Ch1(100:500));
Ch11=Ch1-dcVout;

%Iout
dcIout=mean(Ch2(100:500));
Ch12=Ch2-dcIout;

%Iin
dcIin=mean(Ch3(100:400));
Ch13=Ch3-dcIin;

%% Build the iddata objects
%Z(s)   (Vout,Iout,Ts)
z1=iddata(Ch11,Ch12,Ts);

%H(s)   (Iin,Iout,Ts)
h1=iddata(Ch13,Ch12,Ts);

% Plot the preprocessed data for verification
figure

subplot(3,1,1); plot(time,Ch1);%plot(time,z1.y)
title('Vout with mean removed')

subplot(3,1,2); plot(time,Ch3);%plot(time,h1.y)
title('Iin with mean removed')

subplot(3,1,3); plot(time,Ch2);%plot(time,z1.u)
title('Iout Step with mean removed')

% Set the rate at which the data is to be re-sampled
 decimationRateH=[32,32,32,32,32,32,32];
 decimationRateZ=[32,32,32,32,32,32,32];

 % Resample the data
 z=resample(z1,1,decimationRateZ(x));
 h=resample(h1,1,decimationRateH(x));

 % Resample the time vector to match the data length
 zTime=0:Ts*decimationRateZ(x):length(Ch11)*Ts-Ts;
 zTime=zTime';
```

```
 hTime=0:Ts*decimationRateH(x):length(Ch13)*Ts-Ts;
 hTime=hTime';
 %% Estimate the transfer function of each parameter

 % Set the desired order of the estimated transfer functions
 thetaZ=[2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1;];
 thetaH=[2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1; 2 3 1;];

 % Run the System Identification Tool

Z=oe(z,thetaZ(x,1:3),'MaxIter',100,'Tolerance',0.001,'searchmethod','lm
');

H=oe(h,thetaH(x,1:3),'MaxIter',100,'Tolerance',0.001,'searchmethod','lm
');

 % Convert from Z to S
 if type == 1
 Z=d2c(Z);
 H=d2c(H);
 end

 % Generate numerator and denominator polynomials for Z parameter
 numZ(x,1:length(Z.b))=Z.b;
 denZ(x,1:length(Z.f))=Z.f;

 % Generate numerator and denominator polynomials for H parameter
 numH(x,1:length(H.b))=H.b;
 denH(x,1:length(H.f))=H.f;

%% Plot Measured, Filtered and Estimated transient responses to
%% the screen

% Z parameter verification
figure

% Apply a step to the estimated tf
[y,t]=step(Z,400e-6);

% Plot the step response
plot(time,z1.y,'r')
hold on
plot(zTime,z.y,'color','black','LineWidth',2)
hold on
plot(t+1e-4,0.5*y,'LineWidth',2)

% Set the handles of the figure
legend('Measured','Filtered','Estimated')
axis([0 4e-4 -0.04 0.03])
title(['Verification of Z(s) based on a current step at decimation rate
'...
    ,num2str(decimationRateZ(x))])
```

```
xlabel(['Num. Poly. order is ',num2str(thetaZ(x,1)),...
    '  Den. Poly order is ',num2str(thetaZ(x,2)),' run ',num2str(x)])
ylabel('Output Voltage')

% H parameter verification
figure

% Apply a step to the H parameter tf
[s,t]=step(H,400e-6);

% Plot the response
plot(time,h1.y,'r')
hold on
plot(hTime,h.y,'color','black','LineWidth',2)
hold on
plot(t+1e-4,0.5*s,'LineWidth',2)

% Set the handles of the figure
legend('Measured','Filtered','Estimated')
axis([0 4e-4 -2 3])
title(['Verification of H(s) based on a current step at decimation rate
'...
    ,num2str(decimationRateH(x))])
xlabel(['Num. Poly. order is ',num2str(thetaH(x,1)),...
    '  Den. Poly order is ',num2str(thetaH(x,2)),' run ',num2str(x)])
ylabel('Input Current')


end
figure

%% Frequency response of estimated transfer functions at specific
%% operating points

for x=1:7
%Bode Information for the Output Impedance Z(s)
if type == 1
    % Continuous time
    kZ=tf(numZ(x,1:4),denZ(x,1:4));
else
    % Discrete time
    kZ=tf(numZ(x,1:3),denZ(x,1:4),Ts*decimationRateZ(x));
end

