A NONLINEAR DIGITAL CONTROL SOLUTION FOR A DC/DC POWER CONVERTER

MINSHAO ZHU

Bachelor of Science in the Department of Electronics Science

NanKai University, P.R. China

July 1992

Submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

CLEVELAND STATE UNIVERSITY

February, 2002

This thesis has been approved

For the Department of Electrical and Computer Engineering

And the College of Graduate Studies by

_____

Thesis Committee Chairperson, Dr. Zhiqiang Gao

_____

Department/Date

_____

Thesis Committee Member, Dr. Dan Simon

_____

Department/Date

_____

Thesis Committee Member, Dr. Ana Stankovic

_____

Department/Date

DEDICATION

To my parents and lovely wife…

It is their anticipation that stimulates me to pursue higher education!

ACKNOWLEDGMENT

First, I would like to express my deepest appreciation to my advisor, Dr. Zhiqiang Gao, for his supervision throughout the course of my study. It was an invaluable experience working on his industry oriented projects and learning from his problem-solving methodology and approach to research.

I would like to thank Dr. Dan Simon and Dr. Ana Stankovic, who are on my committee, for their time in reading and evaluating this thesis.

I would also like to thank Jack Zeller, Marcelo Gonzalez, Greg Tollis, Arthur Stachowicz, Ivan Jurcic, Adrienne Fox, Charles Alexander, Dave Gerdeman, Dave Wladyka, Aaron Radke, Bosheng Sun, Zhan Ping and the rest of the electrical engineering department at Cleveland State University for the endless support given to me while at Cleveland State University.

I would like to thank my friends Weiwen Wang, Shaohua Hu, Yi Hou, Wei Xiong, Jing Liu and Shahid Parvez. They provided help and encouragement when I was performing my research and writing this thesis.

A special thanks to my lovely wife, FangFang Qu and our parents, for their continuous support, encouragement, and inspiration to finish my degree.

A NONLINEAR DIGITAL CONTROL SOLUTION FOR A DC/DC POWER CONVERTER

MinShao Zhu

ABSTRACT

A digital Nonlinear Proportional-Integral-Derivative (NPID) control algorithm was proposed to control a 1-kW, PWM, DC/DC, switching power converter. The NPID methodology is introduced and a practical hardware control solution is obtained. The design of the controller was completed using Matlab® Simulink, while the hardware-in-the-loop testing was performed using both the dSPACE® rapid prototyping system, and a stand-alone Texas Instruments® Digital Signal Processor (DSP)-based system. The final Nonlinear digital control algorithm was implemented and tested using the ED408043-1 Westinghouse DC-DC switching power converter. The NPID test results are discussed and compared to the results of a standard Proportional-Integral (PI) controller.

Keywords: PWM, DC/DC power converter, nonlinear PID (NPID), Tracking Differentiator (TD)

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Switched-mode power supplies (SMPS) became accepted and broadly applied around the mid-1970s [1] for many reasons. The benefits of SMPS over their linear counterpart include reduced weight, reduced size, and improved efficiency. Switched-mode DC/DC power converters are usually based on Pulse-Width Modulation (PWM) control schemes. Designers can achieve a stable, well-regulated DC output for a wide variety of applications. Since the very beginning, SMPS controllers have been implemented almost exclusively using complex analog circuitry.

Since the designed regulator is to be combined with a PWM generator that produces discrete signals, a simple PI controller does not seem feasible due to the nonlinear operation of the components used in the power converter. The complexities of various advanced control algorithms impede the implementation of these controllers in the practical analog control circuit [4].

With the recent advancement in the speed and size of digital technology (DSP, microprocessors, etc.), it is now relatively easy to implement digital control design for DC/DC converters. The intelligent digital control will have many benefits that will allow Power Management and Distribution (PMAD) systems to achieve higher efficiency, lower weight, better power quality and higher fault tolerance and reliability [3]. The flexibility of digital signal processors allows us to readily achieve implementation by software coding. The digital controller is also highly immunized to environmental changes such as temperature and aging of components. These advances in implementation and design capability can be obtained at low cost because of the widespread availability of inexpensive and powerful processors and their related peripheral devices. Modern computer network concepts also allow for easy communication between digitalized systems to achieve intelligent system management.

The program in Advanced Engineering Research Lab (AERL) funded by NASA Glenn Research Center concentrates on developing advanced digital controls and distributed processing algorithms for PMAD components and systems to improve their size, weight, efficiency, and reliability.

1.2 Literature Review

In the area of power converter based control, there are many papers on nonlinear and digital control. The papers discussed below are good examples of applications of both nonlinear control and digital control of power converter systems.

The paper, "CUK Converter Global Control via Fuzzy Logic and Scaling Factors," by A. Balestrino, focuses on the implementation of a digital PI-Fuzzy controller. The controller is applied to a CUK converter with variable output voltage capability by applying a suitable variation of the scaling factors. Specific PI techniques were introduced followed by a discussion of the properties criteria for a CUK power converter topology. Next, a PI-Fuzzy controller was proposed to achieve these properties. After applying membership and Fuzzy Association Memory (FAM) concepts to the converter system and controller, the PI-fuzzy controller was implemented using MATLAB Real-Time Workshop. In order to improve the variant dynamic performance for various user-defined setpoints, non-linear scaling factors were introduced in terms of duty ratio according to a small signal linear model for a general power converter. Finally, successful results were presented and discussed for various operating conditions.

In a related paper, extended linearization techniques were proposed by Hebertt Sira-Ramirez for the design of nonlinear PI controllers for PWM controlled converters with constant setpoints [8]. The Ziegler-Nichols method was applied to obtain a family of parameterized transfer function models of the linearized average converter behavior around a constant operating "equilibrium" point. The linearized transfer function is then used to design a nonlinear PI controller. Implementation of the designed nonlinear PI controller using a PWM control scheme was also discussed. The boost and the buck-boost converters were treated separately, and the regulated performance was illustrated through computer simulation experiments.

Gupta, Tarun implemented a fuzzy controller for dc-dc converters using an inexpensive 8-bit microcontroller [9]. An "on-chip" analog-to-digital (A/D) converter and

PWM generator eliminate the external components needed to perform these functions. Implementation issues include limited on-chip program memory of 2 kB, unsigned integer arithmetic, and computational delay. The duty cycle for the dc-dc converter can only be updated every eight switching cycles because of the time required for the A/D conversion and the control calculations. However, it is demonstrated here that stable responses can be obtained for both buck and boost converters under these conditions. Another important result is that the same microcontroller code, without any modifications, can control both converters because their behavior can be described by the same set of linguistic rules. The contribution shows that a nonlinear controller such as fuzzy logic can be inexpensively implemented with microcontroller technology.

1.3 Problem Formulation and Motivation

Power converter designers are concerned with high-efficiency conversion of electric power from the form available at the input, or power source, to the form required at the output or load. Commonly, one talks of DC/DC converters where "DC" here typically refers to a nominally constant voltage waveform. The goal of high efficiency dictates that the power processing components in the circuits are close to lossless.

Switches, capacitors, inductors, and transformers are therefore the typical components in a power electronic converter. The switches are operated cyclically, and serve to vary the "Duty-Ratio" over a complete switching cycle. The capacitors and inductors perform filtering actions, regulating power flows by temporarily storing or supplying energy. The transformers scale voltages and currents, and also provide galvanic isolation between the source and load.

The main control decision for a DC/DC converter is when to open and close the power switches. It is by modulating the instants at which the switches are opened and closed that dictates the output of a power electronic converter. Feedback and feedforward control can be easily integrated with PWM control schemes to improve regulation of the converter.

Figure 1 shows a full-bridge, full-wave, power converter schematic. This is an appropriate topology for power applications between 400 and 2000+ Watts [5]. This topology requires four power switches, two of which have floating drive circuits, and is the most costly to implement compared with other topologies. There are a few other PWM switching power supply topologies including buck, boost, buck-boost, half-forward, flyback, and half bridge. At high power levels, the added cost is necessary due to the topology that is required.

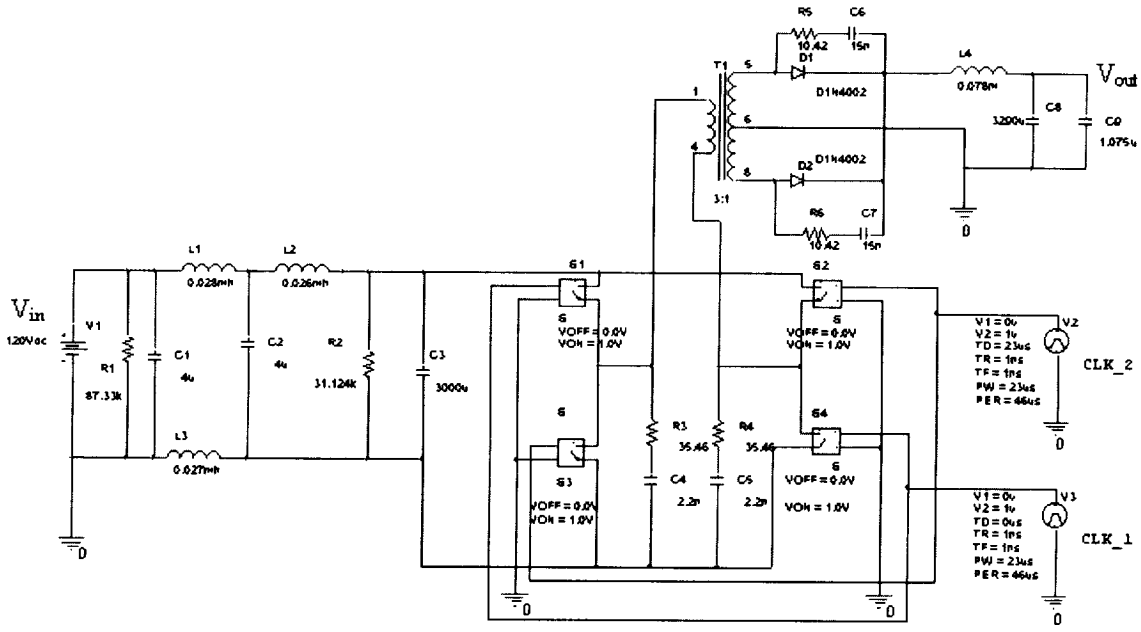


Figure 1 Full-Bridge PWM Switching Mode DC/DC Converter

The components connected across the 120V DC input power supply lines include the four switches (MOSFET bridge) and the input filter. The input filter is divided into two parts, an Electromagnetic Interference (EMI) filter, and a Radio Frequency Interference (RFI) filter. Following the MOSFET switches is a 3:1 step-down isolation transformer. The final section of the converter following the transformer includes the rectifier and output filter.

Figure 2 shows bi-phase PWM signals for a full-bridge power converter. T_c represents the switching period, T_a is phase 1's on time, and T_b is phase 2's on time. The phase 1 and phase 2 signals link to CLK_1 and CLK_2 in Figure 1 respectively. CLK_1 and CLK_2 control the open and close of four MOSFETs switches, where each CLK signal controls two of the four MOSFETs. The overall duty ratio for the power converter is shown below in Eq. (1.1). The input voltage V_{in}, output voltage V_{out} and Duty Ratio relationship is described in Eq. (1.2).

$$\text{Duty Ratio} = (T_a + T_b) / T_c \tag{1.1}$$

$$V_{out} = V_{in}/3 * (T_a + T_b)/T_c \tag{1.2}$$

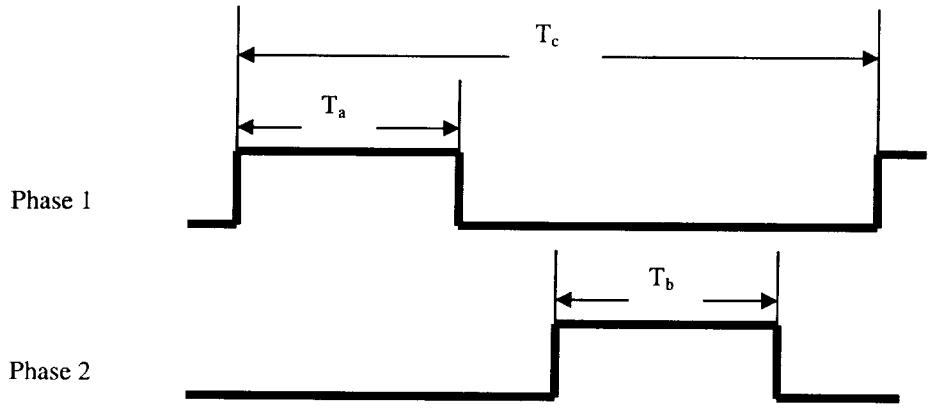


Figure 2 Bi-Phase PWM Signal Definitions

The output voltage of the DC-DC switching power converter can be controlled by the duty ratio found in the bi-phase PWM signal. The normal duty ratio control loop is shown in Figure 3.

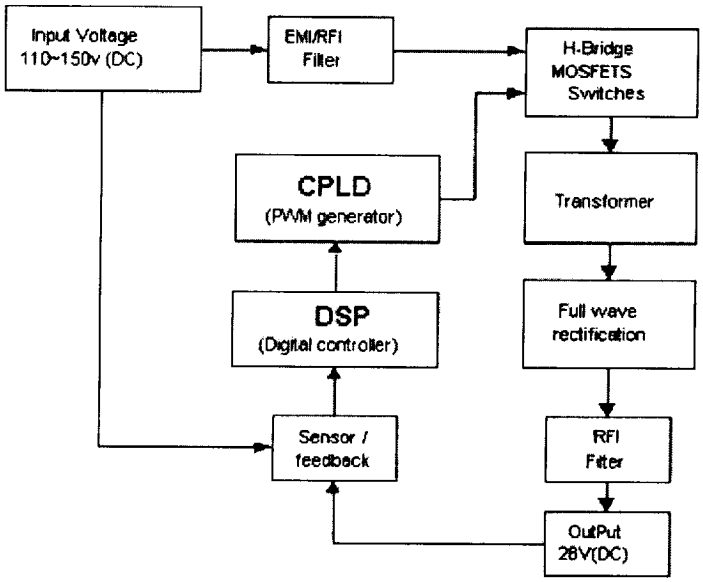


Figure 3 DSP Convert Block Diagram

The two PWM signals, Phase 1 and Phase 2 were generated by an ALTERA® Complex Programmable Logic Device (CPLD) based on signals received from the dSPACE® rapid prototyping system. In addition, the CPLD is programmed to provide a blanking time to prevent the two switch pairs from being on simultaneously. The control algorithm is executed in the DSP with feedback from the output voltage and input voltage. The control result is converted to pulse counter for the CPLD to create the PWM signal.

The objective of this research is to develop a digital control algorithm for a DC-DC switching power converter. The performance of the DC-DC switching power converter is based on the accuracy and stability of the output voltage under any possible condition.

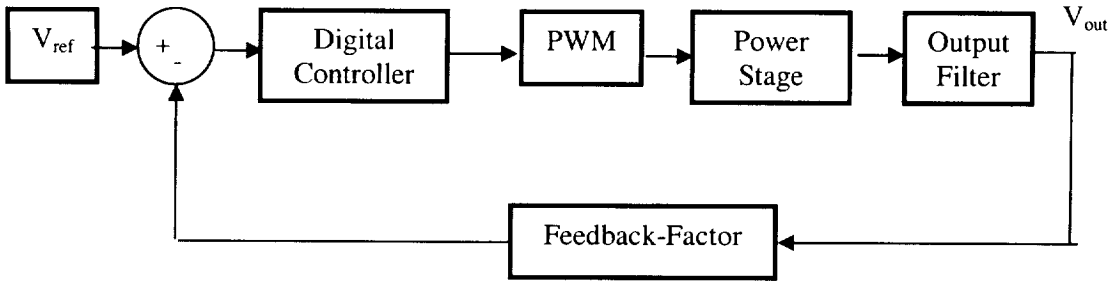


Figure 4 Digital Control Loop

In Figure 4, the V_{ref} in the control loop is the desired output voltage for the DC/DC switching power converter. The actual measured output voltage of the DC/DC switching power converter is compared to the V_{ref} point to generate the error signal. The objective of the digital controller is to regulate the error in the output voltage so that it approaches zero within a given time and voltage constraint. The following tasks were completed:

- Design of digital control algorithm
- Software simulation
- Hardware-in-the-loop simulation
- Implementation
- Analysis of results

A DSP solution was employed to control a switch mode DC/DC power converter. A novel NPID control scheme was proposed to enhance the digital PI control. The NPID digital controller design and simulation results are discussed in Chapter II and Chapter III. The NPID hardware implementation and the details of the experiment are discussed in Chapter IV. In Chapter V, a comparison of the test results obtained for the PI and

NPID controllers are presented. Finally, Chapter VI will summarize the achievements of this research and suggest some further objectives for the study of digital control.

CHAPTER II

NPID CONTROL DESIGN

In this chapter, the NPID methodology is introduced and analyzed. A PID controller was first implemented to provide a foundation for the NPID control design. The PID design was based on pole placement concepts using a 2^{nd} order transfer function model. Finally, the Nonlinear PID control design and implementation procedures are discussed.

2.1 NPID Control Concept

The classical PID controller is simple and easy to implement. PID controllers are currently used for 90% of all industrial control applications. The mathematical description of a PID controller is shown below in Eq. (2.1).

$$u = K_p e + K_i \int e dt + K_d \dot{e} \tag{2.1}$$

The u stands for the control output and represents the sum of three parts. The e, $\int e$ and $\dot{e}$ terms represent the actual error, the integral of the error, and the derivative of the error respectively. The three terms, K_p, K_i, and K_d represent the proportional, integral, and derivative gains for the PID controller.

Due to the high popularity of PID controllers in industry, it is common to find controllers that are not optimized for the particular system they are controlling. The reason for this is that the controller is often poorly tuned by the user. It is quite common that derivative action is not used due to unavoidable noise disturbance in practical implementation.

Nonlinear PID control schemes have been recently proposed for use with PWM controlled power converters. Since the designed regulator is to be combined with a PWM actuator (MOSFET, IGBT, etc.) producing discrete signals, a simple PID (linear) controller does not seem optimal due to the nonlinear characteristics of the power converter [8].

A novel NPID [7] control algorithm was introduced in [10] and was investigated in this research. The focus of the research was to implement the NPID control design on a

Westinghouse power converter unit used in the Advanced Engineer Research Lab (AERL). The NPID controller was developed as an alternative control strategy to the previously developed PI controller.

The control law for the novel NPID is based on the classical PID control law, and is shown below in Eq. (2.2):

$$u = K_p fal(e,\alpha_p,\delta_p) + K_i \int fal(e,\alpha_i,\delta_i)dt + K_d fal(\dot{e},\alpha_d,\delta_d) \qquad (2.2)$$

Again, the u, e, $\int e$, $\dot{e}$, K_p, K_i and K_d terms have the same definition as the PID controller described in equation (2.1). The $fal(x,\alpha,\delta)$ term is a nonlinear function illustrated in Figure 5, and is defined by Eq. (2.3).

$$y = fal(x,\alpha,\delta) = \begin{cases} sign(x)\cdot|x|^{\alpha}, & \text{when } |x| > \delta \\ \delta^{\alpha-1}\cdot x, & \text{when } |x| \le \delta \end{cases} \qquad (2.3)$$

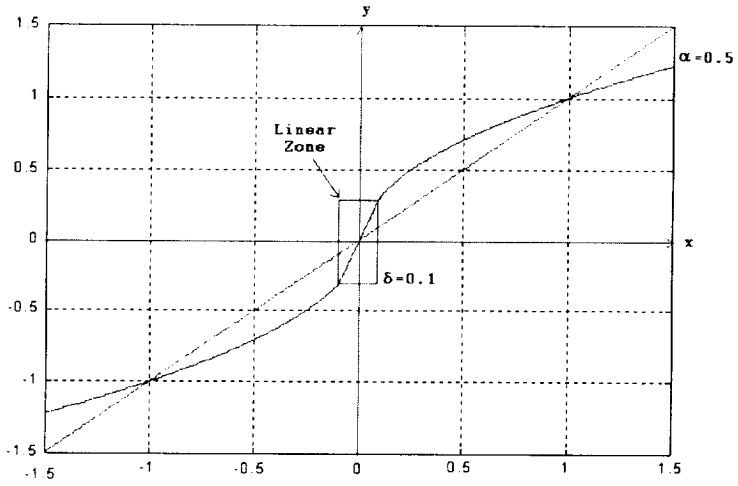


Figure 5 Nonlinear *fal* Function Track

Here, x is the controller input (actual error) and y is the control output (modified error). The α and δ terms are two curve-shaping parameters of the nonlinear function.

Usually, α is chosen between 0 and 1 ($0<\alpha\leq 1$). When $\alpha = 1$, the fal function becomes a linear function described by y = x. The δ term is a small positive number applied to create a small linear area in this nonlinear function where x is near zero. This is to prevent excessive gain when error is small, which was known to cause high frequency chattering in some simulation studies.

The idea of the NPID controller is to use a nonlinear combination of e, $\int e$ and $\dot{e}$ in place of the linear combination used in the classical PID controller. The $fal(x,\alpha,\delta)$ function is an exponential function where α is the exponent. The $fal(x,\alpha,\delta)$ function is able to achieve limited high gain while the error is small (in small area δ), and low gain while the error is large. This algorithm makes it easy to remove overshoot in the start-up transient response (with high error and low gain), and achieve strong disturbance rejection after settling at the desired setpoint (with small error and high gain).

For the proportional and the integral part, a commonly used value for α is around 0.5, which gives a nonlinear mapping between e and y shown in Figure 5. Compared with the linear function y = e, the nonlinear function $fal(x,\alpha,\delta)$ gives high gain for small error and small gain for large error. This strategy will help solve the integral wind-up problem, which is frequently faced in practice. The value selected for δ is related to system behavior and the noise levels where the equipment is installed. A recommended value for δ is around 50% of the set point value. If δ is set too low, the gain will become excessive, causing oscillations or complete loss of control. If δ is set too high, the system response will look similar to the traditional PID response, and the benefits of nonlinear control will be lost. The goal is to tune δ as small as possible without causing oscillation or instability.

The derivative part of the controller is designed to provide quick response to system transients, and minimize overshoot. Since the derivative part acts only during transient periods, it provides little action while the system has reached steady state. However, the differentiator is sensitive to system noise because noise is generally made up of very fast voltage spikes. Using a nonlinear controller allows us to minimize the differentiator output that is caused by noise, while maintaining the differentiator performance during system transients. The δ_d Setting is related to noise level and sample rate. Usually, we set $\delta_d > Vnoise_{p-p}/2h$, where h is the sample rate, and $Vnoise_{p-p}$ is the peak-to-peak voltage of the noise. So by choosing $\alpha_d > 1$, it makes the derivative gain small when error is small, and big when error is large.

In practice, most systems are nonlinear. Therefore, classical (linear) PID controllers are not the best choice for most control applications. One approach is to linearize the system dynamically, i.e., to approximate the nonlinear model by a series of linear ones. Based on the dynamically linearized models, several linearization-based PID controllers can be designed for various operating conditions. The local design data can then be interpolated to yield an overall nonlinear controller. This procedure is known as gain scheduling [7]. It is an intuitively appealing but heuristic process, which is used in a wide variety of control applications for nonlinear systems. The main goal of gain scheduling is to achieve high gain for small error and low gain for large error throughout the nonlinear region. The novel NPID controller uses an exponential function to implement this idea simply and systematically. The tuning of an NPID controller is similar to the tuning of a PID controller and can also be performed on-site for optimization. The simulation results in CHAPTER III show that an NPID controller for a

PWM DC/DC power converter (nonlinear system) achieves better performance than a classical PID controller. Improvements include increased robustness and an enhanced ability to deal with the large variations of line and load changes. A block diagram of the NPID controller is shown below.

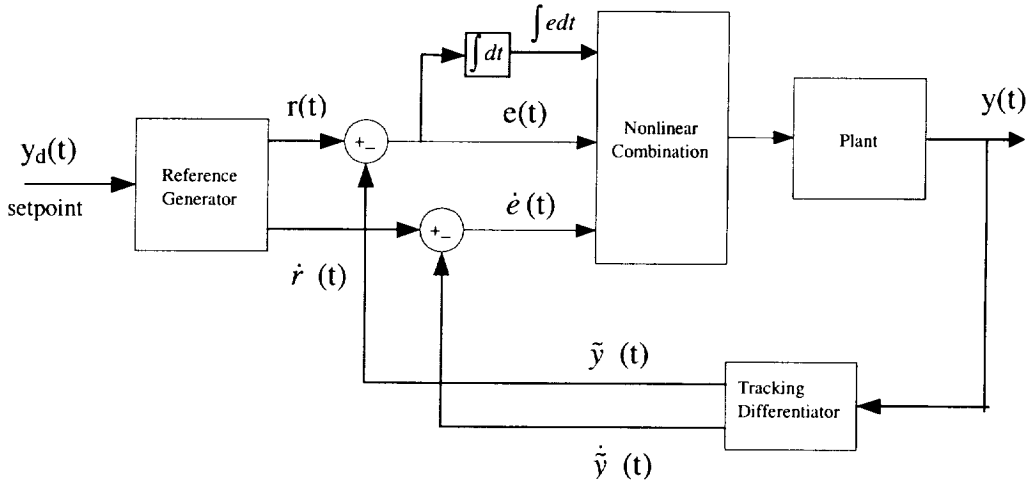


Figure 6 NPID configuration

The r(t) and $\dot{r}(t)$ terms are the desired trajectories for $y(t)$ and $\dot{y}(t)$ respectively. Here $\tilde{y}(t)$ and $\dot{\tilde{y}}(t)$ are the approximate trajectory and approximate derivative trajectory of the output signal $y(t)$ respectively. The Reference Generator idea comes from motion profile, which is regularly applied in the motion control industry. Such a method is implemented to provide better control of position, velocity, and acceleration. The tracking differentiator will be discussed later in this chapter.

2.2 The NPID Design

Based on the novel NPID method discussed in the previous section, an NPID controller for a PWM DC-DC Power Converter can now be designed. In this case, we use

a 2nd order transfer function (linear) model of the form shown in Eq.(2.4) to approximate the nonlinear power converter. The NPID controller will be derived based on the classical PID controller.

$$G_p(s) = \frac{l}{ms^2 + ns + 1} \tag{2.4}$$

2.2.1 Initial PID Tuning Parameters

Using a simplified 2nd order model, the PID controller parameters can be obtained by a direct calculation using pole placement. There are three parameters in the model Eq. (2.4). By using a PID controller, which also has three parameters, it is possible to arbitrarily place the three poles of the closed-loop system.

The approximate derivative term is implemented using a $s/(s/N+1)$ transfer function block. Making N large can cause the system to be more sensitive to noise. We can now design the controller based on its transfer function shown in Eq.(2.5).

$$\begin{aligned} G_c(s) &= K_p + \frac{K_i}{s} + \frac{K_d s}{s/N+1} \\ &= \frac{(NK_d/K_i + K_p/K_i)s^2 + (NK_p/K_i + 1)s + N}{s} \frac{K_i N}{s+N} \end{aligned} \tag{2.5}$$

By using the pole placement design concept, we can make

$$G_c(s)G_p(s) = k/s(s+N) \tag{2.6}$$

Figure 7 Closed-Loop configure

The closed loop transfer function is shown in Eq.(2.7).

$$G(s) = G_c(s)G_p(s)/(1+G_c(s)G_p(s)) = k/(s^2+Ns+k) \tag{2.7}$$

A suitable closed-loop control was designed to obtain a critically damped impulse response within a required settling time T_s. An approximation for the settling time that will be used is $T_s = 4/\xi\omega_n$. Here ξ is the damping ratio, and ω_n is the natural frequency. A more standard notation for the 2nd order transfer function is shown in Eq.(2.8).

$$G(s) = k/(s^2+Ns+k) = \omega_n^2/(s^2+2\xi\omega_n s+\omega_n^2) \tag{2.8}$$

For a critically damped output, we choose $\xi = 1$, and $\omega_n = 4/T_s$. To identify the coefficients k and N in Eq.(2.8), we can define them as $k = \omega_n^2$ and $N = 2\xi\omega_n = 2\omega_n$.

According to Eq's (2.4), (2.5), and (2.6), we obtain

$$G_c(s)G_p(s) = \frac{(K_d/K_i + K_p/K_i/N)s^2 + (K_p/K_i + 1/N)s + 1}{s} \frac{K_i N}{s+N} \frac{l}{ms^2+ns+1}$$

$$= \frac{k}{s(s+N)}$$

$$= \frac{\omega_n^2}{s(s+N)}$$

After simplifying the equation, we are left with

$$\frac{(K_d/K_i + K_p/K_i/N)s^2 + (K_p/K_i + 1/N)s + 1}{ms^2+ns+1} \times \frac{K_i N l}{\omega_n^2} = 1$$

So

$$K_i = \frac{\omega_n^2}{Nl} = \frac{\omega_n}{2l} \qquad \frac{K_d}{K_i} + \frac{K_p}{K_i}\frac{1}{2\omega_n} = m \qquad \frac{K_p}{K_i} + \frac{1}{2\omega_n} = n$$

Finally, we obtain

$$K_i = \frac{2}{lT_s} \qquad \frac{K_d}{K_i} = m + \frac{T_s^2}{64} - \frac{nT_s}{8} \qquad \frac{K_p}{K_i} = n - \frac{T_s}{8} \tag{2.9}$$

The k_p, k_i and k_d relationship shown in Eq.(2.9) can give us rough initial gain parameters for PID controller.

The values obtained for k_p, k_i and k_d during the fine-tuning process can be used as a starting point for tuning the NPID controller. If the $fal(x,\alpha,\delta)$ function parameter, α, is set equal to 1, then the nonlinear $fal(x,\alpha,\delta)$ function becomes equivalent to linear PID function. Therefore, as a starting point, α was set equal to 1 to confirm the previously obtained PID results.

2.2.2 NPID Design

In practice, a pure differentiator cannot be implemented because of its high sensitivity to noise. Many approximate differentiators are proposed, and in this case, we chose a Tracking Differentiator [7] for calculating the approximate derivative of the output voltage signal.

The NPID provides high gain when error is small yet offers good disturbance rejection and robustness by lowering the gain when the error is large. However, if α is set too high, the controller may become too sensitive to noise, making the system unstable.

For the proportional part of the NPID controller, we simplify the $fal(x,\alpha,\delta)$ function by representing it as a G-function, which is a nonlinear gain function made up of two linear regions shown in Figure 8. The straight-line represents a linear function and the curve represents the nonlinear G function, shown as the combination of two linear

regions. The design philosophy is fully explained in [12]. Here, the intuition is that the gain should be high when the error is small, making the controller "more stiff." That is, the proportional control is made more sensitive to the small errors. This will also reduce the dependency on the integral part of the control to eliminate steady state errors. Note that instability is often caused by the 90-degree phase lag in the integral control.

The mathematical description of the proportional part can be expressed in Eq. (2.10). As the progression from linear PID to NPID occurs, a good starting point for the proportional G-function parameters is to set $k_{p1} = 1$, $k_{p2} = 0.2$ and $\delta_p = 0.5$. The next step is to tune k_p as large as possible while reducing k_{p2} and δ_p as small as possible.

$$k_p \cdot G_p(e) \qquad G_p(e) = \begin{cases} k_{p2} \cdot e + (k_{p1} - k_{p2}) \cdot \delta_p \cdot \mathrm{sgn}(e) & |e| > \delta_p \\ k_{p1} \cdot e & |e| \le \delta_p \end{cases} \tag{2.10}$$

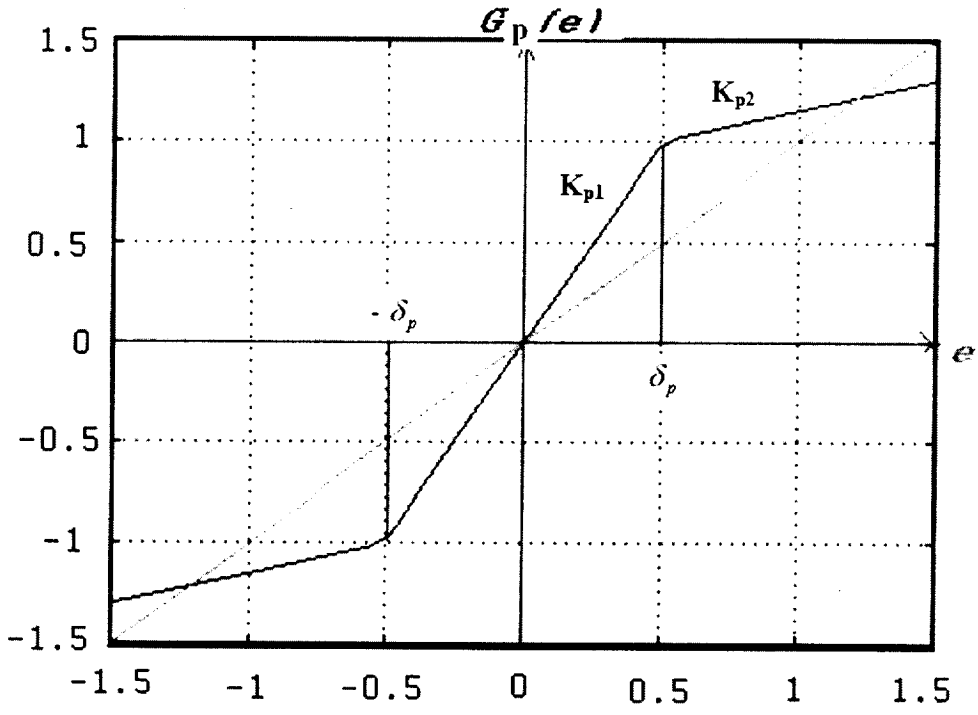


Figure 8 G-Function for proportional control

The main purpose for using integral control part is to eliminate the steady state error. Setting the integral gain too high can cause overshoot, wind-up and phase lag problems. A mathematical description of the integral part of the NPID controller is shown in Eq.(2.11). The corresponding G-function for the integral part of the controller is shown in Figure 9. The integrator is allowed to integrate only when the error is "small," typically when the output is within 10% of the set point. This design strategy allows the control to effectively avoid undesirable overshoots and integrator wind-up during transients. The $fal(x,\alpha,\delta)$ function has the ability to increase gain when error is small. However, these functions do not have the ability to limit the gain to zero beyond a specified error level. Therefore, the G-function for the integral part of the controller has been modified to eliminate any integral gain when the error is greater than 10% of the setpoint. The modified nonlinear G-function shown in Figure 9 can effectively solve this problem.

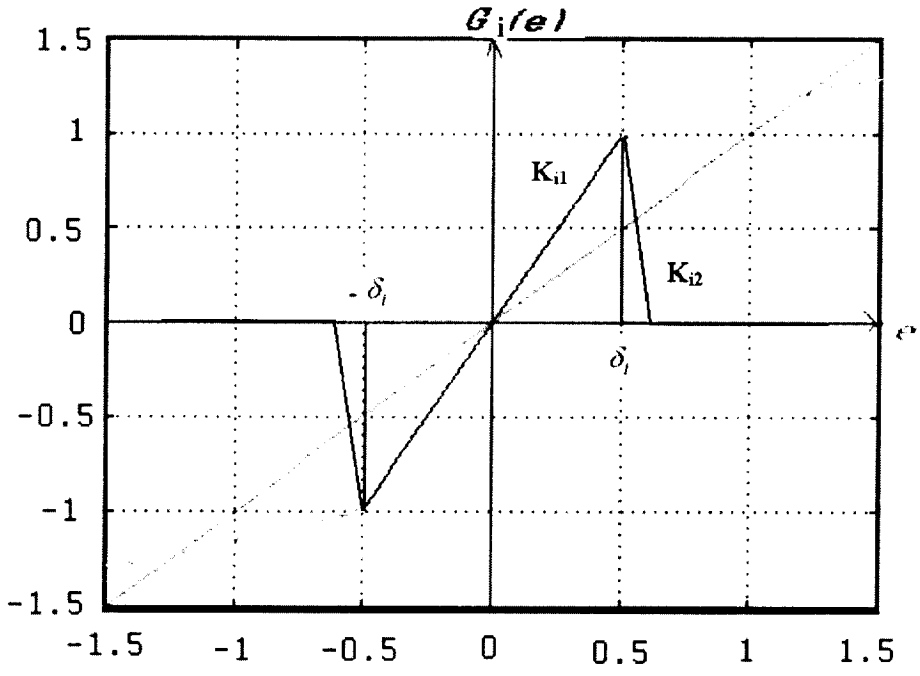


Figure 9 G-Function for integral control

$$\begin{matrix} k_i \cdot \int G_i(e)dt & G_i(e)\cdot e \geq 0 \\ k_i \cdot 0 & G_i(e)\cdot e < 0 \end{matrix} \qquad G_i(e) = \begin{cases} k_{i2}\cdot e + (k_{i1}-k_{i2})\cdot \delta_i \cdot \mathrm{sgn}(e) & |e| > \delta_i \\ k_{i1}\cdot e & |e| \leq \delta_i \end{cases} \tag{2.11}$$

As the integral part of the controller is improved from linear to nonlinear, a good starting point for the integral G-function parameters is to set $k_{i1} = 1$, k_{i2} = -5, and δ_i = 0.5. The next step is to tune k_i as large as possible and fine tune k_{i2} and δ_i as small as possible.

The G function and damping concept can be applied to simplify the derivative part of the controller [12]. The mathematical description of the nonlinear derivate part of the controller is shown in Eq. (2.12).

$$k_d \cdot G_d(-\dot{y}, \delta_d) \qquad G_d(-\dot{y}) = \begin{cases} k_{d2}\cdot(-\dot{y}) + (k_{d1}-k_{d2})\cdot\delta_d \cdot \mathrm{sgn}(-\dot{y}) & |\dot{y}| > \delta_d \\ k_{d1}\cdot(-\dot{y}) & |\dot{y}| \leq \delta_d \end{cases} \tag{2.12}$$

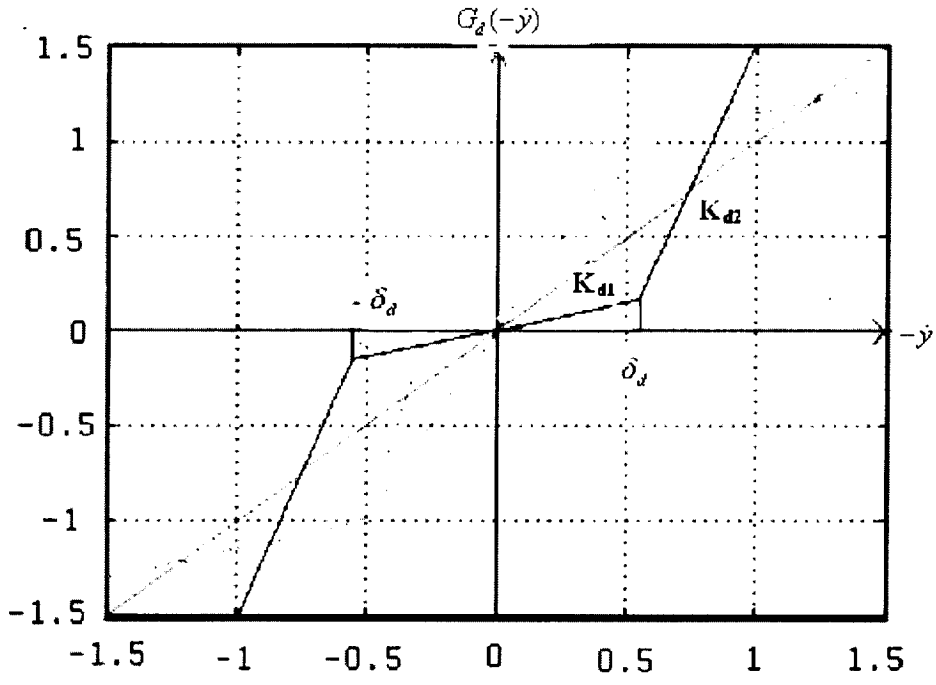


Figure 10 G-Function for derivative control

By applying a nonlinear gain, it is possible to increase the gain of the derivative part while greatly reducing its sensitivity to system noise. This means that the derivative control is only active during transient periods to prevent overshoot and oscillations. At the same time, it can almost ignore signal noise after the system has settled at the setpoint. As the controller was modified from linear PID to NPID, the starting values for the G-function were $k_{d2} = 1$, $k_{d1} = 0.2$ and $\delta_d = 70$. The next step was to tune k_d as large as possible while fine-tuning k_{d1} and δ_d as small as possible.

Another important transient response is the system start-up response. That is, the step response of the converter for a step in the setpoint, which occurs when the converter is started. Due to the initially large error, a technique is implemented to limit this error and bring the converter to steady state in a much more controlled manner. The method is known as a profile, and is very popular in the motion control industry. We call this method "soft start" because the setpoint is not a step. Rather, the step input is applied to a first order transfer function (lowpass filter) $1/(0.00025s+1)$, which has a settling time of about 10 ms. The output of this transfer function is used as the system setpoint. Once the signal settles, the setpoint will remain constant at its final value.

Since the Westinghouse converter does not respond symmetrically to a load increase or a load decrease, a separate control gain is used to optimize the response for each condition. This asymmetric control scheme is implemented for the proportional and integral parts of the controller only. Figure 11 shows the Westinghouse converter response for open loop load changes. The benefit is that the tuning can be conducted separately, one for load step-down or error greater than zero, the other for load step-up or error less than zero.

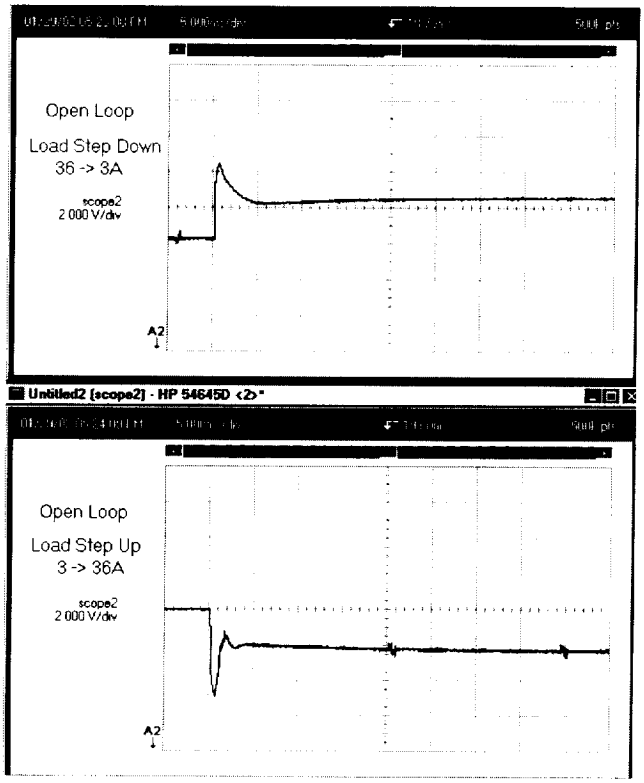


Figure 11 Output Voltages in Open Loop with Load Change

The asymmetric nonlinear control concepts employed above make tuning the positive and negative gains much easier for the proportional and integral parts of the NPID controller. The reason for this is because changing one parameter will not have an affect on the other. This improves the robustness of the controller by optimizing the controller's configuration for the specific asymmetric characteristics of the Westinghouse converter.

2.2.3 Approximate Differentiator - Tracking Differentiator

A nonlinear Tracking Differentiator (TD), also known as a "filter-differentiator," is proposed by J. Han [6] and is shown in Eq.(2.13).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -Msign(x_1 - v(t) + \dfrac{x_2 |x_2|}{2M}) \end{cases} \tag{2.13}$$

Where the output x_1, tracks the reference input $v(t)$ as quickly as possible without overshoot. The term M represents the maximum acceleration the system can obtain and is a function of the maximum actuation available in the system. The term x_2 is the approximate derivative of the signal $v(t)$.

It was shown [7] that $\forall\ \varepsilon > 0$ and $T > 0$ $\exists\ M_0 > 0$ such that if $M > M_0$ $\int_0^T |x_1(t) - v(t)| dt < \varepsilon$. The only design parameter of the filter is the gain M, which corresponds to the upper bound of acceleration assuming x_1 represents a position signal. Parameter x_1 can track $v(t)$ relatively fast as long as M can be chosen large. The parameter x_2 is the derivative of x_1 and therefore approximates the derivative of $v(t)$.

A discrete time realization of the TD (Eq. 2.14) was derived from Eq.(2.13).

$$\begin{cases} v_1(t+h) = v_1(t) + h\ v_2(t) \\ v_2(t+h) = v_2(t) + h\ fst_2(v_1(t), v_2(t), v(t), r, h) \end{cases} \tag{2.14}$$

The terms v_1 and v_2 are the state variables, where $v(t)$ is the input signal, h is the step size and the function fst_2 is defined as:

$$d = r\,h\ ;\ d_0 = d\,h;\ y = v_1 - v + h\,v_2$$

$$a_0 = \sqrt{d^2 + 8\,r\,|y|} \tag{2.15}$$

$$a = \begin{cases} v_2 + \dfrac{y}{h}, |y| < d_0 \\ v_2 + \dfrac{sign(y)(a_0 - d)}{2}, |y| \geq d_0 \end{cases} \tag{2.16}$$

$$fst_2 = \begin{cases} -r\dfrac{a}{d}, |a| \le d \\ -r \cdot sign(a), |a| > d \end{cases} \tag{2.17}$$

The impact of the TD on the system is profound. First, as a noise filter, it blocks any part of the signal with acceleration rates exceeding M. In practice, we often know the physical boundary of a signal in terms of its acceleration rate. This knowledge can be conveniently incorporated into the TD to reject noise based on the understanding of the physics of the plant. On the other hand, the traditional linear filter can only attenuate noises based on its frequency contents. In addition, it is shown [10]that a filter TD also has very desirable frequency response characteristics. In particular, it has a much smaller phase shift compared to linear filters, while maintaining an extremely flat gain within the pass band.

Finally, perhaps the most important role of the TD is its ability to obtain the derivative of a noisy signal with a good signal to noise ratio. It is well known that a pure differentiator is not physically realizable. The error is often not differentiable in practice due to the noise in the feedback signal. This explains why the PID controller is often reduced to a PI controller in most industrial control applications. The use of the "D" part has been quite limited due to the extreme amplification of noise by differentiation, or its approximations. This noise problem is resolved using a TD because x_2 is obtained via integration. M is the only parameter needed to tune the TD, which simplifies the tuning process.

2.3 NPID Control Stability

The introduction of the nonlinear gain to the control algorithm makes it difficult to perform stability analysis. Due to the lack of effective mathematical analysis tools, the close-loop system's stability will be using simulation and experimentation. The simulation was performed using MATLAB® Simulink and the controller were implemented using the digital control module based on the DSP from Texas Instruments.

CHAPTER III

SOFTWARE SIMULATION

In this chapter, the approximate 2nd order linear model (transfer function) is briefly introduced. The simulation set-up of the PI and NPID controllers is described and the results are compared. The controller response to line and load disturbances are also analyzed in this simulation. The simulation and implementation results both show that the NPID controller performs better than the previously developed PI controller does. Therefore, it would be advantageous to implement an NPID controller rather than a PI controller since the NPID performance exceeds that of the PI controller. Implementation issues for the NPID controller are discussed in the following chapter.

3.1 Model Description of the PWM DC-DC Power Converter

3.1.1 Derivation of Plant and Line/Load Disturbance Transfer Functions

An ED408043-1 Westinghouse DC-DC switching power converter was used for this control research. The simulation uses a 2nd order linear transfer function as the model of the Westinghouse power converter. The derivation of this model can be found in [2], and is simply presented in Figure 12, followed by a brief explanation.

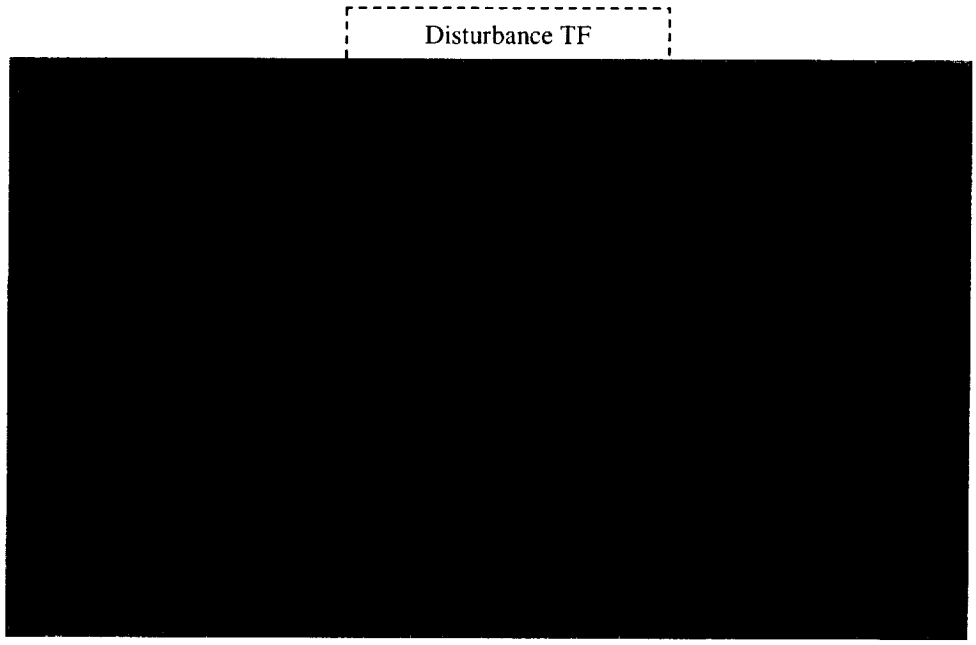


Figure 12 Linear Model of Full-Bridge Converter

The transfer function shown in Figure 12 was obtained from the following converter parameters and waveforms: DC voltage gain, rise time, percent overshoot, damping factor, and total length (in seconds) of the transient. Once collected, this data

was used to establish the transfer function in Figure 12. This linear model can be used for both steady state and transient simulation. The nominal test settings used as a baseline for collecting characteristic converter data was: 120 V input, 4 A load current, and 72.66% duty ratio (186 Pulse Counts). The pulse count ranges from 0 – 256 due to the eight-bit quantization of the hardware implementation. The characteristics of the simulated transient output voltage waveforms were compared to the characteristics of the experimental transient output voltage waveforms to ensure that the responses correlated.

The size and behavior of the transient in the output voltage waveform was dependent upon the magnitude of the line or load step. To totally characterize the operation of the W.S.P.C., the magnitude of the step was changed by ±5% (±9 pulse counts), ±10% (±18 pulse counts), and ±15% (±28 pulse counts) at a constant load current of 4A. The value of 186 pulse counts was used for a nominal output voltage of 28 V with a 4 A load and a 120 V input voltage.

To determine the line change transfer function, the duty ratio was held constant at 186 with a constant load current of 4 A while step changes in magnitude of ±6 (5%), ± 12 (10%), and ± 18 (15%) volts were made in the W.S.P.C.'s supply voltage. The size of the load current had no effect on the transfer function.

To determine the load current change transfer function, the duty ratio was held constant at 186 with a constant supply voltage of 120Vdc. Step changes of ± 2 (5%), ± 4 (10%), and ± 6 (15%) amps were made in the load current. The supply voltage had a negligible effect on the size or shape of the transient produced by the W.S.P.C.'s output voltage.

3.1.2 Discretization, quantization, and normalization

The sample and quantization Simulink Blocks shown in Figure 13 are used to simulate the sampling of an analog signal in a digital system.

Figure 13 Sample-Quantizing-Normalizing Simulink Block

In this research, DSPACE DS2001 A/D (Analog-to-Digital) converter is used to sample the feedback voltage. The signals used in the description of digital control systems are called discrete-time signals. Discrete-time signals are defined only for discrete instants of time, usually at evenly spaced time steps. Discrete-time computer-generated signals have discrete (or quantized) amplitudes and thus attain only discrete values. The A/D converter samples the analog signal and produces an equivalent binary representation of signal. In general, an n-bit binary code can represent only 2^n discrete values. In this case, we set n equal to 16 and the analog input ranges from –10 to 10 V. This sets the quantization interval for this feedback signal is equal to 0.00031 Volts/bit, which is the same as the simulation setup in Figure 13.

Normally A/D converters can sample analog signals at a very high sample rate. However, most CPUs or processors cannot use the sampled data to complete a control algorithm at the same high speed. Therefore, a periodic time interval is allowed for the sample and control algorithm to complete, which is called the step-size. This block of time is related to the CPU interrupt timer setup. The interrupt mechanism is a hardware-

based task scheduler, which will trigger after a user-defined time interval. After the trigger event, the CPU will execute an interrupt task. In this case, the interrupt task involves clearing the collected sample data and running the control algorithm. Therefore, the user must make sure that the sample data collection and control algorithm can be completed within one step-cycle. We must keep in mind that as the step-size is decreased, the performance of the digital controller usually improves. Here we choose a step-size of 50us, which is denoted as h in the Zero-Order Hold simulation block shown in Figure 13. Therefore, the term h must be set equal to 5e-5 before running the simulation.

The setpoint and feedback voltages are normalized to allow the user to define various setpoints. This is especially important for the Nonlinear PID controller since the nonlinear region is related to the setpoint. After normalizing, we can keep the same nonlinear region settings, even for different setpoints. The error obtained from the two normalized signals will allow the control output to be the same for different setpoints. Since the control output determines the plant output, a scaling factor should be applied to the control output. The scaling factor is equal to the (desired set point / tuning set point). The desired setpoint is the current setpoint that is desired by the user, and the tuning setpoint is the setpoint for which the controller gains were tuned. In the case of the Westinghouse converter, the tuning setpoint is 28 Volts DC. The desired setpoint, which is denoted as n, is shown in the simulation setup in Figure 13.

3.1.3 PWM Generator Input

The PWM generator is controlled via four 8-bit registers. Two registers hold the PWM duty cycle for each of the two phase-locked channels. The third register holds the

divisor for the modulo-n clock generator circuit. The fourth register is reserved for future expansion and may be used in part to increase the PWM resolution.

The additional expense of utilizing a CPLD in this application is easy to justify. Most importantly, the architecture of the CPLD-based PWM generator ensures that the PWM signals continue in the event of a DSP software failure. Furthermore, the CPLD approach provides hardware-guaranteed phase lock between the two PWM signals. Finally, the CPLD based PWM generator relieves the DSP of potentially significant processor bandwidth demands. [13] The simulation model of the Controller/PWM Generator Interface is represented in Figure 14.

Figure 14 Controller/PWM Generator Interface

The "In" terminal in Figure 14 is connected to the controller output, and the "Out" terminal is sent to the PWM generator (CPLD). The controller output is duty ratio, which cannot be sent directly to the PWM generator. The Pulse Quantizer block has a quantization step level of 1. The 8-bit resolution of the PWM generator also determines the maximum pulse count, which is 256. Therefore, the exact pulse count number that is sent to the PWM generator is equal to the duty ratio multiplied by 256.

It is dangerous to let all MOSFET switches turn on at same time since damaging levels of current could flow through one or both sides of the h-bridge, destroying the MOSFET switches and their gate drivers. Therefore, saturation is applied to the duty

cycle to ensure that all MOSFETs will not be on at the same time. The maximum duty cycle produced by the PWM Generator will be 240 counts out of a possible 256. This provides a time interval of about 1.56us (50us*(1-240/256)/2) known as deadband, where all four MOSFETs will be turned off. The factor n/28 was discussed in the previous section, and is designed to allow the user to easily adjust the setpoint without having to retune the controller.

3.1.4 NPID Controller

Based on the practical NPID design principle that was discussed in chapter II, a Simulink model of the NPID controller is shown in Figure 15.

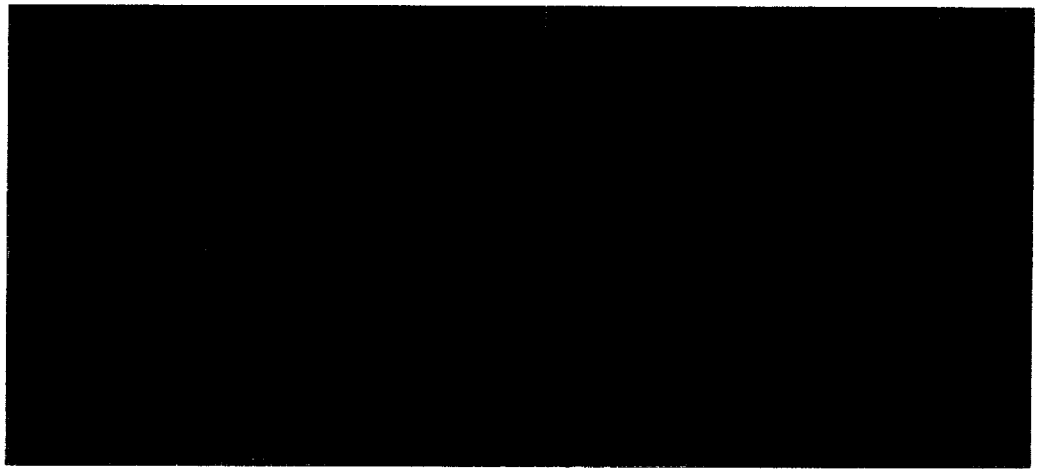


Figure 15 NPID controller Simulink Block

The "setpoint" is the normalized desired converter voltage, and the "feedback" signal is the normalized signal from the output of the converter. The NPID G function was built using a Matlab S-function, which was named Gfunc.c (Appendix B). The tracking differentiator has two outputs. The upper output is the approximate converter voltage, which is not used directly, and the lower output is the approximate derivative of the converter voltage, which is used in the derivative part of the controller. The TD was

built using a Matlab S-function, which was named tdfst2d.c (Appendix A). Figure 16 shows the Simulink block diagram setup for the PI and NPID controllers.

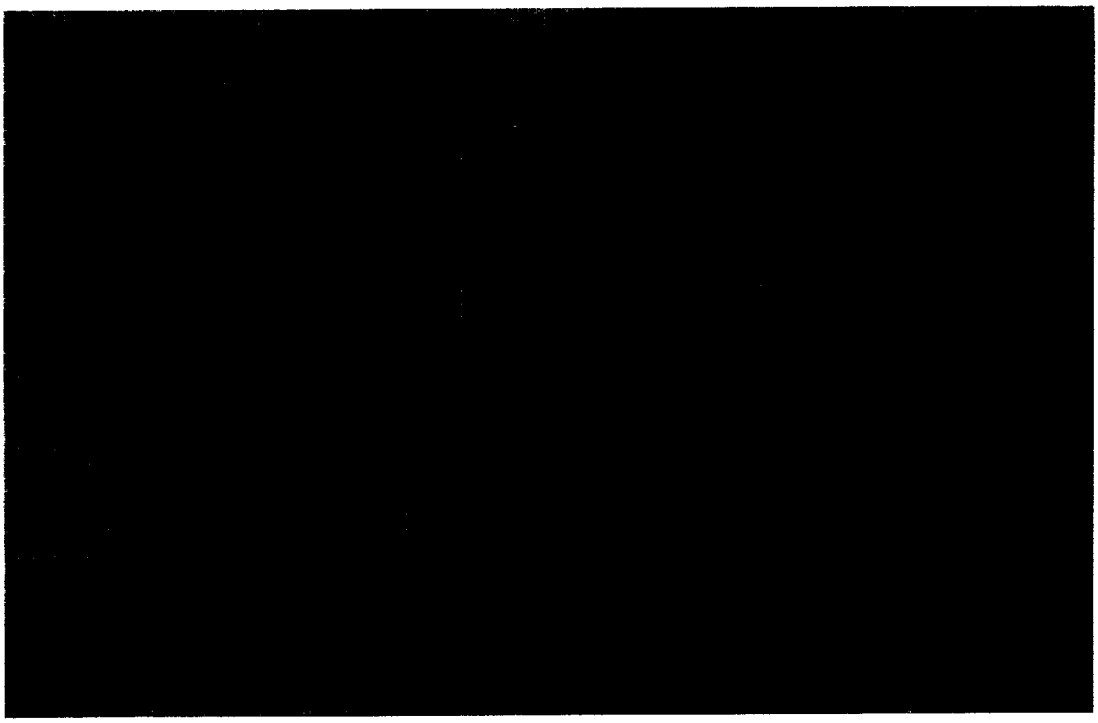


Figure 16 PI/NPID Simulink Blocks

3.1.5 PI/NPID Controller Tuning

Based on equations (2.4) through (2.9), the following values were obtained:

$$l = 0.1569 \quad m = 2.21e-7 \quad n = 7.52e-4$$

Since the setpoint and feedback voltages were previously normalized for the purpose of allowing the user to input various setpoints, the controller must reverse this process so that a correct control output can be obtained. Therefore, the control output must be multiplied by 28 to reverse the previous normalization process. At this point the control signal is given in duty ratio, meaning that the control output is a percentage value. Therefore, the signal must be multiplied by the maximum resolution of the PWM

generator. For example, in this simulation, the PWM resolution is 256 (8-bit). This means that the final output of the controller is in pulse counts, which can be implemented by the PWM generator. If the desired settling time $T_s = 2ms$, we obtain the following values:

$$k_i = \frac{2}{0.1569 \times 0.002} 28/256 \approx 600 \qquad \frac{k_d}{k_i} \approx 2e-7 \qquad \frac{k_p}{k_i} \approx 7e-4$$

Therefore, the initial PID gain parameters were set to $k_p = 0.4 \quad k_i = 600 \quad k_d = 1e10^{-4}$.

Before the tuning process begins, the soft start feature must be disabled. The soft start is applied to avoid possible damage or overshoot caused by large control action. This simulation was designed to give a side-by-side comparison of the PI and NPID controllers. The PI controller is simulated rather than a PID controller so that a comparison can be made with a previously developed PI controller. The main goal for tuning is to obtain a fast transient response with no overshoot, and less than 1% steady state error. System noise is an important consideration when tuning the controller. After manually tuning the controller for the Westinghouse power converter, control gains were found as shown below:

PI: $\quad k_p = 0.5 \qquad k_i = 800$

NPID: $\begin{bmatrix} k_p = 4 & \delta_p = 0.4 \\ k_{p1} = 1 & k_{p2} = 0.2 \end{bmatrix} \begin{bmatrix} k_i = 3800 & \delta_i = 0.5 \\ k_{i1} = 1 & k_{i2} = -5 \end{bmatrix} \begin{bmatrix} k_d = 1\text{e-}3 & \delta_d = 200 \\ k_{d1} = 0.1 & k_{d2} = 1 \end{bmatrix}$

3.2 Simulation Results

3.2.1 Steady State Performance

Figure 17 shows the simulation results of the PI and NPID controllers for a startup condition. It was observed that the NPID controller performed better than the PI

controller for following the soft start profile. Although the NPID response appears to be much closer to the desired setpoint, both the PI and NPID controllers were able to stabilize near the final desired value. However, once again the NPID performance exceeded the PI performance for providing the smallest steady state error. The PI steady state error value was measured to be ± 12mV, and the NPID steady state error was measured to be ± 4mV. Therefore, the PI steady state error is 3 times larger than the NPID steady state error. Another parameter of considerable importance is overshoot. It is apparent from Figure 17 that neither waveform exhibits any overshoot during startup. The settling time for the PI controller is 11ms, while the settling time for the NPID: controller is 10ms, making the PI controller 10% slower than the NPID controller. Finally, one of the most noticeable differences between the two waveforms is the shape. The NPID waveform smoothly follows the desired value, while the PI waveform seems erratic.