[magZ1,phaseZ1,wZ1]=bode((kZ),{2*pi*100,2*pi*200e3});

magZ((1:length(magZ1)),x)=magZ1;
wZ((1:length(wZ1)),x)=wZ1;

bode(kZ,{2*pi*1000,2*pi*200e3})

hold on
end
```

```
% Set the figure handles
legend('0.2A','0.667','1','1.33','2','3','5A')
title('Bode Diagram of Z(s)')
figure

for x=1:7
%Bode Information for the Output Impedance H(s)
if type == 1
    kH=tf(numH(x,1:4),denH(x,1:4));
else
    kH=tf(numH(x,1:3),denH(x,1:4),Ts*decimationRateH(x));
end

% Get the frequency response data
[magH1,phaseH1,wH1]=bode((kH),{2*pi*100,2*pi*200e3});

magH((1:length(magH1)),x)=magH1;
wH((1:length(wH1)),x)=wH1;

% Plot the frequency response
bode(kH,{2*pi*1000,2*pi*200e3})

hold on
end

% Set the bode plot handles
legend('0.2A','0.667','1','1.33','2','3','5A')
title('Bode Diagram of H(s)')

figure

% Set the axis length
for d=1:7
Zyaxis(1:length(magZ),d)=d;
Hyaxis(1:length(magH),d)=d;
end

%Surface plot of Z(s)
surf(wZ,Zyaxis,20*log10(magZ));
colormap hsv;
title('Surface Bode of Z(s) versus Output Current Op. Point')
xlabel('Frequency (Hz)')
ylabel('Output Current (A)')
zlabel('Mag. (dB)')
set(gca,'XScale','log')

%Surface Plot of H(s)
figure
surf(wH,Hyaxis,20*log10(magH));
colormap hsv;
title('Surface Bode of H(s) versus Output Current Op. Point')
xlabel('Frequency (Hz)')
```

```
ylabel('Output Current (A)')
zlabel('Mag. (dB)')
set(gca,'XScale','log')
```

A.3 Small Signal Model Simulation Program

```
%% Simulate Model
%
% WRITTEN BY: CRAIG BILBERRY
% MISSISSIPPI STATE UNIVERSITY
% 10/10/2010
%
% SIMULATES THE FULL SMALL SIGNAL MODEL AT DIFFERENT OPERATING
% POINTS. PRINTS THE RESULTING RESPONSE TO THE SCREEN FOR
% ANALYSIS AND COMPARISON TO THE EXPERIMENTAL DATA
%%

% Clear the workspace
clc
close all

% Define the data sample rate
Ts=80e-9;

% Specify which model that you wish to run
model = 3;

% Specify the desired input voltage
Vin = 10;

% Specify the desired output voltage
VoutDC = 4.963;

% Select exp. Setup: Voltage Step = 1, Current Step = 0
Setup = 0;

% Select type = 1 for Continuous or type = 0 for digital
type = 1;

% Run the state machine
switch model
        case 1
            % Load set at 25 Ohms
            if Setup == 1
                Iout = 0.194;
            else
                Iout = 0.2;
            end
            file = '25_Ohm_0.2A';

        case 2
            % Load set at 7.5 Ohms
```

```
            if Setup == 1
                Iout = 0.622;
            else
                Iout = 0.667;
            end
            file = '7.5_Ohm_0.667A';

        case 3
            % Load set at 5 Ohms
             if Setup == 1
                Iout = 0.946;
             else
                Iout = 1;
             end
            file = '5_Ohm_1A';

        case 4
            % Load set at 3.75 Ohms
            Iout = 1.33;
            file = '3.75_Ohm_1.33A';

        case 5
            % Load set at 2.5 Ohms
            if Setup == 1
                Iout = 1.91;
            else
                Iout = 2;
            end
            file = '2.5_Ohm_2A';

        case 6
            % Load set at 1.67 Ohms
            if Setup == 1
                Iout = 2.844;
            else
                Iout = 3;
            end
            file = '1.67_Ohm_3A';

        case 7
            % Load set at 1 Ohm
            Iout = 4.848;
            file = '1.0_Ohm_5A';
end

%% Load the correct model

% Use this model for Discrete Simulation
if type == 0
 numYS_10 = numY(6,:);
 denYS_10 = denY(6,:);

 numGS_10 = numG(4,:);
 denGS_10 = denG(4,:);
```

```
 numHS_10 = numH(6,:);
 denHS_10 = denH(6,:);

 numZS_10 = numZ(6,:);
 denZS_10 = denZ(6,:);


% THIS SHOULD ONLY BE UNCOMMENTED IF THE DYNAMIC MODELS USED IN
% THE FULL DISCRETE SIMULATION NEEDS TO BE CHANGED 5/9/2011
 %save I:\ap\data1\man_portable\Bilberry\InputVoltageDynComp\...
 %10V_Discrete.mat numYS_10 denYS_10 numGS_10 denGS_10 numHS_10...
 %denHS_10 numZS_10 denZS_10

end

% Use this model for Analog Simulation
if type == 1
 numYS_10 = numY(5,:);
 denYS_10 = denY(5,:);

 numGS_10 = numG(5,:);
 denGS_10 = denG(5,:);

 numHS_10 = numH(6,:);
 denHS_10 = denH(6,:);

 numZS_10 = numZ(6,:);
 denZS_10 = denZ(6,:);

% THIS SHOULD ONLY BE UNCOMMENTED IF THE DYNAMIC MODELS USED IN
% THE FULL CONTINUOUS SIMULATION NEEDS TO BE CHANGED 5/9/2011
 %save I:\ap\data1\man_portable\Bilberry\InputVoltageDynComp\...
 %10V_Continuous.mat numYS_10 denYS_10 numGS_10 denGS_10...
 %numHS_10 denHS_10 numZS_10 denZS_10

end

%% Select the experimental data for analysis
for k=1:4
    if Setup == 1

p=['I:\ap\data1\man_portable\Bilberry\InputVoltageStepData\10_Volt\'...
            ,file,'\Ch',num2str(k),'.dat'];
    else

p=['I:\ap\data1\man_portable\Bilberry\OutputCurrentStepData\10_Volt\'..
.
            ,file,'\Ch',num2str(k),'.dat'];
    end
load(p);
end
```

```
% Set the variables
v_out = Ch1;
i_out = Ch2;
i_in = Ch3;
v_in = Ch4;
%time = 0:Ts:length(Ch1)*Ts-Ts;

% Start the simulation in Simulink
if type == 1
    % Run the continuous model
    sim('Black_Box_Model_Continuous_10')
else
    % Run the discrete model
    sim('Black_Box_model_10')
end

% Setup the simulation parameters in Simulink
sampleTime = 80e-9;  % Set the sample rate of the simulation
decimation = 1; % No neeed to resample
startTime = 14.00016e-3/sampleTime/decimation; % Steady State Reached
stopTime = length(ScopeData.time(:,1)); % Stop time
xMin = startTime * sampleTime * decimation; % Set y axis min
xMax = stopTime * sampleTime * decimation;  % Set y axis max

% Get the simulated Time data from Simulink
time_est = ScopeData.time(startTime:stopTime,1);

% Get the simulated output voltage from Simulink
Vout_est = ScopeData.signals(1,1).values(startTime:stopTime,1);

% Get the simulated Input Voltage data from Simulink
Vin_est = ScopeData2.signals(1,1).values(startTime:stopTime,1);

% Get the simulated Input Current data from Simulink
Iin_1 = ScopeData1.signals(1,1).values(startTime:stopTime,1);

% Get the simulated Output Current data from Simulink
Iout_est = ScopeData3.signals(1,1).values(startTime:stopTime,1);

%% Overlay the experimental data and simulated transient response on
%% a single plot

% Output Voltage response
figure
subplot(2,2,1);plot(time_est,v_out+VoutDC,'Red')
hold on
subplot(2,2,1);plot(time_est+1e-6,Vout_est)
title('Black-Box Model Output Voltage')
axis([xMin xMax VoutDC-0.03 VoutDC+0.03]);