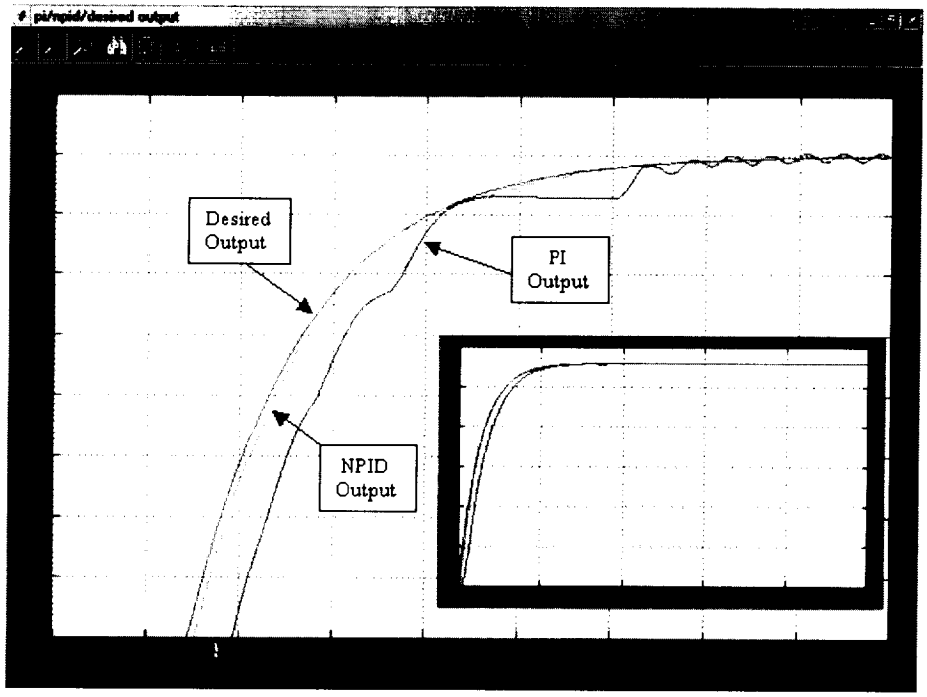


Figure 17 PI, NPID and Desired Output Voltages during Startup Transient

3.2.2 Line Disturbance Performance

Figure 18 shows the simulation results of the PI and NPID controllers under line disturbance. The supply voltage is increased by 30 volts from its nominal value of 120 Volts and nominal load current of 4 amps. The line disturbance is applied at 0.03s. Therefore, the maximum deviation from the setpoint was found to be 320mV for the PI controller and 70mV for the NPID controller. This means that the difference between the PI and NPID controllers is 250mV, which corresponds to a 360% difference in amplitude. Also the NPID controller produces a system response that is much cleaner than the PI controller produces. We do not evaluate recovery time here, since the supply voltage transient time is very long (about 100ms) compared to the normal converter transient response. In either case, the maximum voltage deviation is smaller than 2% of the desired output voltage, which is within the regulation specifications of the converter.

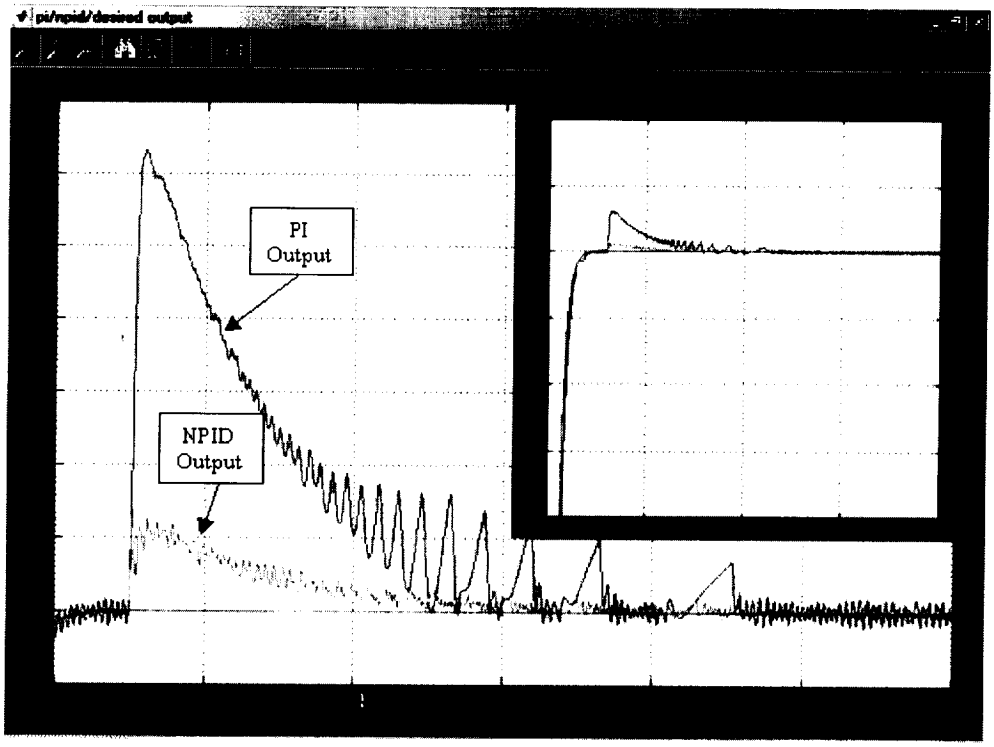


Figure 18 PI and NPID Output Line Disturbance Comparison

3.2.3 Load Disturbance Performance

Figure 19 shows the simulation results of the PI and NPID controllers during a load step-up. The figure shows simulation results for a load increase from 4 Amps to 36 Amps with a nominal supply voltage of 120 Volts. The load disturbance is applied at the 0.03s point on the graph in Figure 19. The maximum deviation from the setpoint is 970mV for the PI controller and 150mV for the NPID controller. This translates into a 550% improvement over the previously developed PI controller. The recovery time for the PI controller is 6ms where the NPID recovery time is 5ms. This corresponds to a 20% improvement over the PI controller. Based on these results, it is obvious that the NPID performance exceeds that of the previously developed PI controller.

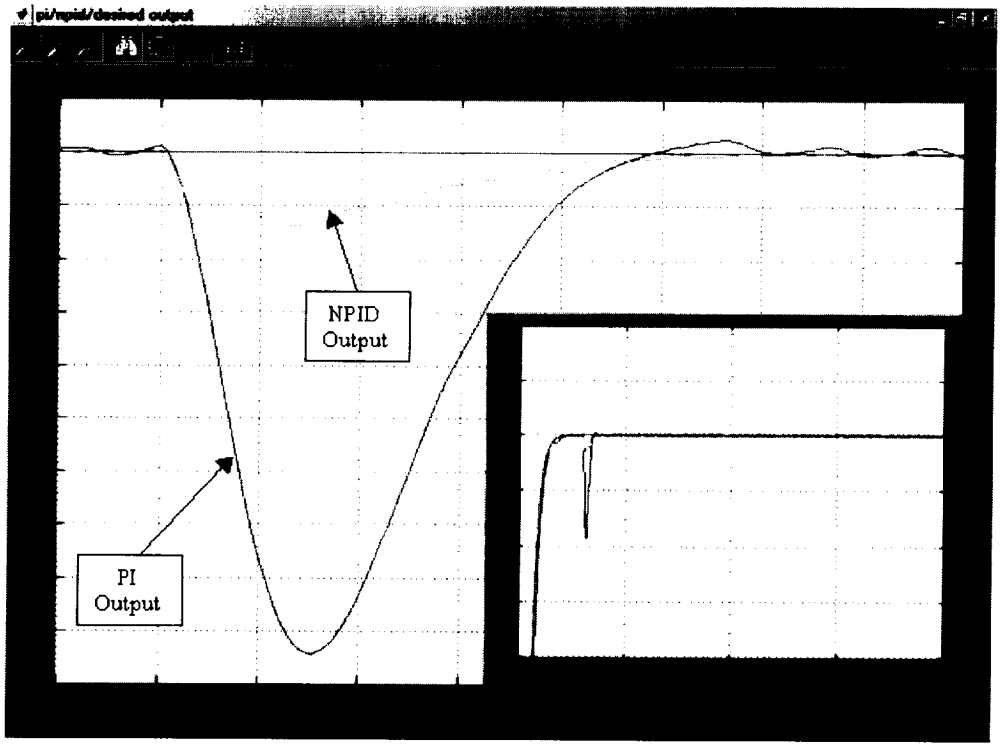


Figure 19 PI and NPID Output Load Step-Up Comparison

3.2.4 Dual Disturbance Performance

Figure 20 shows the simulation results of the PI and NPID controllers under dual disturbance conditions. The load current was increased by 32 Amps from its nominal value of 4 Amps, while the supply voltage was decreased by 20 Volts from its nominal supply voltage of 120 Volts. This corresponds to a worst-case disturbance condition. Both line and load disturbances were applied at the 0.03s point. The maximum deviation from set point for the PI controller was 1150mV, while the maximum deviation from the setpoint for the NPID controller was 180mV. Again the NPID controller exhibited superior performance over the PI controller, resulting in a 540% improvement in maximum voltage deviation. In addition, the NPID output presents a much smoother response than the PI output. Recovery time is again neglected since the voltage supply settling time is very long (about 100ms).

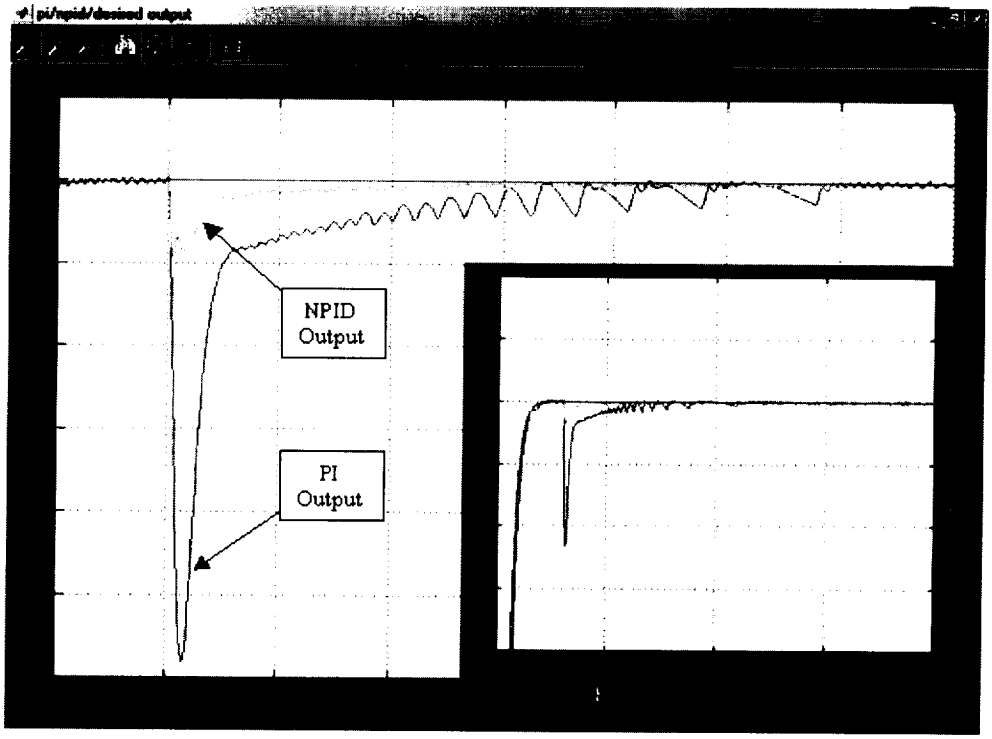


Figure 20 PI and NPID Output dual disturbance Comparison

3.2.5 Noise Response

Figure 21 shows the simulation results of the PI and NPID controllers under noise disturbance. The load current was set to a nominal value of 4 Amps, while the supply voltage was set to a nominal value of 120 Volts. The noise power was set to 0.00000015 to achieve noise levels similar to those measured in the lab. Looking at the output voltage waveforms with the added system noise, there seems to be little difference in either one of the systems. However, this is not the case for each of the controller output signals. The peak-to-peak PWM count was 2 for the PI controller, and 16 for the NPID controller, which corresponds to a 700% increase in performance from the PI to NPID controller. The NPID controller exhibits higher gain than the PI controller in the small error region, which causes the NPID controller to be more sensitive to voltage fluctuations.

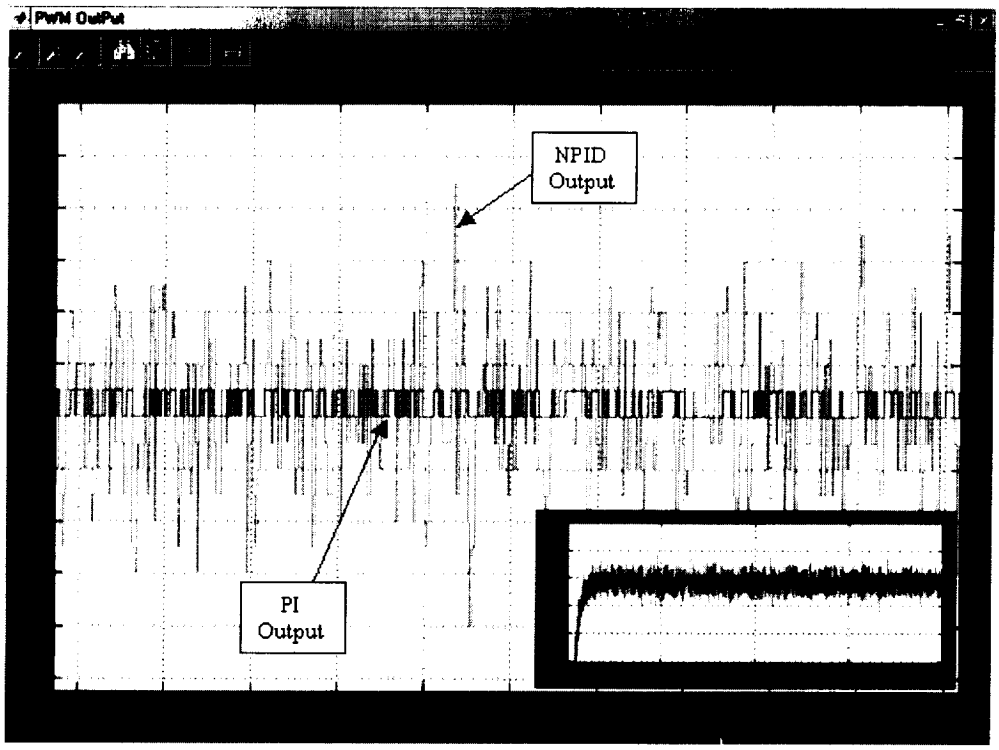


Figure 21 PI and NPID PWM Output Noise Response Comparison

3.3 Summary

The new nonlinear concepts employed in this research make it easier to tune the controller for a wide variety of applications. Compared to the previously developed PI controller, the NPID controller gain allows the user to optimize the performance of the system based on the additional gain parameters. This improves the robustness of the controller and allows the controller to be used in a number of different applications.

In real time, the high load system noise is far greater than the low load system noise. Therefore, the gain must be decreased so that the controller will remain stable for the maximum noise condition. Alternatively, by separating the gains for load increase and load decrease conditions, the converter performance can be optimized for the load removal condition where there is less noise. The digital controller makes it easy to implement this dual gain control system. Therefore, if the load steps down from 36A to 3A, then the recovery time will be fast due to the higher gain settings for this low noise condition. By separating the gain settings for step-up and step-down conditions, a desirable response can be obtained for both conditions.

CHAPTER IV

IMPLEMENTATION

This chapter describes the hardware implementation of the digital controller for dSPACE and the Stackable DSP System. Both hardware and software aspects will be discussed in this chapter. The hardware set up and calibration will be described and discussed in great detail. For the software aspect, flow charts will be used to help describe the operation of the native code. Finally, the noise issue encountered during implementation will be discussed.

4.1 Westinghouse DC-DC Switching Power Converter

The full bridge full wave power converter schematic has been discussed in CHAPTER I. Implementation of the switch-gear is focused on in this section. The switches (power MOSFETs) are controlled by the gate driver board, which is controlled by the CPLD (PWM generator). The specific MOSFETs used in this design were HEXFET ® Power MOSFETs, made by International Rectifier (IR), Model FA57SA50LC. It is a Third Generation HEXFET, which can provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance, low gate charge and cost-effectiveness. Some important specifications are: Drain-to-Source Breakdown Voltage - $V_{DSS} > 500V$, Static Drain-to-Source On-Resistance R_{DS} (on) < 0.08Ω, Continuous Drain Current, $I_D < 36A$ @ $T_C = 100℃$. The Westinghouse power converter unit requires a Drain-to-Source Breakdown Voltage > 180V and Continuous Drain Current > 17A. The FA57SA50LC power MOSFET that was selected for this application can safely achieve this requirement.

The MOSFET gate driver board contains four Integrate Circuits (IC). Two IC's are opto-couplers while the other two are gate driver IC's. The opto-couplers isolate the digital signal from the CPLD and provide an output signal that connects to the corresponding gate driver IC's. Each gate driver IC can drive one low-side MOSFET (common ground), and one high-side MOSFET (floating ground). Therefore, a total of four MOSFET's (2 high-side, and two low-side) can be driven by the gate driver board. The gate driver circuit is shown in Figure 22.

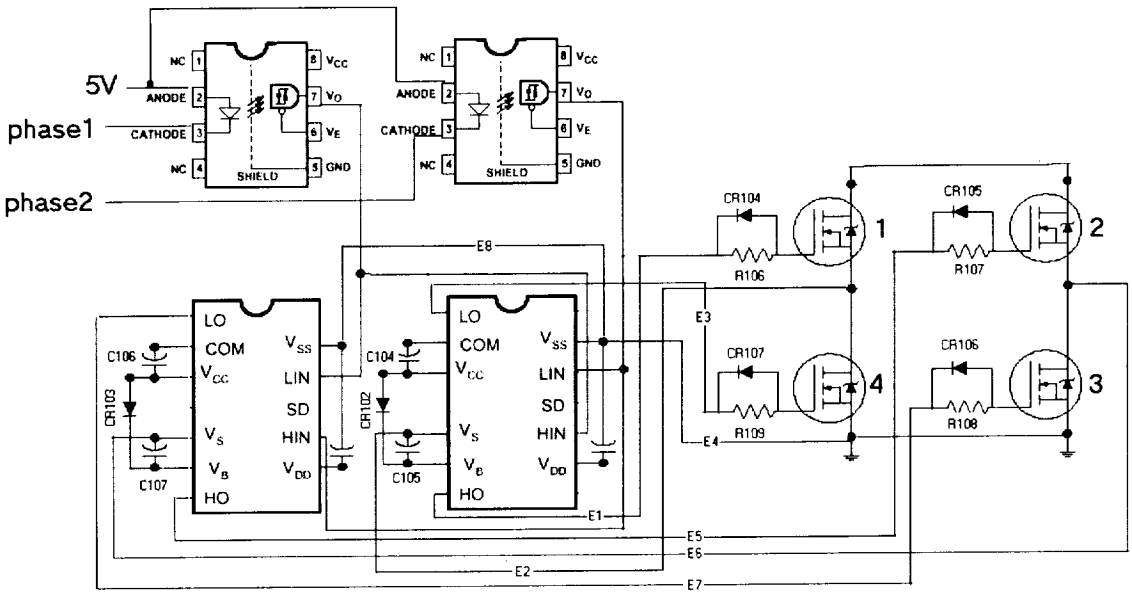


Figure 22 MOSFETS driver board Circuit Diagram

A low input current logic gate optocoupler was chosen that was manufactured by HP, with model number HCPL-2200. This IC has a combined GaAsP LED with integrated high gain photo detector. The detector has a tri-state output stage and has a detector threshold with hysteresis. The tri-state output eliminates the need for a pull-up resistor and allows for direct drive of data busses. The hysteresis provides differential mode noise immunity and eliminates the potential for output signal chatter. The optocoupler used here is to isolate the low voltage (5V) digital circuit and the high voltage (150V) analog circuit for protection purposes.

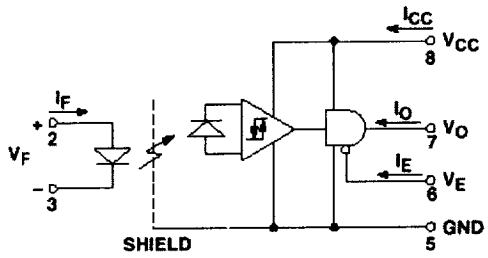


Figure 23 low input current logic gate optocoupler Schematic

The high / low side MOSFET driver used was manufactured by International Rectifier and has the part number IR2113. This is a high voltage, high speed, power MOSFET driver with independent high and low side referenced output channels. The floating channel (HO) can be used to drive an N-channel power MOSFET in the high side configuration, which operates up to 600 volts. Delay matching time (HS & LS turn-on/off) is less than 10 ns.

In Figure 22 a capacitor across the V_B and V_S pins is used to supply gate drive current to the high-side MOSFET. The difference of HO and V_S drives the gate of high side MOSFET. When high side drive is on, the high side MOSFET will be on, then current will go through the MOSFET and cause V_S to go up to around 120V. Without a capacitor across the V_B and V_S pins, the V_B potential will be limited under 15V, so it is impossible to drive the gate of the MOSFET when V_S goes up to 15V gradually. The capacitor just makes the HO node floating when V_S is increasing, so that there is positive voltage cross gate-to-source of high side MOSFET.

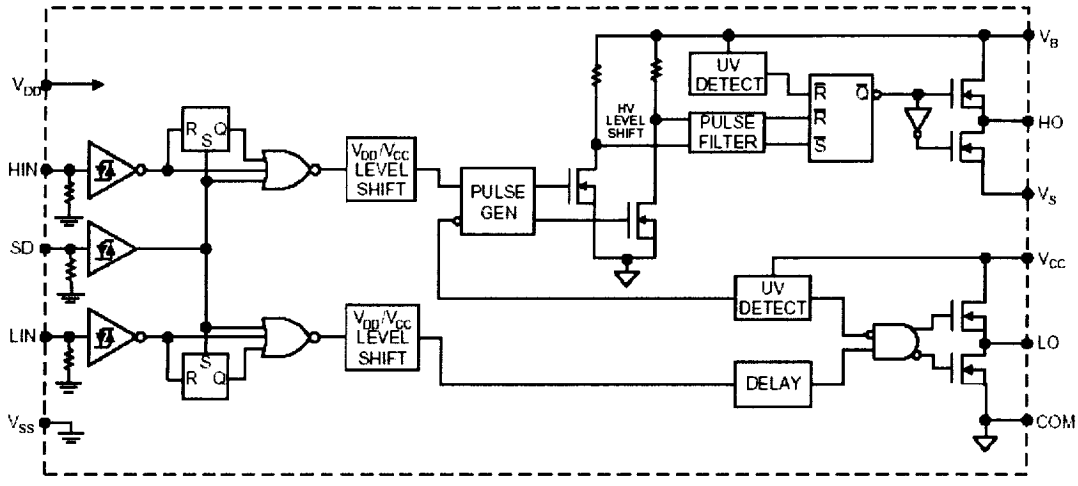


Figure 24 IR2113 Function Block Diagram

4.2 Signal Conditioning Board

The signal conditioning board was designed to scale-down the converter output voltage, isolate the power convert from the DSP board, and filter signal noise to prevent aliasing from the A/D converter.

Figure 25 Single Conditioning Board top view

The main device on board is a large isolation amplifier, AD210, from Analog Devices. The AD210 provides complete isolation with both signal and power isolation supplied via transformer coupling internal to the module. The true three-port design structure permits the AD210 to be applied as an input or output isolator, in single or multichannel applications. In addition, the AD210 provides protection from fault conditions that may cause damage to other sections of a measurement system. The Input/Output voltage Range is ±10 V. This output amplitude can be sent directly to the dSPACE DS2001ACD board after low pass filter occurs. For the stackable DSP system, the ADC board has a 50Ω input impedance and ±1 V input range. Therefore, an additional circuit was added to scale down the 10V signal and provide enough power to drive the 50Ω-terminating resistor. This final scale-down factor for the Stackable DSP

System is about 10:1. Since the power converter output voltage can be up to 40V for 120Vdc input, the first stage scale down factor before the isolation amplifier is around 3:1. This limits the 40V output signal to 10V to avoid damaging the isolation amplifier. The higher the ADC input voltage, the greater signal-to-noise ratio.

4.3 PWM Generator

The PWM Generator generates pulse signals according to the binary data (pulse count) received from the DSP. The pulse count is equal to the duty ratio multiplied by the maximum count of the PWM generator in certain frequency settings. The CPLD used for this system is Altera's EPM7128SLC84-6, which was programmed in VHDL. The working clock is 100MHz.

The CPLD-based PWM generator ensures that the PWM signals continue in the event of a DSP software failure. Under this condition, the PWM duty cycles will not change significantly and the converter will continue to provide a voltage output near the specified value. This allows the DSP to restart the control program without necessarily disrupting the power conversion process. Thus the PWM generator is called a fault tolerant unit.

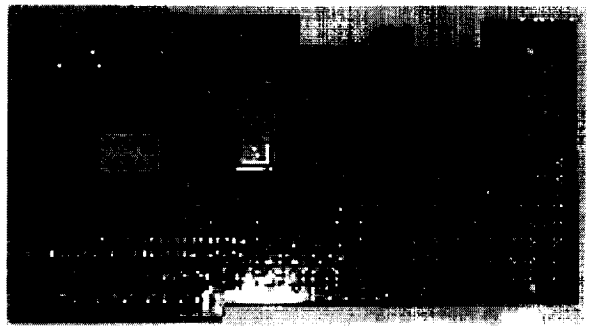


Figure 26 PWM Generator Board Top View

4.4 HP Electronic Load and Sorensen Power Supply

In order to conduct realistic experimental research, NASA provided to Cleveland State Unicersity an HP Electronic Load (6050A) and Sorensen Power Supply (DHP150-20 M9D).

The DHP series DC power supplies provide a wide range of power levels. Individual models provide DC outputs from 2 kW to 30 kW. The DHP modular design allows for easy system upgrades or field repair. All models incorporate self-diagnostics that continuously monitor critical parameters to assure maximum system uptime. For ease of use, the DHP has a front panel keypad control and standard analog remote programming as well as an optional remote digital IEEE-488.2, RS232 and Isolated Analog Input. Extensive programming capability allows storing individual auto sequences, voltage and current settings.

The HP 6050A Multiple Input Electronic Load Mainframes are used for design, manufacturing, and evaluation of dc power supplies, batteries, and power components. The mainframe contains six slots for load modules. The mainframe can dissipate up to 300 watts per slot, to a total of 1800 watts for a fully loaded mainframe. An individual module may have either 1 or 2 channels, each of which has its own channel number. Each module contains its own input connectors. The mainframe contains a processor, GPIB connector and interface circuits, trigger circuits, front-panel keypad and display, and other circuits common to all the load modules.

These two intelligent units are set up to communicate with a computer via the GPIB cable (IEEE-488.2), allowing full programmable remote control and monitoring.

The SCPI command set is used for programming the units. The SCPI supports the IEEE-488.3 status reporting data structures. These structures are comprised of status registers and status registers enable mask pairs.

In order to identify the hardware interface configuration, power up the equipment and run the HP IO Visa Assistant on the computer. Next choose the GPIB port on the left side of the window, which is desired to identify. Next click on the tab labeled "Formatted IO", and choose the instruction language SCPI at the right bottom corner. Finally press "*IDN?" button and the information of the equipment connected on this port will show up in the window. The GPIB address will be used in programming remote control and monitoring software.

Figure 27 shows the test control panel for the supply and load units. The main source code is attached in APPENDIX E.

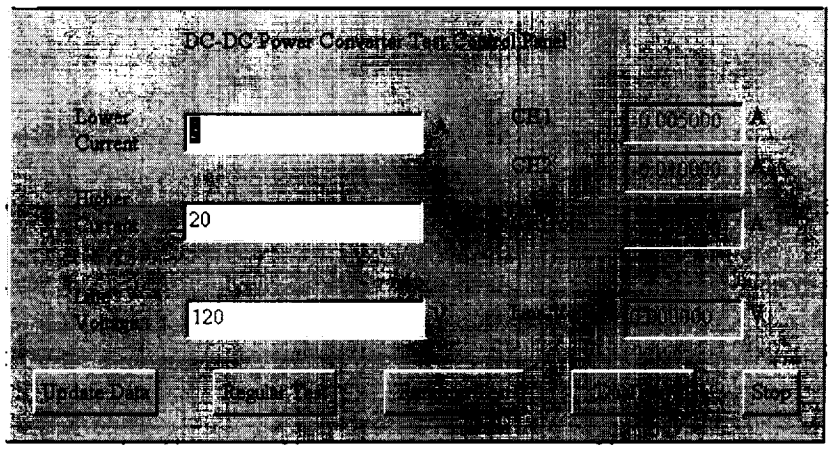


Figure 27 Test Control Panel

4.5 The dSPACE Hardware/Software Setup

dSPACE is the worldwide leading supplier of solutions for development processes with real-time systems for rapid control prototyping, production code

generation, and hardware-in-the-loop testing. dSPACE systems enable manufacturers of controllers to reduce their development times and costs dramatically and to noticeably increase their product quality. The hardware includes DS1003 DSP Board, ds2001 High-Speed A/D Board, and ds4002 Timing and Digital I/O Board. Experiment software used was ControlDesk Version1.1.

DS1003 DSP Board

The DS1003 DSP Board is the core of dSPACE's modular real-time systems for rapid control prototyping and hardware-in-the-loop simulation. It uses a TMS320C40, 60 MHz, TI DSP. The DS1003 allows a great deal of flexibility. The on-board PHS (Peripheral High Speed)-bus interface allows the user to have access to the entire range of dSPACE I/O boards. This means that the user can adapt the system's I/O precisely to specific application needs. If more computing power is required at any time, simply connect further DSP or Alpha boards. Increased memory requirements can be met with up to 3 MWords SRAM on board. Programming the DS1003 board is easy with Simulink and dSPACE's Real-Time Interface. You can add and configure all I/O boards connected to the DS1003 within the Simulink environment without programming one line of code within the Simulink environment. Generating code, compiling and downloading it to the board is reduced to a single mouse click. For those programming directly in C or using code from other sources, basic C functions for initialization and I/O access are included. Debugger, compiler, and loader software help the user to implement the code on the DS1003 board.

For the complicated NPID control algorithm, which requires a fast update time (around 20KHz), we can program directly in native C for highly efficient code execution.

ds2001 High-Speed A/D Board

The DA2001 shown in Figure 28 can provides 5 parallel A/D channels, with 4, 8, 12 or 16-bit resolution (programmable). The input voltage range is ±5 V or ±10 V, (programmable) with 1 M input impedance. Sampling can be conducted by software from the processor board. Physical Requirements include an 8 or 16-bit ISA slot (power supply only). The A/D conversion time is 1.5,2.7,3.8 or 5.0us depend on the resolution setting.

In this application, the input voltage range is set at ±10 V and resolution is set at 12-bit (3.8us conversion time).

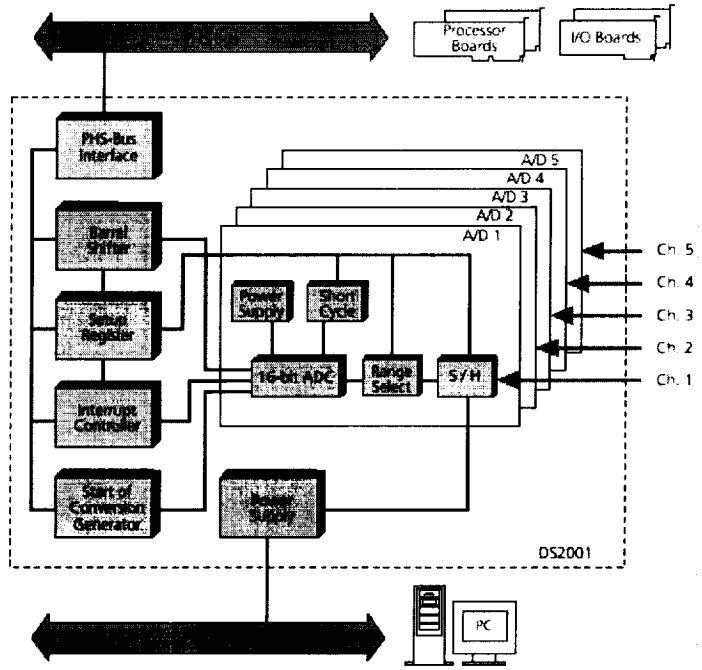


Figure 28 Block diagram of the DS2001 board

ds4002 Timing and Digital I/O Board

The DS4002 Timing and Digital I/O Board combines all the digital I/O tasks you can think of on one board. It provides the functionality of an ordinary digital I/O board plus additional features that help you perform specific control tasks easily. Eight channels can be programmed for either capturing digital signals or generating flexible pulse patterns. 32 additional I/O lines can be used for further digital I/O tasks, for example, to control single input lines (switches, sensors) or output lines (relays, displays).

32 additional digital I/O lines (TTL) were used here. The first 24 lines are for IN/OUT (programmable in 8-bit groups). The next 4 lines were for IN only, and the last 4 lines were for OUT (fixed) only. 12-bit resolution PWM pulses were generated with the first 24 lines.

ControlDesk

ControlDesk offers an interactive control environment for real-time applications up to the most complex automation tasks. Besides Experiment and Platform Manager, ControlDesk comes with an instrumentation set, the Parameter Editor and basic automation features.

The graphical Platform Navigator keeps track of your system setup. With the Experiment Manager, you can handle your experiment data professionally based on central data storage.

Choose a control or plotter instrument from the Instrument Selector and place it on the workspace. Double-click the instrument to open its property page. Choose from various pre-configured instrument settings or powerful configuration options. To assign a variable, drag it from the Variable Browser onto the instrument. Repeating this procedure, an impressive instrument panel connects it with corresponding model variables.

Through instrument panels, we can tune parameters online, display variable values continuously without interruption of experiments, and also capture the data on the hard drive for future research use.

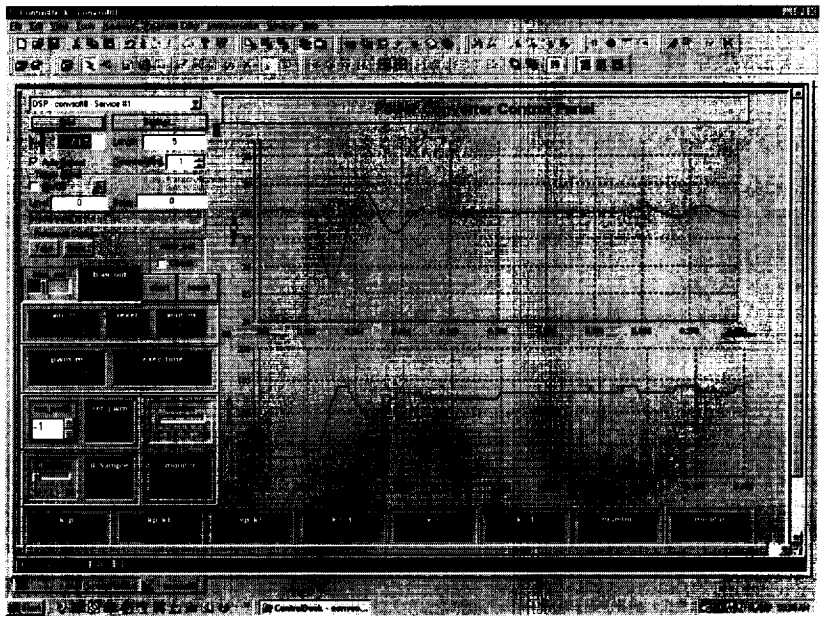


Figure 29 Power Converter Control Panel on dSPACE

4.6 The Stand-Alone DSP Control System

The Stand-alone DSP Control system, which was built by researchers at Advanced Engineering Research Lab(AERL) is comprised of a TI DSP board (TMS320C6711 DSK), TI A/D board (THS1206EVM), and PWM generator board. The

Stand Alone DSP was designed as a compact replacement for the dSpace system. Unlike dSpace, the Stand-Alone DSP Control System lacks powerful development software and support; however, the reduced cost and portability make the stackable system a more flexible solution for practical applications.

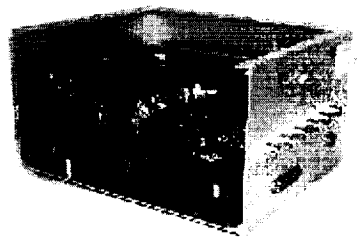


Figure 30 Stand-Alone DSP Control Systems

TMS320C6711 DSK Board:

The new TMS320C6711 DSP Starter Kit (DSK) is powerful enough to use for fast development of networking, communications, imaging and other applications. Operating at 150 MHz, the C6711 provides an impressive 1200 million instruction per second (MIPS) and 600 million floating-point operations per second (MFLOPs).

Code Composer Studio:

The C6711 DSK comes with an array of DSK-specific software functionality (256 KB software image memory limited), including the highly efficient C6000 C Compiler and Assembly Optimizer, Code Composer Debugger and DSK support software. The compiler provides over 80% of the performance of hand-coded assembly on DSP benchmarks using natural C code.

Through watch windows, the variables memory space can be set and retrieved while the system is running. This feature makes tuning control variables on-the-fly possible.

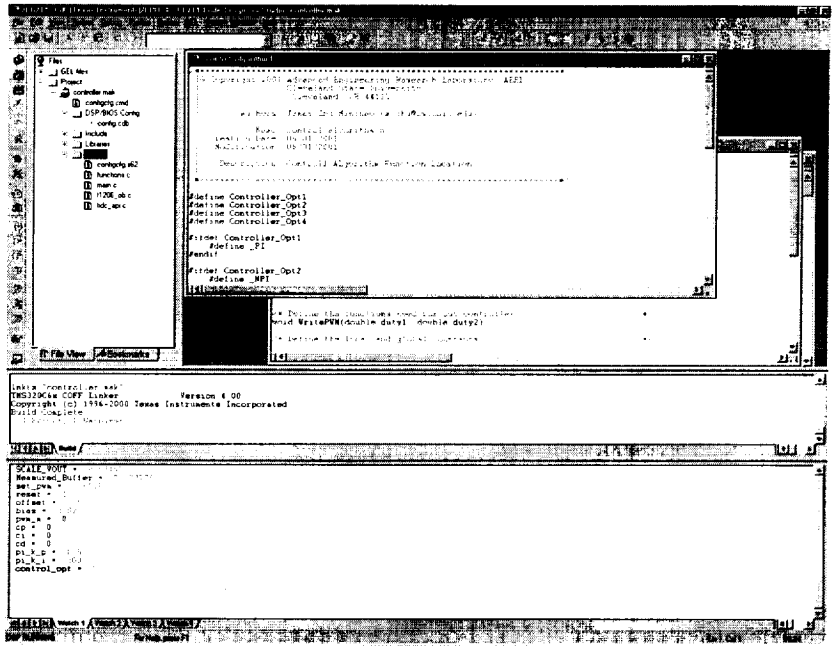


Figure 31 Power Converter Control Panel on stackable system

TH1206 A/D Board:

The TH1206 A/D Board is a 12-Bit, 6 Mega Sample Per Second (MSPS), Simultaneous Sampling, 4 channel Analog-to-Digital Converter (ADC). An integrated 16 words deep FIFO allows the storage of data in order to take the load off the processor connected to the ADC at 6 MHz for 4 channels.

4.7 Derivative of the Output Signal

There are two ways to obtain the derivative of the converter output signal. One is through software, as discussed in CHAPTER II. The other is hardware, from current sensors measuring the current through the large output capacitor, represented in Eq. (4.1).

From $q = cv$, we can obtain $i = dq/dt = c \cdot dv/dt$. (4.1)

Where q is the charge of the capacitor at the output, I represents the current through the capacitor, c represents the value of the capacitor, and v is the output voltage of the converter. We compared these two methods for obtaining the derivative of the converter output voltage. The goal was to determine which method produced the most accurate results.

A current sensor, model BB-100, from F.W. Bell, which can accurately measure dc and ac currents and provide electrical isolation between the output of the sensor and the current carrying conductor. Some features are as following: maximum current < 100A, Sensitivity = 50mV/A and Response time < 2us.

The Non-Linear Tacking Differentiator parameter settings are shown as follows:

$$\begin{bmatrix} M = 2e9 & \delta_d = 100 \\ k_{d1} = 0.2 & k_{d2} = 1 \end{bmatrix}$$

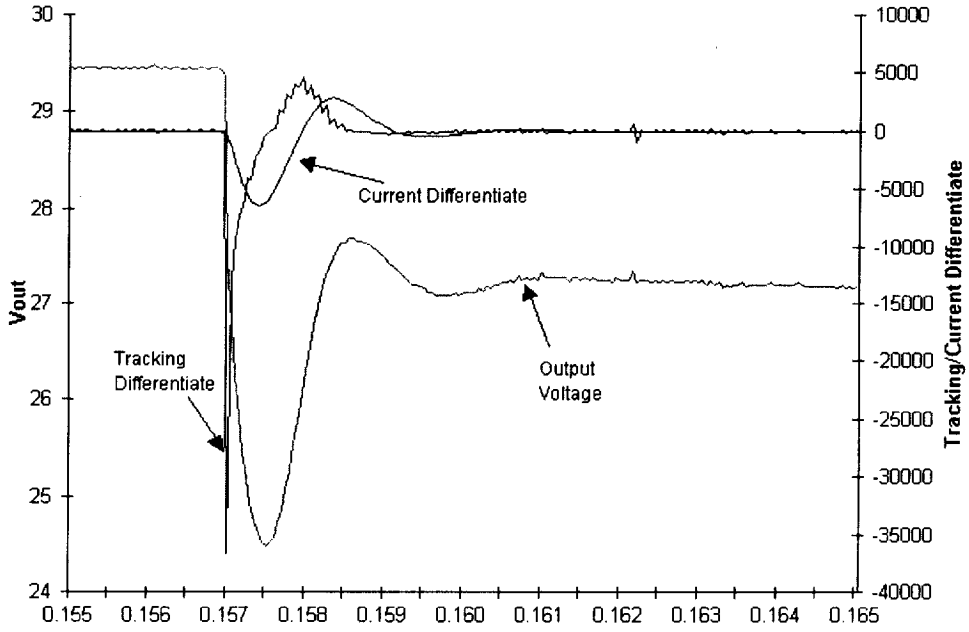


Figure 32 Output signal derivative during Step-Up

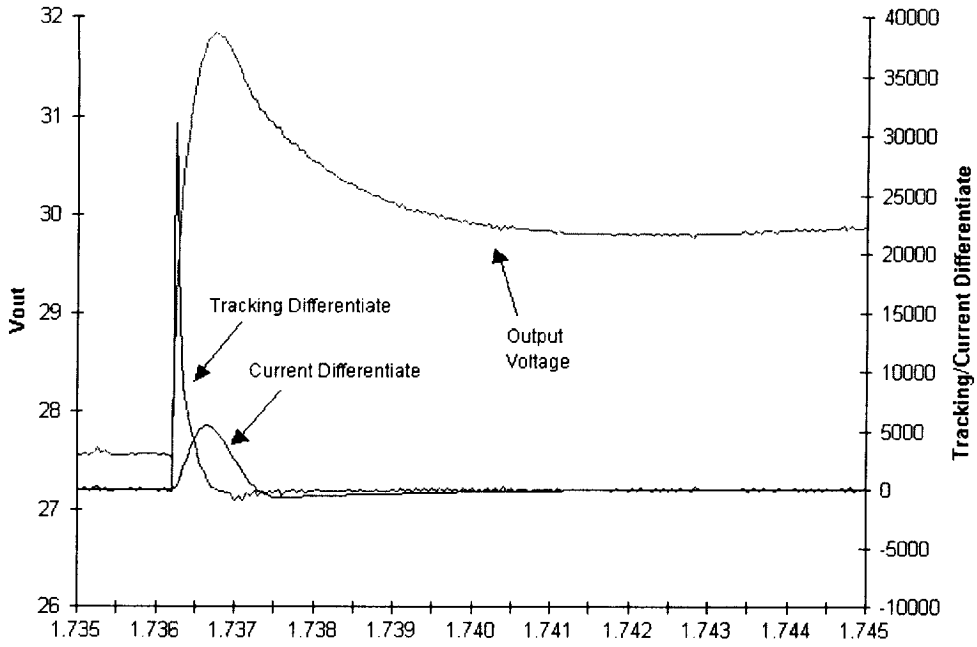


Figure 33 Output signal derivative during Step-Down

We test the two differentiation methods under open loop conditions with load currents alternating between 3A and 36A. The results are shown in Figure 32 and Figure 33. It is obvious that the TD has a much faster and better response compared with the current sensor measurement. Compared to pure differentiator, there is 50us delay for the TD, but 500us delay for the current sensed differentiation. In addition, the TD is a better approximate derivative of the signal than the current sensed differentiation.

4.8 Software Calibration

Figure 34 shows the Power Converter Test Bench set up. Before testing could be conducted, calibration was required for the software measurement system.

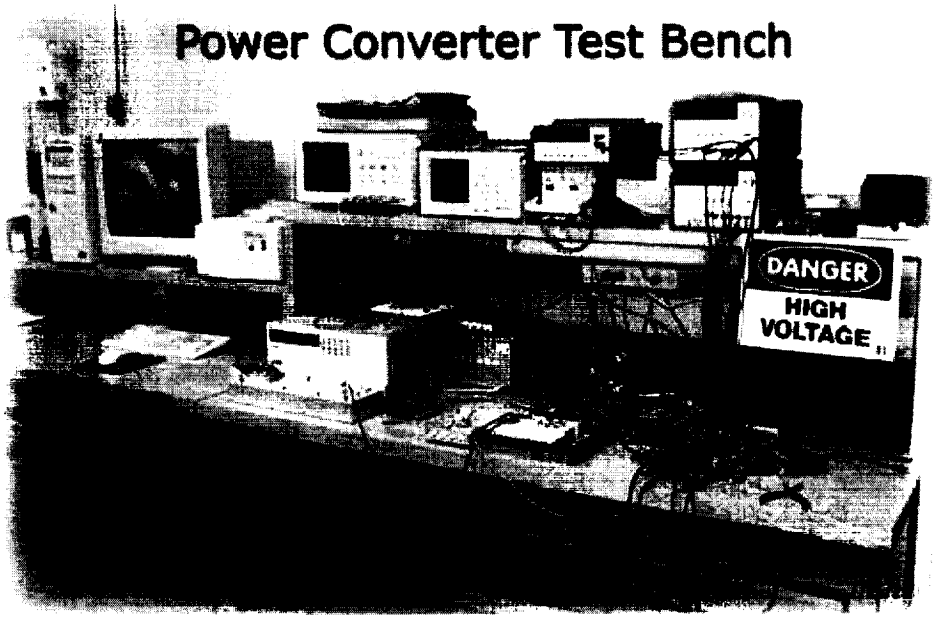


Figure 34 Power Converter Test Bench

The reason for this is due to the fact that the converter output voltage does not go directly into the A/D converter. Therefore, we need to convert the DSP software reading data into actual output measurement data for the control algorithm. The equations listed below are used to scale the sample data accordingly:

$$V_{Measured} = (V_{DSP\ Reading} - V_{offset})\ Scale\_V_{out} - V_{bias}$$

$V_{Measured}$:	Measured data used for control algorithm
$V_{DSP\ Reading}$:	Measured data (using DSP software retrieve from A/D buffer)
V_{offset} :	Intput adjustment for the zero input
$Scale\_V_{out}$:	Scale factor for output voltage
V_{bias} :	Output adjustment for the setpoint (ex 28) output on the converter during closed loop operation.

A multimeter (HP34401A) was used to calibrate the software parameter setting for the output voltage reading. First, set the PWM ratio, Scale_V_{out} and V_{bias} to 0, then tune V_{offset} to get $V_{Measuring} = 0$. Second, set PWM ratio ≈ 0.689, to get an output voltage near 28 Vdc (HP34401A reading), then tune Scale_V_{out} until the $V_{Measured} = 28$. Since the power converter is a nonlinear system, the closed loop response is a little different from the open loop response. Therefore, during closed loop operation, V_{bias} can be tuned to make $V_{Measured}$ match with the multimeter reading as close as possible.

[Acknowledgement]

I would like to thank all the team members in the AERL for their hard work and support during my research. Their contributions are listed below: [15]

- Aaron Radke
  - Work with the Code Composer Studio 1.23 and 2.0
  - Work with the DSP and A/D
- Arthur Stachowicz
  - Enclosure
  - PWM Generator in AHDL
- Dave Wladyka
  - Researched and bought the DSP stack
  - Initial work with Code Composer Studio 1.23 and DSP
- Greg Tollis
  - Signal Condition design, testing, building and improving
- Ivan Jurcic
  - PWM Generator in VHDL
  - Prototype board layout for the PWM
- John Sustersic
  - Signal conditioning board layout
- Tom Stimac
  - Plant modeling

CHAPTER V

TEST RESULTS

First, NPID tuning parameters that were found by actually tuning the system are introduced. Next, the test results will be shown and evaluated for the PI, NPI, PID, and NPID controllers. The main performance measurement for the various controllers is disturbance rejection. Before testing however, we must determine the performance specifications described by NASA for the 1KW, Westinghouse Power Converter. A list of the Westinghouse converter design specifications is shown in Table 1. These specifications deal with different system characteristics from regulation to disturbance rejection.

After the tuning process was completed using the Stand-Alone DSP Control System, the PI and NPID gains were set to the values shown below:

PI: $k_p = 0.3$ $k_i = 230$

NPID: $\begin{bmatrix} \delta_p = 0.4 \\ k_{p1} = 1 \quad k_{p2} = 0.2 \end{bmatrix} \begin{bmatrix} \delta_i = 0.5 \\ k_{i1} = 1 \quad k_{i2} = -5 \end{bmatrix} \begin{bmatrix} k_d = 4.5\text{e-}3 \quad \delta_d = 70 \quad k_{d\_r} = 2\text{e}9 \\ k_{d1} = 0.2 \quad k_{d2} = 1 \end{bmatrix}$

Load Step-Up : $k_{p\_up} = 0.7$ $k_{i\_up} = 600$

Load Step-Down : $k_{p\_dn} = 3$ $k_{i\_dn} = 650$

All test results obtained in this chapter, were based on the above tuning parameters.

Performance Criteria	Definition	Equation	Specification
DC Load Regulation	Maximum deviation of the output voltage as the load is varied within its range	$\Delta V_o = V_{s.s.} - V_{actual}$ $(I_{load} = 0 \ to \ 36 amps)$	± 280 mV
DC Line Regulation	Maximum deviation of the output voltage as the supply voltage is varied within its range	$\Delta V_o = V_{s.s.} - V_{actual}$ $(V_{supply} = 110 \ to \ 140 \ V_{dc})$	± 280 mV
Ripple	Maximum Peak-to-Peak of the output voltage	$\Delta V_o = V_{high} - V_{low}$	200 mV
Load Transient Response	Maximum deviation and settling time of the output voltage due a change in the load	$\Delta V_o = V_{s.s.} - V_{actual}$ $T_{settling} = T_{s.s.(second)} - T_{s.s.(first)}$	± 280 mV 10 ms
Line Transient Response	Maximum deviation and settling time of the output voltage due to a change in the line	$\Delta V_o = V_{s.s.} - V_{actual}$ $T_{settling} = T_{s.s.(second)} - T_{s.s.(first)}$	± 280 mV 10 ms

Table 1. Design Specifications for the 1-kw W.S.P.C.

5.1 NPID Output Voltage Ripple

The output voltage ripple is an important factor to judge the quality of the DC voltage. Figure 35 shows Output Voltage Ripple with 120Vdc supply voltage under 4 load conditions: 1A, 12A, 24A and 36A.

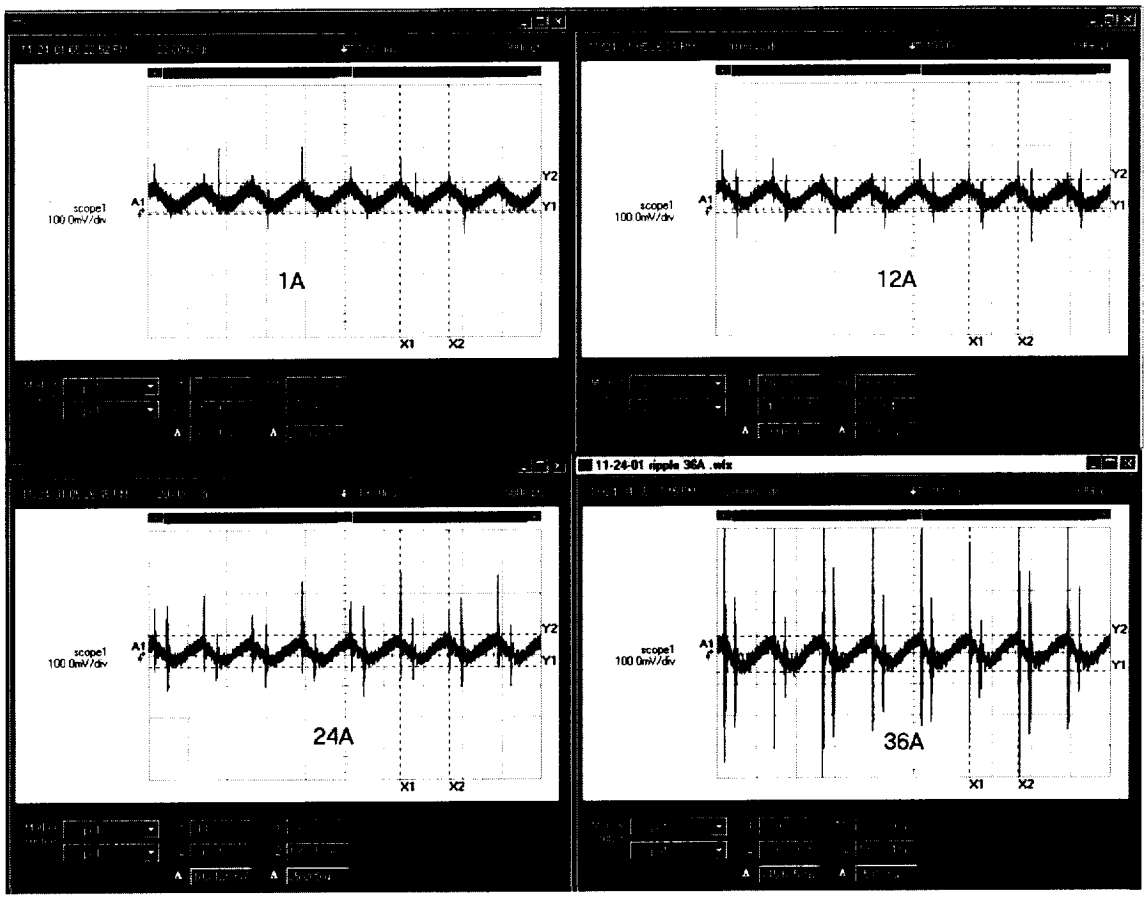


Figure 35 Output Voltage Ripple

The ripple is related to the load current, therefore, large loads will produce large ripple. Under 36A load (100%), the maximum ripple voltage (peak-to-peak) is around 115.625mV, which is far less than the requirement 200mV.

5.2 Transient Response during Start-up

Figure 36 shows the start transient response with the PI and NPID controllers with a supply voltage of 120Vdc and load current 36A using PI & NPID controller. In order to eliminate overshoot, a soft start discussed in CHAPTER II was implemented.

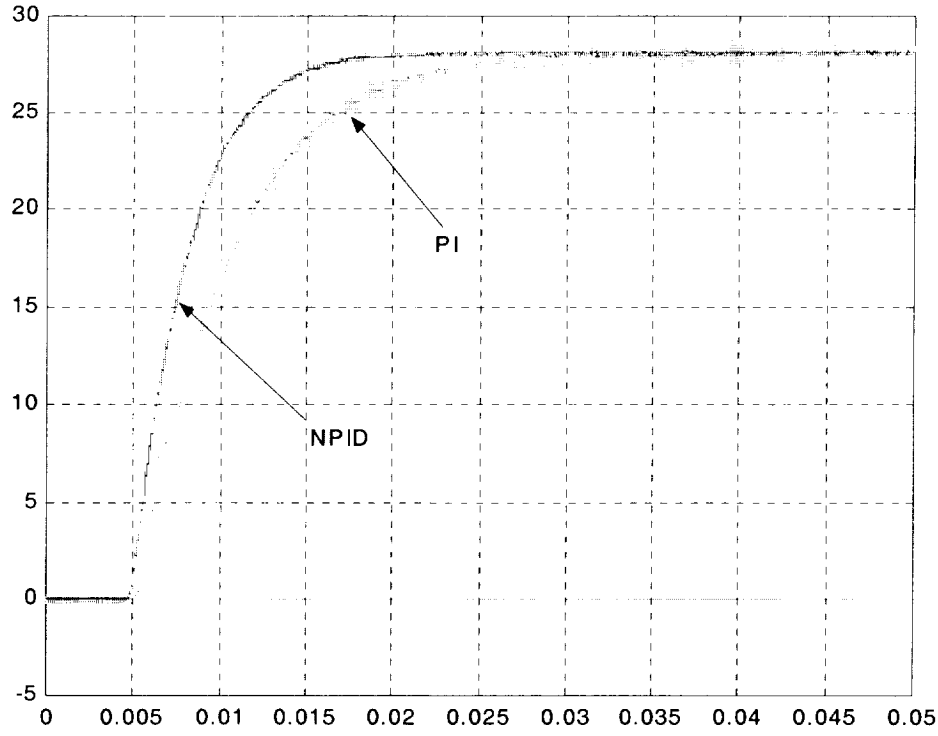


Figure 36 Start Transient Response (PI vs. NPID)

The biggest consideration during start-up is the settling time. The settling time is the period from start point to the 98% of the full step size. For the Westinghouse converter, the 98% point is at 27.44V. The settling time is 18ms for the PI controller and

11ms for the NPID controller. Therefore, the PI controller is 67% slower than the NPID controller.

Another important performance consideration is steady state error. For the Westinghouse converter, the PI controller produces a ± 250mV steady state error, while the NPID controller produces a ± 180mV steady state error. Therefore, the PI controller produces a steady state error that is 39% larger than that produced by the NPID controller. The test results showed a close relationship with the simulation results.

5.3 Line Disturbance Rejection

Figure 37 shows Line Disturbance Rejection when a voltage step-up occurs from 110Vdc to 140Vdc with a load current of 3A. Figure 38 shows Line Disturbance Rejection during a voltage step-down from 140Vdc to 110Vdc with a load current of 36A. The top waveform is with PI controller, and the lower one is with NPID controller.

The output voltage transient response shows that a line disturbance of ±30V has a finite effect on the output voltage of the W.S.P.C. The peak of the transient and total recovery time of the Westinghouse converter output voltage is shown below for the PI and NPID controllers.

Deviation of voltage (mV)	PI	NPID	Improvement
Step-Up	444	169	162.72%
Step-Down	525	225	133.33%

Recovery time (ms)	PI	NPID	Improvement
Step-Up	214	56	282.14%
Step-Down	244	141	73.05%

Table 2. Line Disturbance Rejection Transient

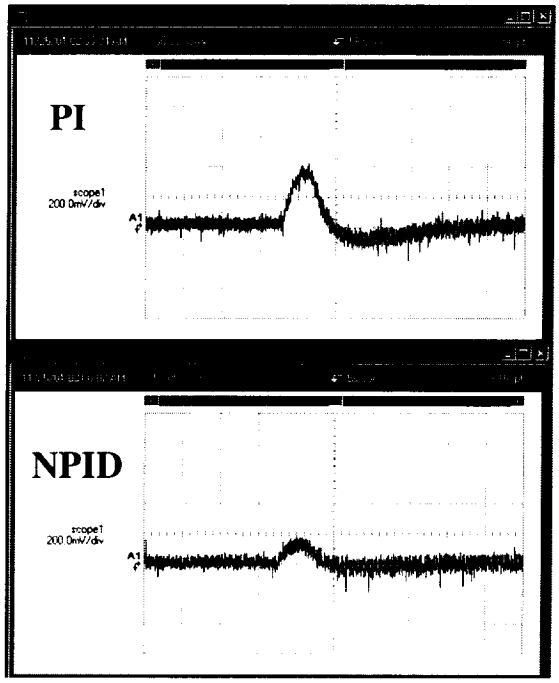


Figure 37 Line Voltage Disturbance Rejection (110v → 140v,3A)

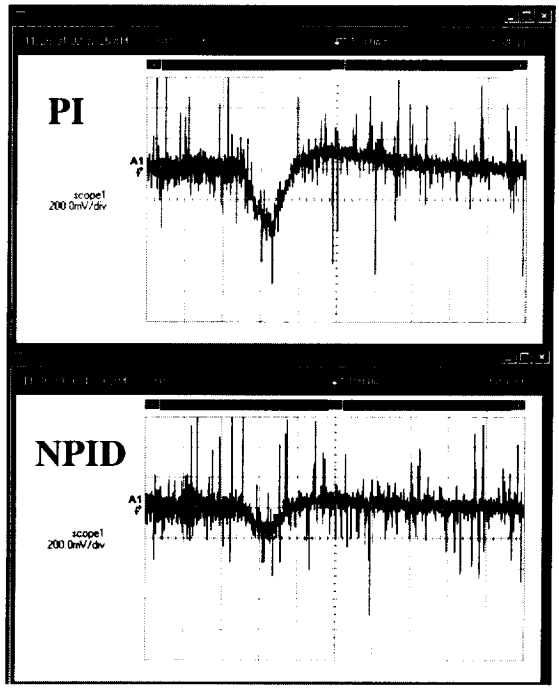


Figure 38 Line Voltage Disturbance Rejection (140v → 110v,36A)

The percentage shows the performance gain achieved by using the NPID controller over the PI controller. It is very important to note that the W.S.P.C.'s output

voltage never left steady state operation; the error in the output voltage was always less than 280 mV during the transient period for NPID controllers. This means that the Westinghouse converter performance is within the specifications for NASA. The main reason that the transient appears in the output voltage waveform for more than 50 ms is that the characteristics of any transient seen at the input of the W.S.P.C. are slowly introduced into the actual components in the W.S.P.C. due to the massive filtering that is taking place within the first stage of the power conversion process.

5.4 Load Disturbance Rejection

Figure 39 shows Load Disturbance Rejection during a load step-down from 36A(100%) to 3A(8.3%) with a supply voltage of 120Vdc. Figure 40 shows Load Disturbance Rejection during a load step-up from 3A to 36A with a supply voltage of 120Vdc. The top waveform is with the PI controller, and the lower waveform is with the NPID controller.

The output voltage transient response shows that a line disturbance of ±33V has a big effect on the output voltage of the W.S.P.C. The peak of the transient, and total recovery time is shown in Table 3 for the PI and NPID controllers.

Deviation of voltage (V)	PI	NPID	Improvement
Step-Up	4.34	3.40	28.64%
Step-Down	3.9	3.22	21.12%

Recovery time (ms)	PI	NPID	Improvement
Step-Up	13	5.9	120.34%
Step-Down	7	2.8	150%

Table 3. Load Disturbance Rejection transient

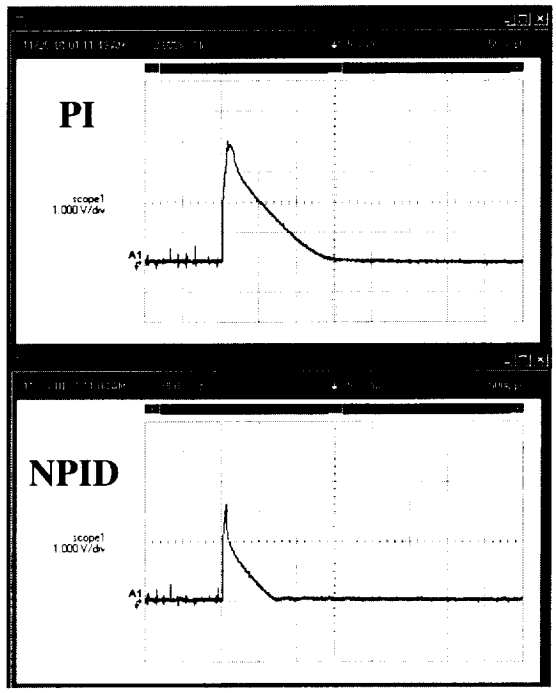


Figure 39 Load Current Disturbance Rejection (36A → 3A)

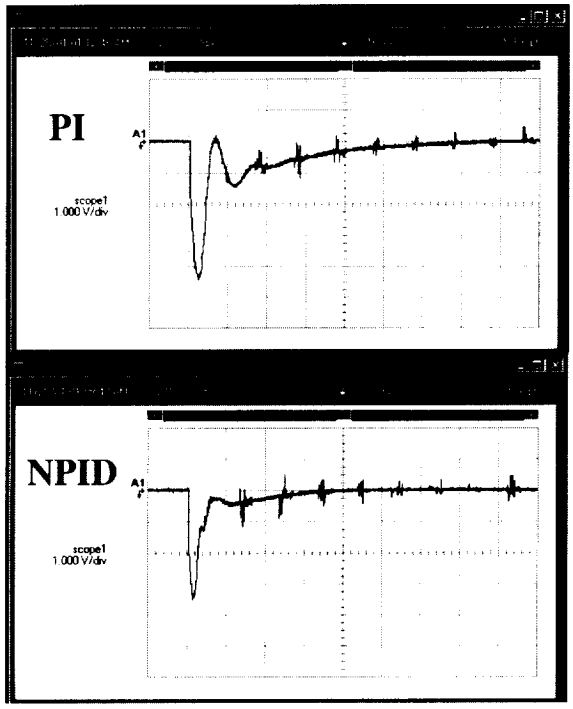


Figure 40 Load Current Disturbance Rejection (3A → 36A)

5.5 Output Impedance

Output Impedance is a good criterion to judge the DC regulation quality.

DC Output Impedance

The steady state output voltage versus the output current in close loop is shown in Figure 41. The supply voltage is 120Vdc. The best linear trend line as computed in Excel is also shown. The slope of the line gives the effective output resistance. There is not much of a difference between the PI controller and NPID controller. The smaller the effective output impedance, the better the Westinghouse converter looks as a source.

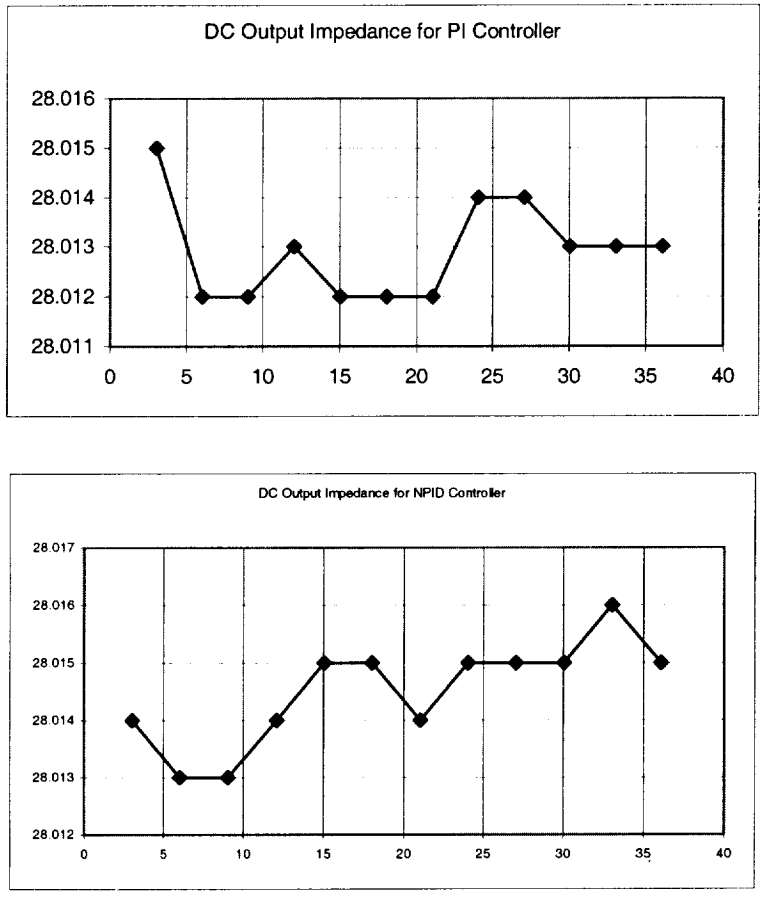


Figure 41 DC Output Impedance

Transient Output Impedance

The output voltage was monitored while a step of load current was applied to the output of the converter. This test was performed at a supply voltage of 120Vdc. The amplitude of the load current step was form 3A to 36A, and reverse. Figure 39 and Figure 40 shows the corresponding load step transients.

Transient Output Impedance (Ω)=ΔV/ΔA	PI	NPID	Improvement
Step-Up	0.13	0.1	30%
Step-Down	0.12	0.098	22.4%

Table 4. Transient Output Impedance

5.6 Robustness Test of NPID

As is mentioned in CHAPTER II, the introduction of nonlinear parameters into control algorithms brings difficulties to convergence and stability analysis. Therefore, a series of critical tests were conducted in order to test the robustness of the power converter.

Regular tests include step changes in the load between 3A and 36A by 1A intervals back and forth, while keeping the supply voltage at 120Vdc. Random tests involved random changes in load current between 3A and 36A keeping the supply voltage at 120Vdc. Dual disturbance rejection is when line and load changes are made at the same time with random settings. Line voltages between 110Vdc and 140Vdc were used for this experiment. Load currents between 3A and 36A were used at the same time

for this experiment. Load and line settings were updated every four seconds for the dual disturbance test. Each test was conducted for more than 1 hour.

Throughout the testing period, the Westinghouse converter was found to be robust and stable under all conditions. The NPID digital controller and the control system proved to be robust and safe.

CHAPTER VI

Conclusion

The process of developing a digital NPID control for a 1-kw W.S.P.C. was demonstrated in this thesis. First, the NPID methodology was studied and discussed. Next, a simulation was created based on the linear transfer function model of the converter. The NPID controller was then implemented using the DSPACE rapid prototyping system and the stand-alone DSP system. Finally, the NPID controller was tested using the ED408043-1 Westinghouse 1-kw switching DC-DC power converter, and was compared to the previously developed PI controller. The performance of the digital NPID controller in this thesis proved that a practical and robust control algorithm could be developed and implemented for the W.S.P.C. Compared to the PI controller, the gains of the NPID controller allow much more flexibility for the user to independently change the gains for different error regions. The nonlinear part greatly improves the performance of the system by allowing the controller to act on a different set of gains for positive or negative error conditions.

Hardware-in-the-loop simulation on the dSPACE and Stackable DSP Systems allow the controller to be tuned in real time while the converter output response is

monitored. The digital control module design makes the new control algorithm easy to implement by software coding, and also easy to reproduce.

Next, our DSP-based research will be conducted so as to evaluate a multitude of controller types that can only be accomplished digitally. The following list of features will be considered for future research on the Westinghouse converter unit.

1. Under/over voltage protection
2. Short-circuit protection
3. Low power efficiency with variable frequency
4. Output current limit
5. Multiple power converters with one DSP processor
6. Component "health monitoring"

Finally, the DSP's potential for providing high-speed communications will be evaluated in terms of developing more reliable power management and distribution (PMAD) systems for future space platforms.

BIBLIOGRAPHY

[1] Marty Brown, "Power supply cookbook", Butterworth-Heinemann, 1994, pp. 25.

[2] Tomislav J. Stimac, "Digital Control of a 1-kW DC-DC switching Power converter", Master's Thesis, Department of Electrical and Computer Engineering, Cleveland State University, December 2000.

[3] Robert M. Button," Intelligent systems for power management and distribution", IECEC, Vol. 1, Jul 2001 pp.193-198.

[4] George C. Verghse, "Dynamic Modeling and control in Power Electronics", The Control Handbook, CRC Press Inc, 1996, pp. 1413-1424.

[5] Marty Brown, "Power supply cookbook", Butterworth-Heinemann, 1994, pp. 33.

[6] J. Han and W. Wang, "Nonlinear Tracking Differentiator", Systems Science and Mathematical Sciences, Vol.14, No. 2, pp.177-183, 1994.

[7] ZhiQiang Gao, "An Alternative Paradigm for Control System Design", 2001 IEEE Conference on Decision and Control

[8] Sira-Ramirez, Herbert," Nonlinear P-I controller design for switch mode DC-to-DC power converters", IEEE Transactions on Circuits and Systems v 38 n 4 1991 p.410-417.

[9] Gupta, Tarun; Boudreaux, R. R.; Nelms, R. M. and Hung, John Y, "Implementation of a fuzzy controller for DC-DC converters using an inexpensive 8-b microcontroller", IEEE Transactions on Industrial Electronics. v 44, n 5. Oct 1997, p 661-668.

[10] J.Q. Han, "Nonlinear Design Methods for Control Systems" The Process of the 14th IFAC World Congress, 1999

[11] Jacob Tal, "Motion Control Systems", The Control Handbook, CRC Press Inc, 1996, pp.1382-1386.

[12] ZhiQiang Gao, "From Linear to Nonlinear control Means - a Practical Procession", ISA transactions September 12, 2001.

[13] John Sustersic, John Zeller, and Zhiqian Gao, "Design and Implementation of a digital Controller For DC-to-DC Power Converters", Society of Automotive Engineers, 2000

[14] Bill Andreycak, "New Driver ICs Optimize High Speed Power MOSFET Switching Characteristics", UNITRODE Integrated Circuits Corporation, U-118

[15] Arthur Stachowicz, Greg Tollis, Aaron Radke and Ivan Jurcic, "Stand-Alone DSP Control System Documentation", AERL-MNUL-0001B Advanced Engineering Research Laboratory (AERL), Cleveland State University

APPENDICES

A. C code for Tracking Differentiator

```
// Block TD for NPID or ADRC
// Using fst2 function
// in discrete format
// 1 input, 2 outputs, 2 discrete states
// 1 vector parameter, includes 5 parameters
// #1: r
// #2: h (normally same as the step size #5)
// #3,#4: initial conditions of the two states
// #5: discrete step size
// parameters must be in the right order

#define S_FUNCTION_NAME tdfst2d

/*
 * Need to include simstruc.h for the definition of the SimStruct and its associated macro
definitions.
 */

#include "simstruc.h"
#include "math.h"

double sat(double x, double delta)
{
 if (x>delta) return(1.0);
 else if (x<-delta) return(-1.0);
 else return(x/delta);
 }

double fst2(double x1, double x2, double u0, double r, double h)
{
   double d,d0,y1,a0,a1;

   d=r*h;
   d0=d*h;
   y1=x1-u0+h*x2;
   a0=sqrt(d*d+8*r*fabs(y1));
   if (y1>d0) a1=x2+(a0-d)/2;
   else if (y1<-d0) a1=x2-(a0-d)/2;
   else a1=x2+y1/h;

   return(-r*sat(a1,d));
}
```

```
/*
 * mdlInitializeSizes - initialize the sizes array
 *
 * The sizes array is used by SIMULINK to determine the S-function block's
 * characteristics (number of inputs, outputs, states, etc.).
 */

static void mdlInitializeSizes(SimStruct *S)
{
   ssSetNumContStates(    S, 0);     /* number of continuous states */
   ssSetNumDiscStates(    S, 2);     /* number of discrete states */
   ssSetNumInputs(        S, 1);     /* number of inputs */
   ssSetNumOutputs(       S, 2);     /* number of outputs */
   ssSetDirectFeedThrough(S, 0);     /* direct feedthrough flag */
   ssSetNumSampleTimes(   S, 1);     /* number of sample times */
   ssSetNumInputArgs(     S, 1);     /* number of input arguments */
   ssSetNumRWork(         S, 0);     /* number of real work vector elements */
   ssSetNumIWork(         S, 0);     /* number of integer work vector elements */
   ssSetNumPWork(         S, 0);     /* number of pointer work vector elements */
}

/* Input Arguments */

#define TD_PARA            ssGetArg(S,0)

/*
 * mdlInitializeSampleTimes - initialize the sample times array
 *
 * This function is used to specify the sample time(s) for S-function.
 * If S-function is continuous, must specify a sample time of 0.0.
 * Sample times must be registered in ascending order.  If S-function
 * is to acquire the sample time of the block that is driving it, must
 * specify the sample time to be INHERITED_SAMPLE_TIME.
 */

static void mdlInitializeSampleTimes(SimStruct *S)
{
   double stepsize;

   stepsize=mxGetPr(TD_PARA)[4];

   ssSetSampleTimeEvent(S, 0, stepsize);
   ssSetOffsetTimeEvent(S, 0, 0.0);
}
```

```
/*
 * mdlInitializeConditions - initialize the states
 *
 * This function initializes the continuous and discrete
 * states for S-function block. The initial states are placed
 * in the x0 variable. Also, any other initialization can be
 * performed.
 */

static void mdlInitializeConditions(double *x0, SimStruct *S)
{
 double x1_0,x2_0;

 x1_0=mxGetPr(TD_PARA)[2];
 x2_0=mxGetPr(TD_PARA)[3];

 x0[0]=x1_0;
 x0[1]=x2_0;
 }
 /* The initial conditions of NPID are all zero.
  * So let them be default.
  */

/*
 * mdlOutputs - compute the outputs
 *
 * In this function, you compute the outputs of your S-function
 * block.  The outputs are placed in the y variable.
 */

static void mdlOutputs(double *y, double *x, double *u, SimStruct *S, int tid)
{
   y[0]=x[0];
   y[1]=x[1];
}

/*
 * mdlUpdate - perform action at major integration time step
* This function is called once for every major integration time step.
 * Discrete states are typically updated here, but this function is useful
 * for performing any tasks that should only take place once per integration
 * step.
 */

static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{
```

```
    double r,h;
    double stepsize;
    double v[2];

    r=mxGetPr(TD_PARA)[0];
    h=mxGetPr(TD_PARA)[1];

    stepsize=mxGetPr(TD_PARA)[4];

   if(ssIsSampleHitEvent(S,0,tid)) {

    v[0]=x[0]+stepsize*x[1];
    v[1]=x[1]+stepsize*fst2(x[0],x[1],u[0],r,h);

    x[0]=v[0];
    x[1]=v[1];
    }
 }

/*
 * mdlDerivatives - compute the derivatives
 *
 * In this function, you compute the S-function block's derivatives.
 * The derivatives are placed in the dx variable.
 */

static void mdlDerivatives(double *dx, double *x, double *u, SimStruct *S, int tid)
{
    }

/*
 * mdlTerminate - called when the simulation is terminated.
 *
 * In this function, any actions that are necessary at the termination of a
 * simulationyou should be performed. For example, if memory was allocated
 * in mdlInitializeConditions, this is the place to free it.
 */
static void mdlTerminate(SimStruct *S)
{
 }

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

B. C code for NPID G Function

```
// G function
// in continues format
// 1 input, 1 outputs, 1 continues states
// 1 vector parameter, includes 3 parameters
// #1~3: K1,K2,delta
// parameters must be in the right order

#define S_FUNCTION_NAME Gfunc

// Need to include simstruc.h for the definition of the SimStruct and
// its associated macro definitions.

#include "simstruc.h"
#include "math.h"

/*
 * mdlInitializeSizes - initialize the sizes array
 *
 * The sizes array is used by SIMULINK to determine the S-function block's
 * characteristics (number of inputs, outputs, states, etc.).
 */

static void mdlInitializeSizes(SimStruct *S)
{
  ssSetNumContStates(    S, 0);     /* number of continuous states */
  ssSetNumDiscStates(    S, 0);     /* number of discrete states */
  ssSetNumInputs(        S, 1);     /* number of inputs */
  ssSetNumOutputs(       S, 1);     /* number of outputs */
  ssSetDirectFeedThrough(S, 0);     /* direct feedthrough flag */
  ssSetNumSampleTimes(   S, 1);     /* number of sample times */
  ssSetNumInputArgs(     S, 1);     /* number of input arguments */
  ssSetNumRWork(         S, 0);     /* number of real work vector elements */
  ssSetNumIWork(         S, 0);     /* number of integer work vector elements */
  ssSetNumPWork(         S, 0);     /* number of pointer work vector elements */
}

/* Input Arguments */

#define GF_PARA            ssGetArg(S,0)

/*
 * mdlInitializeSampleTimes - initialize the sample times array
```

```
 *
 * This function is used to specify the sample time(s) for S-function.
 * If S-function is continuous, must specify a sample time of 0.0.
 * Sample times must be registered in ascending order.  If S-function
 * is to acquire the sample time of the block that is driving it, must
 * specify the sample time to be INHERITED_SAMPLE_TIME.
 */

static void mdlInitializeSampleTimes(SimStruct *S)
{
}

/*
 * mdlInitializeConditions - initialize the states
 *
 * This function initializes the continuous and discrete
 * states for S-function block. The initial states are placed
 * in the x0 variable. Also, any other initialization can be
 * performed.
 */

static void mdlInitializeConditions(double *x0, SimStruct *S)
{
 }

 /* The initial conditions of NPID are all zero.
  * So let them be default.
  */

/*
 * mdlOutputs - compute the outputs
 *
 * In this function, you compute the outputs of your S-function
 * block.  The outputs are placed in the y variable.
 */

static void mdlOutputs(double *y, double *x, double *u, SimStruct *S, int tid)
{
        double K1,K2,delta,alpha;

        K1=mxGetPr(GF_PARA)[0];
        K2=mxGetPr(GF_PARA)[1];
        delta=mxGetPr(GF_PARA)[2];

        if (u[0]>delta) y[0]=K2*u[0]+(K1-K2)*delta;
                else if(u[0]<-delta) y[0]=K2*u[0]-(K1-K2)*delta;
```

```
                    else y[0]=K1*u[0];

        if(K2<0&&u[0]*y[0]<0) y[0]=0;
}

/*
 * mdlUpdate - perform action at major integration time step
 *
 * This function is called once for every major integration time step.
 * Discrete states are typically updated here, but this function is useful
 * for performing any tasks that should only take place once per integration
 * step.
 */

static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{
 }

/*
 * mdlDerivatives - compute the derivatives
 *
 * In this function, you compute the S-function block's derivatives.
 * The derivatives are placed in the dx variable.
 */

static void mdlDerivatives(double *dx, double *x, double *u, SimStruct *S, int tid)
{
   }

/*
 * mdlTerminate - called when the simulation is terminated.
 *
 * In this function, any actions that are necessary at the termination of a
 * simulationyou should be performed. For example, if memory was allocated
 * in mdlInitializeConditions, this is the place to free it.
 */

static void mdlTerminate(SimStruct *S)
{
 }

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

C. C code for dSpace

```
////////////////////////////////////////////////////////////////////////
//    Basic DC-DC Power Converter Controller Implementation (8bit)
//
//    MinShao Zhu ( James )
//    January, 2001
//    Advanced Engineering Research Laboratory
//    Cleveland State University
//      Version 2.0
//
////////////////////////////////////////////////////////////////////////

#include <brtenv.h>
#include <math.h>

// Conversion constants to scale sample data to proper magnitudes


#define period 5.0e-5   // controller period, in seconds

typedef struct
{
      long V_in;  //Converter input voltage in Volts
      long C_out; //Converter output voltage in Volts
      long V_out; //Converter output voltage in Volts

} feedback_data;

// variables for execution time profiling

long V_input_min = 9000;      // minimum input voltage parameter
long V_input_max = 13500;     // maximum input voltage parameter

volatile int k_Vin = 18015 ;   //get input voltage*100's coefficient
volatile int k_Vout = 4537 ;   //get output voltage*100 old"5515 new
5590
volatile int k_Cout = 5100 ;  //get output current*100
volatile float bias_out=-0.025,frequency=21.74 ;      //Converter
output voltage bias setting.
volatile float V_set = 28.0;        // converter set point, in volts

volatile float kp_up = 0.7 ,kp_dn = 8 , kp_k1 = 1,kp_k2 = 0.5, kp_d =
0.1 ; // voltage loop proportional gain, dimensionless
volatile float ki_up = 700 ,ki_dn = 1000 ,ki_k1 = 1,ki_k2 = -5,ki_d=0.3
;
volatile float k_d = 4e-4 , k_d_r =2e9 , k_d_Zone = 70;

volatile int   N_Sample  = 1,reset=1,set_pwm=1 ;     // N_Sample <= 100

feedback_data current,last,total,data[40];      // data storage ,
confirm N_Sample <= 100
float    vout_m,cout_m,vin_m,exec_time,maxcount=1250,cp,ci,cd;
```

```
int      n,pwm_m;
double diff[2];

// SOFTSTART parameter /sub-function zone //SOFTSTART

float xi[2]={0,0},error[3]={0,0,0};
float nout[3]={0,0,0},kin[3]={0,0,0},kout[3]={0,0,0};
//Plant in / out
long   nin[3]={0,0,0};

inline void profile(void) // (1/t*s+1)
{
      kout[2]=((1-
200*period)*kout[1]+200*period*(kin[2]+kin[1]))/(200*period+1);
    kout[2]=28;//?????
}

inline double G_func(double e,double K1,double K2, double delta)
{
      double y;
      if (e>delta) y=(K2*e+(K1-K2)*delta);
            else if(e<-delta) y=(K2*e-(K1-K2)*delta);
                  else y=(K1*e);

      if(K2<0 && e*y<0) y=0;
      return y;
 }

inline double fst2(double v[2], double u0, double mag)
{
    double d,d0,y,a0,a,fst;

    d=mag*period;
    d0=d*period;
    y=v[0]-u0+period*v[1];
    a0=sqrt(d*d+8*mag*fabs(y));

    if (y>d0) a=v[1]+(a0-d)/2;
      else if (y<(-d0)) a=(v[1]-(a0-d)/2);
            else a=v[1]+y/period;

    if (a>d) fst=-mag;
      else if (a<(-d)) fst=mag;
            else fst=-mag*a/d;

    return(fst);
}

inline void TD(double v[2], double u0, double mag)
{
    double x[2];

    x[0]=v[0]+period*v[1];
    x[1]=v[1]+period*fst2(v,u0,mag);

    v[0]=x[0];
    v[1]=x[1];
```

```
}

inline void npid(float pin[3], long pout[3])
{
      double out,int1,error_d,error_p,dif;
    double my_k_p,my_k_i,my_k_d;


    if(pin[2]>0)
    {   my_k_p=kp_up,my_k_i=ki_up;     }
    else
    {   my_k_p=kp_dn,my_k_i=ki_dn;   }

    error_p = G_func(pin[2],kp_k1,kp_k2,kp_d);

    TD(diff,nout[2]/28,k_d_r);

    dif = G_func(diff[1],1,0.2,k_d_Zone);


      pin[2]=G_func(pin[2],ki_k1,ki_k2,ki_d);
      xi[0]=xi[1];
      xi[1]=xi[0]+my_k_i*period*(pin[2]+pin[1])/2;


    cp=my_k_p*error_p;
    ci=xi[1];
    cd=- k_d * dif;
      out= cp+ ci +cd ;   //integrate delay, it's xi[0],not xi[1]

   if (out<0)     out=0;
      else if (out>1)   out=1;
      pout[2]=(long)(maxcount*out*240/256);
}

// SOFTSTART parameter /sub-function zone //SOFTSTART

inline void write_pwm(long phase_1,long phase_2)
{
      ds4002_dio_bit_out(DS4002_1_BASE, 0xffffffff,
0x04000000+((int)(phase_1)&0x1FFF) + (0x2000)*((int)(phase_2)&0x1FFF));
      //Ivan
}

inline void write_freq(void)
{
      long temp;

      maxcount = 50000.0/frequency;
      temp = 0x8000000 | (0x3FFFF & (int)(maxcount-2));
            ds4002_dio_bit_out(DS4002_1_BASE, 0x0C000000, 0x00000000);
            ds4002_dio_bit_out(DS4002_1_BASE, 0xffffffff, temp);
//Output the frequency
            ds4002_dio_bit_out(DS4002_1_BASE, 0x0C000000, 0x00000000);

}
```

```
inline void init_control(void)
{
    write_pwm(0,0);
    pwm_m=0;
    xi[0]=0,xi[1]=0;
    error[0]=0,error[1]=0,error[2]=0;
    nout[0]=0,nout[1]=0,nout[2]=0;
    kin[0]=0,kin[1]=0,kin[2]=0;
    kout[0]=0,kout[1]=0,kout[2]=0;
    nin[0]=0,nin[1]=0,nin[2]=0;

    current.V_in=0,current.V_out=0,current.C_out=0;
    last.V_in=0,last.V_out=0,last.C_out=0;
    total.V_in=0,
    total.V_out=0,total.C_out=0; n=0;
    diff[0]=0,diff[1]=0;
    for ( n=0;n<10;n++) {   data[n].V_in= 0 , data[n].V_out=0;
data[n].C_out=0; }
}

inline void read_data(void)
{
      // save data for next cycle...
      last = data[n];

      // read data from ADC
       data[n].V_in = (long)(k_Vin * ds2001_in(DS2001_1_BASE,3));
      // Voltage input channel 3
       data[n].C_out = (long)(k_Cout * ds2001_in(DS2001_1_BASE,5));
      // Voltage input channel 5
       data[n].V_out = (long)( k_Vout * ds2001_in(DS2001_1_BASE,1));
      // Voltage output channel 1

      //Start next conversion cycle
      ds2001_start (DS2001_1_BASE, DS2001_CVT_ALL);

    // calculate running average

       total.V_in += data[n].V_in - last.V_in ;
      total.V_out += data[n].V_out - last.V_out ;
      total.C_out += data[n].C_out - last.C_out ;

    current.V_in = total.V_in  / N_Sample ;
      current.V_out = total.V_out / N_Sample ;
      current.C_out = total.C_out / N_Sample ;

    vin_m=(float)(current.V_in) /100.0;
    vout_m=(float)(current.V_out) /100.0 -  bias_out;
    cout_m=(float)(current.C_out) /100.0 ;

      ++n;
      if ( n>= N_Sample ) n=0;
}


void isr_t0()                          /* timer0 interrupt service
routine */
```

```
{
  isr_t0_begin();                                        /* overload
check */

  host_service(1,0);                    /* call Host service  */
//  service_mtrace("0");              /* call TRACE service (TRACE >= Vs
3.1) */
  tic0_start();                         /* start execution time
measurement */
//  ds4002_dio_bit_out(DS4002_1_BASE, 0x01000000, 0x01000000);
      //dSPACE implementation

      // acquire data from converter
      read_data();

  if ( reset == 0)
  {

            if ( /*vin_m >= V_input_min && vin_m <= V_input_max*/ 1 )
            {
                  kin[0]=kin[1];kin[1]=kin[2];
                  kout[0]=kout[1];kout[1]=kout[2];
                  nin[0]=nin[1];nin[1]=nin[2];
                  nout[0]=nout[1];nout[1]=nout[2];
                  error[0]=error[1];error[1]=error[2];
                  profile();
                  nout[2]=vout_m;          //plant
                  error[2]=(kout[2]-nout[2])/28; //using for normal

                  npid(error,nin);

                  kin[2]=V_set;

                  if ( set_pwm >= 0 ) nin[2] = set_pwm;

                  if(nin[2]!=nin[1])
                  {     write_pwm(nin[2],nin[2]);
                        pwm_m=nin[2];
                  }
            }
            else
            {
                  init_control();
            }
    }
    else    init_control();

//  ds4002_dio_bit_out(DS4002_1_BASE, 0x01000000, 0x00000000);
      //dSPACE implementation

  exec_time = tic0_read();                        /* calculate execution
time */
  isr_t0_end();                         /* end of interrupt service
routine */
}
```

```
int main(void)
{
      // turn off converter
      init_control();
      atexit(init_control);

      init();

      ds2001_init(DS2001_1_BASE);

      ds4002_init(DS4002_1_BASE);
      ds4002_dio_init(DS4002_1_BASE, DS4002_OUT_0 + DS4002_OUT_1 +
DS4002_OUT_2);
      ds4002_dio_bit_out(DS4002_1_BASE, 0xffffffff,0x00060000);
      //dSPACE implementation
      ds2001_set_wordlen(DS2001_1_BASE, DS2001_CH_ALL,DS2001_LEN12);
      //dSPACE implementation

      write_freq();

      msg_info_set(MSG_SM_RTLIB, 0, "DC-DC Converter Control System
started.");

      isr_t0_start(period);                    /* initialize sampling
clock timer */
      ds2001_start (DS2001_1_BASE, DS2001_CVT_ALL);

      while(1)
      {
        isr_t0_disable();                    /* disable sampling clock
timer */

          while ( (reset ==1) || (msg_last_error_number() !=
MSG_NO_ERROR) )
            {
          if (reset ==1) {   init_control(); isr_t0_enable();
isr_t0_disable(); }
            host_service(0,0);
          }
         /* enable sampling clock timer */
          isr_t0_enable();

        while ( (reset ==0) && (msg_last_error_number() ==
MSG_NO_ERROR) )/* background process */
          {
            host_service(0,0);                              /* call
COCKPIT code */
          }

      }

      return 0;
}
```

D. C code for Stand-Alone System

```
/*=========================================================*
 | Name: control_globals.h
 | Creation Date: 05/04/2001
 | Modification: 05/07/2001
 |  Description: Constants and function prototypes needed for our controllers.
*=========================================================*/

/* Define the functions need for out controller.                    */
void WritePWM(double duty1, double duty2);

/* Define the local and global constants                                    */

/*Global values to change from watch or gel*/
int freq_max_count=2499;
int phase_max_count=2499;

volatile double SCALE_VOUT =  0.02445; /* Conv. scale for output voltage   */
volatile double offset = 5,bias = 0.0; /*global duty variable to change for a open loop
version  */

volatile double set_pwm=0.6858 ; /*Converter openloop PWM setting, negative means
closeloop.*/

/*volatile double period = 0.0001; */          /* Control loop period in seconds.
        */
double period ;  /*Control loop period in seconds; at initialization in main this is set to the
value set by DSP/Bios clk count  */
double exe_time ;  /*exectution time*/

/* Define local and global variables.
        */
int      N_SAMPLES =  16;            /* Num. samples to buffer & avg.      */
short ad_buffer[16];            /* Global buffer for AD data.          */

double Measured_Buffer;              /* Measured output voltage.
        */
int max_pwm_count = 5000; /* ??? 4095 for 12bits Maximum value that a PWM phase
can have.       */

const double V_input_min = 90;       /* minimum input voltage parameter*/
const double V_input_max = 135;      /* maximum input voltage parameter*/
```

```
/*============================================================*
 |      Name: control_algorithm.h
 |      Creation Date: 09/30/2001
 |      Modification: 09/30/2001
 |
 |      Description: Control Parameter
*============================================================*/

#define _PI
#define _NPI
#define _PID
#define _NPID

/* Define Control Parameter,please refer to Variable Description that locates at the end of
the file*/

/* for PI control */
#ifdef _PI
volatile double pi_k_p = 0.5 ;
volatile double pi_k_i = 300 ;
#endif

/* for NPI control */
#ifdef _NPI
volatile double npi_k_p_dn = 0.1 ,npi_k_p_up = 1.2 , npi_kp_k1 = 0.7 , npi_kp_k2 = 0.1,
npi_kp_d = 0.4 ;
volatile double npi_k_i_dn = 300 ,npi_k_i_up = 150 , npi_ki_d = 0.5;
#endif

/* for PID control */
#ifdef _PID
volatile double pid_k_p = 0.5 ;
volatile double pid_k_i = 300 ;
volatile double pid_k_d = 0.5 , pid_k_d_r =2e9 , pid_kd_d = 1500;
#endif

/* for NPID control */
#ifdef _NPID
volatile double npid_k_p_dn = 1 ,npid_k_p_up = 8 , npid_kp_k1 = 0.7 , npid_kp_k2 =
0.1, npid_kp_d = 0.4 ;
volatile double npid_k_i_dn = 350 ,npid_k_i_up = 800 , npid_ki_d = 0.5;
volatile double npid_k_d = 15 , npid_k_d_r =2e9 , npid_kd_d = 1500;
volatile double npid_k_i_d = 700 ,npid_k_i_u = 1000 , npid_ki_d = 0.5;
#endif
```

```
volatile int   reset=1 ;
volatile int   control_opt=3 ;
volatile int   SET_POINT = 28 ;

double cp,ci,cd,pwm_m;

#include "algorithms\_npid_zhu_11a.h"


/*****************************Variable Description**********************

---------------------
Startup, run and miscelaneous control
--------------
SCALE_VOUT        set the a2d factor value
Measured_Buffer   the measured output voltage
set_pwm           >0 manual mode of pwm duty ratio
                  -1 closed loop
reset             0 turn on the controller
                  1 turn off
offset            intput adjust for the zero input on the a2d
bias              output adjust for the set point (ex 28) output on the converter
pwm_m             the exact pwm output values (not really used)

cp                control proportional (control outputs)
ci                control integral (control outptus)
cd                control differential (control outputs)
control_opt       0     PI
                  1     NPI
                  2     PID
                  3     NPID

---------------------
PI controller settings
---------------------
pi_k_p            control proportianal for the PI controller
pi_k_p            control integral for the PI controller

---------------------
NPI controller settings
---------------------
James suffix definitions
_dn: possitive error global slope value
_up: negative error global slope value
k2: outer region slope factor (small error small change only for proportional)
k1: inner region slope factor (hi error big change  only for proportional)
```

_d: region set point of non linear error setting which sets inner and outer region

---

npi_k_p_dn	proportinal +suffix
npi_k_p_up	proportinal +suffix
npi_k_i_dn	integral +suffix
npi_k_i_up	integral +suffix
npi_kp_k1	proportinal +suffix
npi_ki_k2	integral +suffix
npi_kp_d	proportonal +suffix
npi_ki_d	integral +suffix

the td(tracking differential) acceleration can be used to set a psudo corner frequency for setting frequency region, the higher r is the higher the psuedo coner frequency

---

PID: the gains are linear even though the td is there

---

pid_k_p	proportional control
pid_k_i	integral control
pid_k_d	micro differentials:) (mulitplies value by 10^-6)
pid_k_d_r	

NPID: non linear gains

---

npid_k_p_dn	proportinal +suffix
npid_k_p_up	proportinal +suffix
npid_k_i_dn	integral +suffix
npid_k_i_up	integral +suffix
npid_k_d	micro differentials:) (mulitplies value by 10^-6)

----slope factor settings-----

npid_kp_k1	proportinal +suffix
npid_ki_k2	integral +suffix

---region settings------

npid_kp_d	proportional +suffix
npid_ki_d	integral +suffix
npid_kd_d	integral +suffix

```
npid_k_d_r     the td(tracking differential) acceleration can be used to set a psudo corner
frequency for setting frequency region, the higher r is the higher the psuedo coner
frequency

**********************************************************************/

/*=====================================================================*/
|         Ref: AERL-TDOC-0004A Stand Alone DSP Setup
|                        http://www.ti.com

|
|        Name: test_pid_01.c (PID controller test #1)
|  Creation Date: 04/28/2001
|   Modification: 05/08/2001
|
|    Description: This program is designed to test how a PID
|                               controller will run on the stand alone DSP
|                               system, which is the TI 6711 DSK. It will sample
|                               256 values from the THS1206 A/D at 6MHz every
|                               second.
|
|            Settings: Jumper settings for the THS1206 EVM to be used:
|             J1  1-2      J2  1-2      J3  2-5     J4  open
|             J5  open     J6  open     J7  1-2     J10 closed
|             J11 open     J12 2-3      J13 1-2
|
|                               Supply voltage from DSP, CLK from Timer 0,
|                               Input BNC connector = AINP
|                               AD converter address: 0xA0020000
|                               DSP/BIOS II and CSL used
*=====================================================================*/


/*define to do the output real time log to the trace display
#define dothe_logs
*/

/* Include files for data converter support.
        */
#include "dc_conf.h"
#include "t1206_fn.h"

/* Include files for DSP/BIOS.
        */
#include <std.h>
#include <swi.h>
```

```
#include <log.h>

/* Include files for chip support library (CSL).
        */
#include <csl.h>
#include <irq.h>
#include <timer.h>

/* Include files for control algorithm support.
        */
/* Change the include filename to load a new controller.                    */
/* Leave the globals.h as is at all times.
        */
#include "_control_globals.h"

#include "algorithms\_control_algorithm.h"

/* Function prototypes.
                        */
TIMER_HANDLE init_timer0(unsigned int period);
void init_dsk(void);

/* DSP/BIOS objects, created with the Config Tool
  In the "Project View" window, double click the "config.cdb"
  file under the "DSP/BIOS Config" folder.
        */
extern LOG_Obj trace;
extern far SWI_Obj SwiStartConversion;
extern far SWI_Obj SwiDoCalculation;

/*-------------------------------------------------------------------*
 | main
 | All execution starts here. Code in here is only executed once.
 *-------------------------------------------------------------------*/
void main(void)
{
   TIMER_HANDLE hTimer;

   /* CSL_Init - required for the CSL functions of the driver.              */
   CSL_Init();

   /* initialize the DSK and timer0's period.
        */
   init_dsk();
   hTimer = init_timer0(ADC1_TIM_PERIOD);
```

```
   /* configure the data converter
        */
   dc_configure(&Ths1206_1);

   /*set the global variable period to the appropriate value set inside DSP/bios*/
        period = (CLK_getprd()/37.5)/1000000;   /*37.5 prd/ms => divide by 1000000
for seconds*/


   /* start the timer
        */
   TIMER_Start(hTimer);

   /* Let's go... DSP/BIOS takes control and will generate                    */
   /* a "PeriodFunc" software interrupt every second.                */
}

/*function prototype*/
void toggle_led1(void);

void aa_test_func(){
        toggle_led1();
}

/*-----------------------------------------------------------------------*
 | PeriodFunc
 | The function will be called every second by DSP/BIOS and
 | posts a StartConversion SWI to start a new conversion.
 *-----------------------------------------------------------------------*/
void PeriodFunc()
{
        void StartConversionFunc(void);
   static int cnt=0;

   /*toggle_led1();*/
   #ifdef dothe_logs
   LOG_printf(&trace, "\nstart time = %d", (Int)CLK_gethtime());

   LOG_printf(&trace, "Period cnt: %d",cnt++);
   #endif

   exe_time = (Int)CLK_gethtime() ;
   SWI_post(&SwiStartConversion);
   exe_time -= (Int)CLK_gethtime() ;
   /*StartConversionFunc();   */
}
```

```
/*-------------------------------------------------------------------*
| BlockReady
| This function will be called when the dc_rblock routine is
| finished. It posts a DoCalculation software interrupt.
*-------------------------------------------------------------------*/
void BlockReady1206(void *pDC)
{
        void DoCalculationFunc(void);

  #ifdef dothe_logs
  LOG_printf(&trace, "1206 Interrupt");
  #endif
  SWI_post(&SwiDoCalculation);
  /*DoCalculationFunc();        */

}

/*-------------------------------------------------------------------*
| SwiStartConversionFunc
| This software interupt starts a new conversion using the
| dc_rblock function.
*-------------------------------------------------------------------*/
void StartConversionFunc()
{
  dc_rblock(&Ths1206_1, ad_buffer, N_SAMPLES, &BlockReady1206);
}

/*-------------------------------------------------------------------*
| SwiDoCalculationFunc
| This simple routine looks for the max and the min sample. A
| probepoint could be set to update a graphic display.
*-------------------------------------------------------------------*/
void DoCalculationFunc()
{
  int i,value,min,max;
  double myavg,measured_vout;

  min = ad_buffer[0] & 0x0FFF;
  max = ad_buffer[0] & 0x0FFF;

  for (i=0; i<N_SAMPLES; i++) {
    value = ad_buffer[i] & 0x0FFF;
    myavg += value;                                  /* Sum all 256 values.
      */
    if(value < min) min = value;
```

```
      if(value > max) max = value;
    }
    #ifdef dothe_logs
    LOG_printf(&trace, "min = %d   max = %d",min,max);
    #endif

    /* Average the 256 data point and then scale the value.                    */
    myavg = (myavg/N_SAMPLES);
    measured_vout = ((2048-offset-myavg)* SCALE_VOUT-bias);     /*148 offset
counts*/
          Measured_Buffer =  measured_vout;
  // if (measured_vout < 0) measured_vout=0;         /* No neg. numbers.   */

    #ifdef dothe_logs
    LOG_printf(&trace, "My Vout: %lf",measured_vout);
    #endif

  ControlAlgorithm(measured_vout);

}

/*-------------------------------------------------------------------*
 | WritePWM
 | The function will write the binary value of the phases given
 | to a designated location in the onboard SDRAM, which will
 | then make the data bits available on the EMIF bus.
 *-------------------------------------------------------------------*/

void WritePWM(double duty1, double duty2)
{
   int *mem_phase = (int *)0xB0000000;
   int *mem_freq =  (int *)0xB0000004;

   int phase1=200;
   int phase2=200;

   /*
   Both
   phase_max_count
   freq_max_count
   are defined as global variables for real time adjustment
   */

   phase1=phase_max_count*duty1;
   phase2=phase_max_count*duty2;
```

```
        *mem_phase = (phase2 << 16) | ( (phase1) & 0x0000FFFF) ;

        *mem_freq = freq_max_count;

  #ifdef dothe_logs
        LOG_printf(&trace, "end time = %d", (Int)CLK_gethtime());
        LOG_printf(&trace, "duty1 = %lf duty2= %lf", duty1,duty2);
        #endif
}

/*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==--=-=-
        Aaron test functions
-=-=-=-=-=-=-=-=-=-=-=-=-=-*/

 #define LED1_on        *(int *)0x90080000 = 0x0E000000
 #define LED2_on        *(int *)0x90080000 = 0x0D000000
 #define LED3_on        *(int *)0x90080000 = 0x0B000000
 #define LEDs_off       *(int *)0x90080000 = 0x07000000

 #define all_logic_low          *(int *)0x90080000 = 0x00000000
 #define only_tp16_on           *(int *)0x90080000 = 0x20000000
 #define only_tp17_on           *(int *)0x90080000 = 0x40000000
 #define only_tp18_on           *(int *)0x90080000 = 0x80000000


void toggle_led1(){
        static int led_on = 0;

  if(led_on){
        led_on=0;
        LED1_on;}
  else{
        led_on=1;
        LEDs_off;}

}

/* End of "main.c" program.
                        */

/*=============================================================*
 |
 |      Name: npid_zhu_11a.h (James NPID controller @ 11 bits)
 |      Compiler : Code Composer Studio 1.23
 |      DSP Chip : TI TMS320C6711 DSP
 |      Creation Date: 09/30/2001
```

```
|     Modification: 09/30/2001
|
|     Description: PI/PID/NPI/NPID control.
*============================================================*/

#include <math.h>

void ControlAlgorithm(double measured_vout);
extern void WritePWM(double control_duty1, double control_duty2);

/* state store buffer*/
double xi[2]={0,0},error[3]={0,0,0};
double nout[3]={0,0,0},kin[3]={0,0,0},kout[3]={0,0,0};
double nin[3]={0,0,0};
double diff[2];
double out=0;

/*------------------------------------------------------------*
 | ControlAlgorithm
 | The function does all the control computations and then outputs
 | its values to the PWM function.
 *------------------------------------------------------------*/

/* Softstart profile generator*/
inline void profile(void)  /* (1/t*s+1) */
{
        kout[2]=((1-200*period)*kout[1]+200*period*(kin[2]+kin[1]))/(200*period+1);
}

/* Nolinear gain modifier */
inline double fal_func(double e,double K1,double K2, double delta)
{
        if (e>delta) return(K2*e+(K1-K2)*delta);
                else if(e<-delta) return(K2*e-(K1-K2)*delta);
                        else return(K1*e);
}

/* Tracking Differentiator */
inline double fst2(double v[2], double u0, double mag)
{
   double d,d0,y,a0,a,fst;

   d=mag*period;
   d0=d*period;
   y=v[0]-u0+period*v[1];
   a0=sqrt(d*d+8*mag*fabs(y));
```

```
    if (y>d0) a=v[1]+(a0-d)/2;
          else if (y<(-d0)) a=(v[1]-(a0-d)/2);
                else a=v[1]+y/period;

    if (a>d) fst=-mag;
          else if (a<(-d)) fst=mag;
                else fst=-mag*a/d;

    return(fst);
}

void TD(double v[2], double u0, double mag)
{
    double x[2];

    x[0]=v[0]+period*v[1];
    x[1]=v[1]+period*fst2(v,u0,mag);

    v[0]=x[0];
    v[1]=x[1];
}

/* PI Control */
#ifdef _PI
inline void _pi()
{
        xi[0]=xi[1];
        xi[1]=xi[0]+pi_k_i*period*(error[2]+error[1])/2;

        cp=pi_k_p*error[2];
        ci=xi[1];
        out= cp+ ci ;
}
#endif

/* NPI Control */
#ifdef _NPI
inline void _npi()
{
   double error_p;
   double my_k_p,my_k_i;

   if(error[2]>=0)
   {   my_k_p=npi_k_p_dn,my_k_i=npi_k_i_dn;    }
   else
```

```
    {  my_k_p=npi_k_p_up,my_k_i=npi_k_i_up;  }

    error_p = fal_func(error[2],npi_kp_k1,npi_kp_k2,npi_kp_d);

    if(fabs(error[2])>npi_ki_d)  error[2]=0;
    xi[0]=xi[1];
    xi[1]=xi[0]+my_k_i*period*(error[2]+error[1])/2;

    cp=my_k_p*error_p;
    ci=xi[1];
    out= cp+ ci ;
}
#endif

/* PID Control */
#ifdef _PID
inline void _pid()
{
    double dif=0;

    TD(diff,nout[2],pid_k_d_r);
    dif = fal_func(diff[1],1,0.2,pid_kd_d);

    xi[0]=xi[1];
    xi[1]=xi[0]+pid_k_i*period*(error[2]+error[1])/2;

    cp=pid_k_p*error[2];
    ci=xi[1];
    cd=-pid_k_d * dif*1e-6;
    out= cp+ ci ;
}
#endif

/* NPID Control */
#ifdef _NPID
inline void _npid()
{
    double error_p,dif=0;
    double my_k_p,my_k_i;

    if(error[2]>=0)
    {  my_k_p=npid_k_p_dn,my_k_i=npid_k_i_dn;   }
    else
    {  my_k_p=npid_k_p_up,my_k_i=npid_k_i_up;  }

    error_p = fal_func(error[2],npid_kp_k1,npid_kp_k2,npid_kp_d);
```

```
    TD(diff,nout[2],npid_k_d_r);

    dif = fal_func(diff[1],1,0.2,npid_kd_d);

    if(fabs(error[2])>npid_ki_d)  error[2]=0;
    xi[0]=xi[1];
    xi[1]=xi[0]+my_k_i*period*(error[2]+error[1])/2;

    cp=my_k_p*error_p;
    ci=xi[1];
    cd=-npid_k_d * dif*1e-6;
    out= cp+ ci +cd ;
}
#endif

/* To Initial status */
inline void init_control(void)
{
        WritePWM(0.01,0.01);
        pwm_m=0;
        xi[0]=0,xi[1]=0;
        error[0]=0,error[1]=0,error[2]=0;
        nout[0]=0,nout[1]=0,nout[2]=0;
        kin[0]=0,kin[1]=0,kin[2]=0;
        kout[0]=0,kout[1]=0,kout[2]=0;
        nin[0]=0,nin[1]=0,nin[2]=0;
        diff[0]=0,diff[1]=0;
}

void ControlAlgorithm(double measured_vout)
{
 if ( reset == 0) /* Controller start */
 {
        if ( /*vin_m >= V_input_min && vin_m <= V_input_max*/ 1 )   /* Control loop
*/
        {
                /*  state buffer updating */
                kin[0]=kin[1];kin[1]=kin[2];        /* Before Profile */
                kout[0]=kout[1];kout[1]=kout[2];    /* After Profile */
                nin[0]=nin[1];nin[1]=nin[2];        /* Before plant */
                nout[0]=nout[1];nout[1]=nout[2];    /* After plant */
                error[0]=error[1];error[1]=error[2]; /* Error input */

                profile();
                error[2]=(kout[2]-nout[2])/28;
```

```
                switch(control_opt)
                {
                        #ifdef _PI
                                case 0: _pi(); break;
                        #endif

                        #ifdef _NPI
                                case 1: _npi(); break;
                        #endif

                        #ifdef _PID
                                case 2: _pid(); break;
                        #endif

                        #ifdef _NPID
                                case 3: _npid(); break;
                        #endif
                }

                if (out<0)      out=0;
                else if (out>1) out=1;
                nin[2]=out*240.0/256.0;

                nout[2]=measured_vout;                  /*plant*/
                kin[2]=SET_POINT;

                if ( set_pwm >= 0 ) nin[2] = set_pwm;   /* Openloop setting */

                WritePWM(nin[2], nin[2]);
                pwm_m=nin[2];
        }
        else    init_control();
 }
 else   init_control();
}
```

E. VC++ code for test control panel

```
////////////////////////////////////////////////////////////////
//
// DEVICE_ADDRESS line below to specify the address of the
// device you want to talk to.  For example:
//
//    hpib7,0    - refers to an HP-IB device at bus address 0
//                 connected to an interface named "hpib7" by the
//                 I/O Config utility.
//
////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include "HPELoad.h"
#include "HPELoadDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

#include "C:\Program Files\Agilent\IO Libraries\c\sicl.h"

/*************************************************************************
***/
#define HPELoad_DEVICE_ADDRESS "hpib7,4"
#define DHPPower_DEVICE_ADDRESS "hpib7,6"
/*************************************************************************
***/

class CAboutDlg : public CDialog
{
public:
        CAboutDlg();

        //{{AFX_DATA(CAboutDlg)
        enum { IDD = IDD_ABOUTBOX };
        //}}AFX_DATA

        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CAboutDlg)
        protected:
```

```
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
        //}}AFX_VIRTUAL

// Implementation
protected:
        //{{AFX_MSG(CAboutDlg)
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
        //{{AFX_DATA_INIT(CAboutDlg)
        //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(CAboutDlg)
        //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
        //{{AFX_MSG_MAP(CAboutDlg)
                // No message handlers
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CHPELoadDlg dialog

CHPELoadDlg::CHPELoadDlg(CWnd* pParent /*=NULL*/)
        : CDialog(CHPELoadDlg::IDD, pParent)
{
        //{{AFX_DATA_INIT(CHPELoadDlg)
        // Initial the setting
        m_dCHTotal = 0.0;
        m_dHigher = 20.0;
        m_dLower = 3.0;
        m_dCH1 = 0.0;
        m_dCH2 = 0.0;
        m_dLineVoltage = 0.0;
        m_dVoltage = 120.0;
        Test_Option=0; // 1: regular, 2: Random 3: Hysteric 4: Dual
        //}}AFX_DATA_INIT
```

```
        // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CHPELoadDlg::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(CHPELoadDlg)
        DDX_Text(pDX, IDC_CHTOTAL, m_dCHTotal);
        DDX_Text(pDX, IDC_HIGHER, m_dHigher);
        DDV_MinMaxDouble(pDX, m_dHigher, 0., 40.);
        DDX_Text(pDX, IDC_LOWER, m_dLower);
        DDV_MinMaxDouble(pDX, m_dLower, 0., 36.);
        DDX_Text(pDX, IDC_CH1, m_dCH1);
        DDX_Text(pDX, IDC_CH2, m_dCH2);
        DDX_Text(pDX, IDC_LINEVOLTAGE, m_dLineVoltage);
        DDX_Text(pDX, IDC_VOLTAGE, m_dVoltage);
        DDV_MinMaxDouble(pDX, m_dVoltage, 0., 140.);
        //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CHPELoadDlg, CDialog)
        //{{AFX_MSG_MAP(CHPELoadDlg)
        ON_WM_SYSCOMMAND()
        ON_WM_PAINT()
        ON_WM_QUERYDRAGICON()
        ON_BN_CLICKED(IDC_REGULAR, OnRegular)
        ON_BN_CLICKED(IDC_RANDOM, OnRandom)
        ON_BN_CLICKED(IDC_UPDATEDATA, OnUpdatedata)
        ON_WM_TIMER()
        ON_BN_CLICKED(IDC_STOP, OnStopPower)
        ON_BN_CLICKED(IDC_DUAL, OnDual)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CHPELoadDlg message handlers

BOOL CHPELoadDlg::OnInitDialog()
{
        CDialog::OnInitDialog();

        // Add "About..." menu item to system menu.

        // IDM_ABOUTBOX must be in the system command range.
        ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
```

```
        ASSERT(IDM_ABOUTBOX < 0xF000);

        CMenu* pSysMenu = GetSystemMenu(FALSE);
        if (pSysMenu != NULL)
        {
                CString strAboutMenu;
                strAboutMenu.LoadString(IDS_ABOUTBOX);
                if (!strAboutMenu.IsEmpty())
                {
                        pSysMenu->AppendMenu(MF_SEPARATOR);
                        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);
                }
        }

        // Set the icon for this dialog.  The framework does this automatically
        //  when the application's main window is not a dialog
        SetIcon(m_hIcon, TRUE);                 // Set big icon
        SetIcon(m_hIcon, FALSE);          // Set small icon


  // Install a default SICL error handler that logs an error message and
  // exits.  On Windows 95 view messages with the SICL Message Viewer,
  // and on Windows NT use the Windows NT Event Viewer.
  ionerror(I_ERROR_EXIT);

  // Open a device session using the DEVICE_ADDRESS
  id = iopen(HPELoad_DEVICE_ADDRESS);
  id1 = iopen(DHPPower_DEVICE_ADDRESS);

  // Set the I/O timeout value for this session to 1 second
  itimeout(id, 1000);
  itimeout(id1, 1000);

  // Write the *RST string (and send an EOI indicator) to put the instrument
  // in a known state.

  iprintf(id, "*CLS\n");
  iprintf(id1, "*CLS\n");
  iprintf(id, "*RST\n");
  iprintf(id1, "*RST\n");

/**************************************************************************
***/
        // Set reading window updating period
        m_nTimer= SetTimer(1,1000,NULL);
```

```
        ASSERT(m_nTimer !=0 );
        srand( (unsigned)time( NULL ) );
        //GetDlgItem(ID_GETDATA)->EnableWindow(FALSE);
/**************************************************************************
***/

        return TRUE;  // return TRUE  unless you set the focus to a control
}

void CHPELoadDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
        }
        else
        {
                CDialog::OnSysCommand(nID, lParam);
        }
}

void CHPELoadDlg::OnPaint()
{
        if (IsIconic())
        {
                CPaintDC dc(this); // device context for painting

                SendMessage(WM_ICONERASEBKGND, (WPARAM)
dc.GetSafeHdc(), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics(SM_CXICON);
                int cyIcon = GetSystemMetrics(SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width() - cxIcon + 1) / 2;
                int y = (rect.Height() - cyIcon + 1) / 2;

                // Draw the icon
                dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
                CDialog::OnPaint();
        }
```

```
}

// The system calls this to obtain the cursor to display while the user drags
//  the minimized window.
HCURSOR CHPELoadDlg::OnQueryDragIcon()
{
        return (HCURSOR) m_hIcon;
}


void CHPELoadDlg::OnRegular()
{
        Test_Option=1;
}

void CHPELoadDlg::OnRandom()
{
        Test_Option=2;
}

void CHPELoadDlg::OnDual()
{
        Test_Option=3;
}

void CHPELoadDlg::OnUpdatedata()
{
        Test_Option=0;

        while(!UpdateData (TRUE)) return;

/*************************************************************************
***/

        // Set synchronous toggled transient operation

  iprintf (id, "%s\n", "MEAS:CURR:DC?");
  UpdateData  (TRUE);
  iscanf (id, "%lf", &m_dCHTotal);
  UpdateData  (FALSE);

  iprintf(id, "CHAN 1;:INPUT OFF\n");
  iprintf(id, "MODE:CURR\n");

  strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
  iprintf(id, strText);
```

```
  strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
  iprintf(id, strText);

  iprintf(id, "TRAN:MODE TOGG\n");
  iprintf(id, "TRAN ON;:INPUT ON\n");
  iprintf(id, "CHAN 2;:INPUT OFF\n");
  iprintf(id, "MODE:CURR\n");

  strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
  iprintf(id, strText);

  strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
  iprintf(id, strText);

  iprintf(id, "TRAN:MODE TOGG\n");
  iprintf(id, "TRAN ON;:INPUT ON\n");
  iprintf(id, "TRIG:TIM 4\n");
  iprintf(id, "TRIG:SOUR TIM\n");

  iprintf(id1, "SOUR:CURR 18.0\n");
  strText.Format("%s %f %s", "SOUR:VOLT",m_dVoltage, "\n");
  iprintf(id1, strText);

/*************************************************************************
***/

}

BOOL CHPELoadDlg::DestroyWindow()
{

  KillTimer(m_nTimer);
  OnStopPower();
  iclose(id);
  iclose(id1);

  // For WIN16 programs, call _siclcleanup before exiting to release
  // resources allocated by SICL for this application.  This call is a
  // no-op for WIN32 programs.
  _siclcleanup();

  return CDialog::DestroyWindow();
}

void CHPELoadDlg::OnTimer(UINT nIDEvent)
```

```
{
        static count= 0, hysteric_connt = 0;

        if(++count > 5 && Test_Option)
        {
                count = 0;


                switch(Test_Option)
                {
                        case 1: Regular();      break;
                        case 2: Random();       break;
                        case 3: Dual(); break;
                }
        }
        else
        {

                // Read  new data via IEEE-488.2 GPIB
                iprintf (id1, "%s\n", "MEAS:VOLT?");
                iscanf (id1, "%lf", &m_dLineVoltage);
                strText.Format("%lf",m_dLineVoltage);
                SetDlgItemText (IDC_LINEVOLTAGE,strText);

                iprintf(id, "CHAN 1\n");
                iprintf (id, "%s\n", "MEAS:CURR:DC?");
                iscanf (id, "%lf", &m_dCH1);
                strText.Format("%lf",m_dCH1);
                SetDlgItemText (IDC_CH1,strText);

                iprintf(id, "CHAN 2\n");
                iprintf (id, "%s\n", "MEAS:CURR:DC?");
                iscanf (id, "%lf", &m_dCH2);
                strText.Format("%lf",m_dCH2);
                SetDlgItemText (IDC_CH2,strText);

                m_dCHTotal = m_dCH1 + m_dCH2 ;
                strText.Format("%lf",m_dCHTotal);
                SetDlgItemText (IDC_CHTOTAL,strText);
        }
}

void CHPELoadDlg::OnStopPower()
{
        Test_Option=0;
```

```
        // reset setting
  iprintf(id1, "*CLS\n");
  iprintf(id1, "*RST\n");
}

void CHPELoadDlg::OnOK()
{
        // TODO: Add extra validation here

        UpdateData (TRUE);
//      CDialog::OnOK();
}

// gradually increse and decrese by 1 between 3A and 36 A on electric load
void CHPELoadDlg::Regular()
{

        static int increment = 2,direct = 1;

        while(!UpdateData (TRUE)) return;


        if(increment == 3 )     direct = 1;
        if(increment == 36 )    direct = 0;
        if(direct)          increment ++;
        else     increment --;

        if(increment < 3 || increment > 36)     increment = 3;

        m_dLower = increment;
        m_dHigher = m_dLower;
        m_dVoltage = 120;
        UpdateData  (FALSE);

  iprintf(id, "CHAN 1;:INPUT OFF\n");
  iprintf(id, "MODE:CURR\n");

  strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
  iprintf(id, strText);

  strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
  iprintf(id, strText);

  iprintf(id, "TRAN:MODE TOGG\n");
  iprintf(id, "TRAN ON;:INPUT ON\n");
  iprintf(id, "CHAN 2;:INPUT OFF\n");
```

```
 iprintf(id, "MODE:CURR\n");

 strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
 iprintf(id, strText);

 strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
 iprintf(id, strText);

 iprintf(id, "TRAN:MODE TOGG\n");
 iprintf(id, "TRAN ON;:INPUT ON\n");
 iprintf(id, "TRIG:TIM 100\n");
 iprintf(id, "TRIG:SOUR TIM\n");

 iprintf(id1, "SOUR:CURR 15.0\n");
 strText.Format("%s %f %s", "SOUR:VOLT",m_dVoltage, "\n");
 iprintf(id1, strText);

}

// Random setting of electric load
void CHPELoadDlg::Random()
{

        int increment ;

        while(!UpdateData (TRUE)) return;

        increment = rand()%34+3;
        if(increment < 3 || increment > 36 )    increment =3;

        m_dLower = increment;
        m_dHigher = m_dLower;
        m_dVoltage = 120;
        UpdateData  (FALSE);

  iprintf(id, "CHAN 1;:INPUT OFF\n");
  iprintf(id, "MODE:CURR\n");

  strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
  iprintf(id, strText);

  strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
  iprintf(id, strText);

  iprintf(id, "TRAN:MODE TOGG\n");
  iprintf(id, "TRAN ON;:INPUT ON\n");
```

```
  iprintf(id, "CHAN 2;:INPUT OFF\n");
  iprintf(id, "MODE:CURR\n");

  strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
  iprintf(id, strText);

  strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
  iprintf(id, strText);

  iprintf(id, "TRAN:MODE TOGG\n");
  iprintf(id, "TRAN ON;:INPUT ON\n");
  iprintf(id, "TRIG:TIM 100\n");
  iprintf(id, "TRIG:SOUR TIM\n");

  iprintf(id1, "SOUR:CURR 15.0\n");
  strText.Format("%s %f %s", "SOUR:VOLT",m_dVoltage, "\n");
  iprintf(id1, strText);

}

// Random setting on eletric load and supply power
void CHPELoadDlg::Dual()
{
        int increment,input ;

        while(!UpdateData (TRUE)) return;

        increment = rand()%34+3;
        if(increment < 3 || increment > 36 )    increment =3;

        input = rand()%41+100;
        if(input < 100 || input > 140 ) input =120;

        m_dLower = increment;
        m_dHigher = m_dLower;
        m_dVoltage = input;
        UpdateData  (FALSE);

  iprintf(id, "CHAN 1;:INPUT OFF\n");
  iprintf(id, "MODE:CURR\n");

  strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
  iprintf(id, strText);

  strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
  iprintf(id, strText);
```

```
    iprintf(id, "TRAN:MODE TOGG\n");
    iprintf(id, "TRAN ON;:INPUT ON\n");
    iprintf(id, "CHAN 2;:INPUT OFF\n");
    iprintf(id, "MODE:CURR\n");

    strText.Format("%s %f %s", "CURR",m_dLower/2, "\n");
    iprintf(id, strText);

    strText.Format("%s %f %s", "CURR:TLEV",m_dHigher/2, ";SLEW MAX\n");
    iprintf(id, strText);

    iprintf(id, "TRAN:MODE TOGG\n");
    iprintf(id, "TRAN ON;:INPUT ON\n");
    iprintf(id, "TRIG:TIM 100\n");
    iprintf(id, "TRIG:SOUR TIM\n");

    iprintf(id1, "SOUR:CURR 15.0\n");
    strText.Format("%s %f %s", "SOUR:VOLT",m_dVoltage, "\n");
    iprintf(id1, strText);

}
```

Final Report

Volume II

Reference Documentation

Development of Digital Control Technology for Power Management and Distribution Systems

Grant No. NCC3 – 699

Submitted to NASA Glenn Research Center

Submitted by:

Jack Zeller, Senior Research Scientist
Zhiqiang Gao, Principle Investigator

Department of Electrical and Computer Engineering
Fenn College of Engineering
Cleveland State University
Cleveland, Ohio 44115

December 11, 2001

Overview of Volume II

This volume of the final report contains six (6) reference documents that provide detailed documentation to support the abbreviated technical summaries of Volume I. Included are: two (2) conferences publications (References [1] & [2]), one (1) contractor-type engineering report (Reference [3]), and three (3) Master's Theses (References [4], [5], & [6]). One of these thesis (Ref. [4]) has already been defended (December 2000). One (Ref. [5]) will be defended in mid-Dec 2001, while the third thesis (Ref [6]) will be defended in early 2002.