% Input Voltage response
subplot(2,2,2);plot(time_est,v_in,'Red')
hold on
```

```
subplot(2,2,2);plot(time_est,Vin_est)
title('Black-Box Model Input Voltage')
axis([xMin xMax Vin-.25 Vin+0.3]);

% Input Current response
subplot(2,2,3);plot(time_est,i_in,'red')
hold on
subplot(2,2,3);plot(time_est,Iin_1)
title('Black-Box Model Input Current')
axis([xMin xMax 0 2]);

% Output Current response
subplot(2,2,4); plot(time_est,i_out,'red')
hold on
subplot(2,2,4); plot(time_est,Iout_est)
title('Black-Box Output Current')
axis([xMin xMax Iout-0.5 Iout+0.8]);
```

A.4 Full Model Simulation

```
%% Full Model Simulation
%
% WRITTEN BY: CRAIG BILBERRY
% MISSISSIPPI STATE UNIVERSITY
% 12/21/2011
%
% THIS PROGRAM EXECUTES THE FULL BLACK BOX MODEL INCLUDING LARGE
% AND SMALL SIGNAL MODELS.  BOTH THE DISCRET VERSION AS WELL AS
% THE CONTINUOUS VERSIONS ARE SIMULATED FROM HERE.  THIS PROGRAM
% CONTROLS THE EXECUTION OF THE FULL_DISCRETE/CONTINUOUS_MODEL.MDL
% IN SIMULINK

% Clear the current workspace
clc
close all
clear all

% Specify the sample rate
Ts=80e-9;

%% Specify which model that you wish to run
model = 6;
Vin = 10;
VoutDC = 4.963;

%% Select exp. Setup: Voltage Step = 1, Current Step = 0
Setup = 0;

%% Select Continuous = 1 or Discrete = 0
type = 1;
```

```
% Adjust the step time which varies in the experimental data
if VoutDC == 15
    if Setup == 0
        StepTime = 0.0945-230*80e-9;
    else
        StepTime = 0.0945-260*80e-9;
    end
else
    StepTime = 0.0945;
end


% Going ahead and loading all of the estimated models for later use
% load
I:\ap\data1\man_portable\Bilberry\InputVoltageDynComp\12V_Discrete.mat
if type == 0
    load
I:\ap\data1\man_portable\Bilberry\InputVoltageDynComp\10V_Discrete.mat
else
    load
I:\ap\data1\man_portable\Bilberry\InputVoltageDynComp\10V_Continuous.ma
t
end
%load
I:\ap\data1\man_portable\Bilberry\InputVoltageDynComp\8V_Discrete.mat

% Begin the state machine
switch model
        case 1
            if Setup == 1
                Iout = 0.194;
            else
                Iout = 0.2;
            end
            file = '25_Ohm_0.2A';
        case 2
            if Setup == 1 %The data in voltage step mode 12V is off
(LOW)
                Iout = 0.622;
            else
                Iout = 0.667;
            end
            file = '7.5_Ohm_0.667A';
        case 3
             if Setup == 1
                Iout = 0.946;
                if Vin == 8
                    Iout = 0.96;
                end
                if Vin == 12
                    Iout = 0.92;
                end
                if Vin == 30
                    Iout = 1;
                end
```

```
        elseif Setup == 0
            Iout = 1;
            if Vin == 12
                Iout = 0.976;
            end
        end
        if VoutDC== 4.963
                file = '5_Ohm_1A';
        elseif VoutDC == 12
                file = '12_Vout\12_Ohm_1A';
        elseif VoutDC ==15
                file = '15_Vout\15_Ohm_1A';
        end
case 4
    Iout = 1.33;
    file = '3.75_Ohm_1.33A';
case 5
    if Setup == 1
        Iout = 1.91;
        if Vin == 8
            Iout = 1.952;
        end
    else
        Iout = 2;
    end
    file = '2.5_Ohm_2A';
case 6
    if Setup == 1
        Iout = 2.844; % Default for 10V Input Voltage
        if Vin == 8
            Iout = 2.96;
        end
        if Vin == 12
            Iout = 2.846;
        end
        if Vin == 30
            Iout = 3;
        end
    elseif Setup ==0
        Iout = 3;
        if Vin == 8
            Iout = 3.083
        end
        if Vin == 12
            Iout = 2.956;
        end
    end
        if VoutDC== 4.963
            file = '1.67_Ohm_3A';
        elseif VoutDC == 12
            file = '12_Vout\4_Ohm_3A';
        elseif VoutDC ==15
            file = '15_Vout\5_Ohm_3A';
            Iout = 3.06
        end
```