As included, References [4] and [5] are final drafts required minimal modifications before they are ready for publication. Reference [6], as included herein, is at a stage where extensive editing remains to be done. However, all included reference documentation is in a form which serves the intent Volume II. Each provides accurate technical detail for those specific sections of Volume I which only provide summaries or overviews of a particular research item.

Reference List

[1] John Sustersic, et al.; "Design and Implementation of a Digital Control For DC-to-DC Power Converters"; 2000 SAE Power Systems Conference, San Diego, CA, Oct. 31, 2000. (Paper 00PS – 75).

[2] Jack Zeller, Minshao Zhu, Tom Stimac, and Zhiqiang Gao; "Nonlinear Digital Control Implementation for a DC-to-DC Power Converter'; Proceedings of IECEC'01 36th Intersociety Energy Engineering Conference, July 29 – August 2, 2001, Savannah, Georgia. (Paper IECEC2001-AT-64).

[3] Arthur Stachowicz, Aaron Radke, Greg Tollis, Ivan Jurcic, et al.; "Stand-alone DSP Control System Documentation"; AERL-MNUL-0001B, Publication of the Advanced Engineering Laboratory, Cleveland State University.

[4] Tomislav J. Stimac; "Digital Control of a 1- Kw Switching Power Converter"; MSEE Thesis, Cleveland State University, December, 2000.

[5] Marcelo C. Gonzalez; "Simulation and Mathematical Modeling of a DC-DC Power Converter"; MSEE Thesis, Cleveland State University, December, 2001.

[6]] Minshao Zhu; "A Nonlinear Digital Control Implementation for a 1 – Kw PWM DC-DC switching Power Converter"; MSEE Thesis, Cleveland State University, March, 2002.

1.0 Introduction

In May 1999, Cleveland State University's Electrical and Computing Engineering Department (ECE) was awarded a research Grant (NCC3-699) by the NASA Glenn Research Center (GRC) to study digital control of DC - DC converters for space power applications. Presently most DC - DC power converters accomplish their control capability using analog circuit techniques. Thus they are limited in providing a wide range of adaptability, communication, and health monitoring that will be needed as space power system complexities and requirements increase. The objective of this research is the investigation of the issues and benefits related to implementing intelligent digital computer control of power converters and full power management and distribution (PMAD) systems for space platforms.

Upon receiving this Grant, CSU put together a faculty/student team and dedicated experimental laboratory space to conduct this research. The facility was designated as the Advanced Engineering Research Laboratory (AERL) and the team identified as the AERL team. One major research thrust of the grant activity will be the study of the benefits of nonlinear control strategies on power converter regulatory control. Nonlinear control is an area heavily developed by CSU recently for many industrial applications. NASA-GRC wanted this technology investigated as part of the digital controls research. The tasks undertaken to accomplish this thrust and others included: converter modeling, linear and nonlinear control design, experimental controller evaluation, power converter circuit design, real-time software design, etc. The members of the AERL team, therefore, were carefully selected to provide the skill mix needed to carry out these tasks.

This final report will document the activities and technical accomplishments of this 30+ month effort. It will also identify future tasks that would continue to advance the technology of digital power converter control. The report will be divided into two volumes.

Volume I will provide summaries of each of the major technical accomplishments of this Grant effort. This volume will include the following:

1) A summary table of the milestones and the time frame in which they were achieved.

2) A description of the AERL laboratory and research team.

3) An overview of the power converter modeling and simulation support studies.

4) A description of the AERL PWM generation methodology and implementations.

5) Design summary of the operational digital control algorithms.

6) Design description of a stand-alone DSP-based converter controller unit.

7) Design of a modular medium power converter unit (for parallel converter studies).

8) Discussion of future converter and PMAD controller research tasks.

Volume II will contain detailed documentation on all of the research activities completed under this Grant. It will include: technical papers, contractor-type AERL engineering reports and several Master theses. Each enclosure in Volume II will be identified as a numbered reference document and will be referred to as such in the Volume I discussions. Both Volumes will be made available in printed and electronic media formats

2.0 Table of Completed Milestones

Table 2 identifies the major milestones that have been completed during the research period. Included are the completion dates for each milestone.

Table 2 - Completed Milestones	Date
Literature review of digital power converter	8/31/99
Characterization and modeling of the existing analog controller	10/15/99
Setting up the DSP-based digital control platform for DC-DC power converters	11/15/99
Digital Control Design and Simulation Completed	11/30/99
Hardware-in-the-loop Simulation Completed	1/8/00
Native C code developed on dSpace platform, sampling raised to 20 kHz	6/1/00
Linear Transfer Function Model obtained	6/1/00
Successful Full-power operation of Digital Converter completed	7/28/00
DSP-board design and layout completed	8/31/00
A technical paper presented at the SAE power conference	11/1/00
Nonlinear Control Algorithm designed and tested successfully in simulation	12/20/00
Nonlinear Simulation Model obtained	4/1/01
12 bits Variable Frequency PWM Board Completed	4/1/01
Nonlinear PID Algorithm Implementation completed	4/4/01
State Space Mathematical Model obtained	6/1/01
A Paper on NPI Control of the Converter is presented at IECEC	8/1/01
A Stand alone DSP Controller Design, Built, and Tested	8/21/01
A Modular DC-DC converter designed and built	9/31/01
Signal Conditioning Board Completed	10/1/01
GUI Interface Set Up for the DSP Controller Using Labview	11/25/01

3.0 Laboratory Facility and AERL Research Team

3.1 Description of AERL Laboratory -- In order to provide an effective research environment, CSU's ECE department allocated one of its laboratories to this project to function as a combined laboratory and office in which to conduct this research. This facility has been designated as the Advanced Engineering Research Laboratory (AERL). In order to conduct realistic experimental research, NASA provided to CSU a Westinghouse-designed 1 KW "brassboard" power converter. This SMPS unit was designed to accept an input voltage between 100 and 160 volts DC and provide a regulated and isolated output DC voltage of 28 volts for loads up to 36 Amps. Galvanic voltage isolation was obtained with a step-down (3:1) transformer whose primary winding was pulse-width-modulated (PWM) with an H-bridge switching configuration of power MOSFET transistors. The lower voltage secondary winding was rectified and filtered to provide the 28 volt DC output. Pulse-width-modulation (PWM) of the switching devices was used to accomplish closed loop voltage regulation. The analog PWM generation circuitry and analog controller circuitry were removed, since the intent of the research is to accomplish these two functions digitally.

It was decided that a DSP-based digital system would be used rather than a microcontroller approach. Equipment needed to support this approach was put into place and configured to realize a versatile research environment. The DSP development system selected was dSpace Inc.'s [1] rapid-prototyping development system. This system is equipped with a high-performance TI DSP chip, A/D conversion capability as well as digital I/O circuitry. To expedite the development and evaluation of digital control strategies, Mathwork's Matlab/Simulink/Real-Time Workshop toolbox software was selected. Simulink provides the ability to model and accurately simulate the transient performance of dynamic processes to arrive at a set of acceptable closed loop control strategies. Mathworks' Real-Time Workshop will convert a controller, modeled in Simulink, into 'C" code which will run on dSpace's DSP processor to control actual experimental hardware (in this case the 1 KW Westinghouse power converter). This is termed hardware-in-the-loop simulation. The control laws can also be programmed in native "C" code. Then dSpace's compiler and libraries were used to generate the code for

the TI DSP chip on dSpace's processor board. This second approach has been found to generate faster operating real-time control code.

To complete the experimental research facility, appropriate test equipment was acquired. This included: power supplies, signal generators, digital voltmeters, digital oscilloscopes, and an electronic load bank. The photograph in Figure 1 shows how this array of equipment is configured in our facility. A more detailed description of all of this equipment is included in [1].

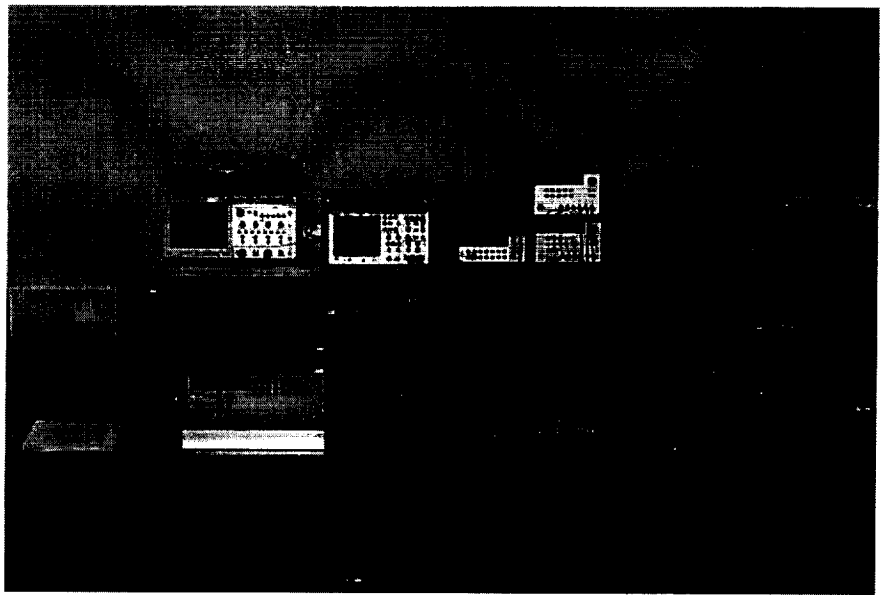


Figure 2. Photograph of AERL Experimental Equipment
For Digital Control Development

3.2 AERL Research Team -- The team assembled to conduct the NASA sponsored converter digital control research work at CSU is unique in terms of the makeup of its student researchers and faculty advisors/supervisors. The AERL team has as many as four experienced graduate student research assistants. Some are former CSU undergraduate students while some come from other educational institutions and from industry. In addition to these graduate assistants, CSU has complemented the team with an equal number of talented undergraduate engineering students in their final year of undergraduate study. Even though they are employed only part-time throughout the year, they have played a valuable role in the research accomplishments of the team. They have applied knowledge and skills obtained in recent course work to solve many of the

project's difficult design problems. The unique blend of talents and the teamwork exhibited by the team members has resulted in the grant's research successes thus far.

It should be pointed out that over the 30+ month duration of this grant at least one generation of student researchers have received their degrees and graduated. The transition of work assignments to the next generation of student researchers has occurred seamlessly. This is further evidence of the enthusiasm and teamwork that has become part of the AERL team.

The team is guided by two senior faculty advisors with broad experience in modern feedback control system design methodology as well as experience in design and testing of power electronic circuitry. Each faculty member has experience managing a variety of research projects. Such experiences are a valuable asset in guiding the efforts of a large part-time team of dedicated university student researchers.

Finally the AERL team utilizes outside industrial expertise as warranted by specific project tasks. A broad knowledge of the Northeast Ohio industrial base allows us to easily identify the needed experts.

4.0 Overview of Modeling and Simulation Activities

4.1 Modeling Overview --Simulation and modeling play an important role in any design activity especially the design of feedback control systems. Thus the development of complete analytical models and computer simulations of the Westinghouse one Kw DC-to-DC power converter was a major activity in this research grant. To expedite the development and design of the converter digital control strategies, two distinct modeling activities were conducted.

The first effort resulted in the development of a linear transfer function model based upon power converter experimental data. The resultant model and its development are reported in [4]. The linear transfer function model was included in a Simulink computer simulation to study new converter digital control strategies. The simulation enables the control design efforts to proceed effectively. The results of the simulation studies and the performance resulting from the digital control laws developed are reported in [2] and [6].

A second, more in-depth mathematical modeling and computer simulation effort was undertaken to provide converter computer models with which converter power circuit topology and component tradeoffs could be evaluated in effectivly. This work is documented in [5]. A summary of this effort follows.

4.1 Converter Circuit Modeling -- The main thrust of this work was to derive two nonlinear models for the 1-kW ED408043-1 Westinghouse Full-Bridge DC-DC Power Converter, with a center-tapped transformer, as part of a converter digital control study. The models will be used in the evaluation of nonlinear and linear control strategies.

The first model is a nonlinear SABER® simulation model. This is a component-level model and, as such, it also lends itself to the study of performance tradeoffs due to modifications in the converter topology. The second model is a circuit-level, piecewise-linear, mathematical model implemented in MATLAB®. Due to its mathematical nature, this model is better suited for nonlinear mathematical analysis. Both the simulation model and the mathematical model are valid for Continuous Conduction Mode (CCM) and Discontinuous Conduction Mode (DCM) of operation. For comparison purposes, a linear

transfer function model was also derived. However, this third model is only valid for the neighborhood about the operating point for which it was derived.

The SABER® model yielded the best overall steady-state and transient response results. For example, its steady-state output voltage was within 1.5% of the actual output voltage, while the other two models predicted steady-state output voltages within 4% of the actual values. Of all three models, the Matlab mathematical model requires the longest computing time due to the fixed integration step, but yielded the second best overall results.

5.0 PWM Generation Methodology and Implementations

5.1 PLD PWM Generation--As stated earlier in this report, NASA provided to the CSU-AERL team a 1.0 Kw power converter that utilized analog circuitry for voltage regulation, overcurrent protection, and generation of the pulse-width-modulation (PWM) switching functions. The AERL team removed this analog circuitry and replaced it with a digital controller architecture. A DSP device (as opposed to a microcontroller) was chosen for the controller's intelligence. As part of this digital architecture a decision was made by one of the original AERL researchers to delegate the PWM signal generator task to a programmable logic device (PLD). An Altera Complex Programmable Logic Device (CPLD) was selected to accomplish this task.. The intent of the design decision to separate the PWM signal generation task from the DSP was twofold: 1) to remove a potentially large computational task from the intelligent processor's control workload and 2) to provide an architecture in which pulse-width switching sinals would continue to be provided (at a fixed pulse width) even if the processor failed.

5.2 Converter PWM Signal Generator Requirements --The converter's PWM generator's task is to provide two variable width pulses (pulse1 and pulse2) whose job it is to turn on and off the power MOSFET switches of the converter's H-Bridge power circuitry. (See figure 5.1). Each pulse will have a maximum width of slightly less than half of the period of the modulating frequency with pulse1 being 180 degrees out of phase with pulse2. The PWM signal generator circuitry will also be required to provide a deadband to prevent both switches in the same leg of figure 5.1's H-bridge from being turned on at the same time. Such an occurrence would have disastrous consequences for the power circuitry.

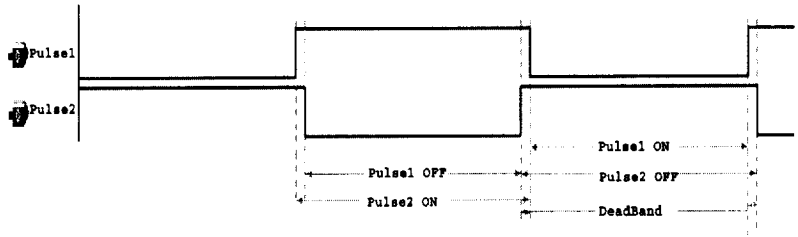


Figure #1: Pulses and DeadBand

5.3 Design History --The AERL's first operational PLD-based PWM signal generator design provided a fixed frequency pulse-width modulator operating at 20 kHz with 8 bits of quantization for each of the two output pulses. It also provided the necessary deadband interval. Early tests, however, showed that an eight (8) bit quantization (being able to resolve each pulse into 255 elements) was too course a resolution for good high-performance closed loop voltage control. Thus design activity were undertaken to produce an improved PWM generator while still using the CPLD device available.

5.3 Improved PWM Generator Designs -- Two new designs were accomplished, programmed and verified experimentally with the converter hardware. Each design was capable of at least 12 bits of quantization and had the capability of providing variable frequency pulse generation. One of the designs used less CPLD logic and provided for cooler chip operation. It was selected more than a year ago and has been used for all control studies since that time. It has become part of the Stand-alone DSP-based Converter Controller reviewed in Section 7.0 of this report. Reference 3 found in Volume II of this report has a section (Section III) devoted to a detailed technical discussion of this improved CPLD-based PWM generator now in use.

The AERL researchers believe that the design decision to use a separate PLD device (CPLD, FPGA, etc.) for PWM generation provides an avenue for efficiently researching many potential converter digital control innovations. One example is phase stagger for converters made up of multiple power modules. This and many other strategies can be investigated through efficient PLD programming languages (VHDL and Verilog) while having minimal impact on the DSP's software and workload.

6.0 Digital Control Design

6.1 Nonlinear Controller Design --The new nonlinear concepts employed in PWM DC-DC Power converter makes us more easily tuning NPID parameters arbitrarily. Compared to PI control, the gain of NPID has much more tolerances. It strengthens the robust of the controller and is more practicable to apply on the actual converter system. Some tests were conducted to compare their performance. Reference [6] in Volume II of this report is dedicated to a thorough description of the digital control mode studies and results.

6.2 Start-up Transient Performance -- A comparison was made of the converter's start-up time response. Figure 6.1 provides this comparison.

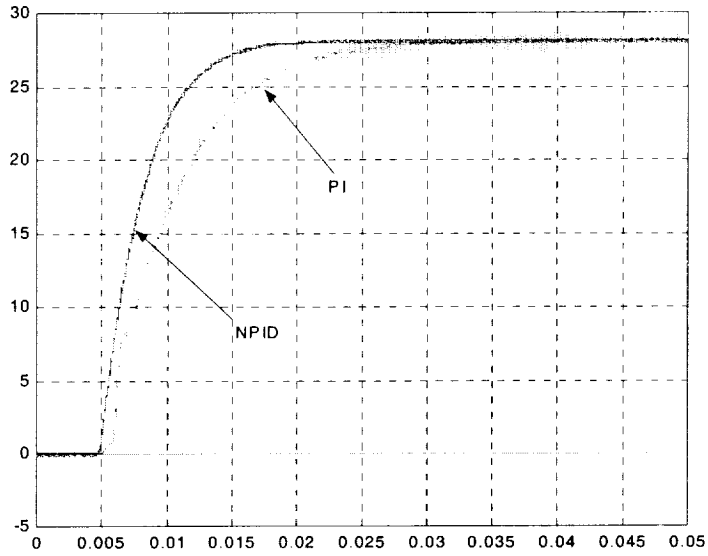


Figure 6.1. Start Transient Response (PI vs. NPID)

First the settling time is observed. The settling time is the period from start point to the – 2% of the step size, which for this example is 27.44V. The settling time is: PI -- 18ms, NPID-- 11ms. That means the PI controller is 67% slower than the NPID controller. The steady state error is: PI-- ±250mV, NPID-- ±180mV. That means the PI controller is 39% slower than the NPID controller.

6.2 Input Voltage Disturbance Rejection -- In figure 6.2, the output voltage transient response shows that for a input voltage disturbance of ±30V has a small maeasurable transient effect on the output voltage of the converter. Table 6.1 summarizes the peak transient voltage deviation and the total recovery for the two different control modes.

Deviation of voltage (mV)	PI	NPID	% (NPID vs. PI)
Step-Up (110->140)	444	169	162.72%
Step-Down (140-110)	525	225	133.33%

Recovery time (ms)	PI	NPID	% (NPID vs. PI)
Step-Up (110->140)	214	56	282.14%
Step-Down (140-110)	244	141	73.05%

Table 6.1 - Voltage Disturbance Rejection (PI vs. NPID)

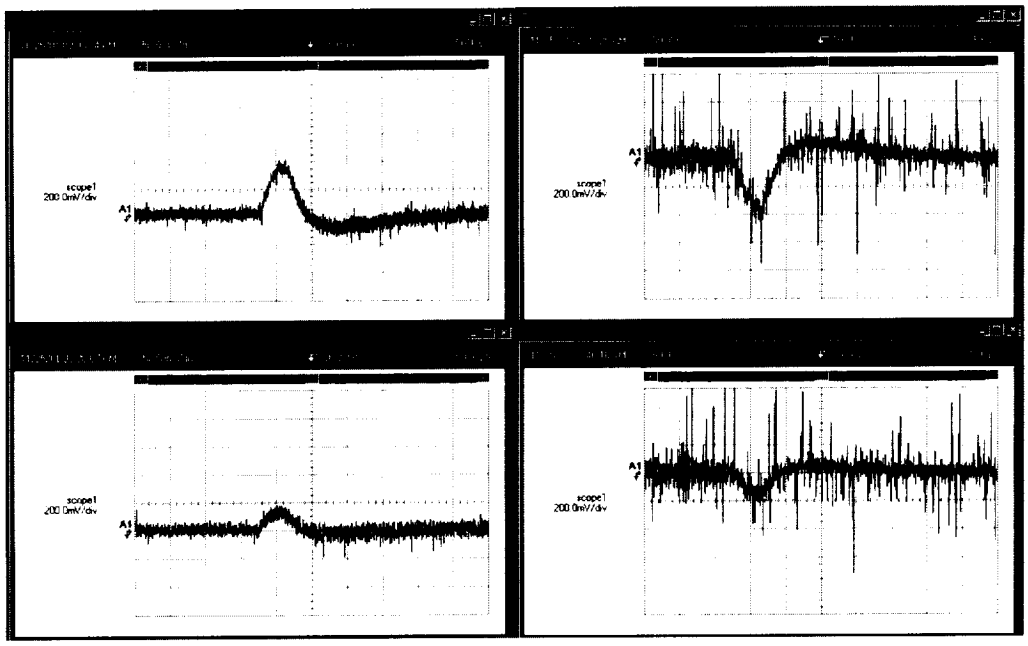


Figure 6.2 – Input Voltage Disturbance Rejection (PI vs. NPID)

6.3 Load Current Disturbance Rejection -- In figure 6.3, the output voltage transient response shows that a load current disturbance of ±33A has a significant transient effect on the converter's output voltage. Table 6.2 identifies the maximum transient deviation of the output voltage and the total recovery time.

Deviation of voltage (V)	PI	NPID	% (NPID vs. PI)
Step-Up (3->36A)	4.34	3.40	28.64%
Step-Down (36->3A)	3.9	3.22	21.12%

Recovery time (ms)	PI	NPID	% (NPID vs. PI)
Step-Up (3->36A)	13	5.9	120.34%
Step-Down 36->3A)	7	2.8	150%

Table 6.2 - Load Disturbance Rejection transient (PI vs. NPID)

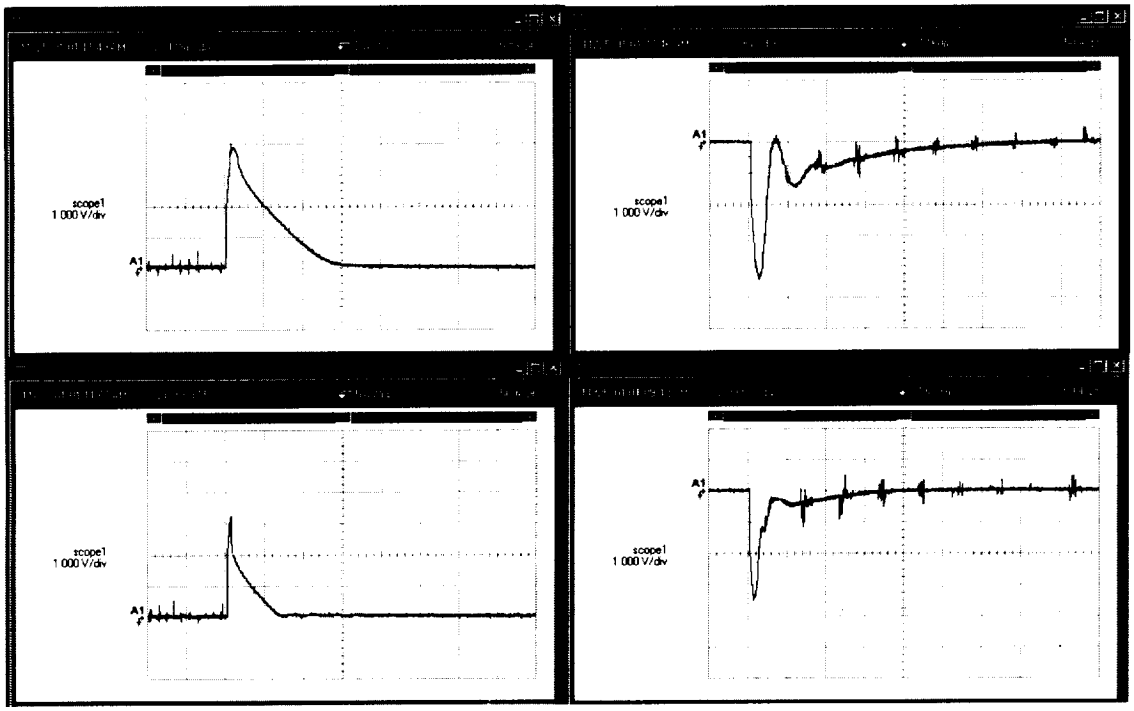
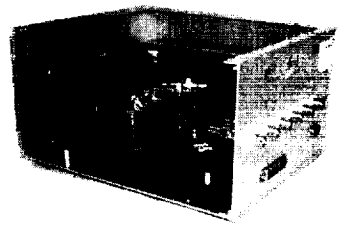


Figure 6.3 - Load Current Disturbance Transient Recovery (PI vs. NPID)

7.0 Design of Stand-alone DSP-based Converter Controller.



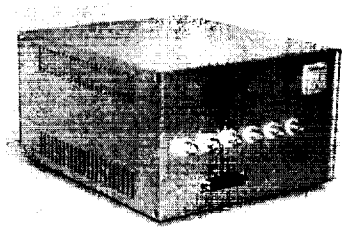
7.1 Hardware Description --The Stand-Alone DSP Control System development was completed in fall of 2001 in the Advanced Engineering Research Laboratory (AERL) at Cleveland State University. The unit was designed to replace the original power converter digital controller using the dSpace DSP rapid-prototype development system. Unlike the dSpace system, the Stand-Alone DSP Control System was designed to control power converters as a single stand-alone unit. The design is based on Texas Instruments' Digital Signal Processing (DSP) board (TMS320C6711 DSK) and Altera's Complex Programmable Logic Device (CPLD) (EPM7128SLC84-6). The DSP is used to monitor and control the power converter while sending the data to the CPLD. The CPLD takes input from the DSP and generates a Pulse Width Modulation (PWM) signals to control the 'H' bridge on the power converter.

This DSP solution with the CPLD provides much more flexibility than the analog PWM and control circuitry used in the past. This system can be easily reprogrammed to control any power converter using any control algorithm (NPID, PID, etc.). The Stand Alone DSP system is faster than the previously mentioned dSpace digital controller. The CPLD-based PWM Generator can be set to control all the legs of the 'H' bridge separately on the power converter, and this control system could be reprogrammed to control multiple power converters simultaneously.

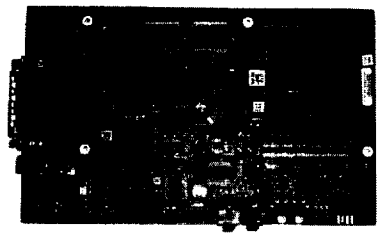
The Stand-alone DSP Control System report [3] documents the design and implementation of the Stand-Alone DSP Control System in five areas: 1) the enclosure, 2) TI DSP board, 3) prototype development board with the PWM Generator, 4) TI Analog-to-Digital (A/D) board, and 5) signal conditioning board. This report is included in Volume II of this final report as Reference [3].

The enclosure for the Stand-alone DSP Control System was purchased from Lansing with part number B3H10-V62A, but many modifications ware made to fit it to the design specifications. The enclosure had to include openings for the inputs and outputs, power

connector, power switch, fan, LED, and screws. The topics covered in detail in Sec. I of [3] include enclosure building, necessary parts, triple-output power supply, connections, wiring, and box layout. The finished enclosure is shown in the picture on the right.

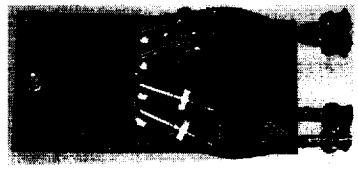


The TI Digital Signal Processing board (TMS320C6711 DSK) and the Code Composer Studio software are discussed in Sec. II of [3]. The DSP board description includes an overview and the data sheets for the board. The picture of the DSP board is included on the right. The Code Composer Studio description will include a software overview and setup.

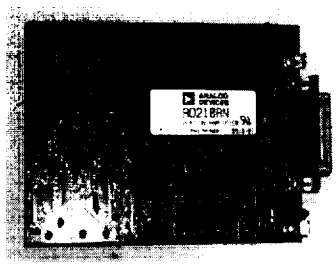


Section III of [3] will cover the TI prototype development board and PWM Generator that is programmed in the CPLD on the board. To program the PWM Generator Altera MaxPlus II software and the Altera CPLD (EPM7128SLC84-6) are used. The picture on the right shows the prototype board with the CPLD.

Section IV of [3] covers the TI A/D board (THS1206EVM) including a brief overview of the board, its data sheets, and the Code Composer Studio code to make the A/D board work with the stackable system. The TI A/D board is shown at the right.



Finally, section V of [3] discusses in detail the signal conditioning board. The discussion includes: its design, schematic drawing, performance, and test results. A picture of the signal conditioning board is on the right. The signal conditioning board has the task of properly conditioning all of the signals from the converter process variables (presently only voltages and currents) so that they can be used in the digital control algorithm or as performance measurements.



Normally this task would be limited to filtering the measurement signals to remove unwanted noise. However, the isolated or above ground nature of many of the power converter's measurements require that galvanic isolation of all measured voltages and currents be a part of the signal conditioning task. All current measurements are accomplished with Hall-effect current transducers to provide the galvanic isolation of these signals. The outputs are passed through appropriate analog low-pass filters to provide usable measurement information. The voltage signals are scaled as necessary, filtered appropriately and passed through a transformer-based analog isolation amplifier.

The AERL team designed signal conditioning board [4] was designed to accomplish these objectives and mate with the TI A/D board as part of the stackable system. Presently only the output voltage measurement channel is being used as part of the closed loop digital controller. Filter bandwidths have been selected to avoid aliasing of the sampled signals and provide the highest possible closed loop controller bandwidth.

7.2 Software Components -- Two major software components are needed to operate the Stand-alone DSP equipment as a closed loop controller operating in a research environment. The converter digital control algorithms are one of these components. Linear and nonlinear algorithms were designed and implemented in "C" code for real-time closed loop voltage regulation. This control software's implementation and the experimental performance it demonstrated are the subject of [6]. Section 6 of this volume of the final report provides an overview of the Reference [6] control design activity.

The other necesary software component is one that provides an easy-to-use operator interface and a data collection capability. A versatile well-designed graphical user interface (GUI) can satisfy this need. The Texas Instruments DSP selected by the team comes with a powerful development software suite called Code Composer Studio. This suite includes in part: an optimizing "C"compiler, a debugger, and a real-time data exchange capability.

7.3 GUI Requirements -- Initially the team expended time working with Visual Basic

attempting to provide a GUI based upon using Code Composer Studio's real-time data exchange (RTDX) functions. Before too much effort was expanded it became apparent that, if we were going to design a GUI with all the features required, the Visual Basic programming task would be significant. As the AERL team began to assess the problem and investigated alternative approaches, it became obvious that Labview software from National Instruments had the capability to expedite the GUI design task. Development of the GUI with Labview has begun. Already its potential benefits are impressive. This design choice should significantly enhance software productivity and provide a feature-rich GUI. Users will be able to: 1) modify several controller settings online and 2) collect data for observing the controller's performance graphically.

Since no reference in volume II discusses this GUI module, the next subsection will provide a brief description of how the Labview software functions with Code Composer Studio. An example GUI screen has already been developed is presented.

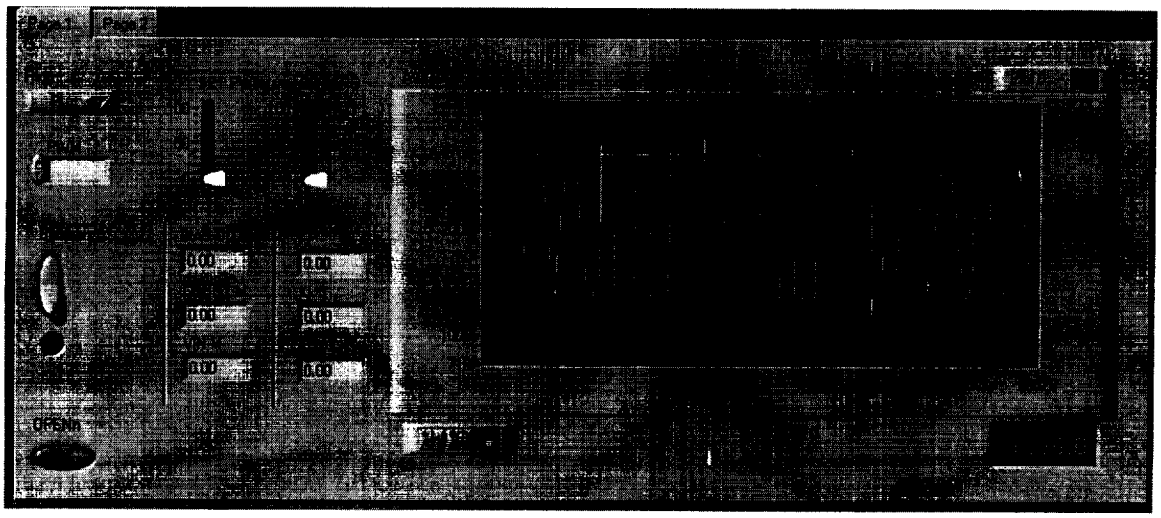
7.4 GUI Fundamental Operation – The block diagram below is intended to describe how real-time information is communicated between a Host and a Target DSP-based computer system. For our application the Host is a Windows-based personal computer. The Target is the Stand-alone DSP Controller described above. The block diagram is intended to describe the flow of digital information and identify where the major software modules and functions are located.

Code Composer Studio controls the flow of data between the host (PC) and the target (TI processor). Code Composer Studio's Real-Time Data Exchange (RTDX™) functions

provide real-time, continuous visibility into the way DSP applications operate in the real world. RTDX allows system developers to transfer data between a host computer and DSP devices without interfering with the target application. The data can be analyzed and visualized on the host using application specific software generally defined as a "COM automation client". For our application Labview will be used for the Com automation client GUI application software.

RTDX itself consists of both target and host components. A small RTDX software library runs on the target DSP. The DSP application makes function calls to this library's application interface (API) functions in order to pass data to or from it. This library makes use of a scan-based emulator to move data to or from the host platform via a JTAG interface. Data transfer to the host occurs in real-time while the DSP application is running.

The Labview (COM automation client) software can receive the data from the DSP target and can send control information. Through the Labview GUI interface the operator can set controller configuration settings such as a control strategy (PI, PD, PID, NPID etc.) or controller gain parameters. The operator can also observe steady-state data values and graphical representations of transient data. A sample Labview GUI interface window has been designed and is shown below. The data observed was produced with a signal generator connected to the one of the Stand-alone DSP's A/D input channels.



8.0 Modular Power Converter Design

8.1 Overview --One of the thrusts of the power converter digital control research is to evaluate digital control strategies needed for a converter comprised of identical converter modules. These moderate power modules would be operated in parallel to provide a high power converter unit in which the load is distributed intelligently between the modules. Also the modular design would provide for fault-tolerance should an individual module fail.

As a starting point for research in this area, a nominal 250 watt modular power converter was designed and built by one of the AERL team's student researchers as an undergraduate Senior Design Project. Figure 8.1 is a photograph of the modular converter brass board unit.

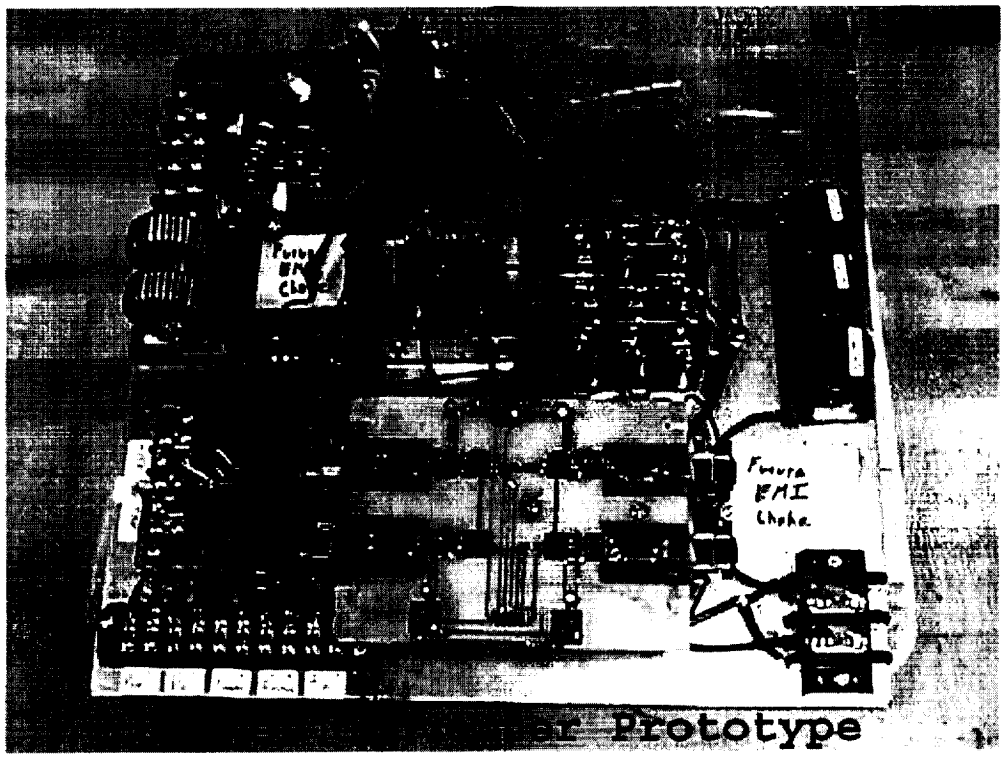


Figure 8.1 – Photograph of Modular Power Converter

Within the last several months, valuation testing of the modular unit has begun. Some preliminary performance results have been obtained. They will be summarized in this

section of Volume I of this final report. Planned future enhancements for this unit will also be discussed in this section. A complete report documenting the unit's performance will be prepared upon completion of a more rigorous series of tests.

8.2 Converter Configuration-- The modular converter uses the same full H bridge topology as the Westinghouse unit discussed earlier. As stated earlier it was designed for areduced power rating of 250 watts. The input voltage range was the same as that of the 1.0 Kw Westinghouse unit (100 - 160 volts). The output voltage was likewise designed to be 28 volts DC. The layout was designed to facilitate easy access to all the components and connections for easy measurement and replacement of components. This layout can be observed in the photograph of Figure 8.1. Figure 8.2,below is the schematic drawing of the unit's circuitry.

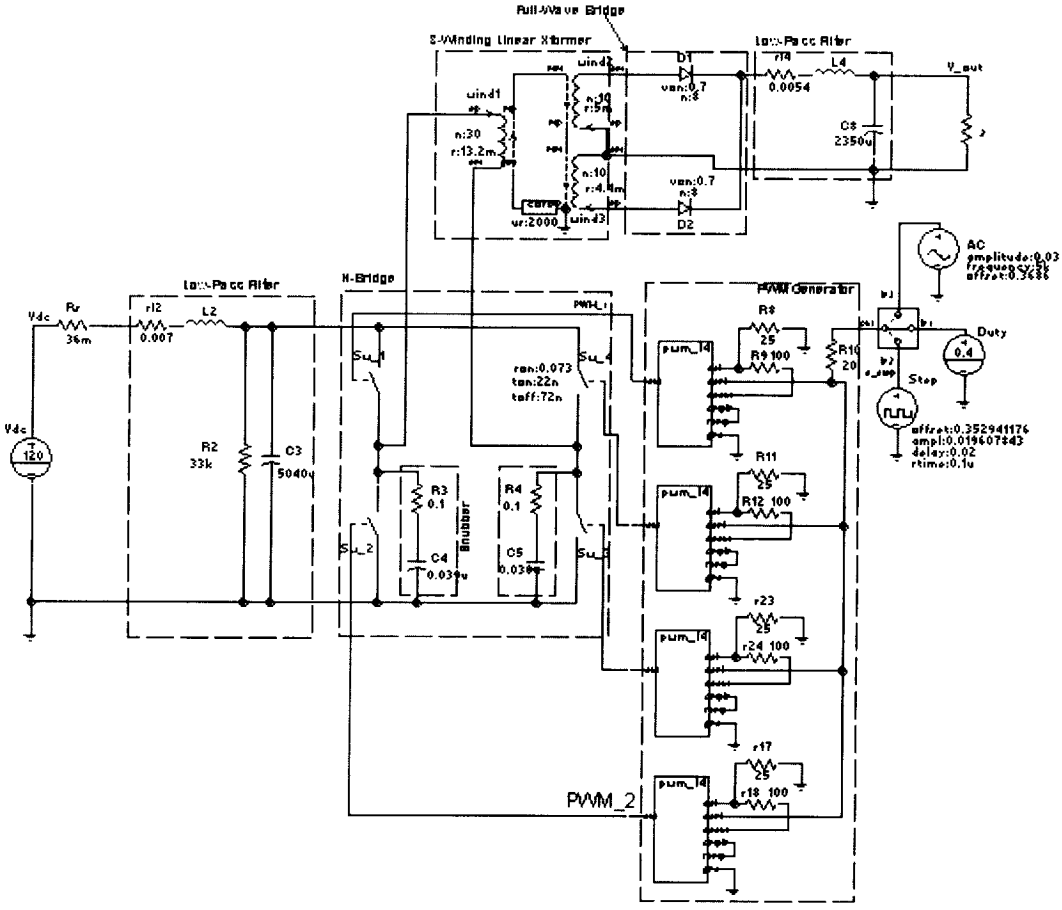


Figure 8.2 Modular Converter Schematic

Some design modifications to the original Westinghouse configuration were made to facilitate research in the area of digital power converter control. One example of this is the fully independent gate drive capability. It has the ability to drive each of the four MOSFET switches' gates independently. This allows for the study of different gating configurations that may or may not improve efficiency and reduce switching noise. A printed circuit board to accommodate the specialized gate drive circuitry was designed and fabricated in the AERL facility. Figure 8.3 is a photograph of the gate driver board and MOSFET power switch circuitry.

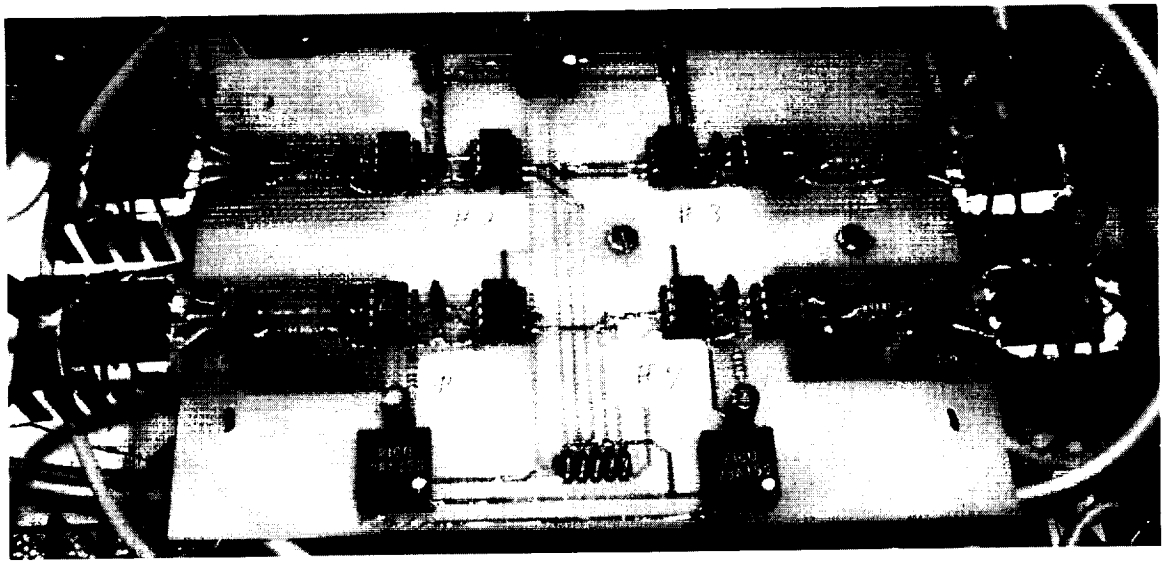


Figure 8.3 Photograph of MOSFET and Gate Drive Circuit Board.

8.3 Operational Performance and Limitations --Because of the lower power rating a new transformer was designed, wound and assembled as part of the Senior Design. Once testing of the new modular converter unit was begun, it was found that the transformer could not sustain 250 watt operation continuously without a large temperature rise in the windings. As a result some of the testing was limited to approximately 150 watts of operation. Even with this limitation some useful results of the converter's efficiency and how it performs at different frequencies were obtained. These limited results will not be reported at this time. The major benefit of this initial testing was the determination that the overall converter was operational at full input voltage and could provide the necessary output power. If could not, however, sustain full power continuously due to the transformer limitation.

In the last few weeks design of a new transformer has been initiated. Following some discussions with NASA-GRC engineering personnel, it was decided that we would raise the output power rating of the modular unit to 350 watts. The new transformer was designed to provide this level of power output. The new higher power transformer was wound and assembled in the AERL laboratory. As this report is being written, evaluation tests of the new transformer design are underway. Initial evaluation tests indicate that the new transformer can adequately handle full power (350 watts) continuously with only a modest temperature rise. These results enable the research team to proceed with the full series of evaluation tests of the modular unit at the higher power level.

8.4 Next Steps -- Future plans for the modular converter unit are as follows. Once a fully functional circuit design has been verified, a compact brassboard package design, including needed circuit boards will be undertaken. Preliminary layouts of this unit are already being discussed. Once completed, several more identical modular converters fabricated.

9.0 Future Research Tasks in Converter Digital Control.

9.1 Overview of Future Research – The research discussed in this final report has established foundational technology for designing and evaluating digital control of power converters and complete PMAD systems. With this background the AERL team can identify near term and long term research tasks needed to enhance this technology base. Those thrusts or important next steps in the research are:

a) Evaluate converter performance improvements available by using enhanced nonlinear controller strategies. Converter units studied should include those with a single high power module as well as those comprised of multiple lower power modules.

b) Utilize developed simulation and mathematical power circuit models to develop and analyze advanced converter digital control modes.

c) Evaluate novel and effective methods for acquiring, isolating, and signal conditioning converter sampled measurements needed for control, health monitoring and fault accommodation

d) Evaluate data communication protocols and methodologies as they relate to the regulatory control and supervision needs of systems of power converters.

The remaining subsections in this section of the report will provide additional detail related to each of these major thrusts. The perspective will be based upon what the AERL team has learned in the past 2 ½ years of research.

9.2 Digital Controller Enhancement—Thus far the research has shown that a single feedback loop using a nonlinear PID control mode provides tight output voltage regulation with responsive recovery to disturbances. (See Reference 6 in Volume II.) The next logical step should focus on performance improvements that can be realized by 1)the closure of inner loops utilizing additional sensed variables, 2) use of variable frequency PWM as it

relates to low power efficiency improvements, and 3) the influence that nonlinear control can have on converter input and output impedance values. Finally control modes for overcurrent, undervoltage, and overvoltage protection must be designed and evaluated.

9.3 Multiple Module Converter Digital Control – Using the AERL designed modular converter, with improvements (Section 8.0), as a starting point, a converter comprised of multiple modules should be developed. Control strategies such as phase stagger between modules to reduce input (and output) filter size and complexity should be studied. Also the study should include determination of the appropriate control hierarchy formultiple module based power converters. Major objectives of the final hierarchy would be to provide intelligent load sharing for efficiency optimization and to provide fault tolerance.

9.4 Modeling Utilization – Accurate and easy to use power converter simulation and mathematical models developed over the past 2½ years can be used to support analytical evaluations of new control strategies where appropriate. The benefits of multiple loops and variable PWM frequency are just two control enhancements whose benefits can be evaluated through simulation studies. The impact that modular converters can have on input and output filter sizing can also be investigated with already developed simulation models.

9.5 Measurement Signal Methodology --Much work was done in the just reported research program on measurement signal conditioning. The isolation requirements for the converter's measured variables, and their filtering to prevent aliasing of the signal to be digitized were accommodated in the digital controller design. In the future innovative methods for satisfying the galvanic isolation, filtering and anti-aliasing needs of sampled measurements should be investigated. Consider must be given to the latest component technologies (high speed serial A/D's and digital isolators) and how the AERL control architecture that incorporates a sophisticated PLD device might benefit from a newer approach to converter measurement signal acquisition and conditioning.

9.6 Data Communication Protocols --A longer term research need relates to the data communication requirements for intelligent PMAD systems. As the above research in

advanced digital control of power converters and converter systems progresses, much valuable information related to data communication needs can be acquired. However, the researchers must be diligent and stay continuously aware of this longer term need if meaningful data communications requirements are to evolve.

Stand-Alone DSP Control System Documentation

AERL-MNUL-0001B

Arthur Stachowicz, Aaron Radke, Greg Tollis, Ivan Jurcic, Jack Zeller, Dr. Zhigiang Gao
Advanced Engineering Research Laboratory (AERL)

Cleveland State University

Released 12/05/2001

Stand-Alone DSP Control System Documentation

Arthur Stachowicz, Aaron Radke, Greg Tollis, Ivan Jurcic, Jack Zeller, Dr. Zhigiang Gao
Advanced Engineering Research Laboratory (AERL)
Cleveland State University

Released 12/05/2001

Table of Contents:

Stand-Alone DSP Control System Documentation

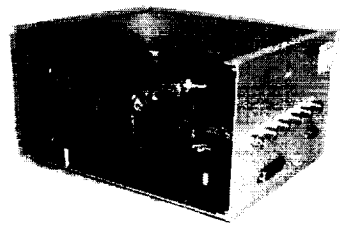
Arthur Stachowicz, Aaron Radke, Greg Tollis, Ivan Jurcic, Jack Zeller, Dr. Zhigiang Gao
Advanced Engineering Research Laboratory (AERL)
Cleveland State University

Released 12/05/2001

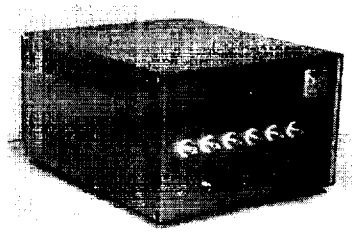
Abstract

Design details of a Stand-Alone DSP Control System. The design is based on Texas Instruments' Digital Signal Processing (DSP) board (TMS320C6711 DSK) and Altera's Complex Programmable Logic Device (CPLD) (EPM7128SLC84-6). The DSP is used to monitor and control the power converter while sending the data to the CPLD. The CPLD takes input from the DSP and generates a Pulse Width Modulation (PWM) signals to control the 'H' bridge on the power converter.

Summery

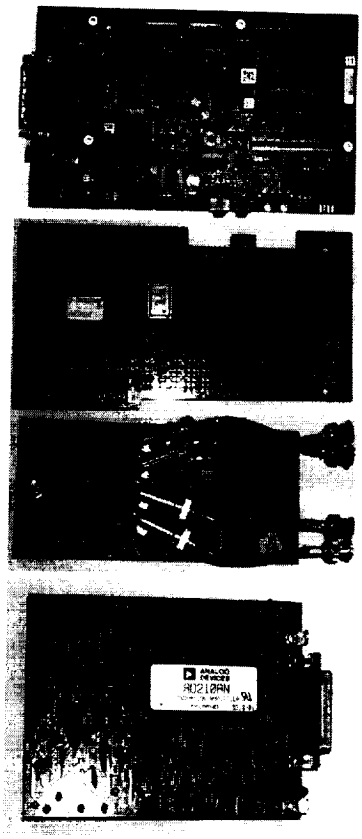


The Stand-Alone DSP Control System development was completed in fall of 2001 in Advanced Engineering Research Laboratory (AERL) at Cleveland State University. The unit was designed to replace the analog circuitry that controls power converters today. This design replaces the existing digital controller using dSpace, but unlike the dSpace the Stand-Alone DSP Control System was designed to control power converters as a single stand-alone unit. This DSP solution with the CPLD provided a lot more flexibility than analog circuitry used in the past. This system can be easily reprogrammed to control any power converter using any control algorithm (NPID, PID, etc.); Stand Alone DSP system is faster then analog system or the previously mentioned digital controller dSpace; the PWM Generator can be set to control all the legs of the 'H' bridge separately on the power converter, also this control system could be reprogrammed to control multiple power converters simultaneously.



This Stand-alone DSP Control System report documents the design and implementations of the Stand-Alone DSP Control System in five areas: the enclosure, TI DSP board, prototype development board with the PWM Generator, TI Analog to Digital (A/D) board, and signal conditioning board. First, Enclosure for the Stand-alone DSP Control System was purchased from Lansing with part number B3H10-V62A, but many modifications ware made to fit it to the design specifications. The enclosure had to include openings for the inputs and outputs, power connector, power switch, fan, LED, and screws. The topics covered in this part of the report include enclosure building, necessary parts, triple-output power supply, connections, wiring, and box layout. The finished enclosure is shown in the picture on the right. Secondly, the TI Digital Signal Processing board (TMS320C6711 DSK) and the Code Composer Studio are discussed in

the report. The DSP board description includes the overview and data sheets for the board. The picture of the DSP board is included on the right. Also, in this part the Code Composer Studio will be described. The code composer studio description will include software overview and setup. The third part of the report will cover the TI prototype development board and PWM Generator that is programmed in the CPLD on the board. To program the PWM Generator an Altera MaxPlus II software and the Altera CPLD (EPM7128SLC84-6) were used. The next picture on the right shows the prototype board with the CPLD. The fourth section covers the TI A/D board (THS1206EVM) including brief overview of the board, data sheets, and Code Composer studio code to make the A/D board work with the stackable system. Also, the A/D board is shown at the right. Finally, the signal conditioning board is covered including the design, schematic, performance, and test results. The picture of the signal conditioning board is last on the right.

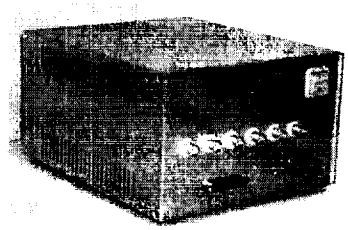


Stand-Alone DSP Control System Documentation

Arthur Stachowicz, Aaron Radke, Greg Tollis, Ivan Jurcic, Jack Zeller, Dr. Zhigiang Gao
Advanced Engineering Research Laboratory (AERL)
Cleveland State University

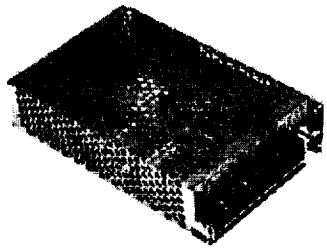
Released 12/05/2001

I. Stand-Alone DSP Control System Enclosure



The enclosure for the stand-alone DSP was purchased from Lansing with part number B3H10-V62A, the exact ordering information is included in the appendix. Next, the box from Lansing was modified to fit the requirements. All the drawings and measurements of the modifications are included in the appendix. The front has: 6 bi-directional isolated BNC connectors, which have female BNC on both sides of the box wall; 2 25-pin connector openings, P1 is for communication between the Code Composer Studio on the PC and the DSP board in the box, P2 is for the input signal to the signal conditioning board that comes from the power converter; The P3 is the 9-pin connector opening for the output from the PWM Generator going to the power converter; S1 is the power switch; LED1 is the power LED; Also, there are 4 screw holes, 2 by P1 and 2 by P2, which hold the boards in place. The rear has only few openings made, first one is P4 for the power plug connector and the second one for the FAN, also there are 4 screw holes to hold the fan in place. The bottom of the box has only few holes drilled to hold the DSP board and the power supply in place.

Mean Well Power Supply



Mean Well switching power supply T-60C series was used for this system. The power supply has 3 output voltages ±15Volts for the signal conditioning board and +5Volts for the stackable system, fan, and the power LED. This power supply was chosen for few reasons: It met all the necessary voltage requirements; The output current was enough to run the support all the devices; Ripple and noise was fairly small; The power supply it self was shielded; The power supply was small and affordable. The specs on the power supply can be found below in the Power Supply Specs Section.

Figure # I - 1: Power Supply Specs

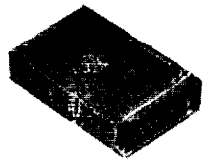
MEAN WELL

SWITCHING POWER SUPPLY

ISO-9001 CERTIFIED MANUFACTURER

T-60 SERIES

.LOW COST, HIGH RELIABILITY
.105°C OUTPUT CAPACITOR
.INTERNATIONAL AC INPUT RANGE
.HIGH EFFICIENCY, LOW WORKING TEMPERATURE
.SOFT-START CIRCUIT, LIMITING AC SURGE CURRENT
.SHORT CIRCUIT, OVERLOAD, OVER VOLTAGE PROTECTED
.COMPACT SIZE, LIGHT WEIGHT
.100% FULL LOAD BURN-IN TEST



MODEL	T-60A			T-60B			T-60C		
SPECIFICATION	CH1	CH2	CH3	CH1	CH2	CH3	CH1	CH2	CH3
DC OUTPUT VOLTAGE	5V	12V	-5V	5V	12V	-12V	5V	15V	-15V
OUTPUT V. TOLERANCE	±2%	±6%	±6%	±2%	±6%	±6%	±2%	±6%	±6%
OUTPUT RATED CURRENT	5A	2.5A	0.5A	5A	2.5A	0.5A	5A	2A	0.5A
OUTPUT CURRENT RANGE	0.5-7A	0.2-3.5A	0-1A	0.5-7A	0.2-3.5A	0-1A	0.5-7A	0.2-3A	0-1A
RIPPLE & NOISE(Vp-p)	100m	100m	100m	100m	100m	100m	100m	100m	100m
LINE REGULATION	±0.5%	±1%	±0.5%	±0.5%	±1%	±0.5%	±0.5%	±1%	±0.5%
LOAD REGULATION	±1%	±4%	±1%	±1%	±4%	±1%	±1%	±4%	±1%
RATED OUTPUT POWER	57.5W			61W			62.5W		
EFFICIENCY	72%			72%			72%		
DC VOLTAGE ADJ.	CH1:+10,-5%			CH1:+10,-5%			CH1:+10,-5%		
INPUT VOLTAGE RANGE	85~264VAC 47~63Hz ; 120~370VDC								
AC CURRENT	2A/115V 1A/230V								
INRUSH CURRENT	COLD START 30A/115V 60A/230V								
LEAKAGE CURRENT	<1mA/240VAC								
OVERLOAD PROTECTION	105%~150%/115VAC TYPE:PULSING HICCUP SHUTDOWN RESET:AUTO RECOVERY								
OVER VOLTAGE PROTECTION	CH1:5.75~6.75V								
TEMP. COEFFICIENT	±0.03% / °C (0~50°C) ON +5V OUTPUT								
SETUP,RISE,HOLD UP TIME	800ms, 50ms, 10ms / 115VAC 300ms, 50ms, 80ms / 230VAC								
VIBRATION	10~500Hz, 2G 3AXES 10min. / 1cycle (1 HOUR / EACH AXES)								
WITHSTAND VOLTAGE	I/P-O/P:3KVAC I/P-FG:1.5KVAC O/P-FG:0.5KVAC 1min.								
ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:500VDC / 100M Ohms								
WORKING TEMP., HUMIDITY	-10°C~+60°C(REFER TO OUTPUT DERATING CURVE), 20%~90% RH								
STORAGE TEMP., HUMIDITY	-20°C~+85°C, 10%~95% RH								
DIMENSION	159*97*38mm CASE:901								
WEIGHT	0.56Kgs								
SAFETY STANDARDS	UL1012, TUV EN60950, IEC950, UL1950 APPROVED								
EMC STANDARDS	CISPR22(EN55022) CLASS B, IEC801-2,3,4, IEC555-2 VERIFICATION								

NOTE : 1.ALL PARAMETERS ARE SPECIFIED AT 230VAC INPUT, RATED LOAD, 25°C 70% RH AMBIENT.

2.TOLERANCE : INCLUDE SET UP TOLERANCE, LINE REGULATION, LOAD REGULATION.

3.RIPPLE & NOISE ARE MEASURED AT 20MHz BY USING A 12"TWISTED PAIR TERMINATED WITH A 0.1uF & 47uF CAPACITOR.

4.LINE REGULATION IS MEASURED FROM LOW LINE TO HIGH LINE AT RATED LOAD.

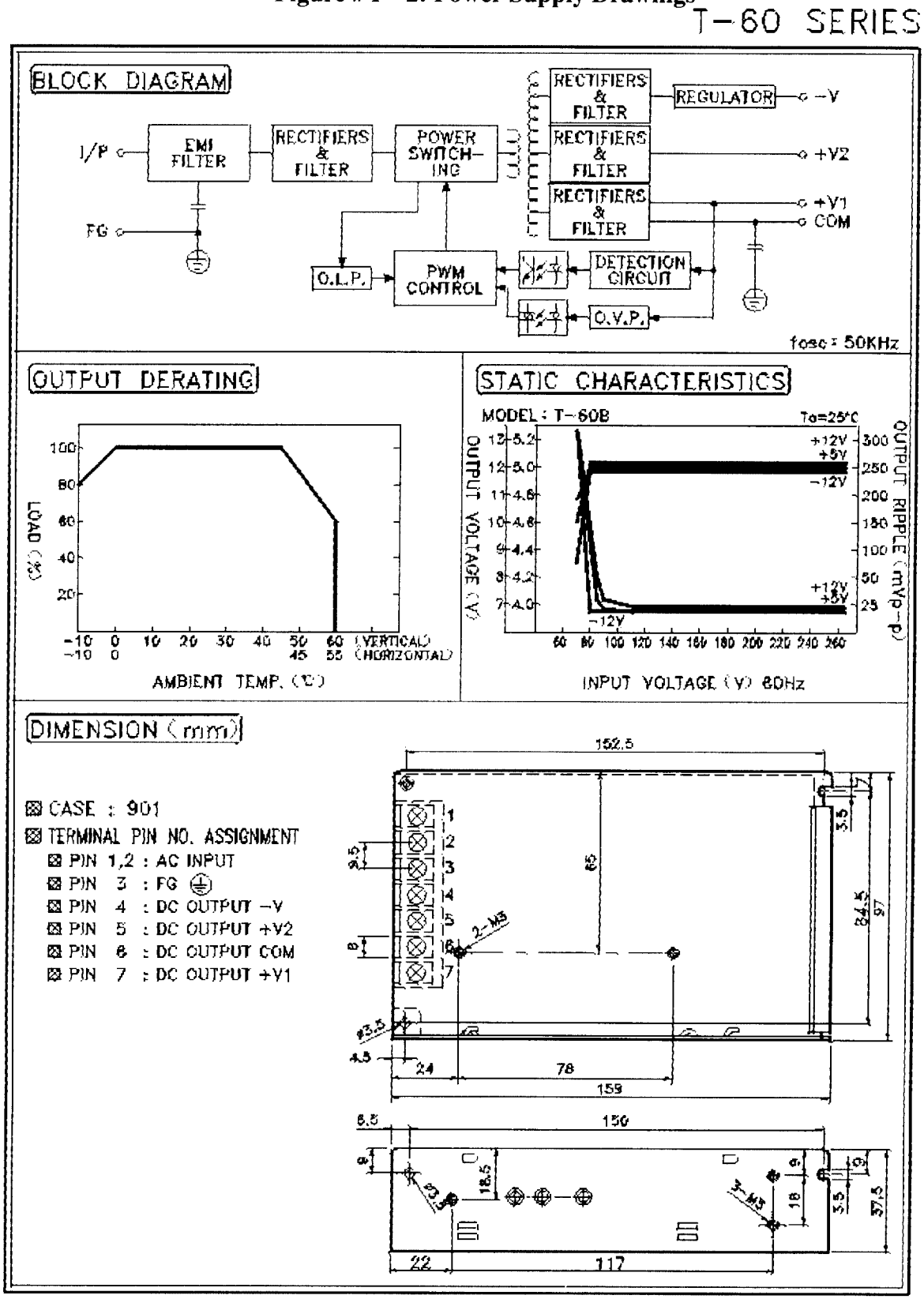
5.LOAD REGULATION IS MEASURED FROM 20% TO 100% RATED LOAD. AND OTHER OUTPUT AT 60% RATED LOAD

6.EACH OUTPUT PROVIDE UP TO MAXIMUM CURRENT. BUT TOTAL LOAD CAN NOT EXCEED MAX. OUTPUT POWER.

7.C23.6 MUST BE REMOVED.

2000-10-03

Figure # I – 2: Power Supply Drawings



Enclosure Parts

The enclosure needs many parts to connect everything together. The fallowing parts list includes parts like screws, washers, nuts, standoffs, connectors, BNCs, power supply, fan, power connectors, switch, LED, mounting supplies, and other.

Figure # I – 3: Parts, Supplier, and Price.

Part	# of	Description	Supplier	Part Number	Price (each)	Part Total
9-pin male	1	D-Subminiature 9-pin male solder-cup connector	Jameco.com	15747	$0.450	$0.450
BNC cable	1	Coax Cable RG58	Bruce		$0.000	$0.000
BNC female isolated bulkhead feed-thru	6	BNC female isolated bulkhead feed-thru	Jameco.com	148232	$1.750	$10.500
BNC male	8	BNC male with pin and ferrule (RG58)	Jameco.com	71482	$1.190	$9.520
BNC M-F right angle	4	BNC M-F right angle	Jameco.com	127351	$2.250	$9.000
Bracket right angle	4	Bracket right angle, 4-40 thread	Mouser.com	534-616	$0.550	$2.200
Disconnect solderless terminals	7	Disconnect solderless terminals (Insulated), female 0.25", 22-16 wire	Jameco.com	109111	$0.140	$0.980
Enclosure	1	Enclosur, Box	Lansing.com	B3H10-V62A	$108.470	$108.470
Fan	1	3.15" SQ 5V DC fan, CR0805HB-D7, Flight LT	Newark.com	91F7495	$14.980	$14.980
Fan guard	1	3.15" square aluminum filter	Jameco.com	138579	$3.250	$3.250
Feet, 4	1	Black tapered square, pack of 4, SJ5023BLK4	Jameco.com	126981	$1.950	$1.950
Female pins	12	female pins for the 4-pin plugs.	Jameco.com	181673	$0.100	$1.200
Header strght 4-pin	2	Strght header four-pin, connector for the led and fan.	Jameco.com	152741	$0.350	$0.700
Jack Screws	2	0.315" length jack screws for the 9-pin connector	Jameco.com	172603	$0.150	$0.300
LED	1	Red diffused LH3330, T1 3/4	Jameco.com	94511	$0.150	$0.150
LED clip	1	Clip with a ring, T1 3/4, LMH200	Jameco.com	23077	$0.120	$0.120
Nuts 4-40	4	4-40 hex nuts for the boards (DSP and signal conditioning	Jameco.com	40942	$0.016	$0.064
Nuts with washers 4-40	2	4-40 nuts with washers fot the 9-pin plug	Sutton Hardware		$0.040	$0.080
Nuts with washers 6-32	4	6-32 nuts with washer for the fan	Sutton Hardware		$0.050	$0.200
plugs 4-pin	3	plugs 4-pin for the headers and pulse output.	Jameco.com	152733	$0.290	$0.870
Power connector (kit)	1	2-pin power connector, kit of 3-plugs and 3-receptacles	Jameco.com	142181	$4.950	$4.950
Power cord	1	3-conductor detachable power supply cord, 6', 17602.	Jameco.com	159361	$2.950	$2.950
Power inlet plug	1	AC power inlet (male) snap-in	Jameco.com	147109	$0.690	$0.690
Power Supply	1	63 watt triple output switching power supply T-60C	Jameco.com	123490	$44.950	$44.950
Power switch	1	PC rocker switch, 10A, 250VAC	Jameco.com	127028	$1.490	$1.490
Ring solderless terminals	7	Disconnect solderless terminals (Insulated), 22-16 wire,	Jameco.com	103683	$0.090	$0.630
Screws 4-40	16	Slotted pan head machine screws 4-40, 0.25" length	Jameco.com	40951	$0.016	$0.256
Screws 6-32	4	1" screws 6-32 for the fan, flat head	Sutton Hardware		$0.100	$0.400
Screws M3	3	matric screws M3, 0.25" length	Sutton Hardware		$0.110	$0.330
Standoffs	4	0.250" hex threaded female standoffs, 4-40, 0.5" length	Jameco.com	108370	$0.220	$0.880
Tie Mount 0.5"	3	0.5" tie mount square self adhesive, 4 way cable tie.	Jameco.com	171213	$0.150	$0.450
Tie Mount 1.1"	2	1.1" tie mount square self adhesive, 4 way cable tie.	Jameco.com	71394	$0.190	$0.380
Ties	6	tie cable 18 lbs.	Jameco.com	104141	$0.025	$0.150
Washers		Internal tooth lock washers, 0.277	Jameco.com	106868	$0.024	$0.024
Totals	**116**					**$223.514**

Figure # I – 4: Front View of the Enclosure B3H10-V62A

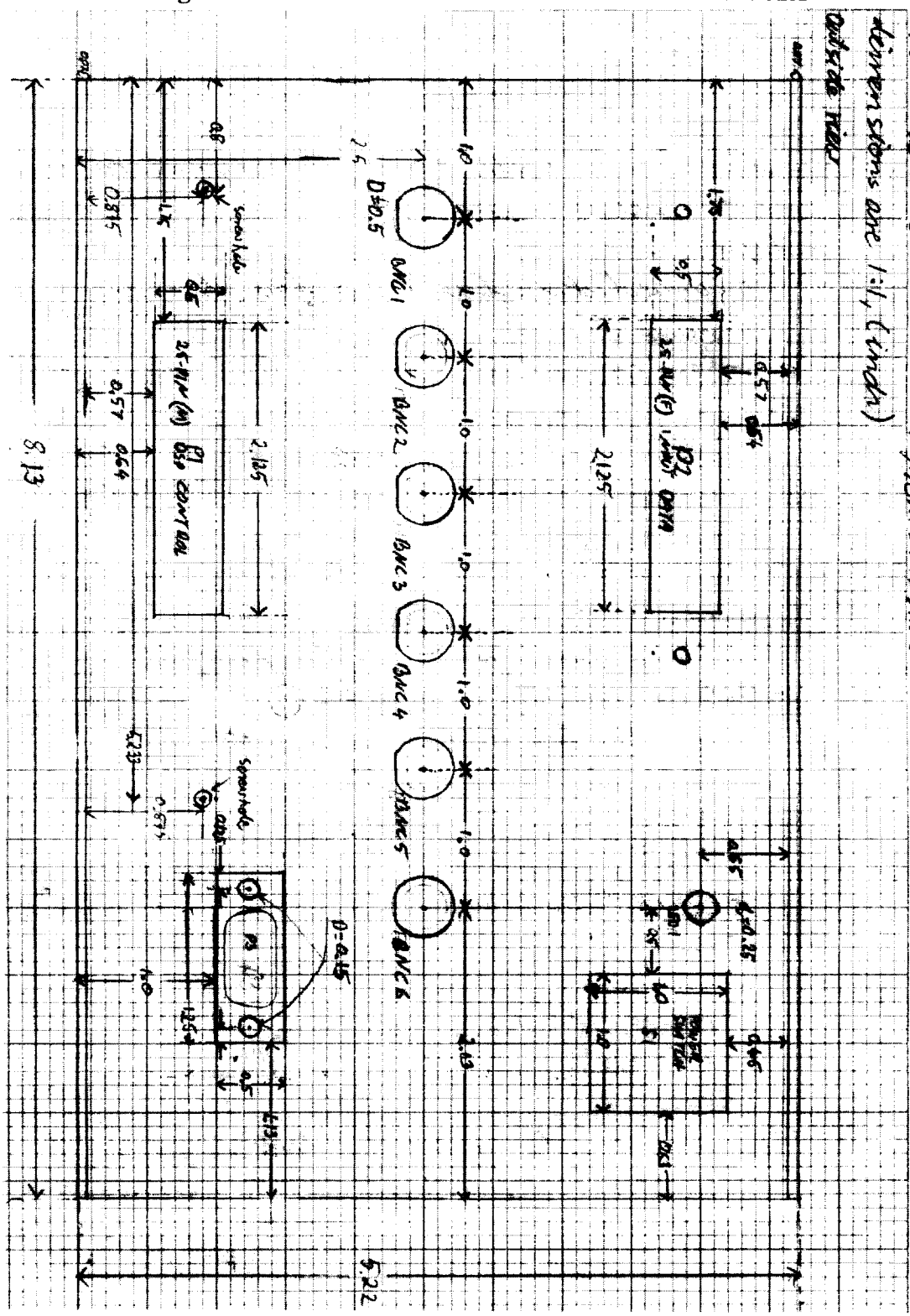


Figure # I – 5: Rear View of the Enclosure B3H10-V62A

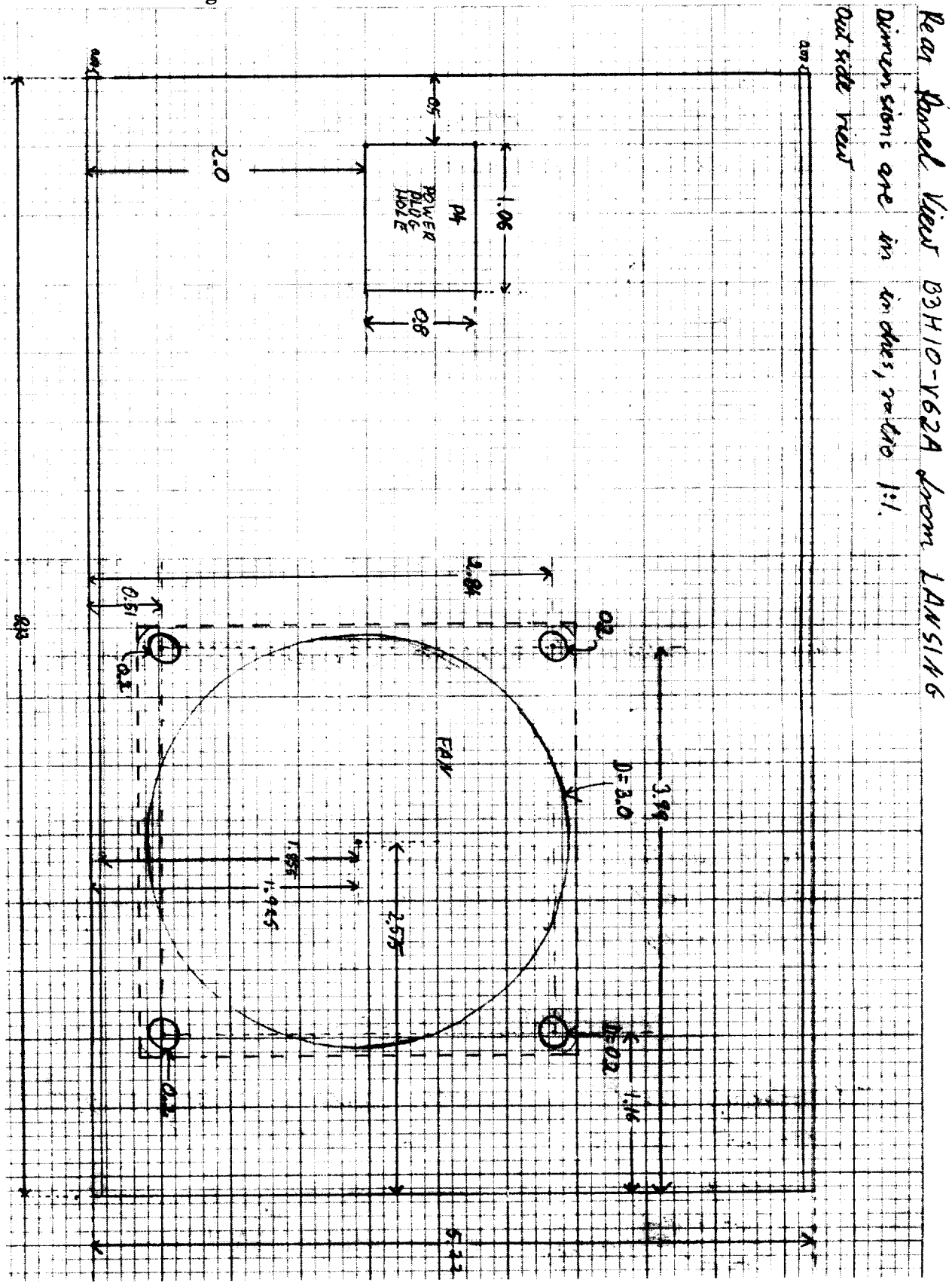


Figure # I – 6: Bottom View of the Enclosure B3H10-V62A

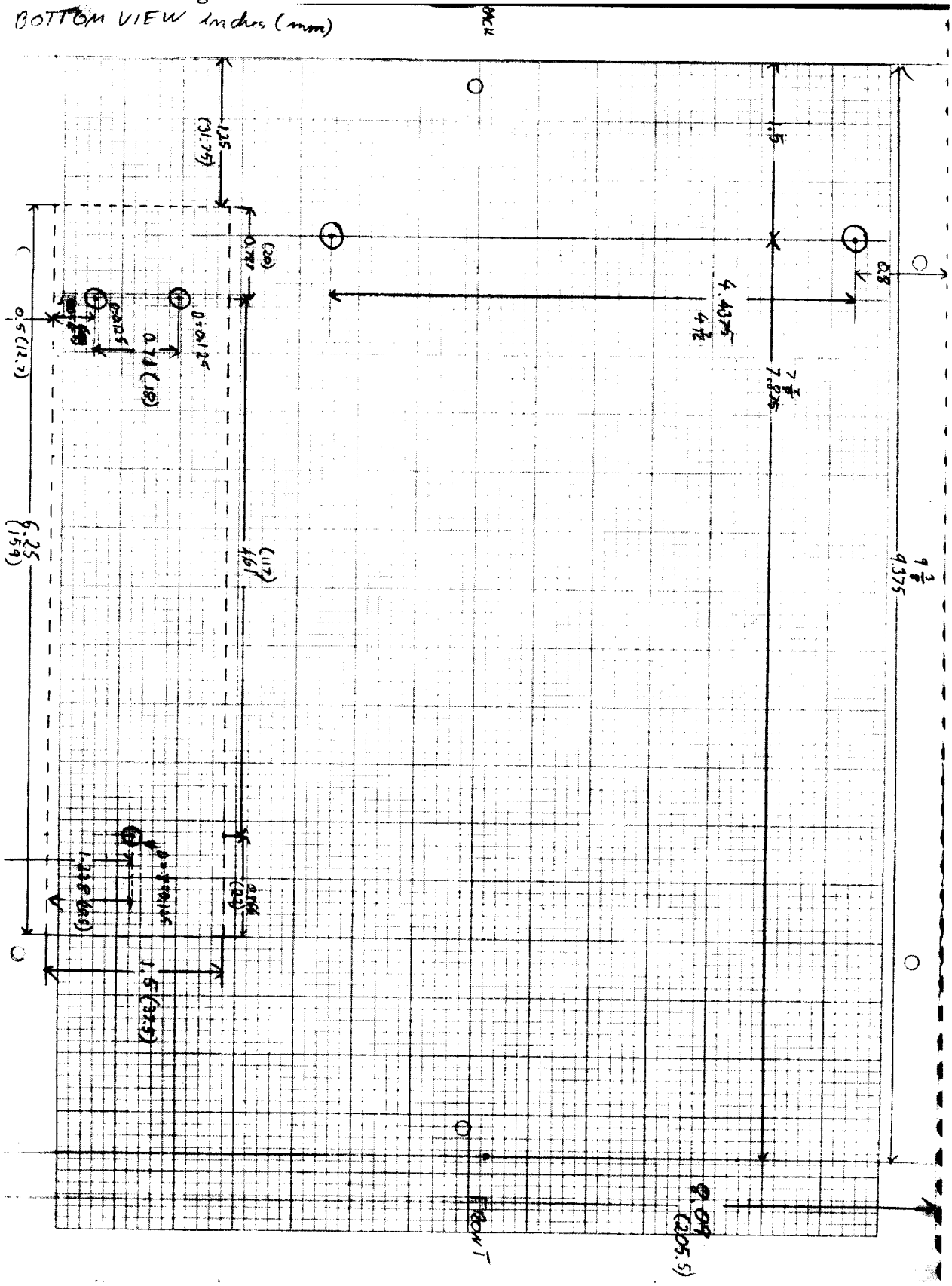
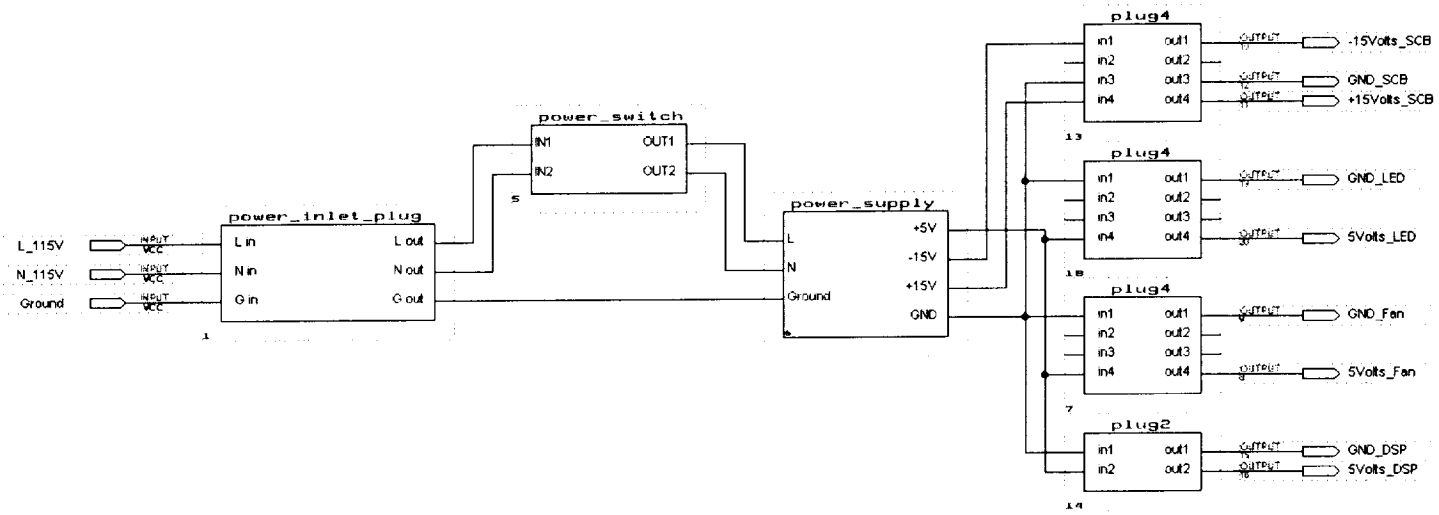


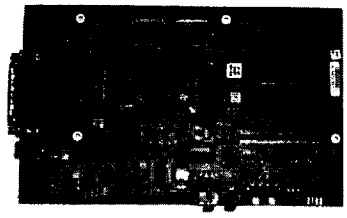
Figure # I – 7: Enclosure Wiring Diagram



II. DSP Board and Code Composer Studio

DSK Board Documentation:

Purpose



The DSK (DSP Starter Kit) was chosen for several important reasons. First of all it was an economic and quick way to have an entire control system built without have to create a large digital DSP board design. Second the starter kit is made to help the design be implemented and does this with sample code to base a design around. The final major benefit of the DSK board is the stackable feature to add on modules for future expansion. With the stackable feature we bought a 3rd party prototyping board to plug add onto the stack for a custom CPLD to interface with the converter.

Usage

No settings or set up is needed other than the configurations done through software in Code Composer, which is described in the Code Composer settings.

Features

The following product description can be found at:
http://focus.ti.com/docs/tool/toolfolder.jhtml?PartNumber=TMDS320006711

Description

The C6711 DSP starter kit available from Texas Instruments for US $295, provides system design engineers with an easy-to-use, cost-effective way to take their high-performance TMS320C6000 designs from concept to production. The new TMS320C6711 DSP Starter Kit (DSK) not only provides an introduction to 'C6000 technology, but is also powerful enough to use for fast development of networking, communications, imaging and other applications. Operating at 150 MHz, the 'C6711 delivers an impressive 1200 MIPS and 600 MFLOPs for only U.S. $22* (1K units). The to use an innovative 2-level cache memory configuration, which provides high-performance in a very cost-effective solution. The C6711 DSK replaces and is a superset of the C6211 DSK. The C6711 is binary code compatible with the C6211. I.E. C, assembly and executable code written for the C6211 will run without modification on the C6711.
The TMS320C6711 DSP Starter Kit includes a C6711-based DSP target hardware module and Code Composer Studio V1.based DSK Specific software.

Contents

Hardware Included

* 'C6711 DSK Board - Easily connects to a PC through a parallel port cable (included): 150 MHZ 'C6711 DSP

* 16 MB External SDRAM and 128 KB External Flash - Provides additional program and data storage
* TI'S TLC320AD535 16-bit Data Converter
* TI'S TPS56100 Power Management Device
* JTAG Controller - Provides easy emulation and debugging
* Expansion Daughter Card Interface - Provides extensible system development
* CE-Compliant Universal Power Supply DSK
* Ordering Information: TMDS320006711 includes a standard US power cord, TMDS32006711E version includes both UK & European power cords

Software Included

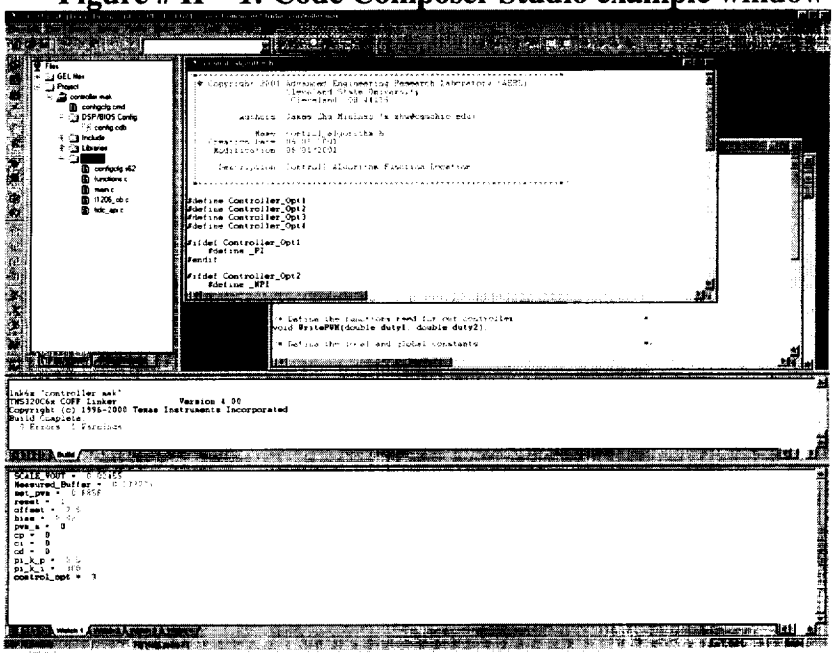
The 'C6711 DSK comes with an array of DSK-specific software functionality (256 KB software image memory limited), including the highly efficient 'C6000 C Compiler and Assembly Optimizer, Code Composer Debugger and DSK support software (flash utility, sample programs and confidence tests). The 'C6000 platform's performance is captured by its highly efficient optimized C Compiler and the industry's first Assembly Optimizer. The compiler provides over 80% of the performance of hand-coded assembly on DSP benchmarks using natural C code without intrinsics or modifications to the algorithms. The Assembly Optimizer enables the developer to write linear, RISC-like assembly code and schedule it to deliver optimum efficiency and performance.

Code Composer Studio:

Purpose

Code Composer is used for the code generation of the control system for several reasons. First of all it is bundled with the DSK and tightly integrated with the use of the DSK. It is also tightly integrated with the A/D as well as the DSP, which creates an effective system for testing and implementing the control algorithm.
For now, the control variables can be changed and evaluated by using the watch window while the converter is running for tuning and evaluation purposes. See the section on stack tuning setup steps for more information on this tuning process. A sample screenshot of Code composer is shown below to show the real time tuning control algorithm via the watch window.

Figure # II – 1: Code Composer Studio example window



Features

The following product information can be found at:

http://dspvillage.ti.com/docs/sdstools/sdscommon/showsdsinfo.jhtml;$sessionid$1VHZ21QAAB0JBTMNZSFBF2Q?path=templatedata/cm/ccstudio/data/description_demo&templateId=57

Code Composer Studio (CCStudio) software is a fully integrated development environment (IDE) supporting Texas Instruments industry-leading TMS320C6000 and TMS320C5000 DSP platforms. Code Composer Studio is one of the key components of the eXpressDSP Real-Time Software Technology that slashes development and integration time for DSP software. Code Composer Studio IDE v2 is the first intelligent development environment to offer TMS320C2000, TMS320C5000 and TMS320C6000 application development for multi-processor, multi-user and multi-site projects!

Code Composer Studio integrates all host and target tools in a unified environment to simplify DSP system configuration and application design. This easy to use development environment allows DSP designers of all experience levels full access to all phases of the code development process. CCStudio has an open architecture that allows TI and third parties to extend the IDEs functionality by seamlessly plugging-in additional specialized tools.

Such familiar tools and interfaces allow users to get started faster than ever before and add functionality to their application thanks to sophisticated productivity tools The environment integrates traditional tools for editing, building, debugging, code profiling and project management. These tools work tightly with the more advanced features also integrated into the Code Composer Studio IDE user interface such as signal probing, multi-processor support, data and system visualization, and a flexible C-based scripting language for automated testing and customization.

Usage

The installation of CSS and the controller code, stack tuning setup steps and stack usage variables description are shown below.

Installation of CCS

Code composer v1.23 installation:

When prompted for the install directory, verify that it is at C:\ti

Installation of digital controller code

Digital control code installation:

Copy the digital controller source code directory to:
C:\ti\myprojects\

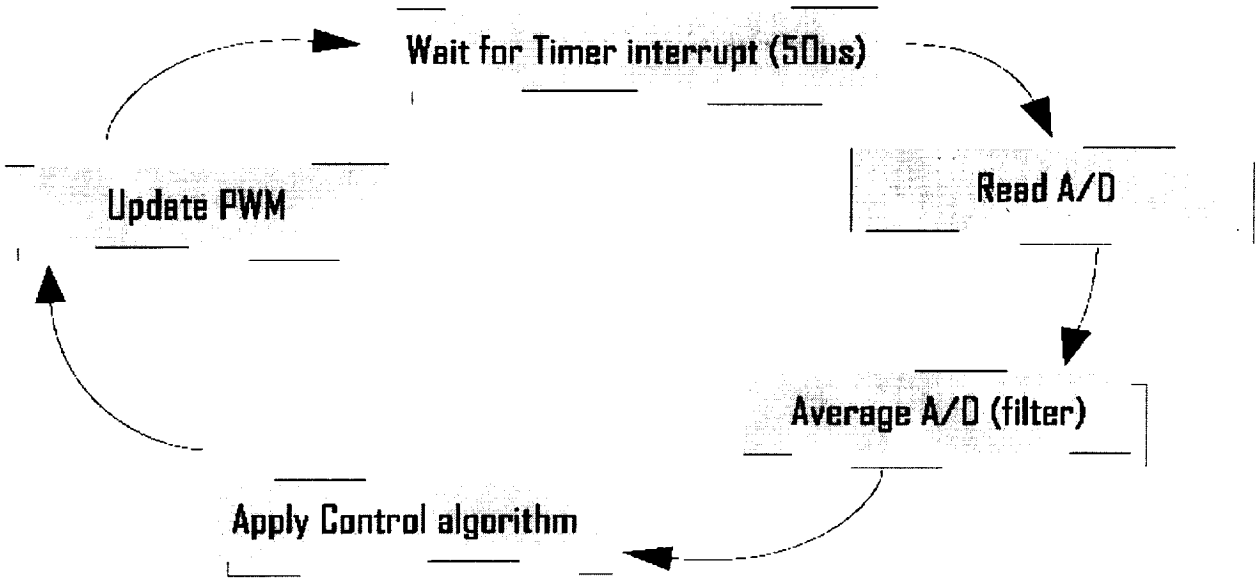
The ccs project file is named controller.prj within the copied directory.

If these files are stored in the proper manner when the code is compiled the dependent files and paths will also be in the proper directories.

Digital Control Software:

The code to implement the converter is actually quite simple. The overview of the basic flowchart diagram is shown below.

Figure # II – 2: Software Flowchart



As you can see there are just a few basic steps the control follows to implement the controller.

The C-code was designed with these steps by modifying, adding and tailoring the read-block DSK example code for the A/D from TI. There were 4 major areas dealt with in order to convert and create the software for the stackable controller from the A/D sample code.

1) Increase the timer loop rate

The sample code had long loop time to update the A/D using the PRD function manager in DSP/Bios. The faster response of 50us was achieved still under the DSP/bios but rather used the clock to directly call the periodic function to run the control algorithm every 50us.

2) Add A/D filtering

A simple averaging filter was added to the receive block to sum and divide over the number of samples. This function also took care of scaling the A/D values to the direct voltage output value. This averaged scaled value is called measured_vout and is then sent to the control algorithm by the function ControlAlgorithm(measured_vout).

3) Add the digital control algorithm

This is the NPID control algorithm that does the work of the control. It is called when the A/D filtering average is finished. The ControlAlgorithm(measured_vout) receives the

measured voltage input and outputs the duty ratios of each PWM by calling the WritePWM function.

4) Output of the PWM and frequency onto the data bus

WritePWM's job after receiving the two PWM duty ratios is to convert the duty rations to the phase count output and send this to the CPLD PWM generator. The phase count output is simply created by a scale factor. The value is then sent to the PWM by writing the value to a memory address, which the CPLD is mapped.

The phase address is: `0xB0000000;`
The frequency address is: `0xB0000004`.--For now this value is constant.

A stripped down version of WritePWM is shown below for the functionality of writing to the CPLD PWM generator.

```
void WritePWM(double duty1, double duty2)
{
int *mem_phase = (int *)0xB0000000;
int *mem_freq =  (int *)0xB0000004;
phase1=phase_max_count*duty1;
phase2=phase_max_count*duty2;
*mem_phase = (phase2 << 16) | ( (phase1) & 0x0000FFFF) ;
}
```

Stack tuning setup steps

The following directions can be used to tune and startup up the stackable DSP system for use with the power converter.

The variables can be adjusted and checked using the watch window in CCS by setting and refreshing the watch window.
In the future it would be nice to use RTDX to control these variables, along with Visual basic to guide and instruct the user through the steps required to get the system running.
You can also use the stack variable description information in the next section for a further description of all of the variables used in the stackable controller.

Set the scale factors and offsets.
---
1)set the offset
The following equation can be used for an understanding of the following steps
(Eq1) measured_vout=(vout - offset)*scale_vout - bias
where vout = 2048-myavg

a)set the bias to zero //so we know the true zero point
b)set set_pwm to zero //zero point
c)set scale_vout to 1 //get rid of scale factor

now (Eq1) becomes:

measured_vout = vout - offset

vout is at the zero point so:

d)the offset should be set to the present vout value

2)set set_pwm so there are 28V out of the converter //set up for step 3
3)adjust scale_vout so measured_buffer = 28V //adjusting the a2d scaling
4)try another set_pwm value to verify the match between the measured_buffer and the output as well as to check the linearity (example around 15V)

Run the converter in closed loop
---
1)make sure the scale factors, offsets and all other converter connections are properly setup
2)reset = 1 //make sure it is shut off
3)set_pwm = -1 //closed loop option
4)reset=0 //run it in closed loop!
5)adjust bias for an even closer match to 28V

Stack Usage Variables Description

The following is a list of the variables used in the non-linear control of the stackable controller. The variables are sorted under section of the types of the variables.
You can also use the stack usage steps in the previous section as a quick guide to remember the steps in setting up the stackable to be run with the power converter.

```
----------------------
Startup, run and miscellaneous control
--------------
SCALE_VOUT set the a2d factor value
Measured_Buffer    the measured output voltage
set_pwm            >0 manual mode of pwm duty ratio
          -1 closed loop
reset              0 turn on the controller
          1 turn off
offset             input adjust for the zero input on the a2d
bias               output adjust for the set point (ex 28) output on the converter
pwm_m              the exact pwm output values (not really used)

cp         control proportional (control outputs to see)
ci         control integral (control outputs)
cd         control differential (control outputs)
control_opt 0      PI
            1      NPI
            2      PID
            3      NPID


---------------------
PI controller settings
----------------------
pi_kp              control proportional for the PI controller
pi_ki              control integral for the PI controller


---------------------
NPI controller settings
----------------------
James suffix definitions
_dn:       load step down (decrease in load)
_up:       load step up (increase in load)
k2: outer region slope factor (small error small change only for proportional)
k1: inner region slope factor (hi error big change only for proportional)
_d: region set point of non linear error setting which sets inner and outer region

----------
npi_kp_dn  proportional +suffix
npi_kp_up  proportional +suffix
npi_ki_dn  integral +suffix
npi_ki_up  integral +suffix
npi_kp_k1  proportional +suffix
npi_kp_k2  proportional +suffix
```

```
npi_ki_k1   integral +suffix
npi_ki_k2   integral +suffix
npi_kp_d    proportional +suffix
npi_ki_d    integral +suffix

    the td(tracking differential) acceleration can be used to set a pseudo corner
frequency for setting frequency region, the higher r is the higher the pseudo corner
frequency

---------------
PID: the gains are linear even though the td is there
------------------
pid_kp              proportional control
pid_ki              integral control
pid_kd              differentiator control
pid_kd_r

-------------
NPID:       non linear gains
-----------------
npid_kp_dn proportional +suffix
npid_kp_up proportional +suffix
npid_ki_dn integral +suffix
npid_ki_up integral +suffix

npid_kd             differentiator control


-----slope factor settings-----
npi_kp_k1   proportional +suffix
npi_kp_k2   proportional +suffix
npi_ki_k1   integral +suffix
npi_ki_k2   integral +suffix
npi_kd_k1   integral +suffix
npi_kd_k2   integral +suffix

---region settings------
npid_kp_d   proportional +suffix
npid_ki_d   integral +suffix
npid_kd_d   integral +suffix

npid_kd_r   the td(tracking differential) acceleration can be used to set a pseudo
corner frequency for setting frequency region, the higher r is the higher the pseudo
corner frequency
```

The fallowing source code included in the apedix.

main.c
_control_globals.h
algorithms_control_algorithm.h
\algorithms_pid_zhu_11a.h
dc_conf.h

Supporting documents

Code Composer Studio User's Guide

(literature number SPRU328) explains how to use the Code Composer Studio development environment to build and debug embedded real-time DSP applications.

TMS320C6000 DSP/BIOS User's Guide

(literature number SPRU303a) describes how to use DSP/BIOS tools and APIs to analyze embedded real-time DSP applications.

TMS320C6000 DSP/BIOS API Reference Guide

(literature numberSPRU403) describes how to use DSP/BIOS tools and APIs to analyze embedded real-time DSP applications.

TMS320C6000 Assembly Language Tools User's Guide

(literature number SPRU186) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the 'C6000 generation of devices.

TMS320C6000 Optimizing C Compiler User's Guide

(literature number SPRU187) describes the 'C6000 C compiler and the assembly optimizer. This C compiler accepts ANSI standard C source code and produces assembly language source code for the 'C6000 generation of devices. The assembly optimizer helps you optimize your assembly code.

TMS320C62x/C67x Programmer's Guide

(literature number SPRU198) describes ways to optimize C and assembly code for the TMS320C62x/C67x
DSP's and includes application program examples.

TMS320C62x/C67x CPU and Instruction Set Reference Guide

(literature number SPRU189) describes the 'C62x/C67x CPU architecture, instruction set, pipeline, and interrupts for these digital signal processors.

TMS320C6201/C6701 Peripherals Reference Guide

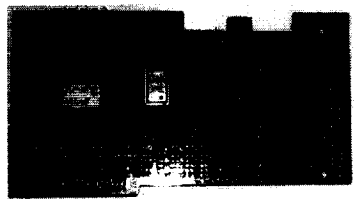
(literature number SPRU190) describes common peripherals available on the TMS320C6201/'C6701 digital signal processors. This book includes information on the internal data and program memories, the external memory interface (EMIF), the host port, multichannel buffered serial ports, direct memory access (DMA), clocking and phase-locked loop (PLL), and the power-down modes.

TMS320C62x Technical Brief

(literature number SPRU197) gives an introduction to the digital signal processor, development tools, and third-party support.

TMS320C6201 Digital Signal Processor Data Sheet

III. PWM Generator and the prototype board



The purpose of the PWM Generator is to create a fault tolerant pulse signal going in to the power converter. In other words, when DSP looses power the CPLD with the PWM will continue to send pulses to the power converter therefore not crushing the system, and also DSP can be restarted and continue to sent input to the CPLD. The CPLD used for this system is Altera's (EPM7128SLC84-6). This section will cover the design of the PWM generator written in VHDL and the layout of the prototype board.

Prototype Board With The PWM Generator Description And Wiring:

Features

100 MHz Clock
Hardware Deadband Enforcement
High resolution: 11.28 bits of resolution with 20kHz PWM

Description

The stackable PWM generator is a board designed to mate with the TMS320C6711 evaluation module (EVM). It contains an EPM7128SLC84-6 CPLD by Altera, and it has a JTAG port for reprogramming. The CPLD is contained in an 84-pin PLCC socket for easy removal. The PWM generator implemented in the CPLD contains two pulse registers mapped at 0xB0000000, and 0xB0000004.

Figure # III – 1: Register Formats

Address	31 16	15 0
B0000000	Pulse 2	Pulse 1
B0000004		Frequency

Operation

The PWM generator contains three countdown counters. One counter is used to control the frequency and the other two are used to trigger the pulse outputs. The frequency register accepts an unsigned 18-bit number that will divide the clock frequency by 2*(frequency+2). The pulse count registers control the duty cycle of the PWM output where the duty cycle equals (Pulse1 + Pulse2) / (Frequency+2). The PWM will enforce a deadband of 2.57us so if the pulse count exceeds the frequency count, the deadband will never be violated preventing shoot-through.

F_{PWM} = 50E6 / (Frequency + 2)

Duty = (Pulse1 + Pulse2) / (Frequency + 2)

In System Programming

The prototype board is equipped with a 10-pin header that can be used to reprogram the device with one of Altera's programming devices providing a quick and easy way to reprogram the device.

External Connections

The PWM board has vertical connectors allowing it to be mounted in a stack, and two more connectors for output and reprogramming.

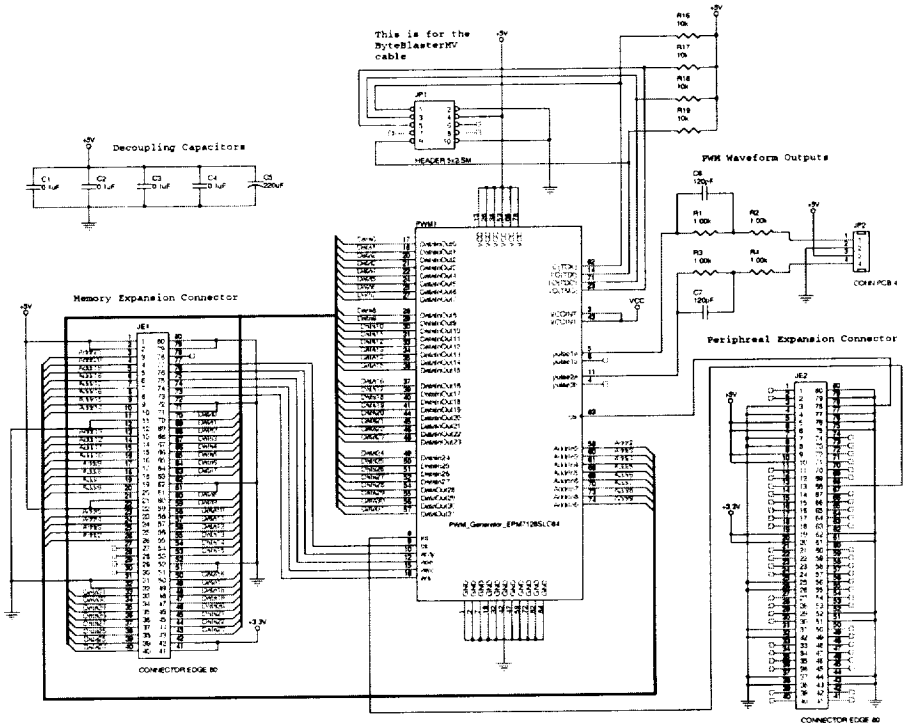
Figure # III – 2: 4-pin output connector

Pin number	1	2	3	4
Connection	Phase 2 Output	GND	5V	Phase 1 Output

Figure # III – 3: Terminals

Designator	Red	Black	1	2
Connection	5V	GND	Phase 1 Output	Phase 2 Output

Figure # III – 4: Stackable PWM Schematic



PWM Generator Program in VHDL:

New PWM Generator

To solve the problems with the 8-bit PWM generator, a new PWM generator was coded. This new PWM generator uses VHDL instead of AHDL. All of the counter registers were widened, and almost all of the control signals were buffered. The output frequency can be changed, and its deadband is enforced at a fixed pulse count.

VHDL vs. AHDL

The new PWM was coded with VHDL using Altera's megafunctions because VHDL has some advantages over AHDL. The megafunctions are a collection of common logic functions optimized for Altera's architecture. Because VHDL is more portable, we will have the flexibility to use CPLDs or FPGAs from other vendors. One disadvantage to using VHDL is that it is not as efficient as AHDL when using Altera's development software. To overcome the performance problem, Altera's megafunctions were used. Using them limits the portability of the VHDL code, but that could easily be modified to use functions from other vendors.

12-bit PWM Components

The goal when coding a PWM generator is to make it run as fast as possible for the given CPLD. To accomplish that the fastest logic blocks, which are logic gates, fast counters, and small FSMs must be used. Logic gates like AND and NOR gates are much faster and smaller than comparators. Loadable decrementing counters are as fast as the clearable incrementing counters used before. Each comparator, counter, and register consumes as much logic as the other so less logic will be consumed (14 before, 9 after). Like a pipeline, flip-flops were used to buffer all of the control signals, except the divider's counter. Because the divider's control signal cannot be buffered, an AND gate was used because they are the faster than NOR gates in Altera's CPLD structure.

PWM Operation

This PWM generator functions differently from the previous design. Maxcount, PulseA, and PulseB registers also act as buffer registers that are updated every cycle during a write. These buffer registers are used to ease the frequency transition between frequencies, and they are updated every cycle during a write. Unlike the other registers, the divider register is updated at the end of every PWM cycle so it uses a buffer to avoid missing a write. The divider's counter is loaded, and all of other counters are enabled whenever the divider's counter counts down to –1. At the end of a pulse width, whenever the maxcount's counter reaches 0 at the start of a new PWM cycle, the PulseA, and PulseB counters are loaded. The PulseB counter is unique in the sense that it is not decremented until the second half of the PWM cycle. The AND logic output for the maxcount, PulseA, and PulseB are buffered, and routed to a finite state machine. It is the job of this state machine to interpret the pulses and generate the pulse width.

Enforcing a Deadband

Enforcing a deadband with this design is very simple. Another NOR gate can be used on the upper bits of the maxcount's counter. That output can be used to force both phase outputs on the finite state machine to off. For example, by ignoring the bottom four bits, the NOR gate will generate and active signal turning off the outputs of the state machine for the last seventeen counts including the delay. With a 100 MHz (10 ns period) clock, that would ensure a deadband of 170 ns. More bits could be ignored resulting in a larger deadband. The limitation of this is that the deadband is a power of two plus another pulse count multiplied by the period of a pulse count. Because deadband is independent of the maxcount, the maxcount can be modified without having to worry about violating the deadband time.

PWM State Machines

The PWM generator is really a pulse generator whose register contents are interpreted by an output state machine. This output state machine is then used to generate the output pulse sequences. There are three state machines in the 12-bit PWM generator. One state machine is used to control the data path in the PWM generator. It controls counter loads, and will enable the PulseB counter when needed. The two other state machines control the phase outputs waveforms, and their role is to interpret the register contents of the data path and generate a PWM signal for each channel. The advantage of this approach is that input registers and the output state machine can be modified without having to modify the core of the PWM generator.

Figure # III – 5: 12-bit PWM RTL

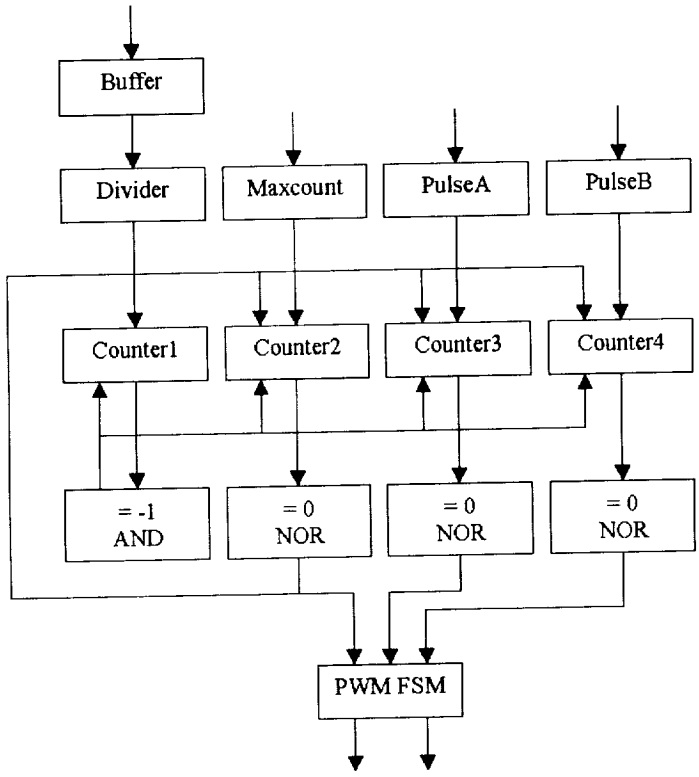


Figure # III – 6: Data Path ASM

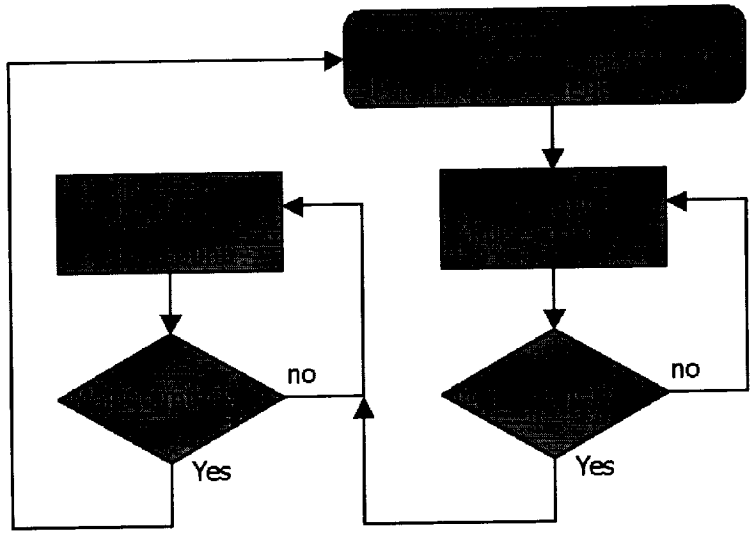
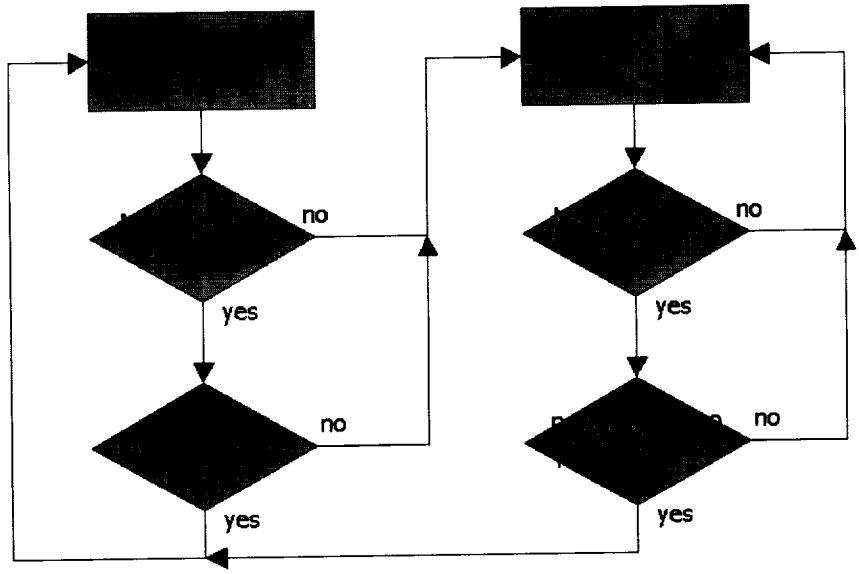


Figure # III – 7: Pulse Output ASM



Problems With the 12-bit PWM

The frequency divider in the 12-bit PWM generator is a problem because it introduces more complexity to the design, and it consumes more logic. It is only useful to generate lower frequencies, but the power converter may never be run below 5kHz so the frequency divider serves little or no purpose. This is also a step backwards because it also reduces the resolution at lower frequencies.

13-Bit PWM

To solve that problem, the 12-bit PWM generator was widened to 13-bits, and the frequency divider removed to improve performance and reduce chip usage. This increase in resolution came at the expense of range. The minimum frequency is now 4.71kHz

assuming a maximum duty cycle of 77.2% (1930/2500). The code was optimized so that the EPM7128SLC84-6 could run with counters larger than 12-bit without sacrificing speed.

Improving Resolution

The resolution is limited by the maximum clock frequency, but it could be overcome indirectly. By varying the pulse count and maxcount, higher resolutions could be emulated at the expense of an accurate output frequency. This would require that the maxcount and pulse registers update at the same time. To accommodate this, the maxcount register was buffered so that it could be updated at the same time as the pulse registers.

13-bit PWM Limitations

Right now, the PWM generator is limited by its clock frequency, and logic capacity. Wider registers are needed to improve the frequency range, and higher clock frequencies are needed to improve the granularity of the PWM generator.

Figure # III – 8: 13-bit PWM RTL

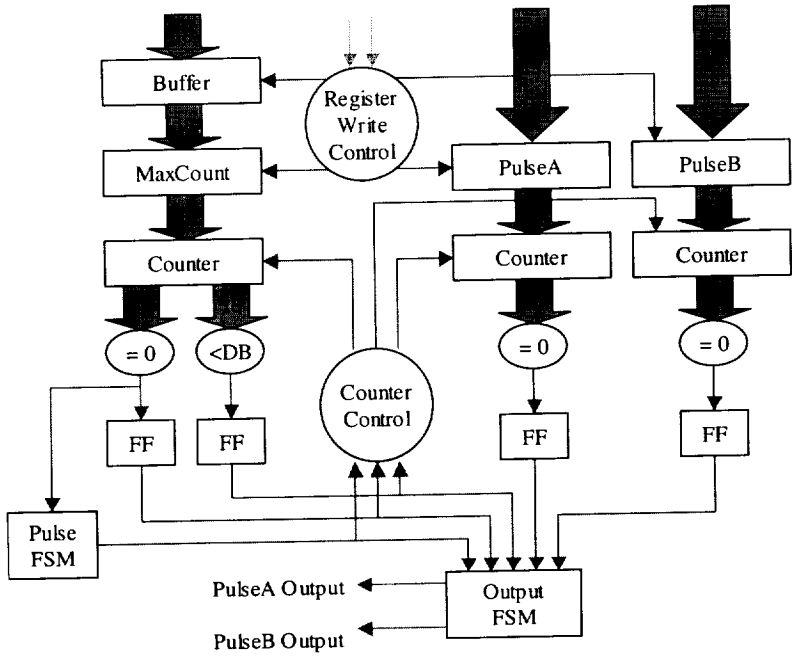
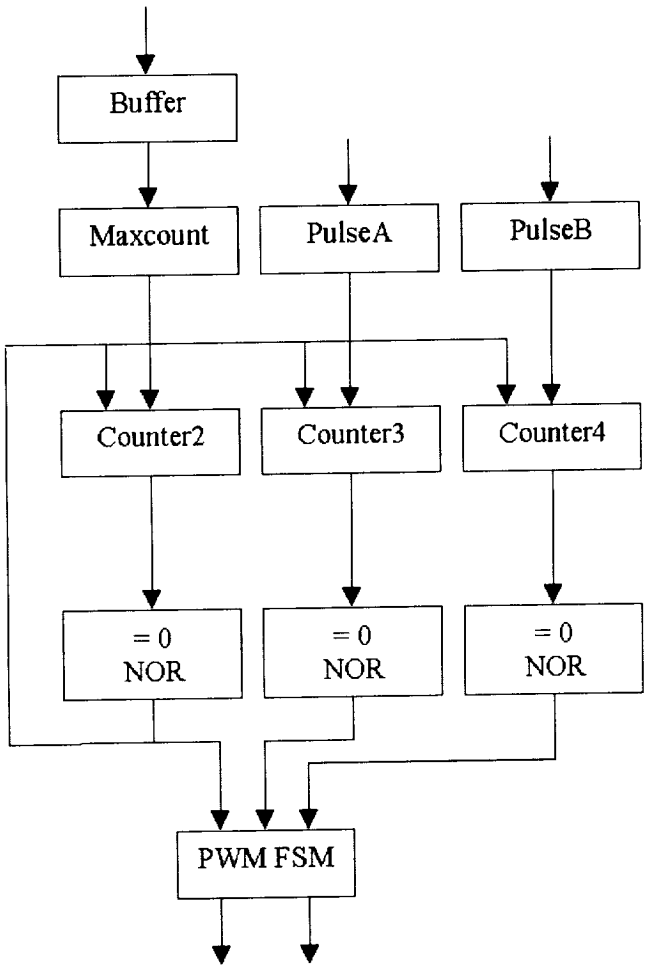


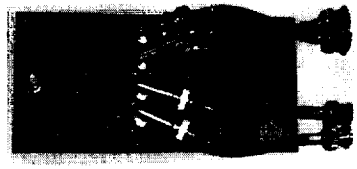
Figure # III – 9: 13-Bit PWM



IV. A/D Board

Purpose

The A/D EVM (Evaluation module was chose because it was specifically designed to work with the TI DSK for the data input for the DSP. This way the circuit was already built and ready to be implemented.



Features

The following product description can be found at:
http://focus.ti.com/docs/tool/toolfolder.jhtml?PartNumber=THS1206EVM

Description

The TH1206 is a 12-Bit, 6 MSPS, Simultaneous Sampling, 4 channel Analog-to-Digital Converter. The EVM allows evaluation of the THS1206 analog-to-digital converters. ... The EVM is specifically designed for interfacing to the DSP starter kits or evaluation modules, which feature the common-connector interface (TMS320C6211, 'C6701, 'C6201, TMS320VC5402).
The different operation modes for the analog input configuration of the THS1206, THS12082, THS10064, and THS10082 are available on the evaluation module. Any channel selection can be done according to the data sheet of each device.

A/D EVM Jumper settings

J1: 1-2	select power source from DSP
J2: 1-2	select power source from DSP
J3: 2-5	select single for AINP not differential inputs
J4: 2-5	select single for AINM not differential inputs
J5: 1-2	single input not differential
J6: 1-2	single input not differential
J7: 1-2	clock conversion from DSP
J8: no connect	just to pull options low
J9: no connect	just to pull options low
J10: closed	CS1 for different CS address settings
J11: open	CS1 for different CS address settings
J12: 2-3	writes signal comes from DSP
J13: 1-2	CS0 chip enable from DSP for address settings

A/D code generator settings

The dc_conf.h header file was created with TI data converter support wizard within Code composer. This wizard automatically generates the header files and functions to use the data converter. This includes the ReadBlock function that we used to read the data converter buffer for the algorithm. The settings for the present one input converter are as follows:

```
/* ADC 1 parameter data */
#define   ADC1_TYPE       THS1206
#define   ADC1_CR0_VALUE   (0x00)
```

```
#define   ADC1_CR1_VALUE    (0xB8)
#define   ADC1_TRIGGER_LEVEL (8)
#define   ADC1_NR_CHANNEL   (1)
#define   ADC1_SAMPLE_FREQ  (6000)
#define   ADC1_SHIFT        (0)
#define   ADC1_INTNUM       (4)
#define   ADC1_BUS_NONE
/* EMIF Interface parameters */
#define   ADC1_ADDRESS      (0xA0020000)
#define   ADC1_RDSETUP      (1)    /* read */
#define   ADC1_RDSTRB       (4)
#define   ADC1_RDHLD        (1)
#define   ADC1_WRSETUP      (4)    /* write */
#define   ADC1_WRHLD        (6)
#define   ADC1_WRSTRB       (3)
#define   ADC1_TIM_PERIOD   (0x0003)

/* DSP parameter data     */
#define   TMS320C6711
/* CSL device build option */
#define   CHIP_6711         (1)
#define   DSP_FREQ          (150)  /* in MHz */
```

Supporting documents

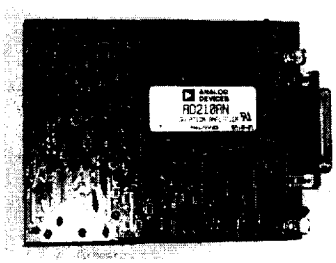
Designing with the THS1206 High-Speed Data Converter

(SLAA094- Updated: 04/04/2000 Application Note

12-Bit, 8 MSPS, Simultaneous Sampling Analog-to-Digital Converters

(SLAS271A- Updated: 06/28/2000) Data Sheet

V. Signal Conditioning Board



The signal conditioning circuitry in a closed-loop control system can have a significant effect on the system speed and response characteristics. Therefore, it is critical that the signal conditioning circuitry be properly designed with the goal of minimizing both signal noise and feedback delay. After applying various techniques to reduce EMI and broadband noise, it became critical to determine a standard method for testing the output of the various signal conditioning circuits. The factors that we used to determine the quality of the signal conditioning circuits from most important too least important were: signal propagation delay, peak-to-peak signal noise, and frequency content. After testing all circuits using the same measurement technique, we were able to compare the various designs to determine the best circuit for our application. To help combat some of the EMI problems, we experimented with ferrites and obtained significant improvements. Once noise levels were decreased, the focus of improvement was aimed more towards decreasing signal propagation delay. In the end, the final circuit decision was made based on a combination of signal noise and signal propagation delay.

Introduction

The signal conditioning board configuration had remained unchanged since the original design in the early stages of the project. Since that time, the signal conditioning circuit seemed to serve its purpose well. However, as more researchers began working in the lab, we were able to focus on previously overlooked issues. One being the EMI/noise problem and the other was loop delay times. Both of these issues have a rather significant effect on the performance of the controller, and ultimately the power converters output characteristics. Therefore, it became absolutely critical that we look deeper into the issues that we recognized as problems from the start. Some of the questions that led to this new area of research were: How much delay does the signal conditioning circuit add to the feedback loop? How much noise can be eliminated from the feedback signal? Noticing that the addition of filter stages did not reduce the amount of noise, we realized that EMI and common mode signals were to blame. We were able to suppress some of this EMI noise by implementing ferrites that act to impede common mode signals. That is, the ferrite acts as a high resistance to common mode signals, which provides a place for the common mode signal energy to be dissipated. Generally, EMI is common mode because each signal line will conduct the same amount of electromagnetic interference unless the wires differ in length, size, or orientation with respect to the radiating source. Since we noticed that the addition of extra filters provided no additional noise reduction, our initial question was: How many poles can be eliminated from the present filter design without compromising any performance? The answer turned out to be a very significant amount, making the new design much faster and more compact.

Key Circuit Parameters

Signal propagation delay was measured with an oscilloscope using a function generator to periodically produce square wave signals that were fed to the input of the signal

conditioning circuit. Using the function generator as a trigger to capture the image, the propagation delay could easily be determined. The peak-to-peak signal noise was much more difficult to get an accurate measurement of. There are a few reasons why it is so difficult. First, the noise spikes, which are normally due to EMI at the switching frequency, are sometimes negative and sometimes positive. If a large positive spike and negative spike appear on the oscilloscope screen at the same time, an extremely large peak-to-peak value will be obtained. At the same time, if a small spike occurs on the screen, then a very small peak-to-peak value will be obtained. Since these extreme values are usually not seen very often (only a few time per minute), it would not be sufficient to record the highest or lowest value as the peak-to-peak noise level for the circuit. Therefore, the peak-to-peak noise values are observed over a time period of a few minutes. An estimated average is then determined for the circuit using the fluctuating values from the scope.

Need For Improvement

When the project was first started in 1999, a breadboard signal conditioning circuit was produced to remove the ripple from the converter output. This circuit was designed with aliasing as the main concern. This led to very low corner frequency, multiple pole filters, that added significant delay to the circuit. This was not a big problem early on due to the fact that the controller gains settings were not extremely aggressive. Therefore, the original circuit board served its purpose for about two years without any significant problems. However, as the controller gains began to increase along with controller complexity, noise levels began to produce undesirable results, leading to an increased interest in optimizing the signal conditioning circuitry.

Signal Measurement Methods

Before we began working to improve the signal conditioning circuit, a method for testing the circuits was devised that would provide consistent measurement results for comparison purposes between the various circuit configurations. Unfortunately, the method was determined to be anything but consistent about half way through the improvement process. It was decided that a scope probe measurement at the output of the signal conditioning board circuit would be taken by clipping the scope probe to the ground and output wires of the various signal conditioning circuits. One problem was that the results of the measurements were discovered to be different depending on the time the measurements were taken. It seemed that, at times, measurements taken one day could be drastically different on the following day for the same test parameters. Another problem was that the probes themselves would pick up some noise depending on their placement or orientation to the power converter while it was running under full load conditions. This led us to look at shielding as a way of minimizing the observed noise, and other possible measurement methods.