```
        case 7
            if Setup == 1 % Voltage Step
                Iout = 4.848; % Default for 10V input voltage
                if Vin == 8
                    Iout = 4.94;
                end
                if Vin == 12
                    Iout = 4.84;
                end
            elseif Setup == 0 % Current Step
               Iout = 5.3; % Default for 10V input voltage
               if Vin == 12
                   Iout =  5.088;
               end
            end
                if VoutDC== 4.963
                    file = '1.0_Ohm_5A';
                elseif VoutDC == 12
                    file = '12_Vout\2.67_Ohm_4.5A';
                    Iout = 4.3;
                elseif VoutDC ==15
                    file = '15_Vout\3_Ohm_5A';
                    Iout = 5.136;
                end
end



%Load up the raw data for comparison
for k=1:4
    if Setup == 1

p=['I:\ap\data1\man_portable\Bilberry\InputVoltageStepData\',...
            num2str(Vin),'_Volt\',file,'\Ch',num2str(k),'.dat'];
    else

p=['I:\ap\data1\man_portable\Bilberry\OutputCurrentStepData\',...
            num2str(Vin),'_Volt\',file,'\Ch',num2str(k),'.dat'];
    end
 % Load up the data
load(p);
end

% Set the variables to the experimental data
v_out = Ch1;
i_out = Ch2;
i_in = Ch3;
v_in = Ch4;

% Select the simulation mode
if type == 1
    sim('Full_Continuous_Model')
else
    %sim('Full_Discrete_Model','StopTime','94.80076e-3')
```

```
    sim('Full_Discrete_Model')
end

%% Begin execution of the Simulink model

% Set the parameters in Simulink
        sampleTime = 80e-9;
        decimation = 1;
        startTime = 94.40084e-3/sampleTime/decimation;
        stopTime = length(ScopeData.time(:,1));
        xMin = startTime * sampleTime * decimation;
        xMax = stopTime * sampleTime * decimation - 0.0002;

% Reserve the data generated in Simulink for comparison
% with the experimental data
time_est = ScopeData.time(startTime:stopTime,1);
Vout_est = ScopeData.signals(1,1).values(startTime:stopTime,1);
Vin_est = ScopeData2.signals(1,1).values(startTime:stopTime,1);
Iin_1 = ScopeData1.signals(1,1).values(startTime:stopTime,1);
Iout_est = ScopeData3.signals(1,1).values(startTime:stopTime,1);


delay = 0;
figure

%% Display Output Voltage
subplot(2,2,1);plot(time_est+delay,v_out+VoutDC,'Red')
hold on
subplot(2,2,1);plot(time_est,Vout_est,'LineWidth',1)

title('Black-Box Model Output Voltage')
axis([xMin xMax VoutDC-0.05 VoutDC+0.03]);
%axis([xMin xMax VoutDC-0.01 VoutDC+0.02]);

%% Display Input Voltage
subplot(2,2,2);plot(time_est,v_in,'Red')
hold on
subplot(2,2,2);plot(time_est,Vin_est,'LineWidth',1)

title('Black-Box Model Input Voltage')
axis([xMin xMax Vin-0.5 Vin+0.5]);
%axis([xMin xMax Vin-.1 Vin+1]);

%% Display Input Current
subplot(2,2,3);plot(time_est+delay,i_in,'red')
hold on
subplot(2,2,3);plot(time_est,Iin_1,'LineWidth',1)

title('Black-Box Model Input Current')
axis([xMin xMax 0 3.5]);
%axis([xMin xMax 0 6]);

%% Display Output Current
```

```
subplot(2,2,4); plot(time_est,i_out,'red')
hold on
subplot(2,2,4); plot(time_est,Iout_est,'LineWidth',1)

title('Black-Box Output Current')
axis([xMin xMax Iout-1 Iout+1.5]);
%axis([xMin xMax Iout-0.1 Iout+0.1]);
```