Shielding

After just a small amount of time working on shielding the signals, it became apparent that proper grounding techniques could produce a significant improvement in the noise content of the signal conditioning circuits. The biggest improvement was achieved when the sensing cable connected from the power converter output to the signal conditioning board input, which had previously been twisted pair, was replaced with a shielded twisted

pair cable. The noise level was reduced by a factor of two in this case using the original scope probe measurement technique. Unfortunately, not all shielding attempts were as productive. At times, the connection of ground or shield wires to what was believed to be the proper ground or shield produced increases in noise levels rather than the expected decrease. This was a bit frustrating, but led us in other research directions. It seemed that the shield connections that produced more noise were actually acting as antennas rather than shields. This directed us more in the direction of electromagnetic interference (EMI) reduction. It may be possible that the missing EMI filter on the converter may add to the electromagnetic compatibility (EMC) problems. To combat the EMI problem, common mode chokes were introduced to the signal conditioning board output signal wires. The main goal here was to reduce any common mode noise just before the A/D converter samples the signal.

Common Mode Chokes

The common mode choke is basically wired like a 1:1 transformer, but not used like a transformer is normally used. Rather, the circuit uses the transformer so that there is no galvanic isolation from input to output. Instead, the "primary" and "secondary" are used to close the path of the signal wire from input to output of the common mode choke. Since the choke is wound 1:1, the current through the "primary" should equal the "secondary" but in the opposite direction. Therefore, by running our forward path of the signal through one side of the choke, and the return path of the signal through the other side of the choke and in the opposite direction, then the choke will have no effect on the signal that is being passed through it. However, if common mode signals appear on both signal paths, the choke will act as a high impedance path to the common mode signal. For example, say that a 100mA signal flows through one side of the choke and return through the other side of the choke so that the path is closed. At this point, the common mode choke will have no effect on the signal passing through it. However, if there is common mode noise of 20mA, then the forward signal current becomes 120mA while the return path drops to 80mA. At this point, the common mode choke acts as a transformer with different current levels flowing through the "primary" and "secondary." This will cause the common mode choke to drive both the forward and return path to the same current level because of the chokes current equalizing effect.

Significant improvements were observed with the addition of a common mode choke to the signal conditioning output path. This method can also have some inconsistent results depending on the orientation and location to the converter or other physical components. One example of this occurs when the core material of the common mode choke it touched by a hand or possibly a metal tool. The noise levels would sometimes double as a result of contact with a person or tool. After researching this topic, it was found that BNC cables act as common mode chokes due to the structure of the cable and its mutual inductance. Therefore, in situations where it is not possible to use common mode chokes due to the size or any other reasons, BNC cable is the next best thing. It would be ideal to use a BNC cable with the common mode core to incorporate the best of both techniques. For this reason, we have tried to use BNC cables wherever it is physically possible. The BNC cable also provides good shielding characteristics in addition to its common mode noise reduction capability.

Final Circuit Design

Now that the various improvement techniques have been discussed, attention can now be turned to the final implementation and design of the signal conditioning circuit. As the design came closer and closer to finalization, there were two main circuits that were tested and modified in parallel. One of theses circuits was breadboarded, while the other circuit was a printed circuit board created with design flexibility in mind. Although the printed circuit board seems like it would be a final design, the board was made so that many variations could be made for future improvements. For a short time, both the breadboarded and printed signal conditioning circuits were configured in the same way. However, it was observed that the printed signal conditioning circuit was not performing as well as the breadboarded signal conditioning circuit. This problem was discovered when the closed loop performance of the converter, using the printed signal conditioning board, was found to have small oscillations. It was discovered that the iso-amp was the cause of part of the problem. When the iso-amp was replaced, similar operation was observed between the two circuits.

The measurements shown on the following page are transient or propagation delay plots captured on an oscilloscope. A function generator generated the step input to the two circuit boards. The result of the measurement proves that both boards exhibit similar performance. It has been verified that both circuits are adequate for use with the advanced nonlinear digital control algorithm. Final tests were being conducted at the same time this report was written. Therefore, a revision will be made at the end of the final testing period that will include frequency response data and other pertinent measurements.

Figure # V – 1: Step Response of the Printed Signal Conditioning Circuit

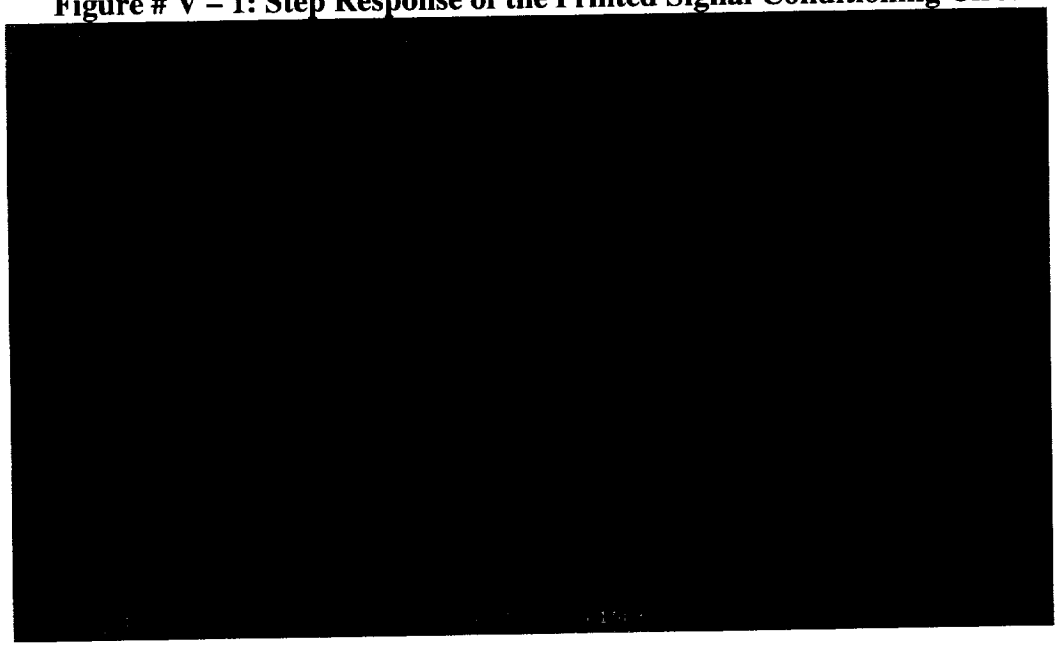
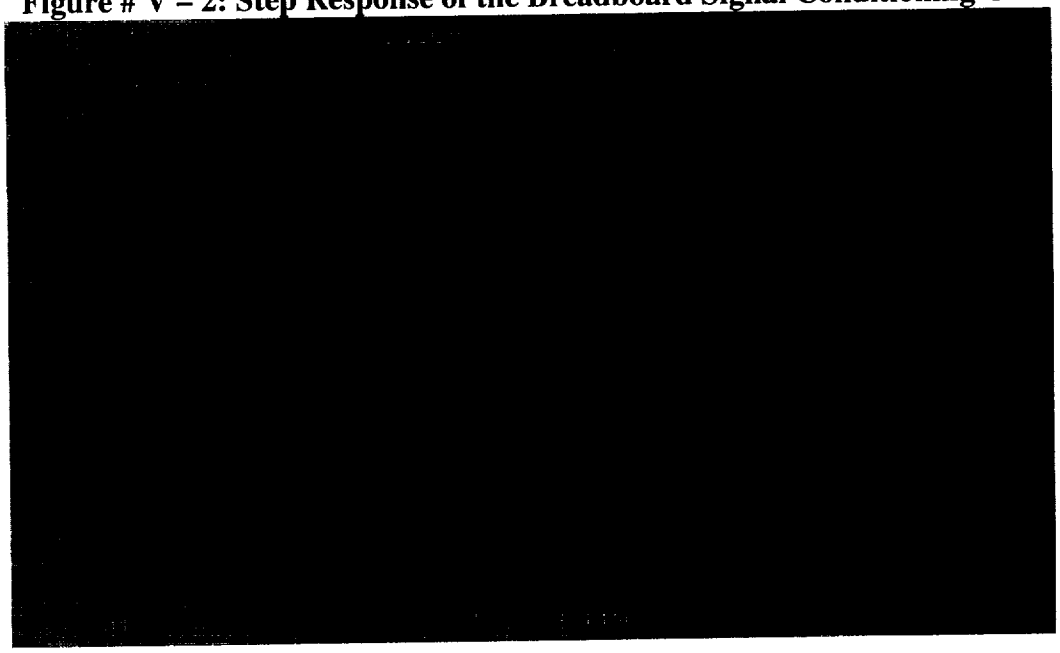


Figure # V – 2: Step Response of the Breadboard Signal Conditioning Circuit



AERL Team: stackable system

- **Arthur Stachowicz**
  - Enclosure
  - PWM Generator in AHDL
- **Aaron Radke**
  - Work with the Code Composer Studio 1.23 and 2.0
  - Work with the DSP and A/D
- **MinShao (James) Zhu**
  - Nonlinear control Algorithm for the DSP
- **Greg Tollis**
  - Signal Condition design, testing, and building.
- **Ivan Jurcic**
  - PWM Generator in VHDL
  - Prototype board layout for the PWM
- **Zhan (John) Ping**
  - Working with the A/D and Code Composer Studio 2.0
- **Dave Wladyka (graduated)**
  - Researched and bought the DSP stack
  - Initial work with Code Composer Studio 1.23 and DSP
- **John Sustersic (graduated)**
  - Signal conditioning board layout

Contact Information

Cleveland State University
Advanced Engineering Research Laboratory, SH 327
1960 East 24th Street
Cleveland, Ohio 44115

Phone: 216-875-9668
Fax: 216-687-5405
Email: z.gao@csuohio.edu
Website: http://academic.csuohio.edu/aerl

Final Report

Volume I

Development of Digital Control Technology for Power Management and Distribution Systems

Grant No. NCC3 – 699

Submitted to NASA Glenn Research Center

Submitted by:

Jack Zelina, Senior Research Scientist
Zhiqiang Gao, Principle Investigator

Department of Electrical and Computer Engineering
Fenn College of Engineering
Cleveland State University
Cleveland, Ohio 44115

December 10, 2001

Table of Contents

MODELING OF A FULL-BRIDGE DC-DC POWER CONVERTER

MARCELO C. GONZÁLEZ

Master of Science in Physics

John Carroll University

August, 1996

Bachelor of Science in Engineering Physics

John Carroll University

May, 1994

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December, 2001

This thesis has been approved
for the Department of Electrical Engineering
and the College of Graduate Studies by

_____

Thesis Committee Chairman, Dr. Zhiqiang Gao

_____

Department/Date

_____

Thesis Committee Member, Dr. Dan Simon

_____

Department/Date

_____

Thesis Committee Member, Dr. Ana Stankovic

_____

Department/Date

To my wife Lynn and daughter Lillian

ACKNOWLEDGMENTS

First and foremost, I express my gratitude to my advisor, Dr. Zhiqiang Gao, for giving me the opportunity to be a member of the Advanced Engineering Research Lab (AERL) and for his patience, guidance, and never-ending support throughout this thesis. I also wish to express similar words of appreciation for the tremendous help provided to me by Mr. John (Jack) Zeller and for his willingness to assist with a smile.

Many thanks to Mr. Jim Soder and Mr. Rob Button at NASA Glenn Research Center for their generous and continued support of the efforts of the Advanced Engineering Research group.

Special thanks to Dr. Ana Stankovic for her assistance in obtaining the simulation software SABER®, to Dr. Louis R. Nerone at GEL-NELA for his assistance in obtaining accurate converter component measurements, and to Dr. Eugenio Villaseca and Dr. Dan Simon for taking the time to answer my numerous questions.

I would like to give thanks to Jan Basch, Adrienne Fox, Minshao Zhu, Charles Alexander, Dave Wladyka, Weiwan Wang, and John Sustersic for their support in various aspects of my studies and thesis work.

Finally, to my wife Lynn, I would like to extend my deepest gratitude for her whole-hearted faith in me and support through these years. I also wish to thank my parents, my in-laws and the rest of my family here in the United States of America and in Belize.

MODELING OF A FULL-BRIDGE DC-DC POWER

CONVERTER

MARCELO C. GONZÁLEZ

ABSTRACT

The main thrust of this work was to derive two nonlinear models for the 1-kW ED408043-1 Westinghouse Full-Bridge DC-DC Power Converter, with a center-tapped transformer, as part of a converter digital control study. The models will be used in the evaluation of nonlinear and linear control strategies.

The first model is a nonlinear SABER® simulation model. This is a component-level model and, as such, it also lends itself to the study of performance tradeoffs due to modifications in the converter topology. The second model is a circuit-level, piecewise-linear, mathematical model implemented in MATLAB®. Due to its mathematical nature, this model is better suited for nonlinear mathematical analysis. Both the simulation model and the mathematical model are valid for Continuous Conduction Mode (CCM) and Discontinuous Conduction Mode (DCM) of operation. For comparison purposes, a linear transfer function model and a state-space average model were also derived. However, these models are only valid for the neighborhood about the operating point for which they were derived.

The SABER® model yielded the best overall steady-state and transient response results. For example, its steady-state output voltage was within 1.5% of the actual output voltage, while the other three models predicted steady-state output voltages within 4% of the actual values. Of all four models, the mathematical model requires the longest computing time due to the fixed integration step, but yielded the second best overall results.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Preamble

Modeling, in general, is the representation (description) of a component or system in terms of mathematical expressions that approximate the actual characteristics (behavior) of the component or system. The accuracy of the approximation is dependent on the modeling approach taken, with highly accurate models resulting in complex mathematical expressions (both) at a component or system level.

There are two approaches that can be taken in modeling power electronic systems, such as the Full-Bridge DC-DC Switching Power Converter. The first is circuit-oriented and can be accomplished with the use of software packages such as PSPICE®, SABER®, or EMTP®.[1] With these simulation packages, the task is to capture the converter schematic with the built-in component models and set the correct values of the component model parameters. Some components, like diodes and transistors, have several built-in models ranging in complexity (accuracy). Which model to use depends on the component parameters available and accuracy of the model desired.

The second approach is to derive a mathematical description by means of algebraic and differential equations, which are then solved with programming languages such as PASCAL, FORTRAN, and C or special software packages such as MATHCAD®, MAPLE®, and MATLAB®. What aspect of the converter to model and the level of accuracy desired will dictate the complexity and approach taken to derive these expressions. An alternative to the second approach is to empirically derive an input-output expression. This method is only applicable if the converter to be modeled has been designed and built.

The principal thrust of this research was to develop a circuit-oriented model using SABER® and a mathematical model, implemented in MATLAB®, of the digitally-controlled ED408043-1 Westinghouse Full-Bridge DC-DC Switching Power Converter to aid in the development and analysis of a control system that will guarantee the operational objectives of the power converter under disturbances. The models are to reflect the inherent nonlinearities of the power converter to as high a degree of accuracy as possible while at the same time keeping the complexity of the models to a minimum. If need be, these models can play a crucial role in improving the design since it is easier to alter the topology or parameters and analyze the effect in the models than in the hardware.

Over the past thirty years, significant work has gone into the area of converter modeling but the majority of the work has focused on the mathematical model of simple converter topologies and the use of some form of averaging in the model.

1.2 Literature Review

Most of the work that has gone into mathematical modeling of DC-DC switching power converters since 1972 has focused on the three basic topologies: Buck, Boost, and Buck/Boost (See Figure 1-1). In addition, of the two main modeling approaches at the time, one was in the domain of equation manipulation to reduce the state-space representation of the converter to a single linear expression and the other employed circuit manipulation to result in a single equivalent linear circuit. The researchers were driven to derive linear expressions or circuits to reduce the computing time due to the limited computing power available at the time. The latter approach is briefly covered in Reference [3], but additional information can be found in its references.

In 1976, F. C. Lee and Y. Yu [2] introduced a new approach that modeled both modes of operation—the CCM and the DCM. Before 1976, only the continuous conduction mode of operation had been modeled. Their goal was to obtain a small-signal model for each of the three converters by calculating the discrete impulse response using the discrete solution of the state-space representation. The discrete impulse response was then transformed to the continuous-time domain. By taking the Laplace Transform of the resulting impulse response, the frequency domain transfer function was obtained. This model, according to the authors, is valid for small-signal low frequency characteristics up to one-half the switching frequency. This is the same as requiring the natural frequency of the converter to be much smaller than the switching frequency.

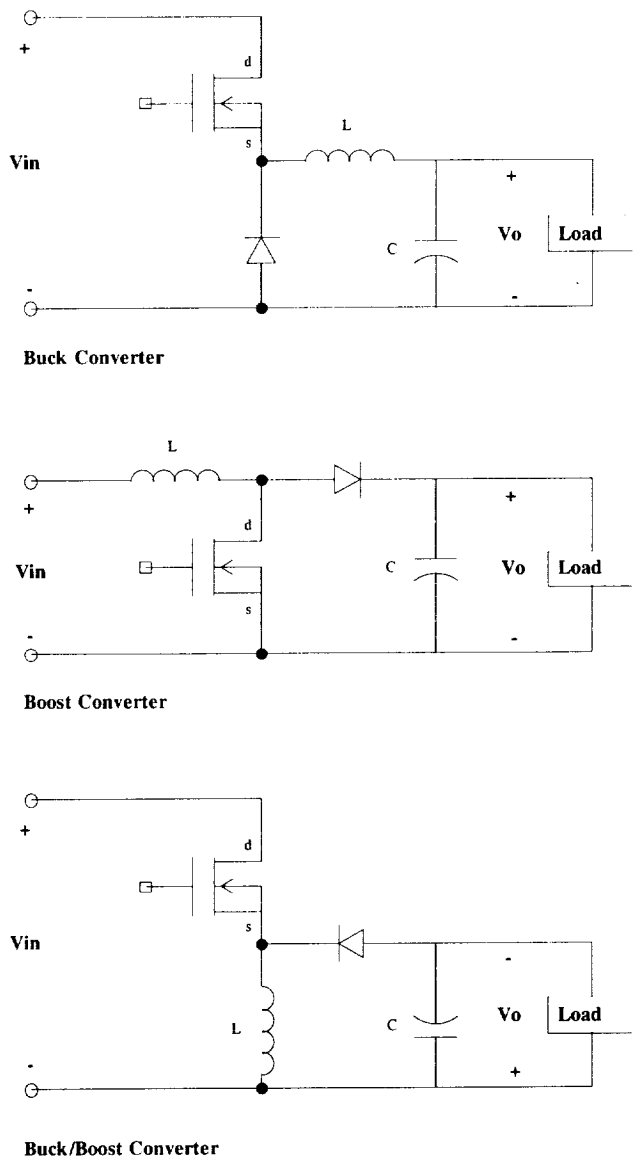


Figure 1-1: Buck, Boost, and Buck/Boost Topologies

That same year, Slobodan Cúk and R. D. Middlebrook [3] published their work on a new averaging technique—State-Space Averaging—that would dominate the area of switching power converter modeling ever since mainly due to the mathematical simplicity in deriving the model. Due to the popularity of this technique, a brief derivation is given below for the CCM.

When the switch is on:

$$\begin{aligned} \dot{x} &= A_1 x + B_1 u \\ y_1 &= C_1 x \end{aligned} \tag{1.1}$$

When the switch is off:

$$\begin{aligned} \dot{x} &= A_2 x + B_2 u \\ y_2 &= C_2 x \end{aligned} \tag{1.2}$$

Equations (1.1) and (1.2) are then weighted by multiplying (1.1) by d and (1.2) by (1-d). The resulting equations are then added to obtain the state-space average. The result is:

$$\begin{aligned} \dot{x} &= [A_1 d + A_2(1-d)]x + [B_1 d + B_2(1-d)]u \\ y &= [C_1 d + C_2(1-d)]x \end{aligned} \tag{1.3}$$

Thus, the average is over a single period, T. Note that the result is in the form of (1.1) and (1.2). The duty-ratio, d, is constant over a switching period and is part of the matrices in (1.3). Because of the inherent approximation of the fundamental (transition) matrix in going from (1.1) and (1.2) to (1.3), this model holds true if the natural frequency of the converter is much less than the switching frequency.

To obtain the input voltage-to-output voltage and duty ratio-to-output voltage transfer functions, the system must be perturbed, that is.

$$\begin{aligned} d &= D + \hat{d} \\ u &= U + \hat{u} \\ x &= X + \hat{x} \\ y &= Y + \hat{y} \end{aligned} \tag{1.4}$$

where D, U, X, and Y are the nominal (steady-state) values and $\hat{d}$, $\hat{u}$, $\hat{x}$, and $\hat{y}$ are the deviations from the nominal values. Substituting (1.4) into (1.3) results in the steady state model (1.5), where $\dot{\hat{x}} = 0$, and the dynamic (1.6) model:

$$AX + BU = 0$$
$$Y = CX \tag{1.5}$$

$$\dot{\hat{x}} = A\hat{x} + B\hat{u} + [(A_1 - A_2)X + (B_1 - B_2)U]\hat{d} + (A_1 - A_2)\hat{x}\hat{d} + (B_1 - B_2)\hat{u}\hat{d}$$
$$\hat{y} = C\hat{x} + (C_1 - C_2)X\hat{d} + (C_1 - C_2)\hat{x}\hat{d} \tag{1.6}$$

where $A = DA_1 + (1\text{-}D)A_2$, $B = DB_1 + (1\text{-}D)B_2$, $C = DC_1 + (1\text{-}D)C_2$. The dynamic model can be linearized by ignoring the terms with products of $\hat{x}$ and $\hat{d}$, and $\hat{u}$ and $\hat{d}$. The result is the bilinear expression

$$\dot{\hat{x}} = A\hat{x} + A2\hat{d} + B\hat{u}$$
$$y = C\hat{x} + D1\hat{d} \tag{1.7}$$

where $A2 = (A_1\text{-}A_2)X + (B_1\text{-}B_2)U$ and $D1 = (C_1\text{-}C_2)X$. To obtain the input voltage-to-output voltage transfer function, $\hat{d}$ is set to zero and the following expression applied

$$\hat{G}_1(s) = C(sI - A)^{-1}B \tag{1.8}$$

For the duty ratio-to-output voltage, $\hat{u}$ is set to zero and the following expression applied

$$\hat{G}_2(s) = C(sI - A)^{-1}A2 + D1 \tag{1.9}$$

Six years later, the authors of Reference [3] teamed up with Robert W. Erickson and published Reference [4]. Their work resulted in a discrete large-signal model which, when converted to time-domain through Euler's forward-differencing approximation ($dx(t)/dt = (x_{n+1}\text{-}x_n)/T_s$) , merely resulted in (1.6) with one twist—u was held constant. Simulation of the discrete model was accomplished with a BASIC program.

Another modeling approach that used the state-space averaging technique as the starting point was published in 1981 by K. Harada and T. Nabeshima.[5] Their work focused on the Buck converter. The time-domain expressions for the inductor current and

output voltage were obtained by taking the Laplace Transform of the state-space averaged model.

Thus far, the switches (Transistors and Diodes) have been assumed to be ideal, that is, there is no on-resistance, no voltage drop, no turn-on or turn-off time, no turn-on time delay, no turn-off time delay (storage time), etc. In 1980, the effect of storage time modulation of a Bipolar Junction Transistor (BJT) was studied.[6] Storage-time modulation is basically the effect in which the base duty-ratio modulation is seen at the collector as a slightly different duty-ratio modulation due to the excess stored charge. The time taken to remove the excess stored charge depends on the type of base drive and collector current. While the excess stored charge is being removed (with reverse base current), the collector current remains constant. The starting point of this approach was state-space averaging. An expression for $\hat{d}$ was then derived to incorporate the storage-time modulation. The limitation of this approach, in addition to that of the state-space averaging technique, is that the expression relies on the transistor current gain and a measured parameter that is only valid for the specific conditions under which it was measured. Also, with the introduction of the MOSFET and continued improvement of these semiconductors, this effect becomes more and more negligible.

The switching-function approach introduced in 1988 attempts to model the inherent switching behavior of the basic converters—rise time, delay time, and storage time of the transistor and reverse-recovery of the diode.[7] The modeling problem was approached by considering the fact that the matrices A, B, C, and D are time-dependent and are varied by the switches in the converter. The switching functions are derived by studying

the switching waveforms of the converters. The drawback of this approach is that it is mathematically involved even for the basic converter topologies.

The transistor and diode in all three basic topologies and the Cúk converter together behave as a single-pole double-throw switch, called the PWM switch.[8] This switch can be replaced with an equivalent circuit model. The derivation of the circuit model starts by looking at the average relationship among the currents and voltages through and across the switch terminals for a specific converter. Once these relationships are obtained, the system is perturbed about an operating point and the effect introduced into the relationships. The currents and voltages are then represented with a Fourier Cosine series. These expressions are then used to derive the dc, fundamental, or harmonic switch circuit models for simulation. So far, this is the only approach that models the actual switching effect on the output voltage and inductor current. However, as in the previous case, this approach is also mathematically involved.

Continued work in the area of modeling of switching power converters has resulted in numerous large-signal modeling techniques some of which incorporate the switching behavior (ripple) of the waveforms. [9]-[14]

1.3 Full-Bridge DC-DC Converter

As opposed to the basic DC-DC converter topologies (Buck, Boost, and Buck-Boost), the ED408043-1 Westinghouse Full-Bridge DC-DC converter is more complicated. This converter topology (See Figure 1-2; the units are in Ohms, Henries, and Farads) has an H-Bridge consisting of four active switches (MOSFETs) that require two 180° out-of-phase

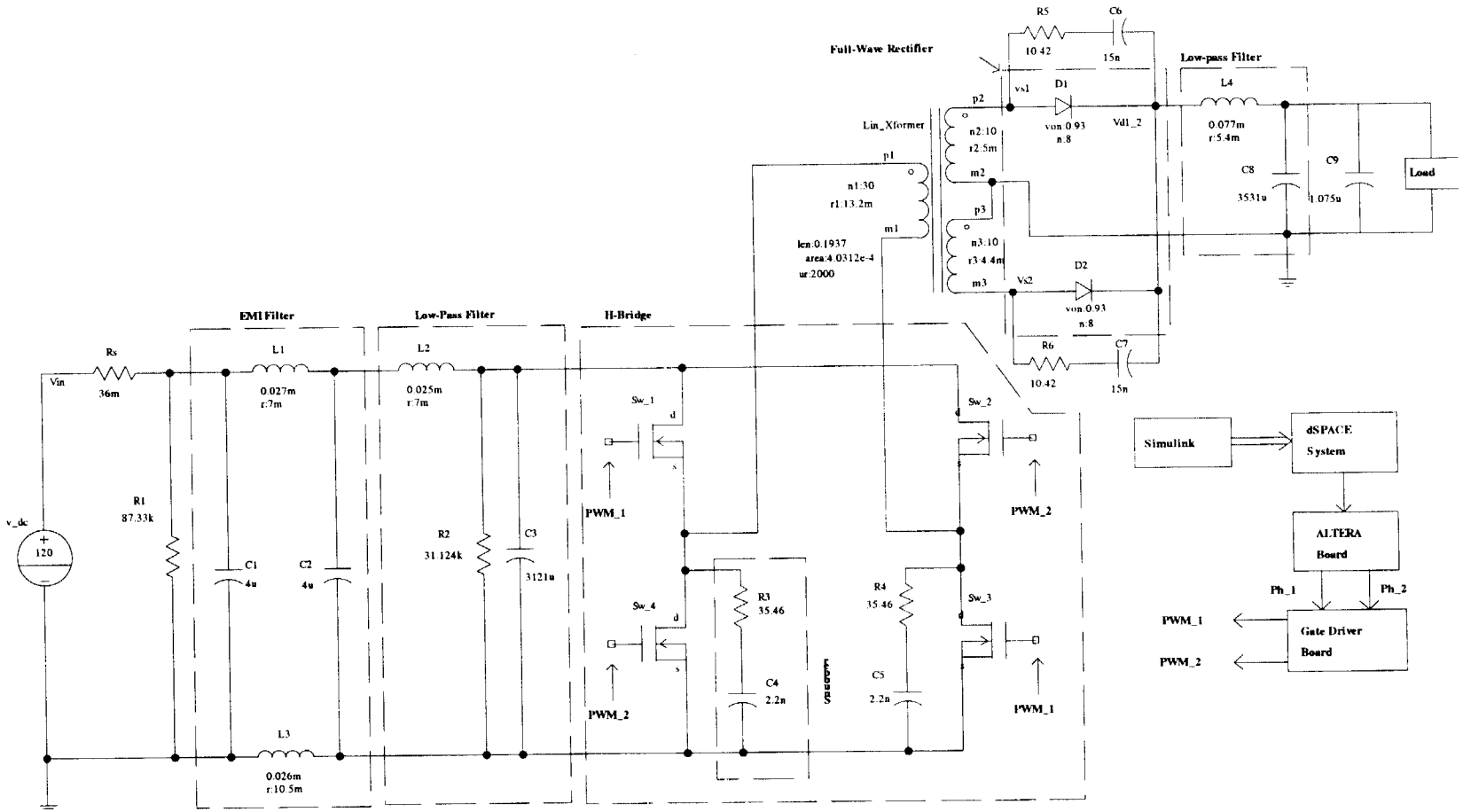


Figure 1-2: Schematic of the Westinghouse Full-Bridge DC-DC Switching Converter

Pulse-Width Modulation (PWM) input signals, two passive switches (Diodes), and a step-down isolation transformer. In contrast, each of the basic converter topologies only has one active switch that requires only one PWM input signal, one passive switch, and no transformer. In addition to the output low-pass filter present in all converter topologies mentioned above, the Full-Bridge DC-DC converter also has an input Electromagnetic Interference (EMI) filter and an input low-pass filter. The output EMI filter originally designed into this converter has been eliminated. It is important to note, that in addition to reducing the EMI conducted to the output, a secondary function of the output EMI filter is to smooth-out the ripple on the output voltage.

Inherent in switching power converters, especially hard-switching designs, is the generation of EMI due to the fast switching action of the MOSFET switches that result in waveforms with high *dv/dt* and *di/dt*. Most of the generated EMI is conducted back to the power supply and to the output and the rest is radiated. EMI standards restrict the level of acceptable conducted noise thus requiring the need of EMI filters and snubbers across the switches. The radiated noise is reduced by the metal enclosure of the switching power converter.

The input supply to the DC-DC converter does not have to be highly regulated. To minimize the variation of the DC voltage across the H-Bridge, an input low-pass filter is added between the EMI filter and the H-Bridge. A similar function is provided by the output filter where the high frequencies present in the square wave at the output of the full-wave rectifier must be filtered out. The result is a DC voltage with ripple that is dependent on the corner frequency of the low-pass filter and the capacitor equivalent series resistance (ESR).

The transformer provides input-to-output electrical isolation for safety reasons and also steps down the voltage at the secondary to a third of the voltage at the primary. This is the result of a 3:1 turns ratio that is derived based on the design requirements, specifically, the DC output voltage level.

The two out-of-phase PWM signals, Ph-1 and Ph_2, used to alternately turn on and off the two switch pairs were generated by an ALTERA® Complex Programmable Logic Device (CPLD) based on the two 8-bit signals received from the dSPACE™ rapid prototyping system. In addition, the CPLD is programmed to provide a blanking time to prevent the two switch pairs from being on simultaneously. For this study, the system was ran in open loop thus the two 8-bit signals were constant, except for PWM step-changes, and generated by downloading a program, created using Real-Time Workshop in SIMULINK®, into the DSP chip in the dSPACE™ system (See Figure 6 of [23]). The CPLD PWM signals were then fed to the Gate Driver Board. On the Gate Driver Board, each signal is fed to an opto-coupler for isolation, the output of which is sent to one of the two high and low side drivers. In Figure 1-2, the output of the high and low side drivers are labeled PWM_1 and PWM_2, where PWM_1 drives Sw_1 and Sw_3 and PWM_2 drives Sw_2 and Sw_4. The fixed switching frequency, f_s, was measured at 21.7 kHz.

When the converter is operating in steady-state, the output inductor (L4) voltage is periodic with period $T_s/2$, where T_s is the switching period and the time integral over one period is zero. In other words, the energy stored is equal to the energy released. Figure 1-3 is the voltage across L4 (v_{L4}) and the load. When either switch pair, (Sw_1, Sw_3) or (Sw_2, Sw_4), is on for t_{on}, the voltage across L4 is

$$v_{L4} = \frac{N2}{N1} V_{dc} - V_{out} \tag{1.10}$$

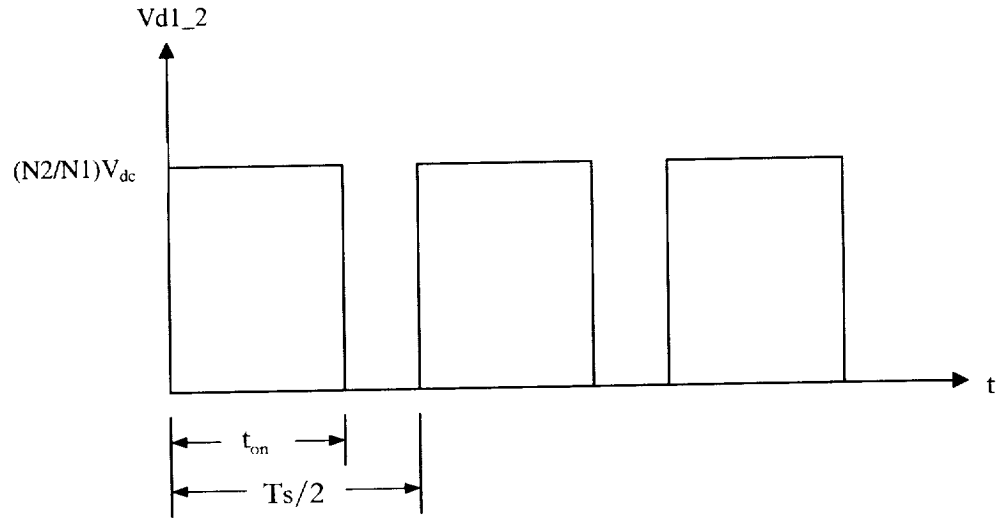


Figure 1-3: The Voltage Waveform at the Full-Wave Rectifier Output

and, when both pairs are off for $t_{off} = T_s/2 - t_{on}$, the voltage across L4 is

$$v_{L4} = -V_{out} \tag{1.11}$$

The integral over one period is written as

$$\int_0^{t_{on}} (\frac{N2}{N1} V_{dc} - V_{out}) dt + \int_{t_{on}}^{\frac{T_s}{2}} -V_{out} dt = 0 \tag{1.12}$$

Solving for V_{out} and setting $D = t_{on}/T_s$ yields the steady-state input-output equation as a function of the turns ratio and the duty ratio (D)

$$V_{out} = (2\frac{N2}{N1} D) V_{dc} \tag{1.13}$$

The Full-Bridge Switching Power Converter under study was designed to operate with a nominal input voltage of 120V with a desired output voltage of 28V at a load ranging from 0 to 40 Amps. Therefore, from (1.13) and a turns ratio (N_1/N_2) of 3, the duty ratio for **each** switch pair should be 0.35, excluding losses.

1.4 Thesis Organization

The following chapter, chapter two, covers the SABER® simulation model. The Full-Bridge Switching Power Converter is broken down into four functional stages: Filters, H-Bridge, Center-Tapped Transformer, and Full-Wave Rectifier. In addition, there is a section on PWM Generation.

In chapter three, the state-space representation of each of the three, switched topologies of the converter is derived for both resistive load and current load by means of differential and algebraic equations. The state-space (MATLAB®) model is the combination of these three state-space representations in an M-file included in the appendix. Additionally, section 3.3 briefly covers the derivation of the input-output based linear transfer function model and the last section goes through the derivation of the state-space averaged model for current load.

The first three models are verified in chapter four by comparing the simulation results to data of the actual converter. The averaged model is compared to the switched state-space model in the last section with the SABER® results as the baseline; it is necessary to compare both the steady-state response and the dynamic response.

CHAPTER II

SABER SIMULATION MODEL

2.1 Introduction

As mentioned previously, two nonlinear models are derived in this work. One is at a component level and the other is at a circuit level. The component-level model is, in this case, achieved using the simulation software SABER®. This approach will yield the most accurate and versatile model. Not only can the model be used for controller design and verification but it can also be used to study variations in the converter topology and component values that may result in an improved design.

This chapter discusses the component-level model and the template selection from those offered by SABER® for each distinct converter component based on the available component parameter(s).

2.2 Converter Stages

The simulation software SABER® offers a few different Graphical User Interfaces (GUI), one of which is SaberSketch™ for capturing schematics for simulation and the

other is SaberScope™ for viewing and analyzing the simulation results. Since SaberSketch™ offers multiple templates for each component, the task will be to select the appropriate template. The choice of template will depend, in this case, mostly on the component parameters available from data sheet or measurements since the goal is to model an existing converter. As mentioned in section 1.4, the Full-Bridge DC-DC Converter under study is divided into four functional sections and each section is addressed separately. The simulation model used to compare to the converter data is shown in Figure 2-1.

2.2.1 Filters

As briefly discussed in section 1.3, the Full-Bridge Switching Power Converter under study has an EMI filter and a low-pass filter at the input and a low-pass filter at the output. The original converter had an output EMI also. Both the EMI filter and the low-pass filter are composed of inductors and capacitors, that is, ideally loss-less components, with the exception that the input low-pass filter has a 31kΩ resistor across the large capacitor bank to discharge the capacitor when the system is turned off. However, since the resistor is in parallel and has a large resistance, the power dissipated is negligible. On the other hand, the parasitic resistances of the inductors (and transformer windings) contribute to losses, especially at high current.

2.2.1.1 Input Filter

The real capacitor can be modeled by an equivalent circuit shown in Figure 2-2.[15] The parasitic components in a capacitor are the equivalent series resistance (ESR), the

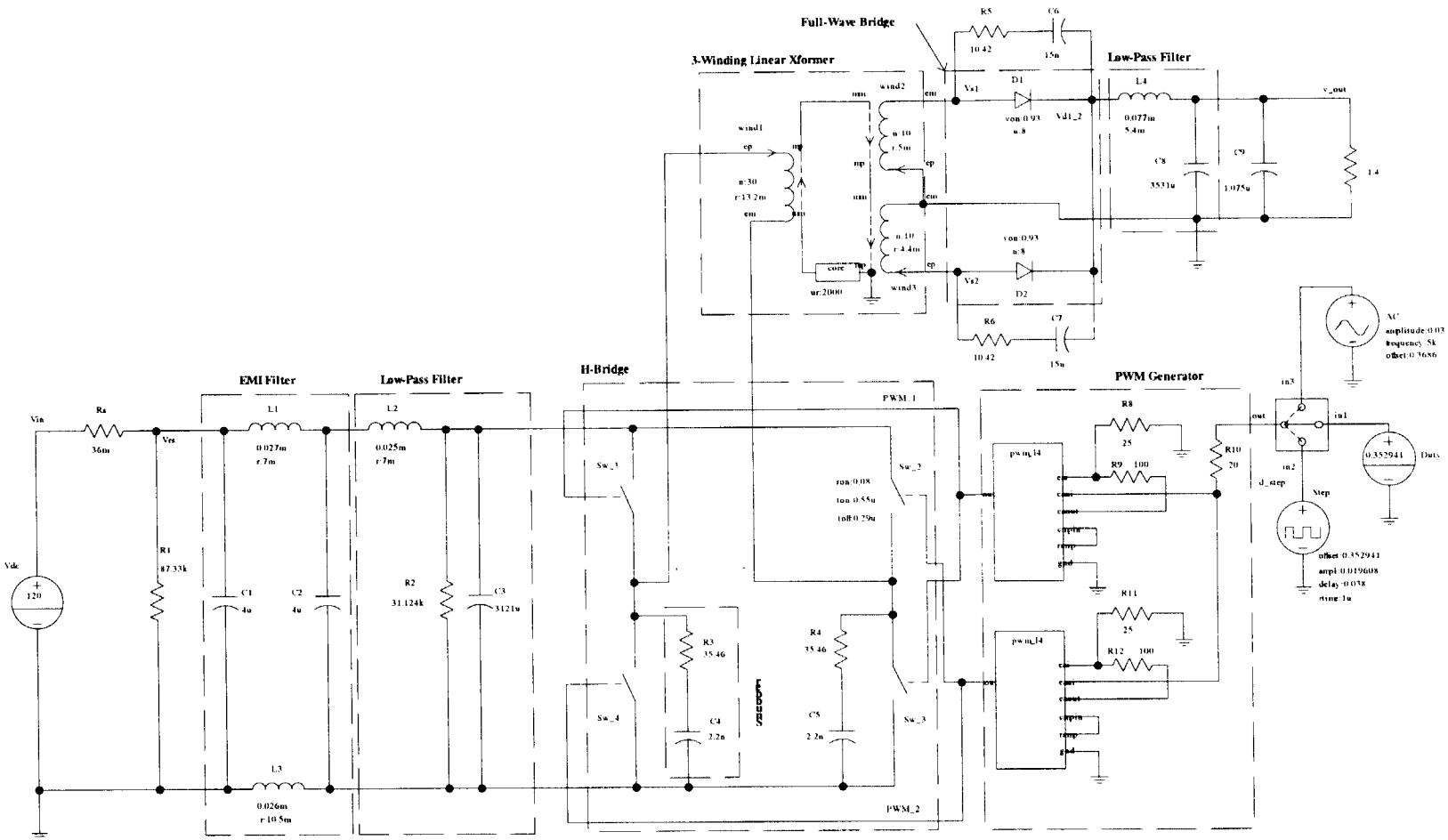


Figure 2-1: SABER® Simulation Model

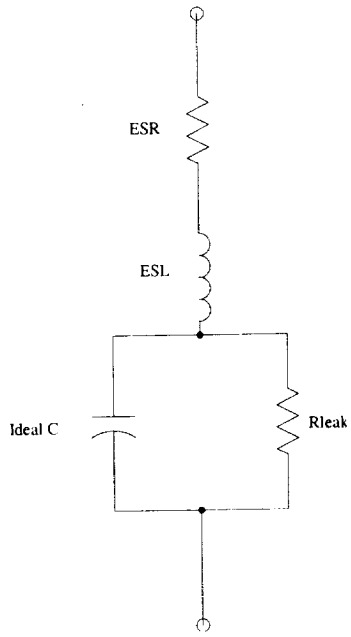


Figure 2-2: Capacitor Equivalent Circuit

equivalent series inductance (ESL), and the leakage resistance, R_{leak}. For C1 and C2, these parasitic parameters were not available and therefore ignored. The nominal capacitance for C1 and C2 was used in the model. The capacitance of C3 was indirectly measured with a handheld FLUKE RLC multi-meter with 5% accuracy by placing a known capacitance, C, in series with C3, which was then calculated using the relationship

$$C3 = \frac{C_s C}{C - C_s} \tag{2.1}$$

where C_s is the measured series capacitance. The nominal capacitance is 3000 μF. Since both C3 and C8 are polarized Tantalum capacitor banks, an ESR value, which was estimated based on the tuned ESR value of C8, was added to the model (See Table 2-1). However, simulation shows that this parameter has no noticeable effect on the output voltage. The other parasitic parameters were not available and, as a result, ignored in the model.

Reference	Resistance(Ω)	Capacitance(F)	ESR(Ω)	% Accu/Tol
Rs	36m			
R1	87.33k			1
R2	31.124k			1
C1		4μ		2
C2		4μ		2
C3		3121μ	0.02	5

Table 2-1: Input Resistor and Capacitor Values

The resistor Rs is the Sorensen DHP Series power supply output DC impedance and supply-converter interconnecting lead resistance. It was indirectly measured by setting the power supply to 120 Volts and measuring the voltage at the input posts of the converter with an HP 34401A multi-meter for different loads (See Figure 2-3 and Appendix A). The magnitude of the slope is the resistance. The resistor R1 is an input voltage sense resistor. It is composed of three discrete resistors to lower the input voltage to a suitable level for use in the feedback loop. The resistor R2 is the bleeder resistor discussed before. The latter two resistors were directly measured with the HP 34401A multi-meter.

The parasitic elements in a real inductor are the series resistance, R_l, of the windings and the parallel capacitance, C_{leak}, as shown in Figure 2-4. Of these two parasitic elements, only the series resistance was measured and incorporated into the model. The measurement was accomplished by applying a variable DC voltage to the winding in series with a known resistance of 1.39 Ω and measuring the voltage across the winding and recording the current delivered by the power supply (See Appendix A for the raw data). The DC voltage was supplied by an HP E3631A and the voltage across the inductor was measured with an HP 34401A. Based on this approach, the series resistances of L1, L2, and

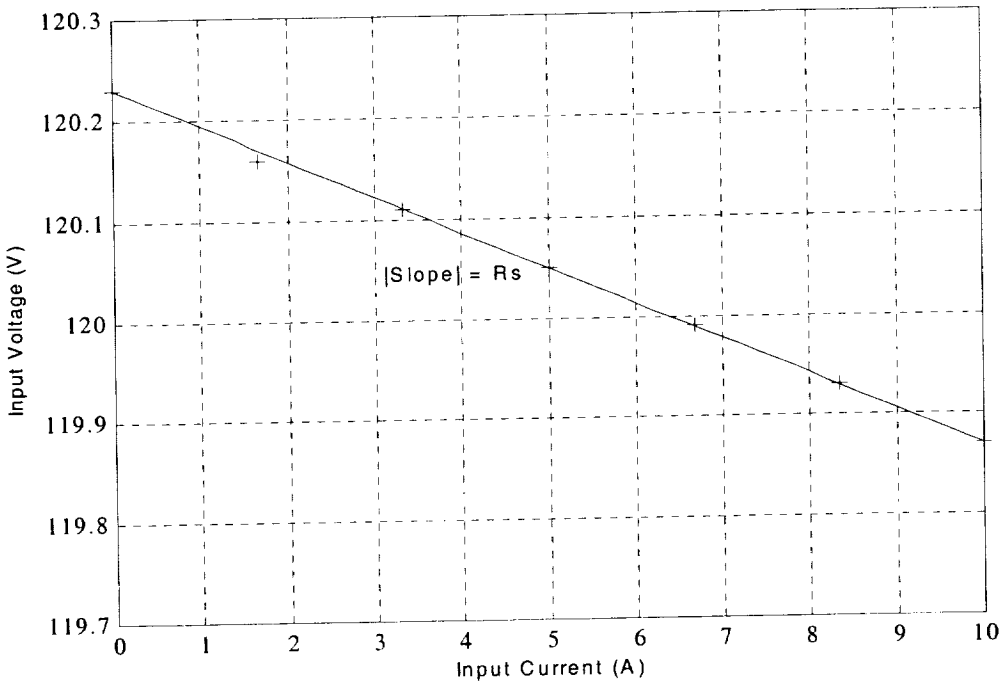


Figure 2-3: Plot of Voltage and Current Input Data and y = -0.036*I + 120.23

L3 were calculated at 7.16 mΩ, 6.52 mΩ, and 10.49 mΩ, respectively. In the simulation model shown in Figure 2-1, these values were set to 7 mΩ, 7 mΩ, and 10.5 mΩ, respectively.

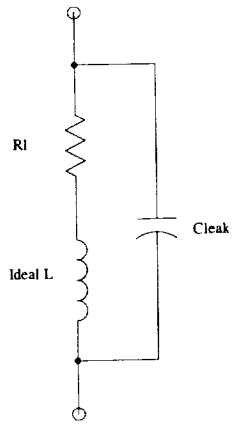


Figure 2-4: Inductor Equivalent Circuit

Due to the fact that the filter inductors will be subjected to a range of DC current bias in the operational range of the converter, it is necessary to make inductance measurements at different bias currents within this range. For an operational range of 0 – 40 Amps, the input filter inductors will be subjected to a maximum of 40/3 Amps average. The factor of 1/3 is due to the 3:1 transformer ratio. Additionally, it is well known that the inductance is also frequency dependent. (For further discussion on this topic please refer to References [16], [17], [18], and [19]) However, since this model is for a Full-Bridge converter operating at 20 kHz, the measurement of interest was carried out at the ripple frequency of 40 kHz. Figure 2-5 shows the plot of the data, which was obtained with an HP 4284A, as well as the plot of the polynomial fit. It should be pointed out that, ideally, three times the data points per inductor would result in a more accurate fit.

The polynomial fit was carried out with the command *polyfit(I,L,n)* in MATLAB®, where I is the bias current array, L is the measured inductance array, and n is the desired order. The best fit for each inductor was a third-order polynomial that resulted in

$$\begin{aligned} L1 &= 0.0027I^3 - 0.0808I^2 + 27.3500 \\ L2 &= 0.0015I^3 - 0.0592I^2 + 25.2000 \\ L3 &= 0.0016I^3 - 0.0517I^2 + 26.4600 \end{aligned} \tag{2.2}$$

From Figure 2-5, this fit is valid for input currents up to about 19 Amps. Reflected to the output, it is valid for output currents up to 57 Amps. Since the load operational range is from 0 – 40 Amps, this will be satisfactory as long as startup transient modeling is not required. Both the SABER® model and the mathematical model in Chapter Three will simulate through the startup transient but the results will not be accurate due to the increase in inductance after about 20 Amps at the input. In reality, the inductance goes to a minimum but would never increase nor go to zero (See Appendix A for raw data).

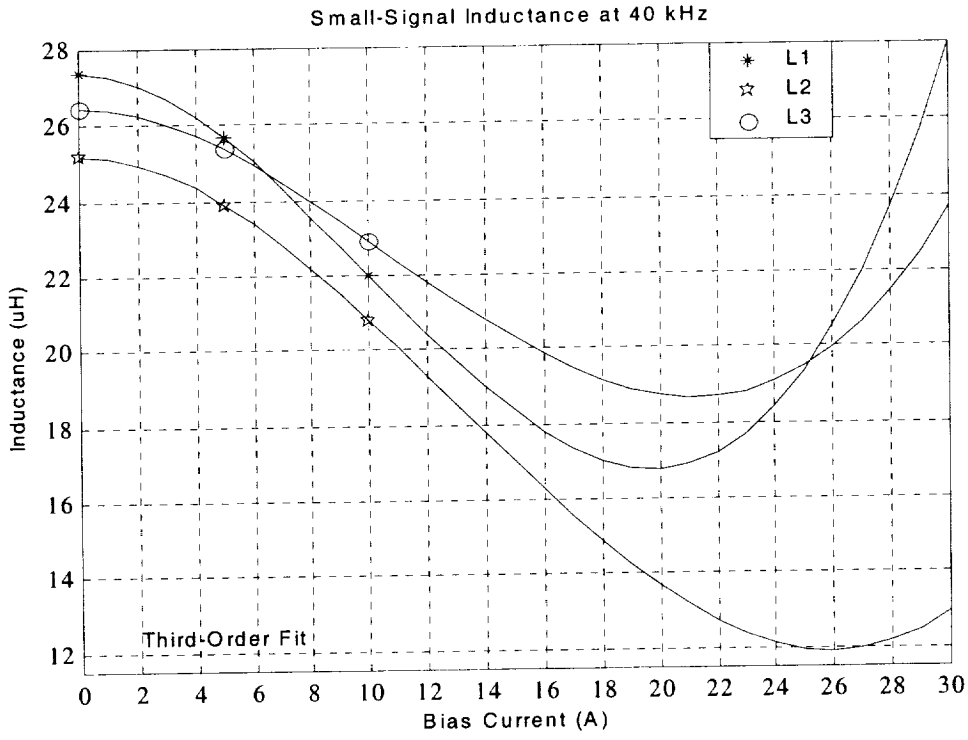


Figure 2-5: Inductance vs. Bias Current and Polynomial Fit for L1, L2, and L3

2.2.1.2 Output Filter

Most of the foregoing discussion in section 2.2.1.1 applies to the output low-pass filter, which is composed of C8 and L4. That is to say, that the capacitance of C8, with a nominal capacitance of 3531μF, and the series resistance of L4 were measured in like manner as C3 and L1 - L3, respectively, and the inductance of L4 was also obtained at 40 kHz but at bias currents of up to 20 Amps—the limit of the HP 4284A. Inductance measurements at up to 40 Amps would yield a more accurate model since L4 is subjected to the full load of 40 Amps and higher during transients. Additionally, Figure 2-1 shows a small, non-polarized capacitor, C9, in parallel with C8. This is for the purpose of aiding in removing high frequencies at the output (See Table 2-2).

Reference	Capacitance(F)	ESR(Ω)	% Accu
C8	3531µ	0.025	
C9	1.075µ		5

Table 2-2: Output Capacitor Values

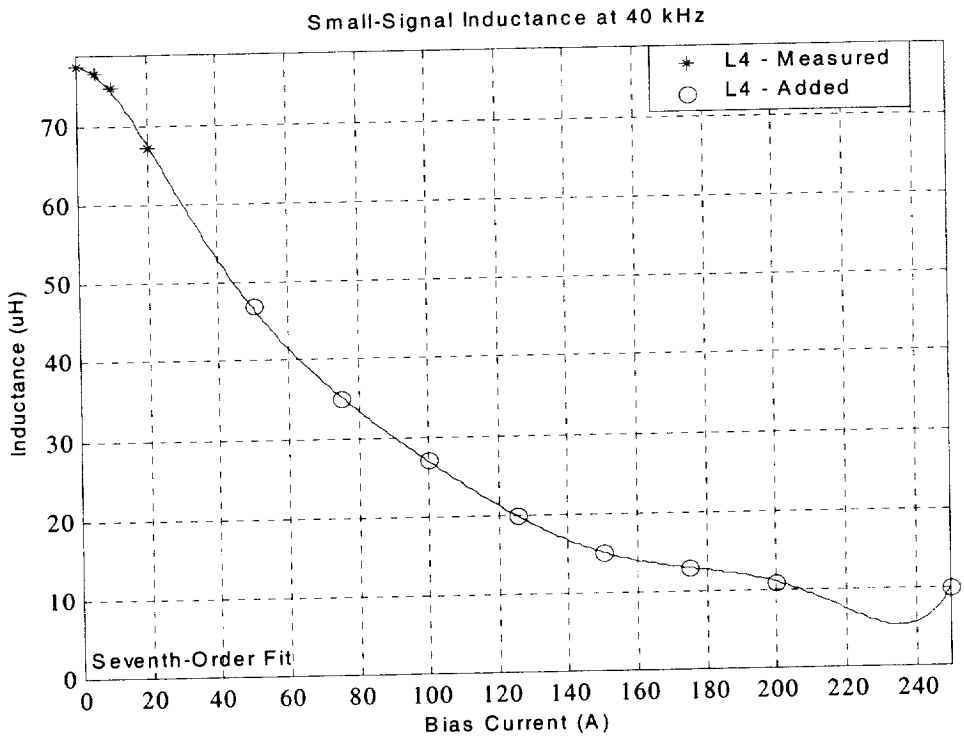


Figure 2-6: Inductance vs. Bias Current and Polynomial fit for L4

A third-order fit or higher for L4 resulted in the inductance going negative for relatively small currents. In order to force the inductance to go to a minimum, a curve was added to that after 20 Amps and then the best fit was found. The result was a seventh-order polynomial:

$$L4 = 3.3376e^{-13} * I^7 - 2.9219e^{-10} * I^6 + 1.0174e^{-7} * I^5 - 1.7885e^{-5} * I^4 +$$

$$1.6000e^{-3} * I^3 - 0.0700 * I^2 + 0.2993 * I + 77.3119 \qquad (2.3)$$

The seventh-order polynomial was necessary to force the inductance to increase instead of decrease. The inductance starts to increase around 235 Amps. The series resistance of L4 was measured at 5.33 mΩ but was set to 5.4 mΩ in the template. The ESR of C8 was obtained by comparing the simulation ripple to the converter ripple for different values of the ESR. It should be noted that the ESR plays a major role in the output voltage ripple and should be kept as small as possible.

The "c" (linear capacitor) template is one of seven templates and it models ESR and leakage resistance and provides for temperature effects and stress analysis. The "r" (nonzero linear resistor) template is one of nine templates. This template provides for temperature effects and stress analysis if desired. For the inductors, the "splp" (SPICE Polynomial Inductor) template was chosen and is one of numerous available. It allows for temperature effects but not stress analysis. For the purpose of creating a simulation model with minimum complexity, these templates should yield suitable results.

2.2.2 H-Bridge

The H-Bridge is composed of four nonlinear MOSFET switches (IR: FA57SA50LC), as shown in Figure 2-1. The first switch pair is composed of Sw_1 and Sw_3 and the second is composed of Sw_2 and Sw_4. In normal operation, the switch pairs alternate but only one switch pair is on at a time with dead time in between. The dead time is due to a duty ratio less than one but a fixed "blanking" time is also designed into the system to prevent all the four switches from being on at the same time (shoot through). The resulting waveform across each switch pair is shown in Figure 2-7 for a resistive load of

5.6Ω, which is equivalent to about 4.8 Amps average, and a pulse-width of 90/255 for each switch pair. Note that the waveform labeled Ph_1 is one of the Altera Board's output

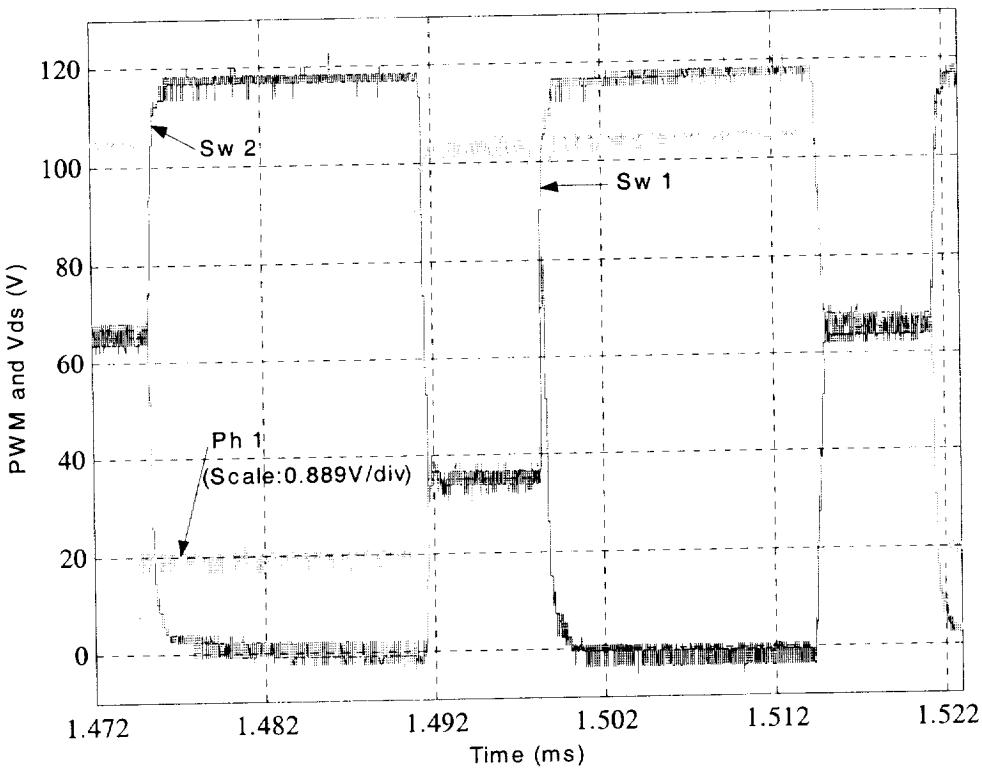


Figure 2-7: Voltage Waveform across Sw_1, Sw_2 and Altera's Phase 1 Signal

signals, that is, one of the two signals **to** the Gate Drivers (See Figure 1-2). When this signal goes low, Sw_1 and Sw_3 turn on. The waveforms across the switches that were not referenced to ground were captured with an HP 1153A Differential Probe and an HP 5063-2145 100:1 adapter connected to an HP Infinium Oscilloscope. Waveforms that were referenced to ground were captured with an HP 1160A 10:1 Probe.

In the SABER® Simulation model (Figure 2-1), the MOSFET switches were simulated with the "sw1_l4" (Digitally-Controlled Ideal Switch) template, which is compatible with the PWM Generator block discussed in section 2.3. This template models the on and off resistance, turn-on and turn-off transition, but not the turn-on and turn-off

delays. The on and off resistance were obtained from the data sheet while the turn on and turn off transition times were obtained from the converter V_{ds} waveform as in Figure 2-7 (See Table 2-3). These transition times were used since the delays introduced by the Altera Board and the Gate Drivers are inherently taken into account. The simulation model does not take into account these delays separately since it uses the PWM Generator block to drive the switch templates. The tweaked turn-on and turn-off transition times used in the simulation model are 0.55 μs and 0.29 μs, respectively. Note that the turn-on and turn-off delays are almost the same.

MOSFET Switches					
On Resistance	Off Resistance	Turn On Time	Turn Off Time	Turn On Delay	Turn Off Delay
0.08 Ω	10 MΩ	0.750 μs	0.320 μs	0.730 μs	0.600 μs

Table 2-3: MOSFET Switch Parameters

There are at least five Power MOSFET templates, in SaberSketch™, with various levels of complexity that model some or all of the components shown in Figure 2-8 but would require extra effort to setup for minimal gain. For a detailed discussion of this equivalent circuit (Figure 2-8), refer to the article "Power MOSFET Basics" in the Technical Information section at International Rectifier's Web site. Note that I_d is a function of V_{gs}.

In addition to the MOSFET Switches, the two lower switches, Sw_3 and Sw_4, each has a turn-off RC snubber to reduce the dv/dt due to stray inductance, such as the transformer leakage inductance, in order to reduce the EMI generation. Note that Sw_3 is in one switch pair and Sw_4 is in the other.

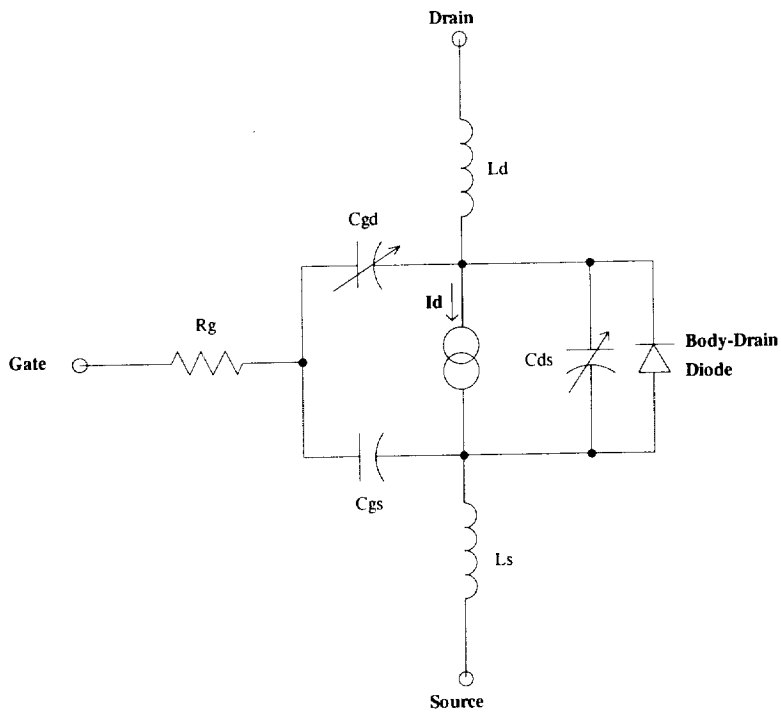


Figure 2-8: MOSFET Equivalent Circuit

2.2.3 Center-Tapped Transformer

Most switching power converters require input-to-output isolation as well as the step-up or step-down of the input voltage to meet the output specifications. The transformer performs these crucial roles and must be properly designed to reduce parasitic elements, such as leakage inductance and winding capacitance. That said, the transformer in the Full-Bridge switching power converter, was designed with a Magnetics® tape-wound cut C-core with part number MC0023-1D, where 1 stands for 1 mil lamination and D for square permalloy 80, and interleaved windings to reduce leakage inductance. The penalty of interleaving is increased inter-winding capacitance.[20] The parameter values are given in Table 2-4 below,

Step-Down Transformer										
A_c (cm^2)	k	A_e (cm^2)	l_m (cm)	B_s (G)	N_p	N_{s1}	N_{s2}	L_p (H)	L_{s1} (H)	L_{s2} (H)
5.039	0.8	4.0312	19.370	8k	30	10	10	4.340m	0.484m	0.484m

Table 2-4: Transformer Parameters

where A_c is the core cross-sectional area, k is the stacking factor, $A_e = k*A_c$ is the core effective cross-sectional area, l_m is the magnetic mean length, B_s is the saturation Flux Density in Gauss, N_p is the primary turns, N_{s1} and N_{s2} are the secondary turns, L_p is the measured primary inductance, and L_{s1} and L_{s2} are the measured secondary inductances corresponding to N_{s1} and N_{s2}, respectively. The winding inductances were measured with an HP 4192A. The series resistances were measured in like manner as for the inductors at 13.15 mΩ, 4.91 mΩ, and 4.39 mΩ for the primary winding and the two secondary windings, respectively. These values were set to 13.20 mΩ, 5.00 mΩ, and 4.40 mΩ in the simulation model. The coupling coefficient was set to 0.98 for each winding template.

The voltage waveform across the primary windings due to the switching function of the H-Bridge is shown in Figure 2-9. It is an AC square pulse with an amplitude of V_{dc}, ignoring losses, resulting in bi-directional core excitation; that is, the Flux Density swing is in all four quadrants of the B-H curve. The solid B-H curve is traced by the core due to the volt-seconds of Figure 2-9 (See Figure 2-10). Some converter topologies, like the Flyback, subject the core to unidirectional excitation, where only the first quadrant is utilized.

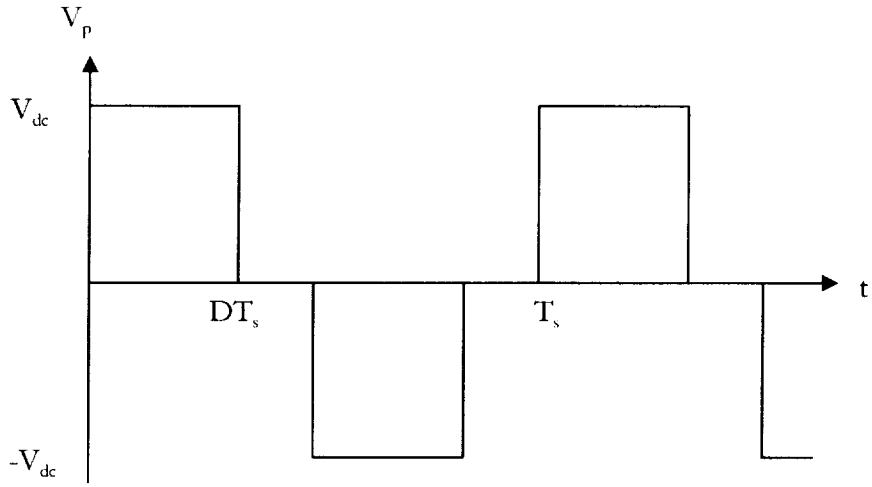


Figure 2-9: Voltage Waveform across Primary Winding

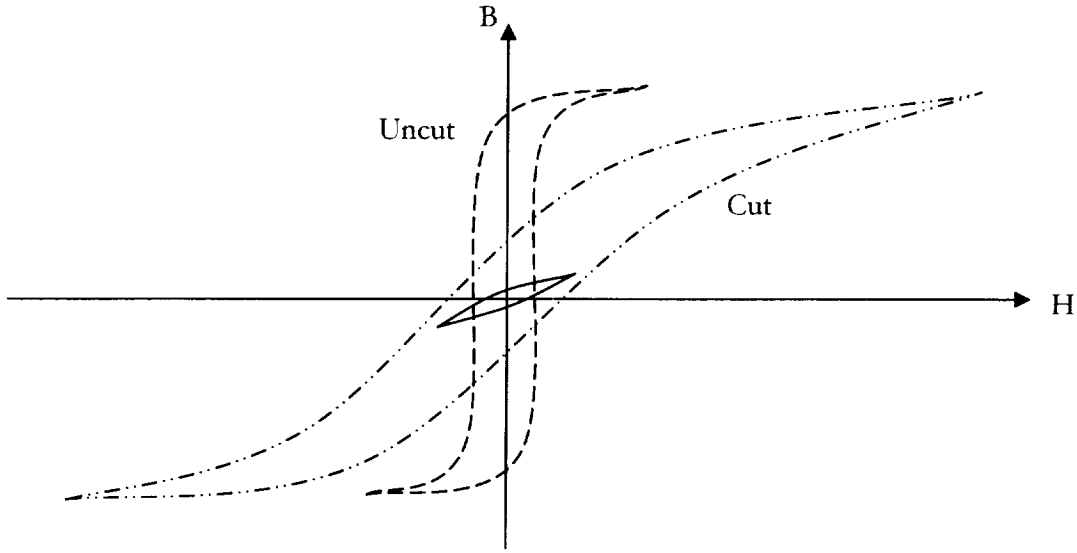


Figure 2-10: B-H Curve of Uncut and Cut Core

The Flux Density swing can be calculated by applying Faraday's Law,

$$V_{dc} = N_p \frac{d\Phi}{dt} \tag{2.4}$$

where

$$\frac{d\Phi}{dt} = \frac{dB}{dt} A_e \tag{2.5}$$

for a constant cross-sectional area. Substituting (2.5) into (2.4) and setting the integral as

$$\int_0^{\frac{D}{f_s}} V_{dc}\,dt = N_p A_e \int_{-B_{max}}^{+B_{max}} dB \tag{2.6}$$

results in

$$V_{dc}\frac{D}{f_s} = N_p A_e 2B_{max} \tag{2.7}$$

The variable of interest is B_{max}, therefore

$$B_{max} = \frac{V_{dc} D}{2N_p A_e f_s} 10^8 \tag{2.8}$$

where f_s is the switching frequency—21.78 kHz. The factor of 10^8 enables B_{max} to be expressed in Gauss and A_c in cm^2; this is the CGS system of units. Equation (2.8) yields 797 Gauss for B_{max} with D = 0.35 and V_{dc} = 120 V.

The value for B_{max} was then used to determine the permeability of the core from the data sheet provided by Magnetics® for this material—square permalloy 80. This permeability, however, is for an uncut core, which is much higher than for a cut core. This can be inferred from Figure 2-10 since the permeability, μ, is B/H, for a linear approximation, and H is the magnetizing force expressed as

$$H = \frac{N_p I_{pk}}{l_m} \tag{2.9}$$

where I_{pk} is the peak magnetizing current and l_m is the mean magnetic length. The permeability given by the chart for 800 Gauss is 22000 Gauss/Oersted.

The permeability of the cut core can be obtained in two ways. The first is by reducing the permeability value from 22000 Gauss/Oe in the core template in Figure 2-1 until the simulation primary winding current is close to the actual transformer primary

winding current (See Figure 4-30). This approach resulted in 2000 Gauss/Oe for the permeability. The second method is to use the accurate inductance measurement of the primary winding in the following linear equation

$$L_p = \frac{0.4\pi N_p^2 A_e}{l_m 10^8}\mu \tag{2.10}$$

where A_e and l_m are in centimeter-squared and centimeters, respectively, and μ is in Gauss/Oe. With a measured primary inductance of 4.340 mH, the permeability was calculated at 1844 Gauss/Oe. This agrees very well with the tweaked value in the first case.

There are two 3-winding transformer templates in SaberSketch™, one is linear and the other is nonlinear. However, since the maximum Flux Density for the given input is only about 800 Gauss while the saturation Flux Density is 8000 Gauss, the decision was made to use a linear model. Furthermore, since the linear 3-winding transformer template produced unsatisfactory results, a linear model was created using the "wind" (Winding) template and the "core" (Linear Magnetic Core) template as shown in Figure 2-1.

2.2.4 Full-Wave Rectifier

Due to the center-tapped Transformer, the full-wave rectifier is composed of only two diodes (Motorola: MURP20040CT), instead of four (See Figure 2-1). This setup is preferred in low output voltage power converters since the tradeoff of using a four-diode rectifier design as opposed to a two-diode rectifier with a less efficient center-tapped transformer favors the latter design, provided the center-tapped transformer is appropriately designed.

One of the simplest ways to model a diode is to use a piece-wise linear model; that is, when the diode is on, it is modeled by a small on-resistance in series with a voltage drop

and, when the diode is off, it is modeled by a large off-resistance. Real diodes, however, do not exhibit such a sharp transition. Figure 2-11 shows the linear-linear plot of the Instantaneous Forward Current versus the Instantaneous Forward Voltage taken from the data sheet for a junction temperature of 25 °C. This data was used since the diodes in the converter ran relatively cool due to the large heat sink (base plate) and at this temperature the diode's voltage drop is higher. From this plot, the on conductance (657 A/V) and the voltage drop (0.93 V) are obtained. The off conductance (0.8 nA/V) was obtained in a similar manner from the Reverse Current versus the Reverse Voltage graph at 25 °C in the data sheet.

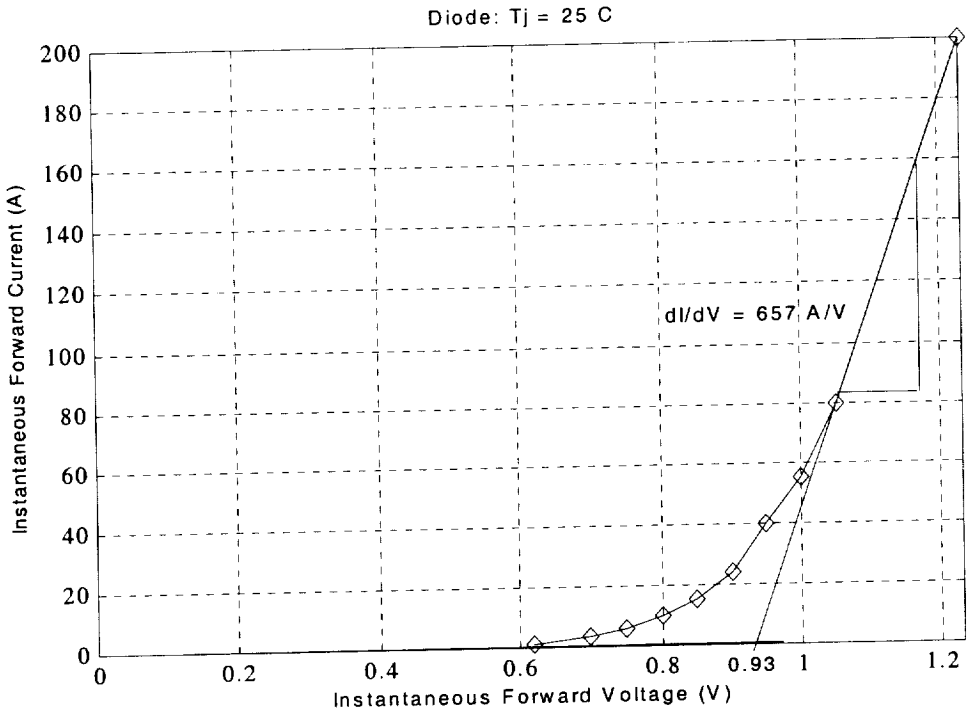


Figure 2-11: I-V Curve of the Diode

The "pwld" (Ideal Diode) template provides for the smooth transition in the transition stiffness factor, n. The value of n = 8 was obtained through simulation by varying n until the I-V data points from simulation matched the I-V data points from the data sheet.

Figure 2-12 shows the simulation schematic and the I-V data points can be found in Appendix A. Of course, other features such as reverse recovery and forward recovery can be simulated with the "dp1" or 'dp1a" template. The "dp1" template models reverse recovery only while "dp1a" models both features. Both of these templates are data-driven, meaning that required parameters could be obtained from data sheets. The data sheet for the particular diode in the Full-Bridge power converter provides information for reverse recovery except one parameter, $R_{snap\text{-}off}$, must be tuned empirically. The data sheet does not provide information for forward recovery, however. Simulation results with the reverse recovery template and an estimated $R_{snap\text{-}off}$ yielded no improvement when compared to the "pwld" template mentioned above.

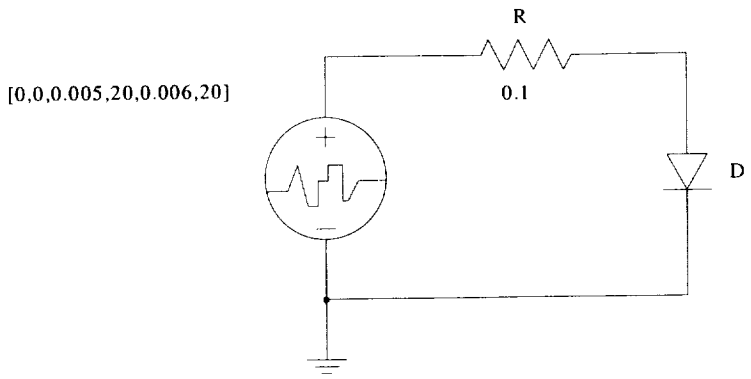


Figure 2-12: Transition Stiffness Factor Tuning

2.3 PWM Generator

In the simulation model, the switches are turned on and off with two "pwm_l4" (Simple Pulsewidth Modulator) templates that output *logic* signals compatible with the switch templates. Furthermore, the delays introduced by the ALTERA® CPLD, the opto-couplers and the high and low side drivers were all aggregated in the switch templates as

mentioned before in section 2.2.2 to simplify the modeling process and since the "pwm_l4" templates do not allow for rise and fall times. This template also enables simulation of duty ratio step-change, which is necessary for model verification as discussed in Chapter IV. Pulse-Width Modulation can be achieved by comparing a control voltage to a sawtooth or triangular waveform with or without a DC offset (See Figure 2-13a). Since the "pwm_l4" template uses a ramp oscillator (sawtooth waveform), the following discussion will center on this waveform. When the control voltage, v_c, is higher than the sawtooth voltage, v_{st}, the output of the comparator is high and, when v_c is lower, the output is low as shown in Figure 2-13b. The relationship among v_c, v_{st}, and the duty ratio (t_{on}/T_s) can be obtained by solving two simple algebraic equations:

$$y_1 = \frac{V_{st}}{T_s} t \qquad (2.11)$$
$$y_2 = v_c$$

where V_{st} is the peak value. Equating y_1 to y_2 and solving for D at $t = t_{on}$ yields,

$$D = \frac{t_{on}}{T_s} = \frac{v_c}{V_{st}} \qquad (2.12)$$

Substituting (2.12) into (1.13) and solving for v_c results in

$$v_c = \frac{3V_{st}}{2V_{dc}} V_{out} \qquad (2.13)$$

for $N_1/N_2 = 3$.

The control voltage is calculated to be 1.75V for a peak sawtooth voltage of 5V, an input voltage of 120V and a desired output voltage of 28V, which corresponds to a duty ratio of 0.35 per switch pair. The input to the "pwm_l4" template is a voltage; therefore, for

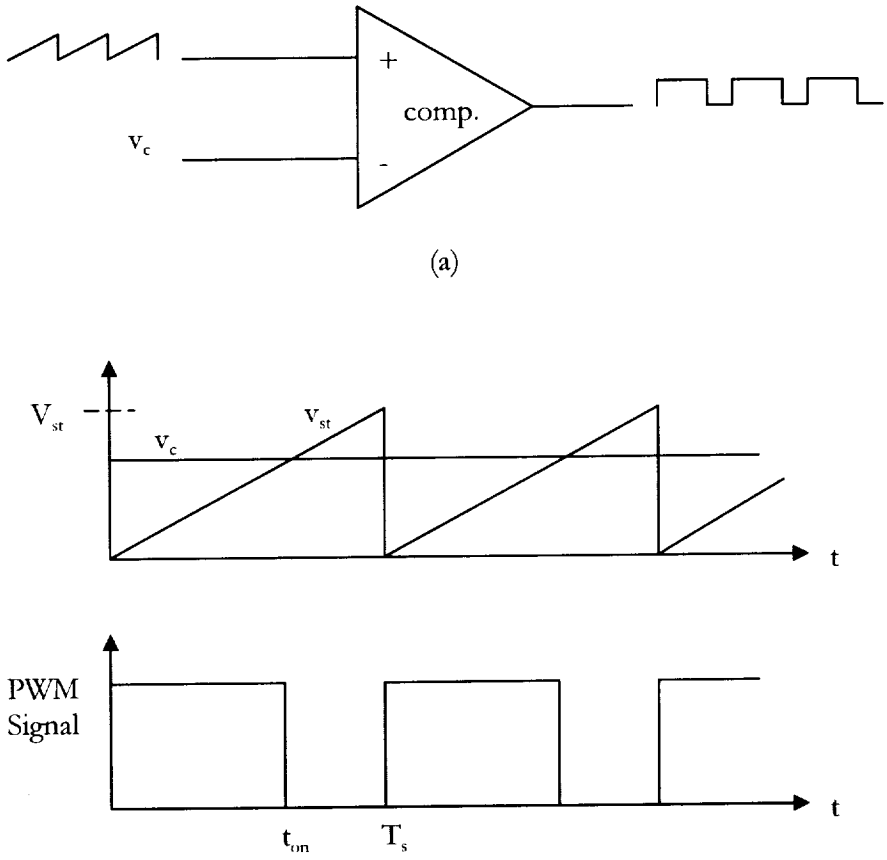


Figure 2-13: Pulse-Width Modulator: (a) Block Diagram, (b) Comparator Signals

this voltage to have a one-to-one relationship with the actual duty ratio, a gain of 1.75/0.35 = 5 is required. Since the "pwm_l4" template has a built-in error amplifier, the output of which is the control voltage, this task is easily accomplished and the gain is set with resistors R8 and R9 in Figure 2-1. The gain of a non-inverting amplifier is given as

$$Gain = 1 + \frac{R9}{R8} \tag{2.14}$$

For a gain of 5, if R9 is 100Ω, then R8 must be 25Ω. The resistor R10 is called the bias current compensation resistor and is given by

$$R10 = \frac{R9R8}{R9 + R8} \tag{2.15}$$

The value of this resistor is 20Ω for R8 and R9 given above.[21]

Some of the required template parameters are shown in Table 2-5. The frequency value was set at 21.7 kHz as measured and the maximum duty ratio had to be less than 1. The value of 0.9999 was chosen to improve the correlation between the input duty ratio voltage and the actual output duty ratio of the template. The parameters "vclow" and "vchigh" set the minimum and maximum value of the sawtooth waveform and the open-loop gain of the error amplifier is obtained from "gain_ae" as follows

$$A_{ol} = 10^{\frac{gain\_ae}{20}} \tag{2.16}$$

Note that PWM_2 is delayed by half a period as required. Other parameters not included in the table are the unity-gain frequency of the error amplifier (bandwidth), the slew rate of the error amplifier, and the DC offset of the sawtooth waveform. These values were set at 5.5 MHz, 12 MV/s, and 0V, respectively.

"pwm_l4" Template							
	freq (Hz)	dutymax	vclow (V)	vchigh (V)	gain_ae (dB)	vcc (V)	td (s)
PWM_1	21.7k	0.9999	0	5	95	10	0
PWM_2	21.7k	0.9999	0	5	95	10	23.041u

Table 2-5: Pulse-Width Modulator Parameters

CHAPTER III

MATHEMATICAL MODEL

3.1 Introduction

An alternative to the component-level model (SABER® model) is the switched state-space model. Because it is a switched model, it is nonlinear and, as such, is valid over a wide range like the SABER® model. However, it is less accurate than the SABER® model due to the linearization of each switched topology and less flexible since it would be necessary to derive the state-space representation anew if the converter topology changes. On the other hand, the model lends itself to advanced mathematical analysis.

This chapter covers the linear equivalent circuit model of the transformer and the derivation of the state-space representation of each of the three, switched topologies of the Full-Bridge switching power converter and the resultant switched state-space model. The third section briefly covers the linear transfer function model derived for comparison purposes. This model is very restricted and is possible only if the converter has been designed and built. An alternative linear transfer function model is derived in the last section through the state-space averaging technique. This averaging method is applied to the state-space representations derived above.

3.2 Switched State-Space Model

The switched state-space model is composed of the state-space representation for each of the three switched converter topologies. It is a switched model due to the fact that no averaging is incorporated. In this case, each topology is reduced to a linear circuit. As a result, some components, such as the transformer, must be replaced with their linear equivalent circuit model.

3.2.1 Linear Transformer

In the SABER® simulation model, a linear transformer template was used. For the mathematical model, however, the transformer is replaced with its linear equivalent circuit model.[15],[16],[20],[22] Figure 3-1 shows an equivalent circuit model for the center-tapped transformer. The physical winding layout constitutes of two 30-turn parallel primary windings, one on each leg of the cut C-core. One 10-turn secondary winding is interleaved with one primary winding on one leg and the other secondary winding with the second primary winding on the second leg. The return leads of each secondary winding are then brought to a common point—the center-tap.

The parallel capacitors, C_p, C_{s1}, and C_{s2}, model the inter-winding capacitance of each of the three windings as a lumped value but in reality the capacitance is distributed. Similarly, the intra-winding capacitances between the parallel primary windings and the secondary windings are modeled by C_{ps1} and C_{ps2} and are higher for interleaved windings. Unfortunately, these values were not available and, since these values are small and the main purpose of the model is for controller design, these model components can be ignored.

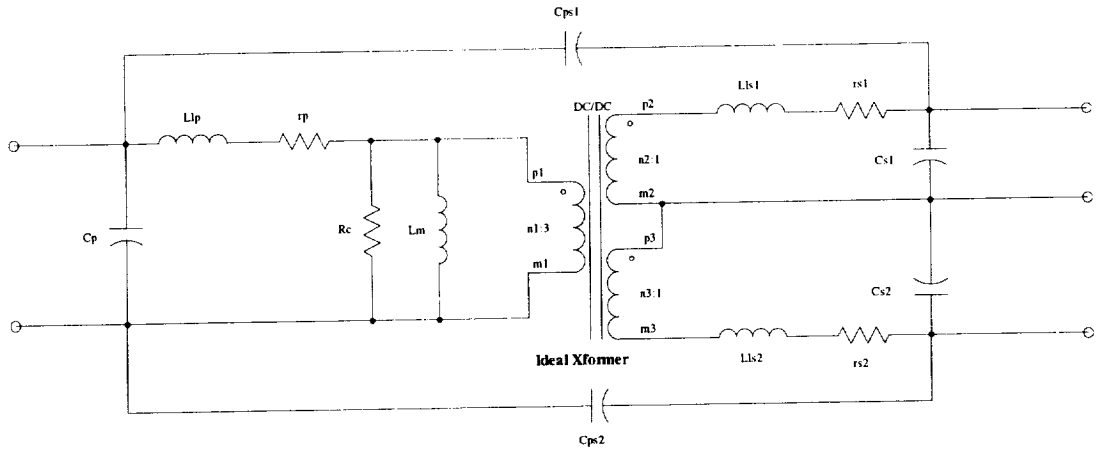


Figure 3-1: Linear Transformer Equivalent Circuit

The winding resistances are given in chapter two, page 26, and are modeled by resistors r_p, r_{s1}, and r_{s2}. The parallel resistor R_c models the hysteresis and eddy current loss, which considered together is termed the core loss. The area enclosed by either B-H curve in Figure 2-10 represents hysteresis loss and is the result of the constant alignment and realignment of the particles in the core. The eddy current loss is due to the current loops in the core, called eddy currents, which are induced by the changing flux. The value of R_c was calculated using the Core Loss (Watt/Pound) versus the Flux Density (Gauss) graph at 20 kHz. Since the Flux Density swing is about 800 Gauss with D = 0.35, the core loss is around 1.9 Watt/Pound. The weight of the core is 1.59 pounds; therefore, the core loss is 3.021 Watts. The primary winding voltage is shown in Figure 2-9 and the root-mean-square value is given by

$$V_{rms} = \sqrt{\frac{2}{T_s}\int_0^{DT_s} V_{dc}^2 dt} \tag{3.1}$$

which calculates to 100.4 V_{rms} for D = 0.35 and V_{dc} = 120 V. Therefore, $R_c = 100.4^2/3.021 \approx 3.3\ k\Omega$.

The inductors L_{lp}, L_{ls1}, and L_{ls2} model the leakage inductance of each of the three windings. These inductances exist due to the fact that not all the flux links all the turns of all the windings. Finally, the current that flows in the primary when no load is attached on the secondary is called the magnetizing current. This current is low for transformer designs that have small air gaps. In the equivalent circuit, the magnetizing current is modeled with the parallel inductor L_m—the magnetizing inductance. It is important to note that a nonlinear model can be obtained by replacing the fixed magnetizing inductance with a nonlinear inductor.

In the SABER® model, a coupling coefficient of 0.98 was used. The coupling coefficient can be calculated from (3.2) if the mutual inductance is known, but attempts to indirectly measure the magnetizing inductance failed due to the lack of accuracy of the inductance meter. Assuming that the value of 0.98 is close to the true value, the mutual inductance is given as [15]

$$M = k_c \sqrt{L_p L_{s1}} \tag{3.2}$$

where L_p and L_{s1} are the measured self-inductances (See Table 2.4) and k_c is the coupling coefficient. Equation (3.2) yields a value of M = 1.42 mH and is the same for the other secondary, L_{s2}, since the self-inductance is the same as L_{s1}. The mutual inductance can then be used to calculate the magnetizing inductance and the leakage inductance of each winding.

The magnetizing inductance is simply the product of the mutual inductance and the turns ratio. Since the turns ratio is 3, the magnetizing inductance, L_m, is 4.26 mH. The equation for the leakage inductance of the primary is

$$L_{lp} = L_p - nM \tag{3.3}$$

The primary winding leakage inductance is calculated to be 80 μH. Similarly, the equation for the leakage inductance of either secondary is

$$L_{lsi} = L_{si} - \frac{M}{n} \tag{3.4}$$

where i = 1 and 2. The leakage inductance of either secondary is 10.67 μH. The leakage inductances were also ignored in the final model due to long simulation run time, that is, very small Δt is required to simulate the model to avoid numerical instability.

3.2.2 Topological States

Due to the H-Bridge, the Full-Bridge power converter loops through three topologies. The first topology exists when the switch pair, Sw_1 and Sw_3, is on; the second topology exists when all the switches (MOSFETs) are off, and the third topology exists when the second switch pair, Sw_2 and Sw_4, is on. As stated before, this MATLAB®-implemented model is valid for both CCM and DCM. However, if a state-space average model is desired that is valid for the DCM, then the fourth topological state that exists when all the switches are off and the current of the inductor L4 goes to zero for a finite time must be specifically considered. This mode of operation is present in the converter when the load is less than one Amp, which is less than 2.5% of the load operational range.

3.2.2.1 First Topology: Sw_1 and Sw_3 ON

In addition to ignoring the winding capacitances and the leakage inductances as mentioned above, the resistor R1, the diode-off branch, capacitor C9, capacitors C1 and C2, and the snubbers on Sw_3 and Sw_4 were neglected. The currents that would flow

through R1 and the diode when it is reversed biased are negligible when compared to the load current and reflected load current. Furthermore, simulation with SABER® indicates that the overall behavior of the model is not significantly affected without C9, C1 and C2 and the snubbers. For example, step changes in the input voltage can still be modeled accurately without C1 and C2. If included in the model, however, the mathematical complexity and simulation run time are significantly increased. Obviously, the more components that are ignored, the less accurate the model is but the goal has always been to obtain a balance between accuracy and complexity.

Resistive Load

The first reduced topology is shown in Figure 3-2 for a resistive load. This topology is achieved by turning the first switch pair, Sw_1 and Sw_2, on while the second switch pair, Sw_2 and Sw_4, is off. The voltage across the primary winding is $+V_{dc}$, causing the diode D_1 to be forward biased and diode D_2 to be reverse biased (See Figure 2-1) and ignored.

The state variables chosen are inductor currents (x_1, x_3, x_4) and capacitor voltages (x_2, x_5). State variables are variables that are associated with energy; other examples of state variables are position, velocity, and acceleration. The goal is to obtain the state-space representation of this topology, which is a set of n, in this case five, simultaneous, first-order differential equations and the algebraic output equations. Mathematically, this is expressed as

$$\begin{aligned} \dot{x} &= A_i x + B_i u + W_i V_d \\ y &= Cx + Du \end{aligned} \tag{3.5}$$

where i = 1, 2, and 3 for each of the three topologies and the matrix D is equal to zero in this particular study. The matrix C does not have a subscript i since the output equations are

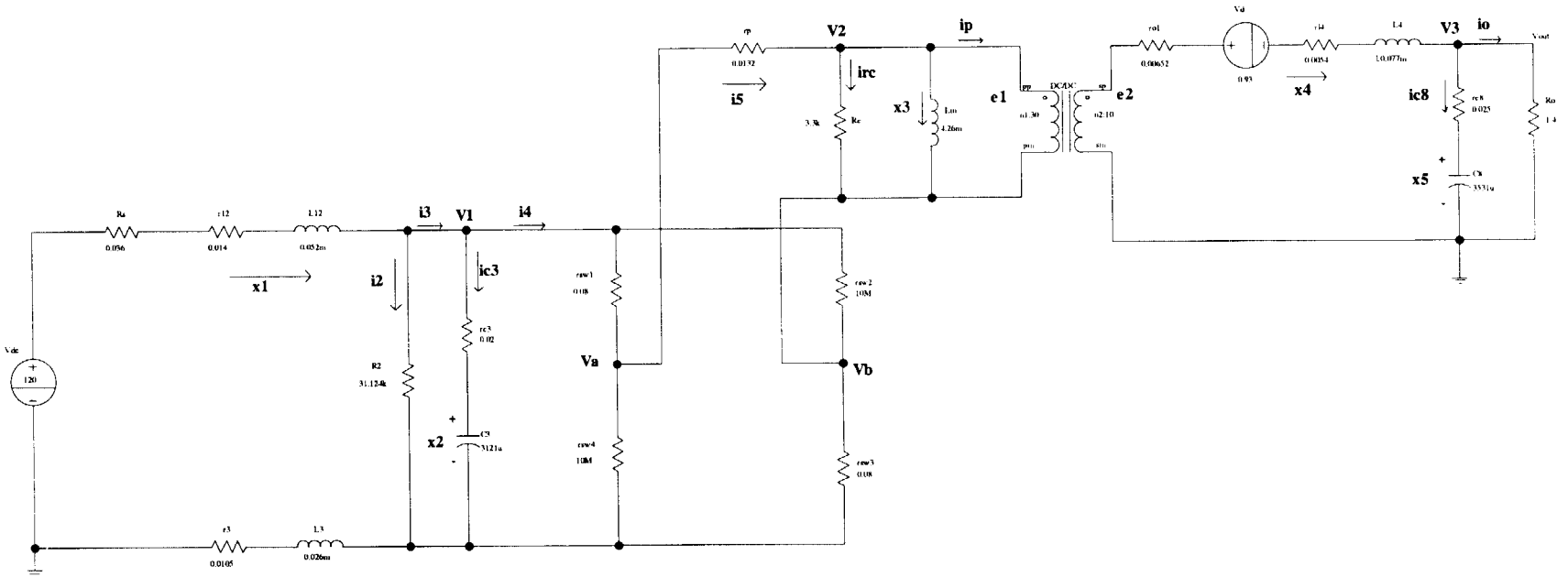


Figure 3-2: The First Switched Topology

the same for each topology. The state-space representation is derived through Kirchoff's Voltage Law (KVL), Kirchoff's Current Law (KCL), and the well-known current-voltage relationship of a resistor, an inductor and a capacitor.

In order to solve for the first two differential equations it is necessary to solve for i_5 first, since $i_4 \approx i_5$, from the following algebraic equations

$$i_5 = \frac{V_{ab} - V_2}{r_p} \tag{3.6a}$$

$$i_5 = \frac{V_2}{R_c} + x_3 + i_p \tag{3.6b}$$

$$i_p = \frac{x_4}{n} \tag{3.6c}$$

Substituting for i_p, equating (3.6a) and (3.6b) and solving for V_2 yields

$$V_2 = -\tau x_3 - \frac{\tau}{n} x_4 + \frac{\tau}{r_p} V_{ab} \tag{3.7a}$$

where

$$\tau = \frac{r_p R_c}{r_p + R_c} \tag{3.7b}$$

Furthermore,

$$V_1 = r_{c3} C_3 \dot{x}_2 + x_2 \tag{3.8}$$

and

$$V_{ab} = V_a - V_b = R_b V_1 = r_{c3} R_b C_3 \dot{x}_2 + R_b x_2 \tag{3.9a}$$

where

$$R_b = \left[\frac{r_{sw4}}{r_{sw1} + r_{sw4}} - \frac{r_{sw3}}{r_{sw2} + r_{sw3}}\right] \tag{3.9b}$$

Substituting for V_2 with (3.7a) and V_{ab} with (3.9a) in (3.6a) gives

$$i_4 \approx i_5 = \frac{\tau}{r_p} x_3 + \frac{\tau}{n r_p} x_4 + r_{c3} \xi R_b C_3 \dot{x}_2 + \xi R_b x_2 \tag{3.10a}$$

where

$$\xi = [\frac{1}{r_p} - \frac{\tau}{r_p^2}] \tag{3.10b}$$

From Figure 3-2, the state variable x_1 is given as

$$x_1 = \frac{V_1}{R_2} + C_3 \dot{x}_2 + i_4 \tag{3.11a}$$

Making use of (3.8) and (3.10a) and solving for x_2-dot results in

$$\dot{x}_2 = \frac{1}{\phi 2} x_1 - \frac{\phi_3}{\phi_2} x_2 - \frac{\tau}{r_p \phi_2} x_3 - \frac{\tau}{n r_p \phi_2} x_4 \tag{3.11b}$$

and

$$\begin{aligned} \phi_1 &= C_3[\frac{r_{c3}}{R_2} + 1] \\ \phi_2 &= \phi_1 + r_{c3} \xi R_b C_3 \\ \phi_3 &= \frac{1}{R_2} + \xi R_b \end{aligned} \tag{3.11c}$$

The expression for x_1-dot is obtained from

$$V_{dc} = (R_s + r_{12} + r_3) x_1 + (L_{12} + L_3) \dot{x}_1 + V_1 \tag{3.12a}$$

Substituting for V_1 with (3.8) and for x_2-dot with (3.11b) and solving for x_1-dot yields

$$\dot{x}_1 = -\frac{\psi_1}{L_{12} + L_3} x_1 - \frac{\psi_2}{L_{12} + L_3} x_2 + \frac{\psi_3}{L_{12} + L_3} x_3 + \frac{\psi_4}{L_{12} + L_3} x_4 + \frac{1}{L_{12} + L_3} V_{dc} \tag{3.12b}$$

and

$$\psi_1 = R_s + r_{12} + r_3 + \frac{r_{c3}C_3}{\phi_2}$$

$$\psi_2 = 1 - \frac{r_{c3}C_3\phi_3}{\phi_2} \qquad (3.12c)$$

$$\psi_3 = \frac{r_{c3}C_3\tau}{r_p\phi_2}$$

$$\psi_4 = \frac{\psi_3}{n}$$

The third state variable is the magnetizing current, x_3, and this is easily obtained from

$$L_m\dot{x}_3 = V_2 \qquad (3.13a)$$

The expression for V_2 is given by (3.7a) and (3.9a) and that of x_2-dot by (3.11b), therefore,

$$\dot{x}_3 = \frac{\Delta_1}{\phi_2}x_1 + [\Delta_2 - \frac{\Delta_1\phi_3}{\phi_2}]x_2 - [\frac{\Delta_1\tau}{r_p\phi_2} + \frac{\tau}{L_m}]x_3 - [\frac{\Delta_1\tau}{nr_p\phi_2} + \frac{\tau}{nL_m}]x_4 \qquad (3.13b)$$

and

$$\Delta_1 = \frac{r_{c3}R_bC_3\tau}{r_pL_m}$$

$$\Delta_2 = \frac{R_b\tau}{r_pL_m} \qquad (3.13c)$$

On the output side, an expression for each of the two remaining state variables, x_4 and x_5, must be found. The first to be determined is x_5. This is accomplished with the following equations:

$$V_3 = r_{c8}C_8\dot{x}_5 + x_5 \qquad (3.14a)$$

and also

$$V_3 = R_ox_4 - R_oC_8\dot{x}_5 \qquad (3.14b)$$

Equating (3.14a) and (3.14b) and solving for x_5-dot yields

$$\dot{x}_5 = \frac{R_o}{C_8(r_{c8} + R_o)}x_4 - \frac{1}{C_8(r_{c8} + R_o)}x_5 \qquad (3.14c)$$

Hence, the output voltage is

$$V_3 = Ex_4 + Fx_5 \tag{3.15a}$$

where

$$\begin{aligned} E &= [R_o - \frac{R_o^2}{r_{c8} + R_o}] \\ F &= \frac{R_o}{r_{c8} + R_o} \end{aligned} \tag{3.15b}$$

Before solving for x_4-dot, an expression for e_2 in terms of the state variables must first be determined. Now,

$$e_2 = \frac{e_1}{n} = \frac{V_2}{n} \tag{3.16a}$$

Again, V_2 and x_2-dot are substituted for, resulting in

$$e_2 = \frac{k_1}{\phi_2} x_1 + [k_2 - \frac{k_1\phi_3}{\phi_2}]x_2 - [\frac{k_1\tau}{r_p\phi_2} + \frac{\tau}{n}]x_3 - [\frac{k_1\tau}{nr_p\phi_2} + \frac{\tau}{n^2}]x_4 \tag{3.16b}$$

and

$$\begin{aligned} k_1 &= \frac{r_{c3}R_bC_3\tau}{nr_p} \\ k_2 &= \frac{R_b\tau}{nr_p} \end{aligned} \tag{3.16c}$$

An expression for x_4-dot is then obtained from

$$L_4\dot{x}_4 = -(r_{01} + r_{L4})x_4 + e_2 - V_3 - V_d \tag{3.17a}$$

Substituting for e_2 with (3.16b) and V_3 with (3.15a) gives

$$\dot{x}_4 = \frac{k_1}{L_4\phi_2} x_1 + \frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}]x_2 - \frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} + \frac{\tau}{n}]x_3 - \frac{1}{L_4}[b + \frac{k_1\tau}{nr_p\phi_2}]x_4 - \frac{F}{L_4}x_5 - \frac{1}{L_4}V_d \tag{3.17b}$$

and

$$b = r_{01} + r_{L4} + \frac{\tau}{n^2} + E \tag{3.17c}$$

This set of five coupled, first-order differential equations are then expressed in matrix form as in (3.5), resulting in the following matrices:

$$A_1 = \begin{bmatrix} -\frac{\psi_1}{L_{12}+L_3} & -\frac{\psi_2}{L_{12}+L_3} & \frac{\psi_3}{L_{12}+L_3} & \frac{\psi_4}{L_{12}+L_3} & 0 \\ \frac{1}{\phi_2} & -\frac{\phi_3}{\phi_2} & -\frac{\tau}{r_p\phi_2} & -\frac{\tau}{nr_p\phi_2} & 0 \\ \frac{\Delta_1}{\phi_2} & [\Delta_2 - \frac{\Delta_1\phi_3}{\phi_2}] & -[\frac{\Delta_1\tau}{r_p\phi_2} + \frac{\tau}{L_m}] & -[\frac{\Delta_1\tau}{nr_p\phi_2} + \frac{\tau}{nL_m}] & 0 \\ \frac{k_1}{L_4\phi_2} & \frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}] & -\frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} + \frac{\tau}{n}] & -\frac{1}{L_4}[b + \frac{k_1\tau}{nr_p\phi_2}] & -\frac{F}{L_4} \\ 0 & 0 & 0 & \frac{R_o}{C_8(r_{c8}+R_o)} & -\frac{1}{C_8(r_{c8}+R_o)} \end{bmatrix} \tag{3.18a}$$

$$B_1 = \begin{bmatrix} \frac{1}{L_{12}+L_3} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18b}$$

$$W_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{L_4} \\ 0 \end{bmatrix} \tag{3.18c}$$

The outputs of interest are the magnetizing current, x_3, the output inductor current, x_4, and the output voltage, V_3. The C matrix is then

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & E & F \end{bmatrix} \tag{3.18d}$$

Current Load

If instead of a resistive load a current load is to be simulated, then the expressions for x_4-dot, x_5-dot, b, and the output voltage must be redefined. The expression for x_4-dot is now

$$\dot{x}_4 = \frac{k_1}{L_4\phi_2}x_1 + \frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}]x_2 - \frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} + \frac{\tau}{n}]x_3 - \frac{1}{L_4}[b + \frac{k_1\tau}{nr_p\phi_2}]x_4 - \frac{1}{L_4}x_5 - \frac{1}{L_4}V_d - \frac{r_{c8}}{L_4}I_o \tag{3.19a}$$

and b is

$$b = r_{01} + r_{L4} + r_{c8} + \frac{\tau}{n^2} \tag{3.19b}$$

For x_5-dot, it is

$$\dot{x}_5 = \frac{1}{C_8}x_4 - \frac{1}{C_8}I_o \tag{3.20}$$

and for V_3, the output voltage, the expression is now

$$V_3 = r_{c8}x_4 + x_5 - r_{c8}I_o \tag{3.21}$$

The state-space representation in (3.5) must now be slightly modified as follows,

$$\begin{aligned} \dot{x} &= A_i'x + B_i'u + W_i'V_d + Z_iI_o \\ y &= C'x + EI_o \end{aligned} \tag{3.22}$$

Note that the matrix E, like C', does not change for all three topologies. The resulting matrices are

$$A_1' = \begin{bmatrix} -\frac{\psi_1}{L_{12}+L_3} & -\frac{\psi_2}{L_{12}+L_3} & \frac{\psi_3}{L_{12}+L_3} & \frac{\psi_4}{L_{12}+L_3} & 0 \\ \frac{1}{\phi_2} & -\frac{\phi_3}{\phi_2} & -\frac{\tau}{r_p\phi_2} & -\frac{\tau}{nr_p\phi_2} & 0 \\ \frac{\Delta_1}{\phi_2} & [\Delta_2 - \frac{\Delta_1\phi_3}{\phi_2}] & -[\frac{\Delta_1\tau}{r_p\phi_2} + \frac{\tau}{L_m}] & -[\frac{\Delta_1\tau}{nr_p\phi_2} + \frac{\tau}{nL_m}] & 0 \\ \frac{k_1}{L_4\phi_2} & \frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}] & -\frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} + \frac{\tau}{n}] & -\frac{1}{L_4}[b + \frac{k_1\tau}{nr_p\phi_2}] & -\frac{1}{L_4} \\ 0 & 0 & 0 & \frac{1}{C_8} & 0 \end{bmatrix} \quad (3.23a)$$

$$Z_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{r_{c8}}{L_4} \\ -\frac{1}{C_8} \end{bmatrix} \quad (3.23b)$$

$$E = \begin{bmatrix} 0 \\ 0 \\ -r_{c8} \end{bmatrix} \quad (3.23c)$$

$$C' = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & r_{c8} & 1 \end{bmatrix} \quad (3.23d$$

and $B_1' = B_1$ and $W_1' = W_1$.

3.2.2.2 Second Topology: all switches are OFF

In the second topology (Figure 3-3), all four switches are off. The voltage across the primary is zero and the LC supplies the power to the output. During this period, both

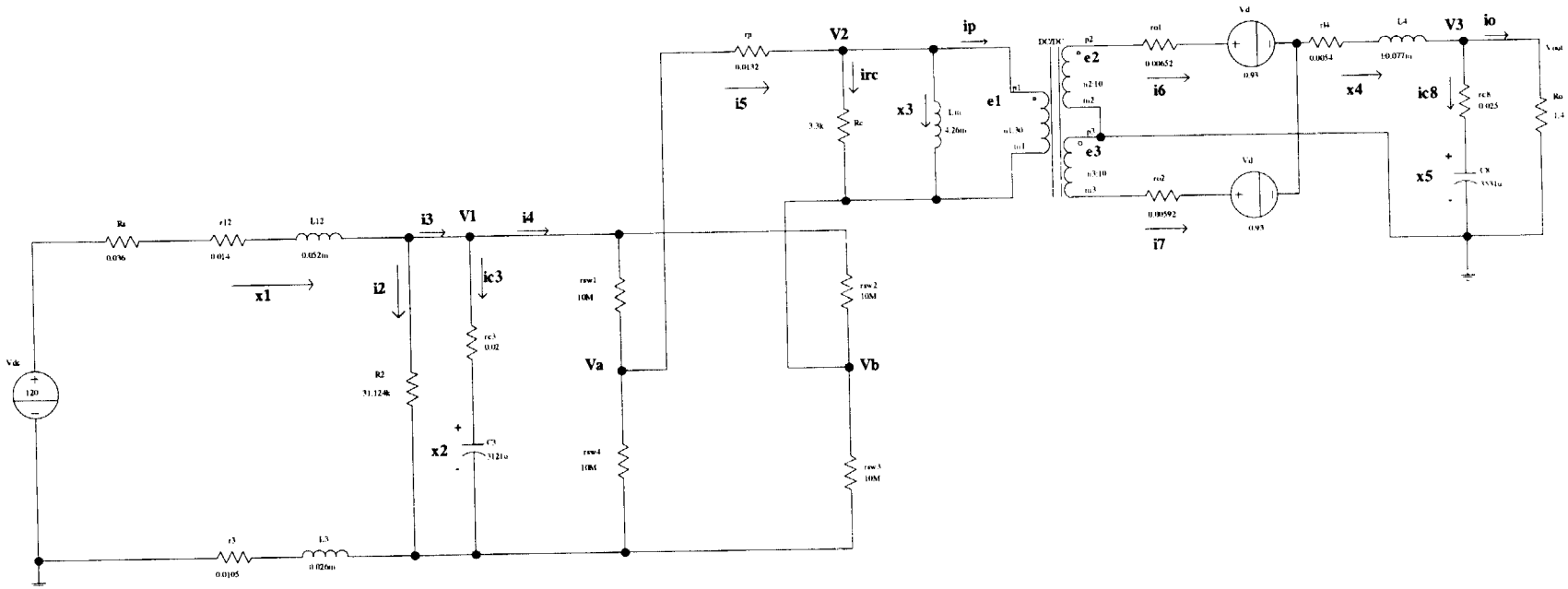


Figure 3-3: The Second Switched Topology

diodes are forward biased each carrying half of the output current, assuming both branches are equally balanced.

Resistive Load

In this topology, $i_5 = 0$, $V_{ab} = 0$, and

$$i_4 = \frac{V_1}{(r_{sw1} + r_{sw4})(r_{sw2} + r_{sw3})} \tag{3.24a}$$

but V_1 is still (3.8); therefore,

$$i_4 = \frac{(r_{sw1} + r_{sw2} + r_{sw3} + r_{sw4})r_{c3}C_3}{(r_{sw1} + r_{sw4})(r_{sw2} + r_{sw3})}\dot{x}_2 + \frac{(r_{sw1} + r_{sw2} + r_{sw3} + r_{sw4})}{(r_{sw1} + r_{sw4})(r_{sw2} + r_{sw3})}x_2 \tag{3.24b}$$

Obviously, this is the leakage current of the MOSFET switches and *could* be ignored to further simplify the expressions since it is a small value. Proceeding with the derivation, the expression for x_2-dot is derived as before using (3.11a) and substituting (3.24b) for i_4, and (3.8) for V_1. Solving for x_2-dot yields

$$\dot{x}_2 = \frac{1}{\phi_2}x_1 - \frac{\phi_3}{\phi_2}x_2 \tag{3.25a}$$

where

$$\begin{aligned} \phi_1 &= C_3[\frac{r_{c3}}{R_2} + 1] \\ \phi_2 &= \phi_1 + \frac{(r_{sw1} + r_{sw2} + r_{sw3} + r_{sw4})r_{c3}C_3}{(r_{sw1} + r_{sw4})(r_{sw2} + r_{sw3})} \\ \phi_3 &= \frac{1}{R_2} + \frac{(r_{sw1} + r_{sw2} + r_{sw3} + r_{sw4})}{(r_{sw1} + r_{sw4})(r_{sw2} + r_{sw3})} \end{aligned} \tag{3.25b}$$

Since the new expression for x_2-dot is now known, x_1-dot is given by (3.12a) and (3.8); hence,

$$\dot{x}_1 = -\frac{\psi_1}{(L_{12} + L_3)}x_1 - \frac{\psi_2}{(L_{12} + L_3)}x_2 + \frac{1}{L_{12} + L_3}V_{dc} \tag{3.26a}$$

where

$$\psi_1 = R_s + r_{12} + r_3 + \frac{r_{c3}C_3}{\phi_2}$$
$$\psi_2 = 1 - \frac{r_{c3}C_3\phi_3}{\phi_2} \tag{3.26b}$$
$$\psi_3 = \psi_4 = 0$$

Note that the expressions for ψ_1 and ψ_2 are the same as for the first topology but their values are different since the expressions for ϕ_2 and ϕ_3 are now different. The third differential equation is again given by (3.13a), but

$$V_2 = -\tau x_3 \tag{3.27a}$$

since $i_p = 0$ and $V_{ab} = 0$ (compare to (3.7a)). The result is then

$$\dot{x}_3 = -\frac{\tau}{L_m} x_3 \tag{3.27b}$$

On the output side, the expression for x_5-dot is given by (3.14c) and the output is given by (3.15a) and (3.15b). By comparison to the first topology, the second topology has two extra variables, i_6 and i_7, that are linearly dependent on x_4 as follows:

$$i_6 + i_7 = x_4 \tag{3.28}$$

Furthermore,

$$r_{02}i_7 + V_d + r_{L4}x_4 + L_4\dot{x}_4 + V_3 = e_3 \tag{3.29}$$

and

$$r_{01}i_6 + V_d + r_{L4}x_4 + L_4\dot{x}_4 + V_3 = e_2 \tag{3.30}$$

but e_2 and e_3 are both zero since e_1 is zero. Expressing i_7 in terms of x_4 and i_6, substituting into (3.29), and solving for i_6 yields

$$i_6 = [1 + \frac{r_{L4}}{r_{02}}]x_4 + \frac{L_4}{r_{02}}\dot{x}_4 + \frac{V_3}{r_{02}} + \frac{V_d}{r_{02}} \tag{3.31}$$

The next step is to Substitute (3.31) and (3.15a) into (3.30) and solve for x_4-dot to yield the fourth differential equation:

$$\dot{x}_4 = -[\frac{\beta+(1+\alpha)E}{\gamma}]x_4 - \frac{(1+\alpha)F}{\gamma}x_5 - \frac{1+\alpha}{\gamma}V_d \tag{3.32a}$$

where

$$\begin{aligned} \alpha &= \frac{r_{01}}{r_{02}} \\ \beta &= r_{01} + (1+\alpha)r_{L4} \\ \gamma &= L_4(1+\alpha) \end{aligned} \tag{3.32b}$$

In matrix form as in (3.5), the resulting matrices are

$$A_2 = \begin{bmatrix} -\frac{\psi_1}{L_{12}+L_3} & -\frac{\psi_2}{L_{12}+L_3} & 0 & 0 & 0 \\ \frac{1}{\phi_2} & -\frac{\phi_3}{\phi_2} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\tau}{L_m} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\beta+(1+\alpha)E}{\gamma} & -\frac{(1+\alpha)F}{\gamma} \\ 0 & 0 & 0 & \frac{R_o}{C_8(r_{c8}+R_o)} & -\frac{1}{C_8(r_{c8}+R_o)} \end{bmatrix} \tag{3.33a}$$

$$W_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{(1+\alpha)}{\gamma} \\ 0 \end{bmatrix} \tag{3.33b}$$

and $B_2 = B_1$ and C does not change.

Current Load

For the case of a current load, the expressions for x_5-dot and V_3, hence the output voltage, are given by (3.20) and (3.21), respectively, and the state-space representation is in the form of (3.22). Consequently, the expression for x_4-dot is now

$$\dot{x}_4 = -[\frac{\beta+(1+\alpha)r_{c8}}{\gamma}]x_4 - \frac{(1+\alpha)}{\gamma}x_5 - \frac{(1+\alpha)}{\gamma}V_d + \frac{(1+\alpha)r_{c8}}{\gamma}I_o \tag{3.34}$$

The current load matrices are

$$A_2' = \begin{bmatrix} -\frac{\psi_1}{L_{12}+L_3} & -\frac{\psi_2}{L_{12}+L_3} & 0 & 0 & 0 \\ \frac{1}{\phi_2} & -\frac{\phi_3}{\phi_2} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\tau}{L_m} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\beta+(1+\alpha)r_{c8}}{\gamma} & -\frac{(1+\alpha)}{\gamma} \\ 0 & 0 & 0 & \frac{1}{C_8} & 0 \end{bmatrix} \tag{3.35a}$$

$$Z_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{(1+\alpha)r_{c8}}{\gamma} \\ -\frac{1}{C_8} \end{bmatrix} \tag{3.35b}$$

and $W_2' = W_2$ and $B_2' = B_1$. As is the case for the resistive load, C' and E remain constant for all three switched topologies.

3.2.2.3 Third Topology: Sw_2 and Sw_4 are ON

In the third topology (Figure 3-4), the second switch pair, Sw_2 and Sw_4, is on while the first switch pair, Sw_1 and Sw_3, is off. This causes the voltage across the primary winding to be $-V_{dc}$ (ignoring losses). As a result, diode D_2 conducts while diode D_1 is reverse biased (See Figure 2-1) and ignored.

Resistive Load

The derivation of the set of first-order differential equations for this topology follows exactly as for the first topology, except that i_p is now negative and $i_4 \approx -i_5$. Of course, V_{ab} is also negative but the sign change is automatically included through the constant R_b given in (3.9b). Hence, equation (3.6b) is now

$$i_5 = \frac{V_2}{R_c} + x_3 - i_p \tag{3.36}$$

Consequently, V_2 is

$$V_2 = -\tau x_3 + \frac{\tau}{n} x_4 + \frac{\tau}{r_p} V_{ab} \tag{3.37}$$

and

$$i_4 \approx -i_5 = -\frac{\tau}{r_p} x_3 + \frac{\tau}{n r_p} x_4 - r_{c3} \xi R_b C_3 \dot{x}_2 - \xi R_b x_2 \tag{3.38}$$

The constant τ is given by (3.7b). Proceeding exactly as before, the expressions for x_1-dot and x_2-dot are derived as

$$\dot{x}_1 = -\frac{\psi_1}{L_{12} + L_3} x_1 - \frac{\psi_2}{L_{12} + L_3} x_2 - \frac{\psi_3}{L_{12} + L_3} x_3 + \frac{\psi_4}{L_{12} + L_3} x_4 + \frac{1}{L_{12} + L_3} V_{dc} \tag{3.39a}$$

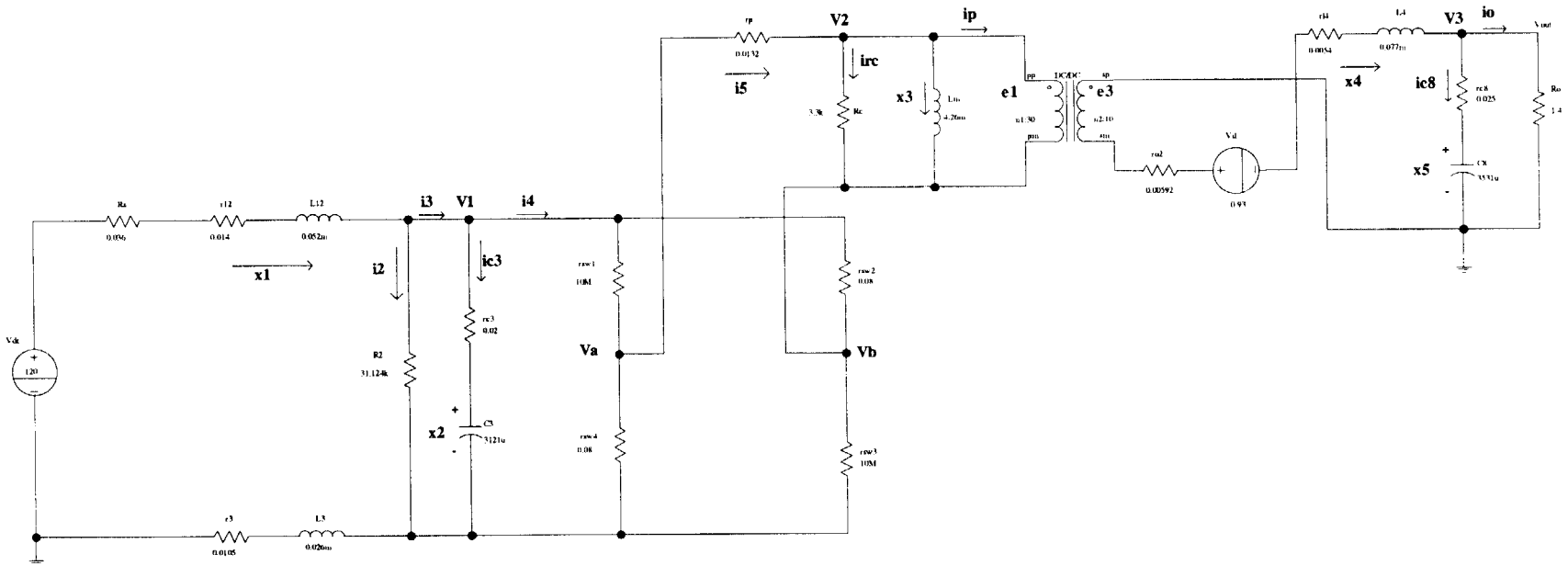


Figure 3-4: The Third Switched Topology

where

$$\psi_1 = R_s + r_{12} + r_3 + \frac{r_{c3}C_3}{\phi_2}$$
$$\psi_2 = 1 - \frac{r_{c3}C_3\phi_3}{\phi_2} \tag{3.39b}$$
$$\psi_3 = \frac{r_{c3}C_3\tau}{r_p\phi_2}$$
$$\psi_4 = \frac{\psi_3}{n}$$

and

$$\dot{x}_2 = \frac{1}{\phi_2}x_1 - \frac{\phi_3}{\phi_2}x_2 + \frac{\tau}{r_p\phi_2}x_3 - \frac{\tau}{nr_p\phi_2}x_4 \tag{3.40a}$$

where

$$\phi_1 = C_3(\frac{r_{c3}}{R_2} + 1)$$
$$\phi_2 = \phi_1 - r_{c3}\xi R_b C_3 \tag{3.40b}$$
$$\phi_3 = \frac{1}{R_2} - \xi R_b$$

In like manner, the expression for x_3-dot is derived as

$$\dot{x}_3 = \frac{\Delta_1}{\phi_2}x_1 + [\Delta_2 - \frac{\Delta_1\phi_3}{\phi_2}]x_2 + [\frac{\Delta_1\tau}{r_p\phi_2} - \frac{\tau}{L_m}]x_3 - [\frac{\Delta_1\tau}{nr_p\phi_2} - \frac{\tau}{nL_m}]x_4 \tag{3.41a}$$

where Δ_1 and Δ_2 are given by (3.13c) but must be recalculated for this topology due to the sign change in R_b.

The expression for the fifth differential equation and the output voltage, V_3, is once again given by (3.14c) and (3.15a) and (3.15b). The voltage across the second secondary winding is similar to (3.16a) and is

$$e_3 = \frac{e_1}{n} = \frac{V_2}{n} \tag{3.42a}$$

After substituting for V_2 and x_2-dot, e_3 is

$$e_3 = \frac{k_1}{\phi_2} x_1 + [k_2 - \frac{k_1\phi_3}{\phi_2}]x_2 + [\frac{k_1\tau}{r_p\phi_2} - \frac{\tau}{n}]x_3 - [\frac{k_1\tau}{nr_p\phi_2} - \frac{\tau}{n^2}]x_4 \tag{3.42b}$$

where k_1 and k_2 are given in (3.16c) and must be recalculated due to R_b. Finally, the expression for x_4-dot is derived from

$$L_4\dot{x}_4 = -(r_{02} + r_{L4})x_4 - e_3 - V_3 - V_d \tag{3.43a}$$

and, after substituting for e_3 and V_3, x_4-dot is given as

$$\dot{x}_4 = -\frac{k_1}{L_4\phi_2} x_1 - \frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}]x_2 - \frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} - \frac{\tau}{n}]x_3 - \frac{1}{L_4}[b - \frac{k_1\tau}{nr_p\phi_2}]x_4 - \frac{F}{L_4}x_5 - \frac{1}{L_4}V_d \tag{3.43b}$$

where

$$b = r_{02} + r_{L4} + \frac{\tau}{n^2} + E \tag{3.43c}$$

The matrices for this topology are as follows:

$$A_3 = \begin{bmatrix} -\frac{\psi_1}{L_{12}+L_3} & -\frac{\psi_2}{L_{12}+L_3} & -\frac{\psi_3}{L_{12}+L_3} & \frac{\psi_4}{L_{12}+L_3} & 0 \\ \frac{1}{\phi_2} & -\frac{\phi_3}{\phi_2} & \frac{\tau}{r_p\phi_2} & -\frac{\tau}{nr_p\phi_2} & 0 \\ \frac{\Delta_1}{\phi_2} & [\Delta_2 - \frac{\Delta_1\phi_3}{\phi_2}] & [\frac{\Delta_1\tau}{r_p\phi_2} - \frac{\tau}{L_m}] & -[\frac{\Delta_1\tau}{nr_p\phi_2} - \frac{\tau}{nL_m}] & 0 \\ -\frac{k_1}{L_4\phi_2} & -\frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}] & -\frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} - \frac{\tau}{n}] & -\frac{1}{L_4}[b - \frac{k_1\tau}{nr_p\phi_2}] & -\frac{F}{L_4} \\ 0 & 0 & 0 & \frac{R_o}{C_8(r_{c8}+R_o)} & -\frac{1}{C_8(r_{c8}+R_o)} \end{bmatrix} \tag{3.44}$$

and $B_3 = B_1$ and $W_3 = W_1$.

Current Load

As in the first and second switched topologies, the expressions for x_5-dot and V_3 are given by (3.20) and (3.21), respectively, and the state-space representation is in the form of (3.22). Substituting (3.21) and (3.42b) into (3.43a) yields x_4-dot as

$$\dot{x}_4 = -\frac{k_1}{L_4\phi_2}x_1 - \frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}]x_2 - \frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} - \frac{\tau}{n}]x_3 - \frac{1}{L_4}[b - \frac{k_1\tau}{nr_p\phi_2}]x_4 - \frac{1}{L_4}x_5 - \frac{1}{L_4}V_d + \frac{r_{c8}}{L_4}I_0 \tag{3.45a}$$

where

$$b = r_{02} + r_{L4} + r_{c8} + \frac{\tau}{n^2} \tag{3.45b}$$

The only difference between equation (3.45b) and (3.19b) is r_{02}. The state-space representation matrices are as follows:

$$A_3' = \begin{bmatrix} -\frac{\psi_1}{L_{12}+L_3} & -\frac{\psi_2}{L_{12}+L_3} & -\frac{\psi_3}{L_{12}+L_3} & \frac{\psi_4}{L_{12}+L_3} & 0 \\ \frac{1}{\phi_2} & -\frac{\phi_3}{\phi_2} & \frac{\tau}{r_p\phi_2} & -\frac{\tau}{nr_p\phi_2} & 0 \\ \frac{\Delta_1}{\phi_2} & [\Delta_2 - \frac{\Delta_1\phi_3}{\phi_2}] & [\frac{\Delta_1\tau}{r_p\phi_2} - \frac{\tau}{L_m}] & -[\frac{\Delta_1\tau}{nr_p\phi_2} - \frac{\tau}{nL_m}] & 0 \\ -\frac{k_1}{L_4\phi_2} & -\frac{1}{L_4}[k_2 - \frac{k_1\phi_3}{\phi_2}] & -\frac{1}{L_4}[\frac{k_1\tau}{r_p\phi_2} - \frac{\tau}{n}] & -\frac{1}{L_4}[b - \frac{k_1\tau}{nr_p\phi_2}] & -\frac{1}{L_4} \\ 0 & 0 & 0 & \frac{1}{C_8} & 0 \end{bmatrix} \tag{3.46a}$$

and $B_3' = B_1$, $W_3' = W_1$, and $Z_3 = Z_1$.

The mathematical model, which incorporates all three switched topologies, was implemented in two M-files in MATLAB®, one for resistive load and the other for current load (See Appendix B). In the simulation M-file, the last state of the previous topology is used as the initial state of the current topology. Hence, the last state of the first topology is

the initial state of the second topology, the last state of the second topology is the initial state of the third topology and the last state of the third topology is the initial state of the first topology and so on. The DCM is accomplished by setting the current of the output inductor (L_4) to zero when it calculates to a negative value.

3.3 Linear Transfer Function Model

A third modeling approach is a transfer function model derived from empirical data. In order to derive this model, actual steady-state and transient data must be collected. This means that a physical converter must exist. It is, therefore, limited to linear analysis and controller design with the restriction that the closed loop stability can only be guaranteed in the neighborhood of the operating point where the model was derived. The main advantage of this model is the very short CPU execution time required to simulate.

The linear model in Figure 3-5 can be used for both steady-state and transient simulation. The operating point around which the Plant transfer function is valid (in terms

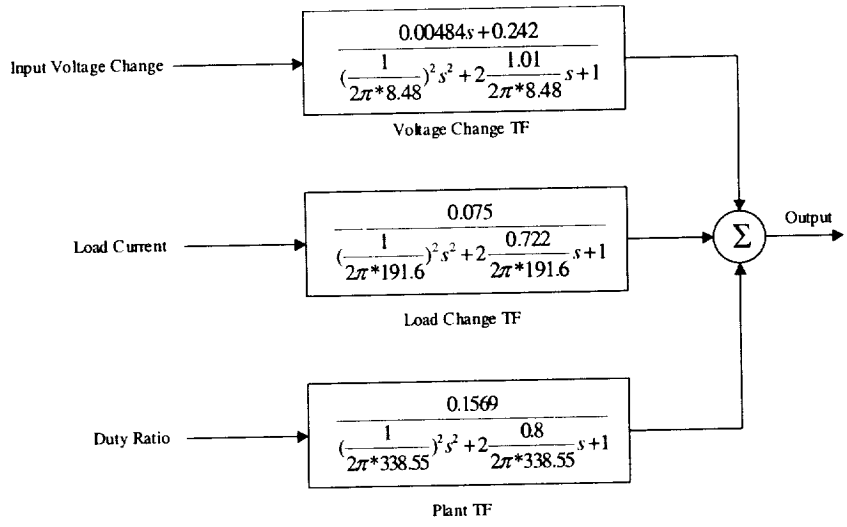


Figure 3-5: Empirical Linear Model of the Full-Bridge Converter

of transient dynamics) is 120 V_{dc} input, 4 Amp load, and 72.94% (186 Pulse Counts) duty ratio. DC gain is the input voltage (120) divided by the product of 3 (the transformer turns-ratio) and 255. It should be noted that the duty ratio is in terms of pulse count, which ranges from 0 – 255 due to the eight-bit quantization. Of course, the pulse count value can be easily converted to percent duty ratio by dividing the pulse count by 255 and multiplying by 100. The DC gain in the Load Change transfer function is the approximate output impedance of the converter under load.[24] As the load increases, the output voltage decreases; thus, the reason for the minus sign in the summation block. The operating point around which this transfer function is valid is 120 V_{dc}, 72.94% duty ratio, and unspecified current load. The third transfer function is the Voltage Change transfer function. The operating point around which this transfer function is valid is 120V_{dc}, 72.94% duty ratio, and 4 Amp load.

If interested in the steady-state response only, the following equation, reported in [24], can be used:

$$V_0 = \frac{V_{in}}{3*255}(PulseCount) - 0.8 - (0.075 * I_L) \tag{3.47}$$

Note that equation (1.13) is the first part of equation (3.47). In terms of steady-state, the only difference between this equation and the linear model at 120 V_{dc} input is the "0.8" term present in equation (3.47), which represents the diode forward drop. In other words, the steady-state value obtained through the linear model above will always be higher than that obtained from (3.47) by 0.8 V. The reason this relationship does not hold at 140 V_{dc} input is that the DC gain in the Voltage Change transfer function is constant when, in reality, it varies due to the "modulation" effect of the duty ratio. Figures 4-9 and 4-10

reflect this fact in the slope of the TF (linear model) graph. Equation (3.47) is limited to steady-state only and for that reason will not be considered further.

An improvement of the linear model in Figure 3-5 in terms of both steady-state and transient response is shown in Figure 3-6 below. This model includes the "modulation" effect on the input voltage change by the duty ratio and the gain of the Voltage Change transfer function is simply the reciprocal of the turns-ratio of the transformer—1/3. This is included in Figures 4-9 and 4-10 as TFi.

The operating point around which the Plant Transfer Function is valid is 120 V_{dc} input, 20-Amp load, and a duty ratio of 70.59% (180 Pulse Counts). Based on the converter response for a 4-amp, 20-Amp, and 32-Amp load, the largest overshoot was obtained for the 20-Amp load and, as a result, the improved linear model was tuned to this response. For the Load Change Transfer Function, the validity is around 120 V_{dc} input, duty ratio of

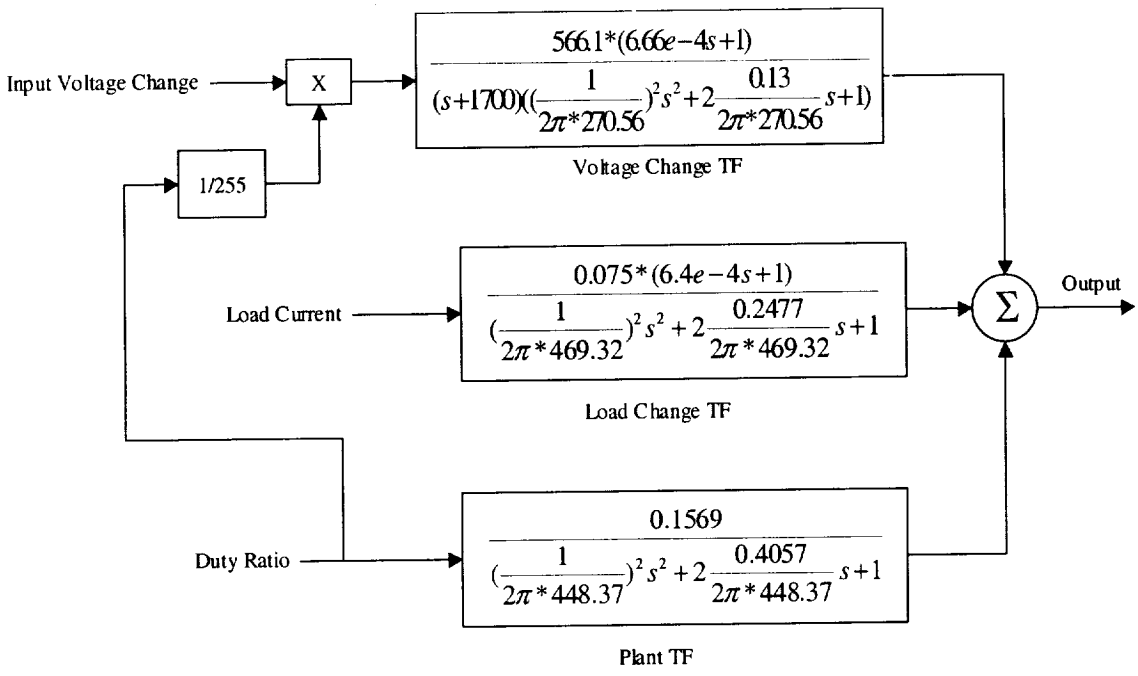


Figure 3-6: Improved Empirical Linear Model

70.59%, and load of 4 Amps. The undershoot was larger when stepping from 4 – 8 Amps than 20 – 24 Amps. Finally, the Voltage Change Transfer Function is valid around 120 V_{dc} input, duty ratio of 72.94% (186 Pulse Counts), and load of 32 Amps. This can also be inferred from the transient output voltage plots in Chapter 4.

3.4 State-Space Average Model

A fourth modeling approach is the averaging technique, discussed in chapter one, that is widely known and used. In order to obtain an average model, it is first necessary to derive the state-space representation of each switched topology as was done in section 3.2. Hence, this approach requires two additional steps compared to deriving a switched state-space model that results in a closed-form solution. The first is the averaging of the state-space representation and, the second step, is the derivation of the transfer function(s) after eliminating any nonlinear terms that may have resulted from the averaging process.

Applying the averaging method to the three state-space representations above will result in an average model that is valid for the CCM of operation. The reason is that a fourth topology would have to be considered for DCM of operation. This topology results when the output inductor (L4) current goes to zero for a finite time when all the switches are off. However, based on actual data, the converter goes Discontinuous around 0.9 Amps, and below, of current load. This is less than 2.5% of the operational current load range. As a result, the DCM of operation is ignored.

For current load and when Sw_1 and Sw_3 are ON for dT_s, the state-space representation is

$$\begin{aligned} \dot{x} &= A_1 x + Bu + WV_d + ZI_0 \\ y_1 &= Cx + EI_0 \end{aligned} \tag{3.48}$$

and, when all switches are OFF for $(1-2d)T_s$, the state-space representation is

$$\begin{aligned} \dot{x} &= A_2 x + Bu + WV_d + ZI_0 \\ y_2 &= Cx + EI_0 \end{aligned} \tag{3.49}$$

The third topology results when Sw_2 and Sw_4 are ON for dT_s. The state-space representation is

$$\begin{aligned} \dot{x} &= A_3 x + Bu + WV_d + ZI_0 \\ y_3 &= Cx + EI_0 \end{aligned} \tag{3.50}$$

In this case, the matrices B, W, and Z are the same for all three topologies. This is not always the case. For example, for resistive load the matrix B is different for all three topologies. The output matrices C and E are also constant.

After averaging and perturbing as discussed in chapter one, the steady-state model and the dynamic model result. The steady-state model is

$$\begin{aligned} AX + BU + WV_d + ZI &= 0 \\ Y &= CX + EI \end{aligned} \tag{3.51}$$

where

$$A = DA_1 + (1-2D)A_2 + DA_3 \tag{3.52}$$

and the dynamic model is

$$\begin{aligned} \dot{x} &= A\hat{x} + B\hat{u} + A_{xd}X\hat{d} + Z\hat{i}_0 + A_{xd}\hat{x}\hat{d} \\ \hat{y} &= C\hat{x} + E\hat{i}_0 \end{aligned} \tag{3.53}$$

where

$$A_{xd} = A_1 - 2A_2 + A_3 \tag{3.54}$$

In order to derive the average linear transfer functions, the nonlinear term, $A_{xd}\hat{x}\hat{d}$, in equation 3.53 is ignored. For the duty ratio-to-output transfer function, $\hat{u}$ and $\hat{i}_0$ are set to zero. The resulting transfer function is

$$G_{dv}(s) = C(sI - A)^{-1} A_{xd} X \tag{3.55}$$

The input voltage-to-output transfer function and the load current-to-output transfer function are obtained in a similar manner. These are given as

$$G_{iv}(s) = C(sI - A)^{-1} B \tag{3.56}$$

and

$$G_{cv}(s) = C(sI - A)^{-1} Z + E \tag{3.57}$$

The transfer function at any operating point is easily derived with the M-file in Appendix D. In Figure 3-7, the Duty Ratio Change transfer function and the Load Change transfer function are valid around 4 amps load, 120V input voltage, and 70.59% duty ratio. The Voltage Change transfer function is valid for 4 amps load, 120V input voltage, and 72.94% duty ratio. In Figure 3-8, the only difference in the operating point is the current load. The Duty Ratio Change transfer function and the Load Change transfer function were

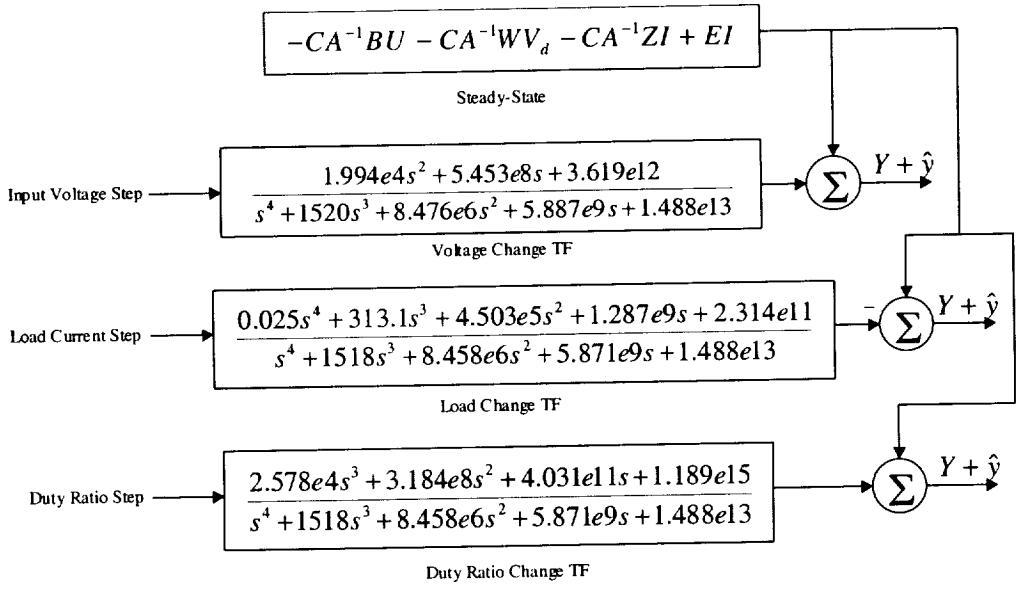


Figure 3-7: Average Linear Model for 4-Amp Load

derived for a 20-amp load whereas the Voltage Change transfer function was derived for 32-amp load. These transfer functions are valid for input step changes and should not be used to calculate the steady state value; the predicted values are about one volt higher. However, the steady-state value is accurately predicted by (3.51). In contrast, the transfer functions in Figure 3-6 can be used to predict the steady-state value since the DC gain of each transfer function was derived based on the physics of the converter.

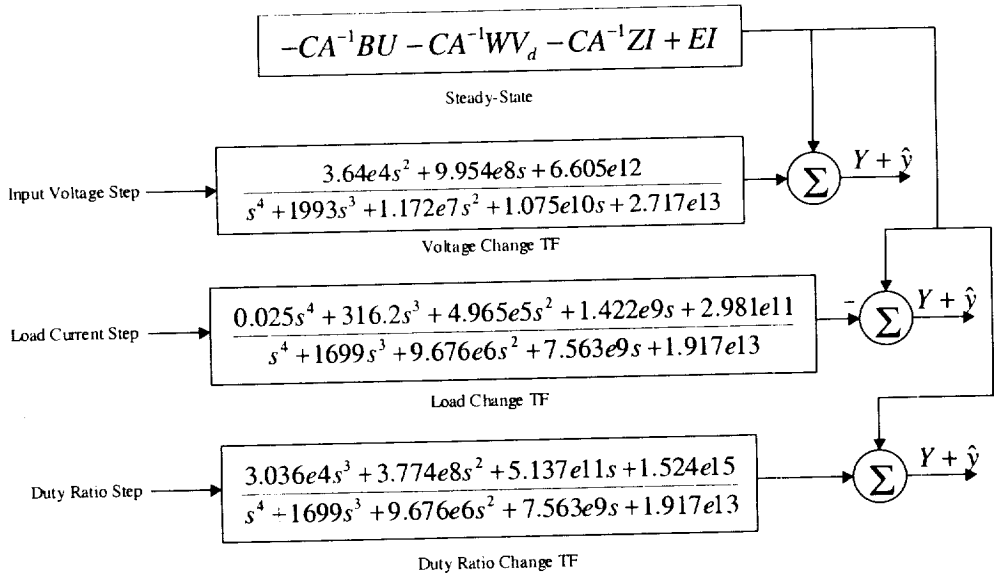


Figure 3-8: Average Linear Model for 20-Amp/32-Amp Load

The following chapter compares the results of the SABER® model and the mathematical model to the actual Full-Bridge converter. Comparison is also made to the linear transfer function model (Figure 3-5), which may have some slight differences to that reported in [23] and [24], for steady-state only and the improved linear model (Figure 3-6). In the section four of chapter four, the average model is briefly compared to the SABER® and MATLAB® (Switched) models as well as to the improved linear model for steady-state only. All the linear transfer function models are for current loads only.

CHAPTER IV

MODEL VALIDATION

4.1 Introduction

An important aspect of modeling is the validation of the model since it is essential to know the accuracy or lack thereof of the model(s). The validation is accomplished by comparing the steady-state response and the transient response of the model(s) to the converter's response. In addition, select converter waveforms, such as the primary current, are compared to that predicted by the SABER® model and, in one instance, to that predicted by the mathematical model.

4.2 Steady-State Response

For a specific set of conditions, a stable system will reach an equilibrium point or a steady state. In the case of the Full-Bridge converter, the steady state is dependent on the input voltage, the load (whether current or resistive), and the duty ratio. Therefore, to validate the steady-state response of the models, it is necessary to collect open-loop data for a range of input conditions. The range is dictated by the operational range of the actual converter and the nominal operating point.

The nominal operating point of the Full-Bridge converter under study is 28 V_{dc} output at 120 V_{dc} input and 70 % duty ratio (35% per switch pair). Under load, however, the duty ratio required to maintain 28 V_{dc} at the output is closer to 75 % for a current load of 20 Amps, for example, to overcome losses in the non-ideal circuit. The operational range, on the other hand, is spelled out in [23] and [24] and is briefly stated here. The input voltage can range from 110 – 150 V_{dc} and the current load can range from 0 – 40 Amps, as

Steady-state data was collected for 120 V_{dc} and 140 V_{dc}. At 120 V_{dc}, data was collected for current loads of 4, 8, 20, and 32 Amps and, for 140 V_{dc}, data was collected at 4 and 32 Amps only. The duty ratio ranged from 56.47% to 87.84% at 6.27% increments. In terms of pulse counts, the range is from 144 pulse counts to 224 pulse counts at 16 pulse count increments (For results with resistive load see Appendix C).

The steady-state response for different loads of the Full-Bridge converter, the SABER® model, the MATLAB® model, and the linear model is shown in Figures 4-1 through 4-4 for 120 V_{dc} input. These figures show that at light loads, the MATLAB® model yields better results than the linear model, labeled TF on the graphs (for duty ratio steady-state at 120 V_{dc}, TF and TFi are the same), but at higher loads, the linear model is better. Throughout the load range, however, the SABER® model is by far the best model. The worst case ΔV for the SABER® model is –0.26 V at a duty ratio of 56.47% and a load of 4 Amps. For the MATLAB model the worst case is 1.12 V at 81.57% and a load of 32 Amps, whereas for the linear model it is 0.92 V at 81.57% and a load of 8 Amps. Note that the steady-state predicted by the MATLAB® model gets progressively higher as the load increases. Compared to the SABER® model, the MATLAB® model does not take into account the losses due to the turn-on and turn-off times of the MOSFET® switches and the

leakage inductance. It might be tempting to simply increase the on-resistance of the switches in the MATLAB® model by 0.05Ω, according to calculations, but the drawback to this is the damping effect of higher resistance. The MATLAB® model will have a longer rise time (of the output voltage) when subjected to disturbances, such as a step in duty ratio.

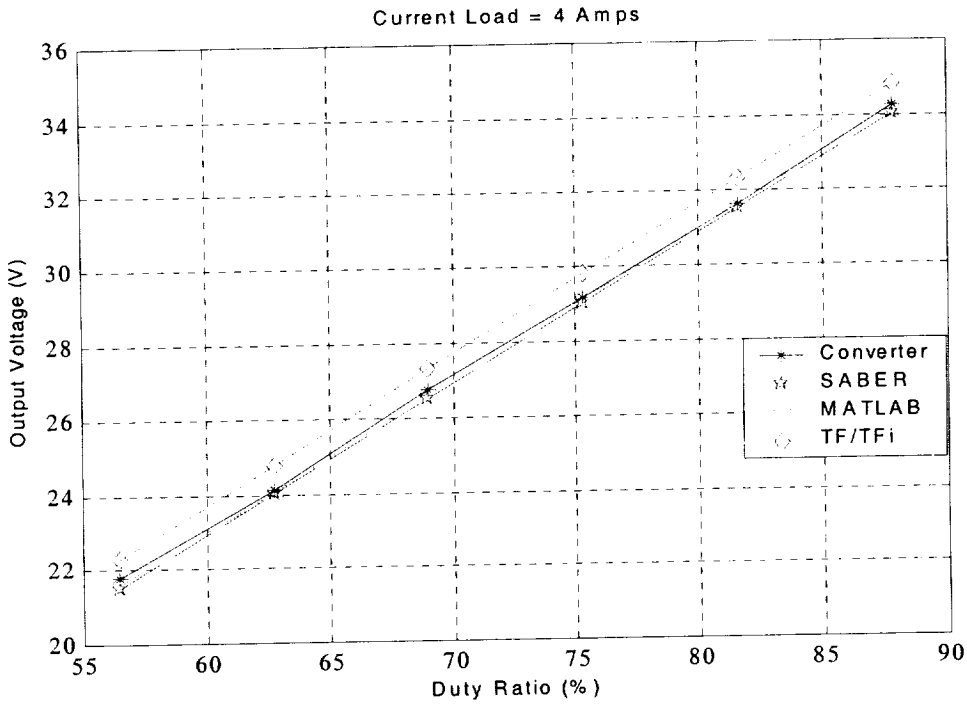


Figure 4-1: Steady-State Response at 4 Amps and 120 V_{dc} Input

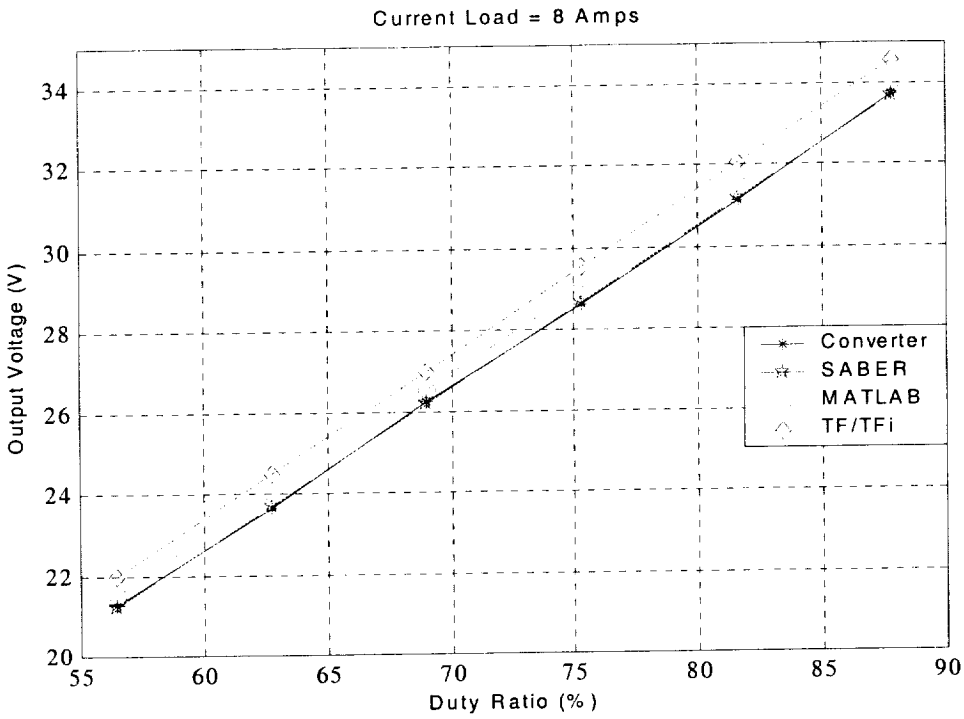


Figure 4-2: Steady-State Response at 8 Amps and 120 V_{dc} Input

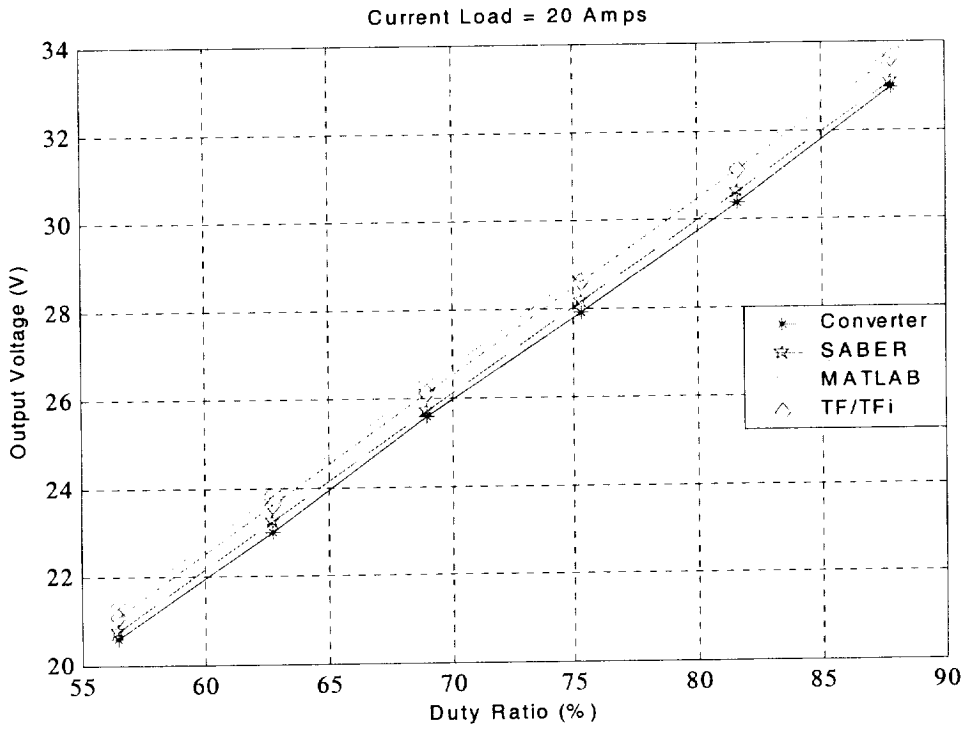


Figure 4-3: Steady-State Response at 20 Amps and 120 V_{dc} Input

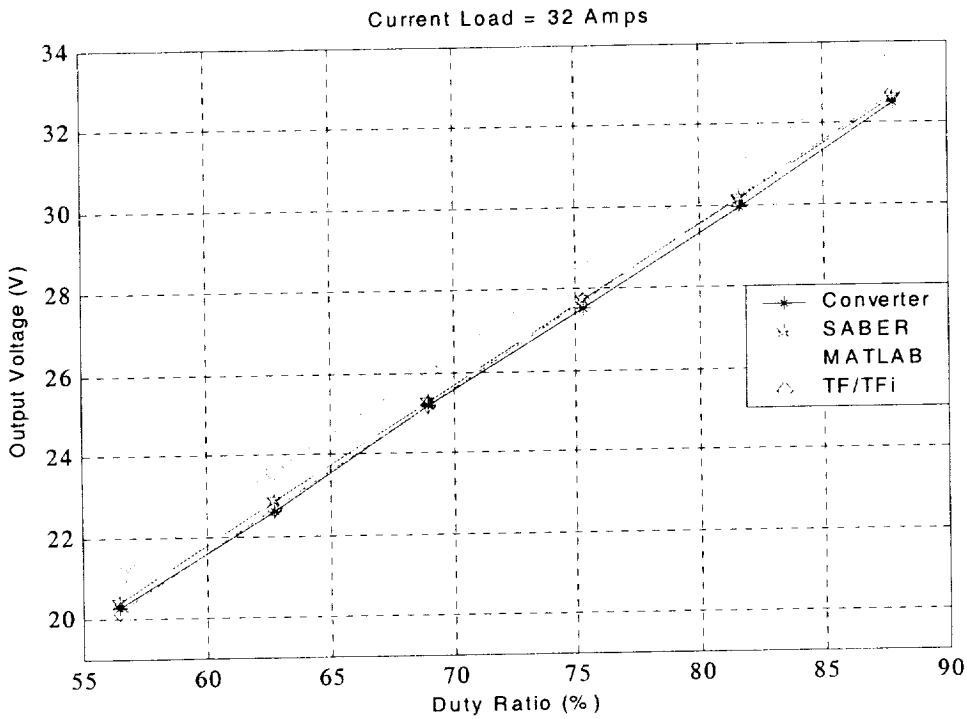


Figure 4-4: Steady-State Response at 32 Amps and 120 V_{dc} Input

The linear model is simply an input-output model with the benefit of fast simulation time. The SABER® model, and to some extent the MATLAB® model, are by far more complex and are capable of simulating other converter waveforms, such as ripple, inductor currents, capacitor voltages, primary winding current, etc. For this reason, the following graphs, Figures 4-5 through 4-8, does not include the linear model. These figures show a good correlation between the converter results and the models' results (See Appendix C, Section C, for the steady-state data). Note that in the more capable models, the ESR of the output capacitor (C_8) was tweaked until the simulation ripple voltage matched the converter's measured ripple voltage. Not also that the ripple increases as the load increases. The main reason for this is the reduction of the inductance of the output inductor as the DC bias increases. In the models, this is accomplished through the inductance polynomial equations.

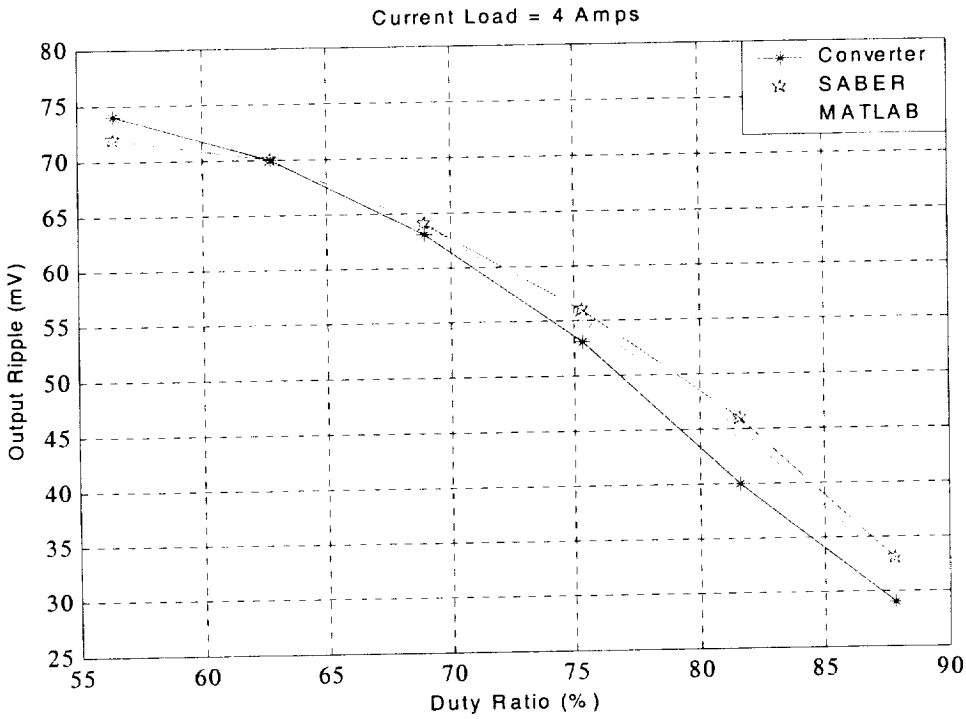


Figure 4-5: Output Voltage Ripple at 4 Amps and 120 V_{dc} Input

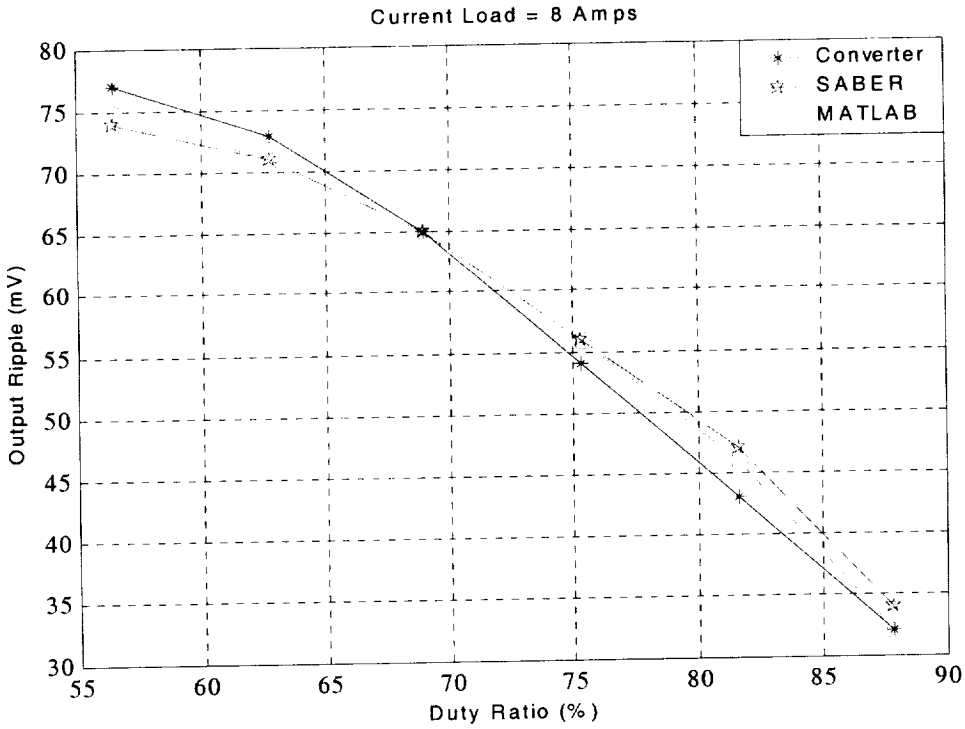


Figure 4-6: Output Voltage Ripple at 8 Amps and 120 V_{dc} Input

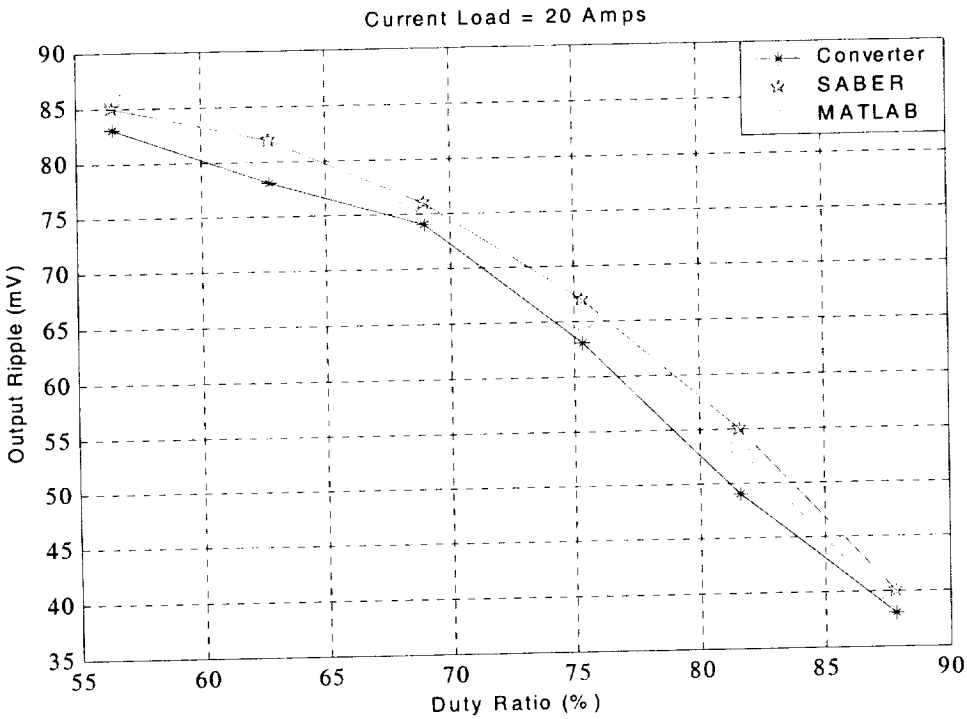


Figure 4-7: Output Voltage Ripple at 20 Amps and 120 V_{dc} Input

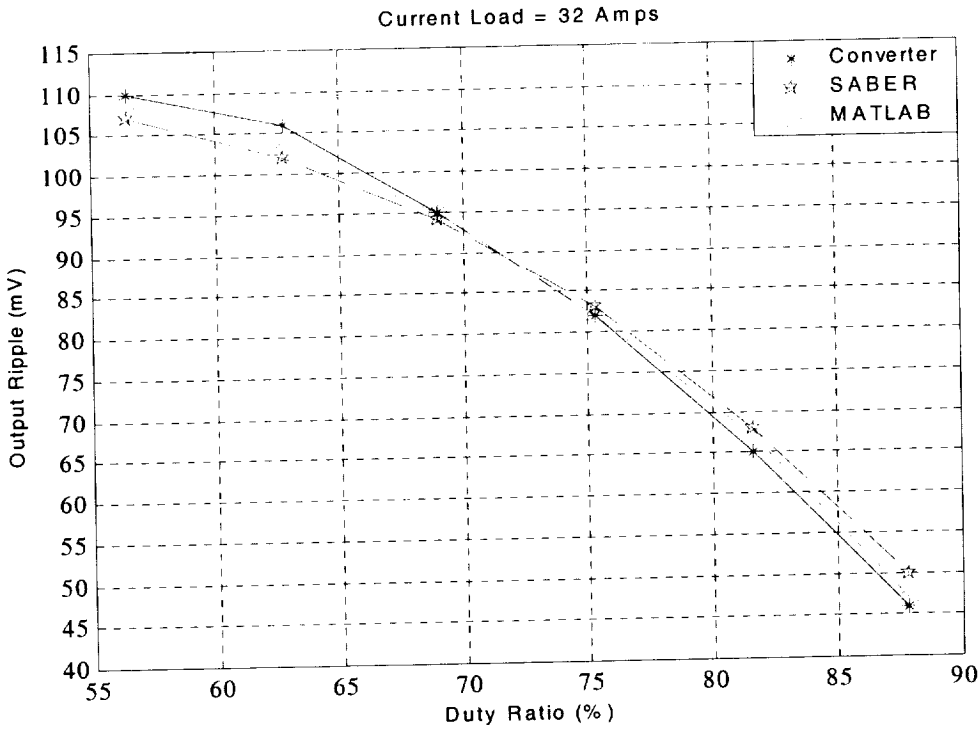


Figure 4-8: Output Voltage Ripple at 32 Amps and 120 V_{dc} input

The steady-state response at 140 V_{dc} input of the converter and all three models plus the improved linear model is shown in Figure 4-9 for a 4 Amp load and Figure 4-10 for a 32 Amp load. Of all four models, the linear model from [23] is the worst. The improved linear model and the MATLAB model are comparable, except the MATLAB model yields better results at low current loads whereas the improved linear model is better at higher current loads. The worst case ΔV for the SABER® model is –0.20 V at 81.57% duty ratio and a load of 4 Amps and for the MATLAB® model it is 1.16 V at 87.84% duty ratio and a load of 32 Amps. The linear model has a worst case ΔV of 1.66 V at 56.47% duty ratio and a load of 4 Amps whereas the improved linear model has a worst case ΔV of 0.59 V at a load of 4 Amps and at a range of duty ratios, starting from 69.02%. The output voltage ripple is shown in Figure 4-11 for a 4 Amp load and Figure 4-12 for a 32 Amp load. The correlation is once again very good but with some disagreement at 32 Amp load and duty ratio below 65%. The SABER® model is by far the best model (See Appendix C, Section C, for the steady-state data).

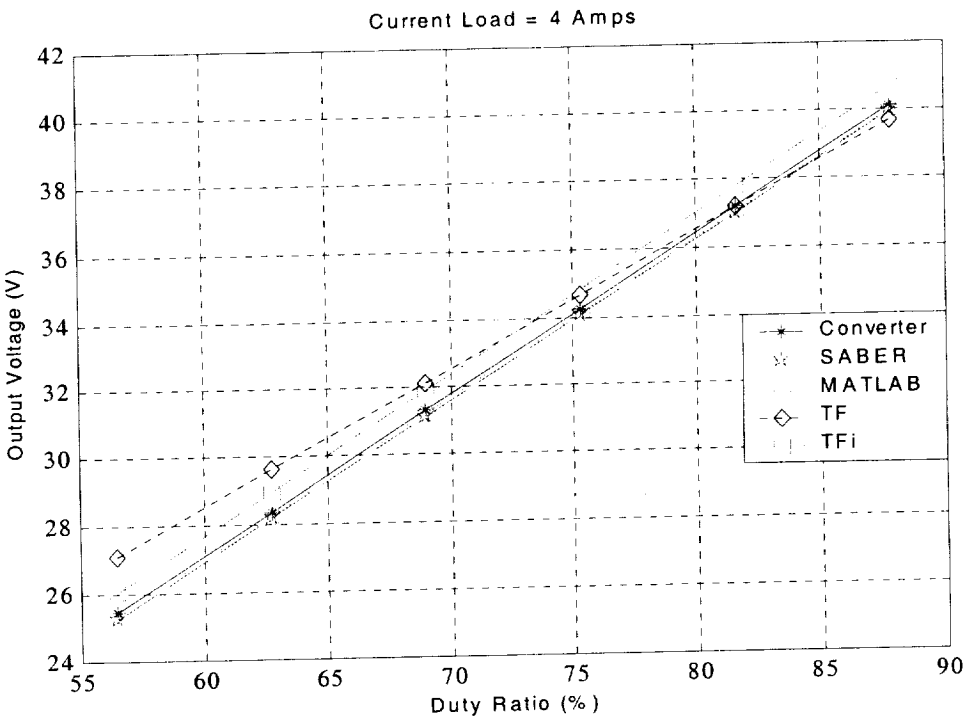


Figure 4-9: Steady-State Response at 4 Amps and 140 V_{dc} Input

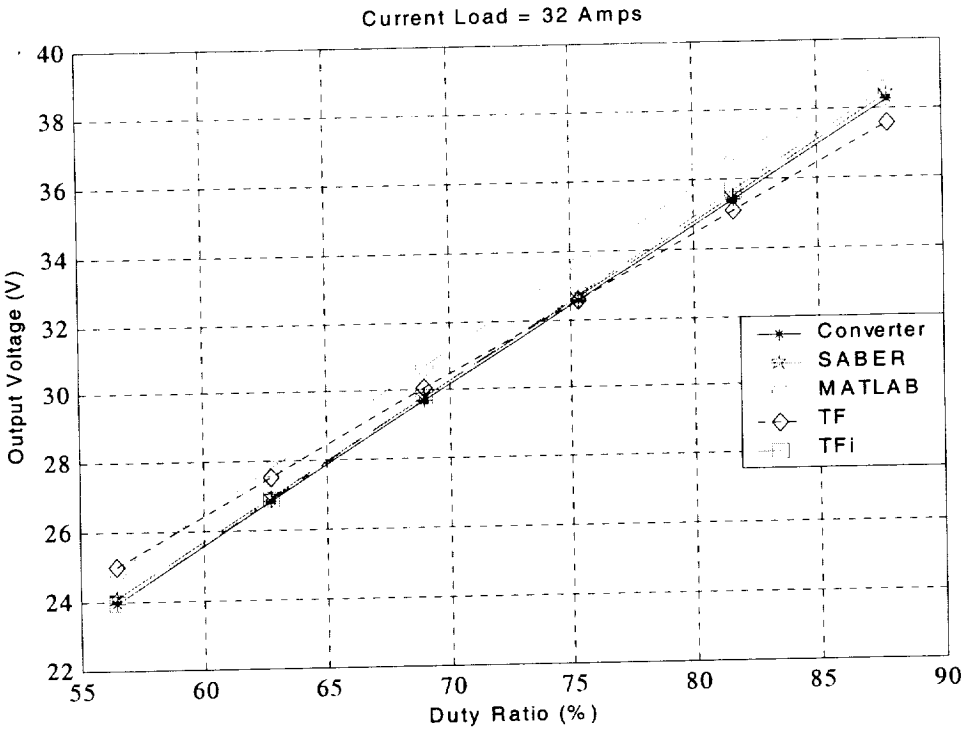


Figure 4-10: Steady-State Response at 32 Amps and 140 V_{dc} Input

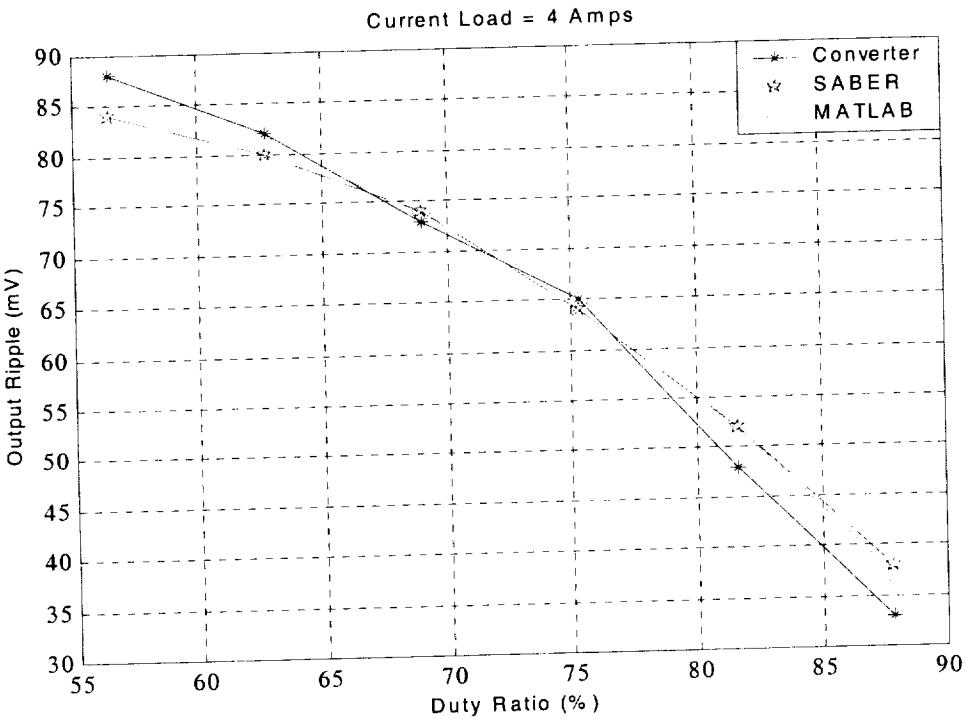


Figure 4-11: Output Voltage Ripple at 4 Amps and 140 V_{dc} Input

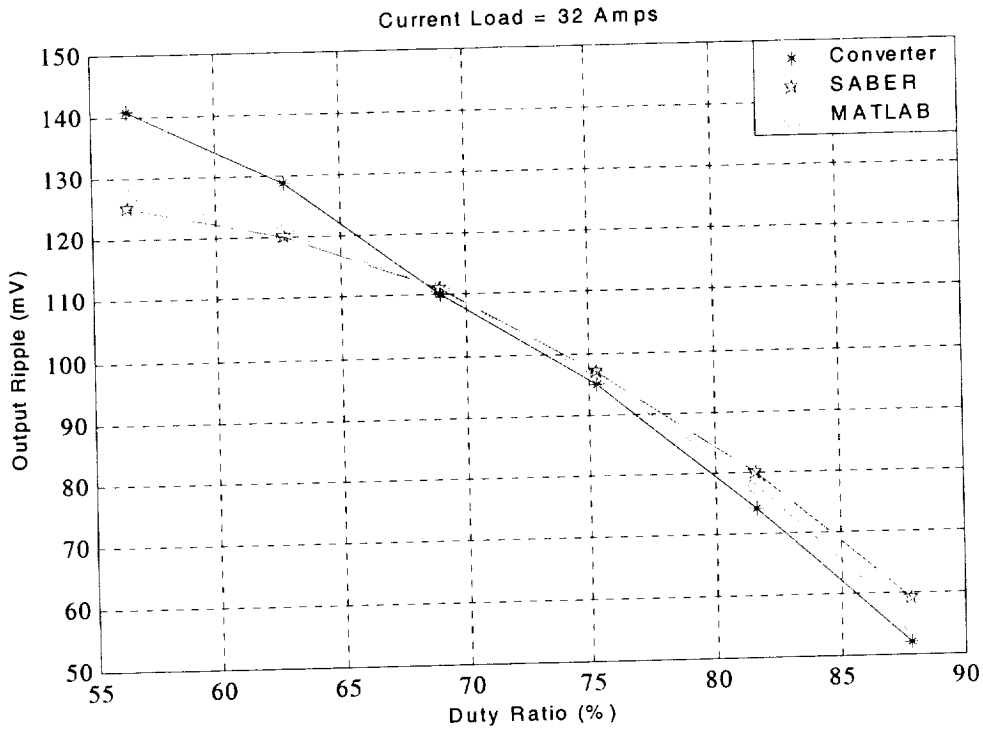


Figure 4-12: Output Voltage Ripple at 32 Amps and 140 V_{dc} Input

4.3 Transient Response

In order to validate the transient response of the models, output voltage converter data was collected for step changes in the duty ratio at different loads and "step" changes in the current load—the quotation marks are due to the fact that the electronic load (HP 6050A) does not step; rather, it ramps to the desired load. This load ramp can be easily modeled. On the other hand, converter data was not collected for "step" change in the supply voltage since the supply behaves as an overdamped system ($1/(\tau s + 1)$). Modeling the supply's dynamics in SABER® would require extra effort and, at this point, it is not warranted. In the SABER® and MATLAB® models, stiff supplies are used. The transfer function model has been configured to simulate supply voltage steps and was accomplished by adding a real-axis pole and *zero* in the Voltage Change transfer function Figure 3-6. The real-axis zero alone does not result in a voltage step transient. Nonetheless, the three models are compared to each other for a step change in the supply voltage (See Figures 4-26 and 4-27).

The parameters V_i, V_f, V_{os}, t_{fv}, t_p, and t_s are used for the qualitative comparison and are defined in Figure 4-13 below. The parameter V_i is the initial voltage, V_f is the final voltage, V_{os} is the overshoot voltage, t_{fv} is the time taken to reach the final voltage (value) for the first time, and t_p is the time taken to reach the peak voltage (value). The parameter t_s is the settling time, which is typically defined as the time taken to reach and stay within ±2% of the final value or $\Delta V = V_f - V_i$. In addition, all the waveforms were manipulated to start at the same initial voltage (value) and step time to facilitate a quick comparison.

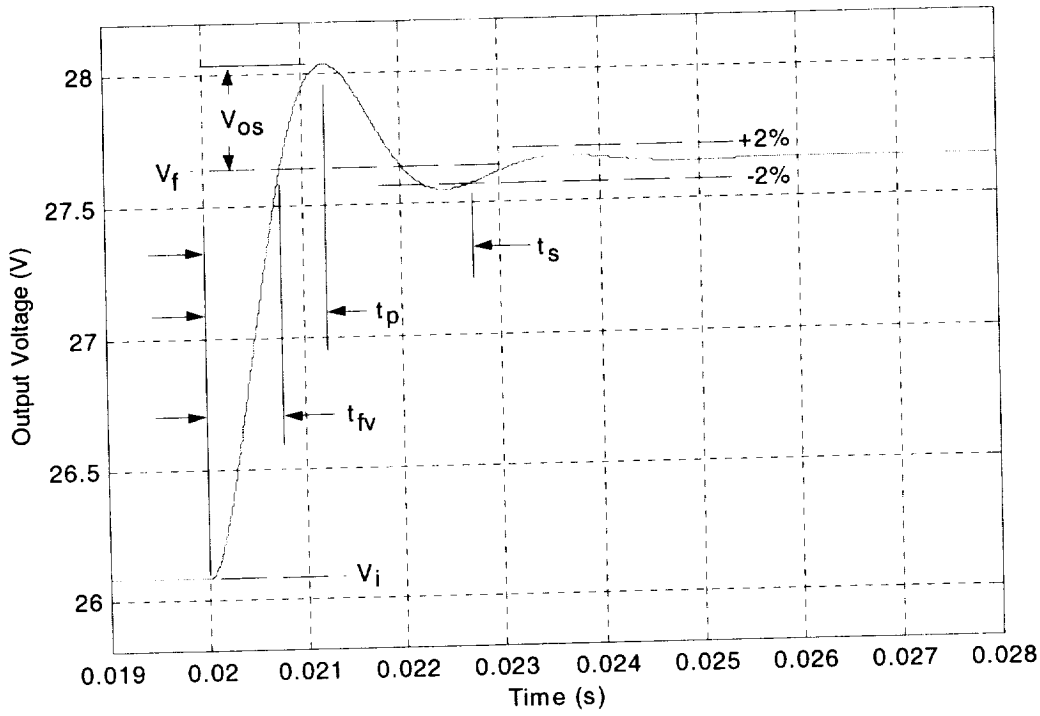


Figure 4-13: Output Voltage Transient Response Parameters

The parameter values for the converter and three models are shown in Table 4-1 for current loads of 4 Amps (Figure 4-14), 20 Amps (Figure 4-15), and 32 Amps (Figure 4-16) and duty ratio step from 70.59% to 74.51%. In terms of t_{fv}, the SABER® model is the best. In going from a 4-Amp load to a 32-Amp load, the time to final value of the converter reduced by 40.9% while ΔV reduced by only 3.06%. The reduction predicted by the SABER® model is 30.6% and 2.11% and that predicted by the MATLAB® model is 24.4% and 4.87%. Both the SABER® and MATLAB® models predict much higher overshoots but only the SABER® model predicts the increase and decrease in overshoot and settling time as the load is increased. The characteristic equation of the improved transfer function model was derived for the 20-Amp load condition and, as a consequence, it yields the best overall result for that case.

	V_i (V)	V_f (V)	V_{os} (V)	t_{fv} (ms)	t_p (ms)	t_s (ms)
Load = 4 Amps						
Converter	27.127	28.664	0.141	1.05	1.55	2.25
SABER	27.127	28.687	0.455	0.98	1.65	4.70
MATLAB	27.127	28.688	0.946	0.82	1.56	7.90
Tfi	27.127	28.696	0.389	0.77	1.23	2.95
Load = 20 Amps						
Converter	26.081	27.603	0.377	0.71	1.23	3.10
SABER	26.081	27.599	0.702	0.77	1.40	5.40
MATLAB	26.081	27.603	0.900	0.72	1.39	6.00
Tfi	26.081	27.650	*0.389*	*0.77*	*1.23*	*2.95*
Load = 32 Amps						
Converter	25.580	27.070	0.350	0.62	1.05	2.60
SABER	25.580	27.107	0.638	0.68	1.25	4.20
MATLAB	25.580	27.065	0.845	0.62	1.22	6.80
Tfi	25.580	27.149	0.389	0.77	1.23	2.95

Table 4-1: Duty Ratio Step Output Voltage Transient Parameter Values

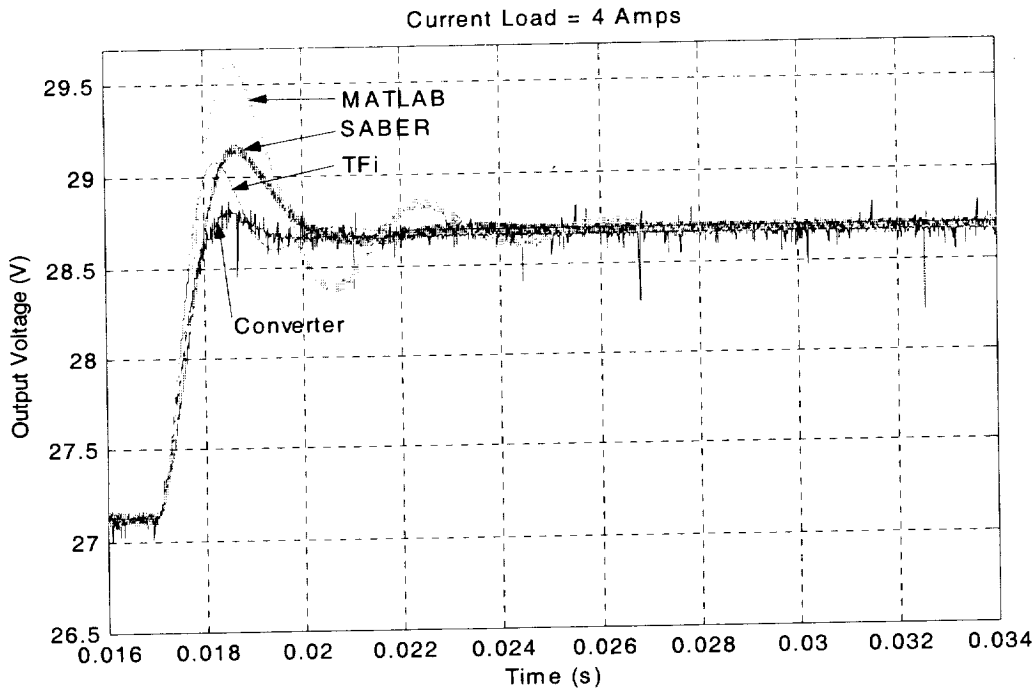


Figure 4-14: Duty Ratio Step Output Voltage Transient at 4 Amps and 120 V_{dc} Input

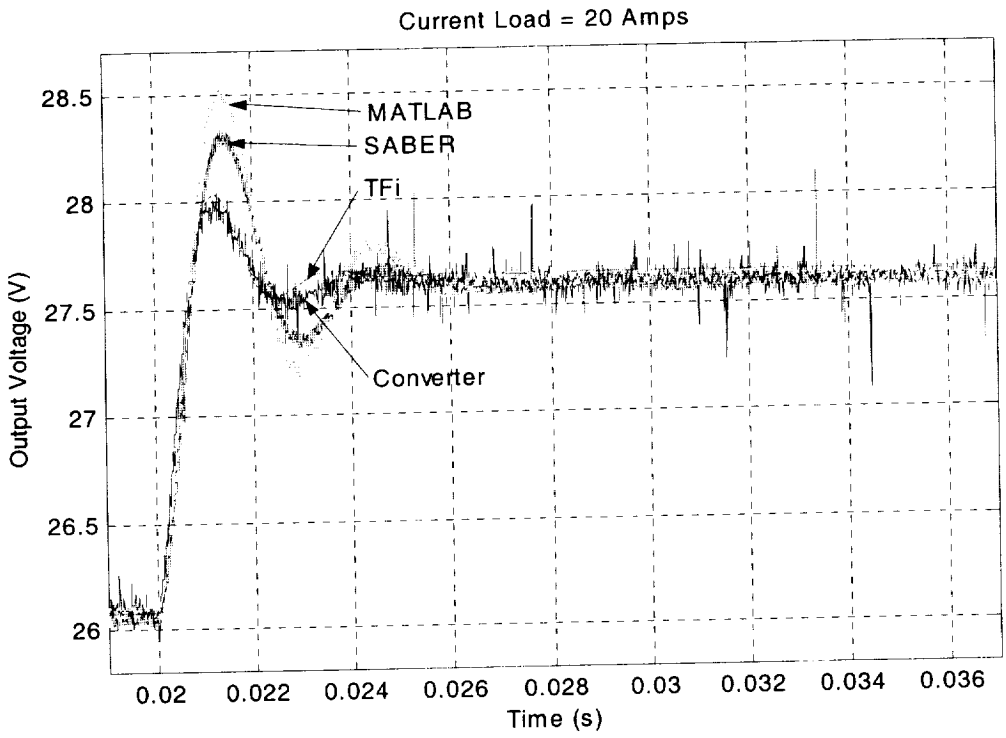


Figure 4-15: Duty Ratio Step Output Voltage Transient at 20 Amps and 120 V_{dc} Input

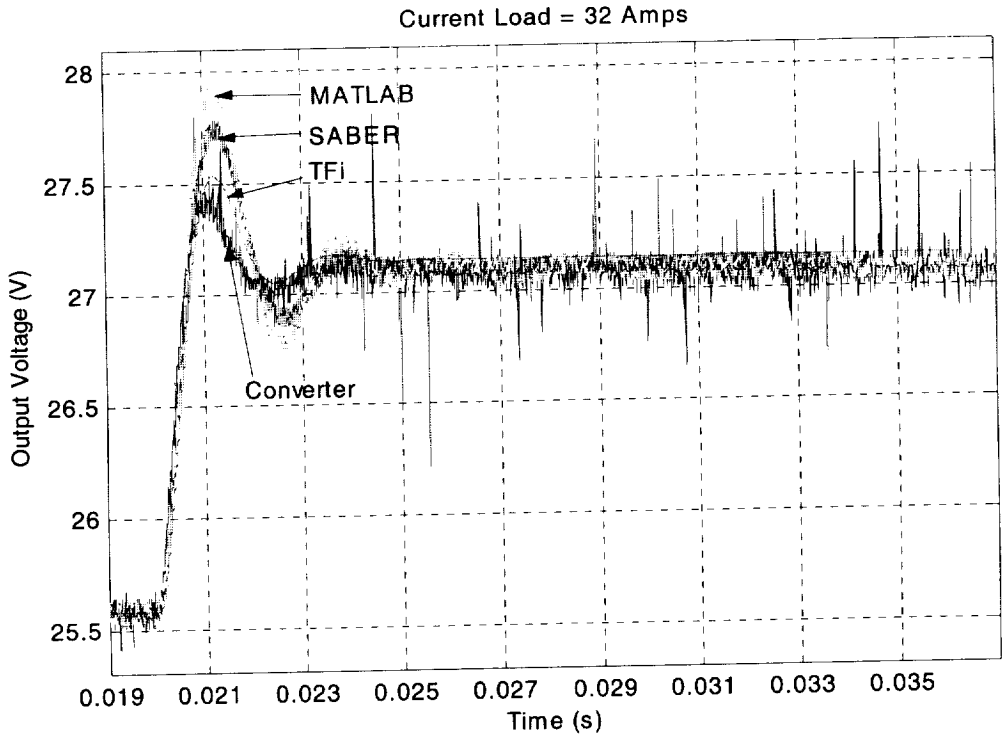


Figure 4-16: Duty Ratio Step Output Voltage Transient at 32 Amps and 120 V_{dc} Input

It is fair to say that of all three models, MATLAB® yields results with the largest discrepancy in terms of overshoot and settling time. Apart from the ignored model components that have been discussed before, the model is simulated with a fixed integration step time (0.7 μs) whereas the SABER® model is simulated with a variable integration step time. Reducing the integration step or applying another fixed integration step algorithm, such as a fourth-order Runge-Kutta algorithm, is not the solution as it will drastically increase the CPU execution time with mixed results. Additionally, if the ignored components are to be incorporated in the model, then a variable integration step algorithm is a must. Other discrepancy-contributing factors are the power supply dynamics (Figures 4-17a and 4-17b) and the electronic load dynamics, which were not accounted for in either model. Figures 4-17a and 4-17b show the input voltage sag for a 20-Amp load and 3.92% (10 Pulse Count) duty ratio step. The power supply reached steady-state after about 0.16 seconds. The input voltage sag is dependent not only on the magnitude of the duty ratio step but also on the load (See Appendix D for resistive load results and note that there is slightly better agreement in the transient results).

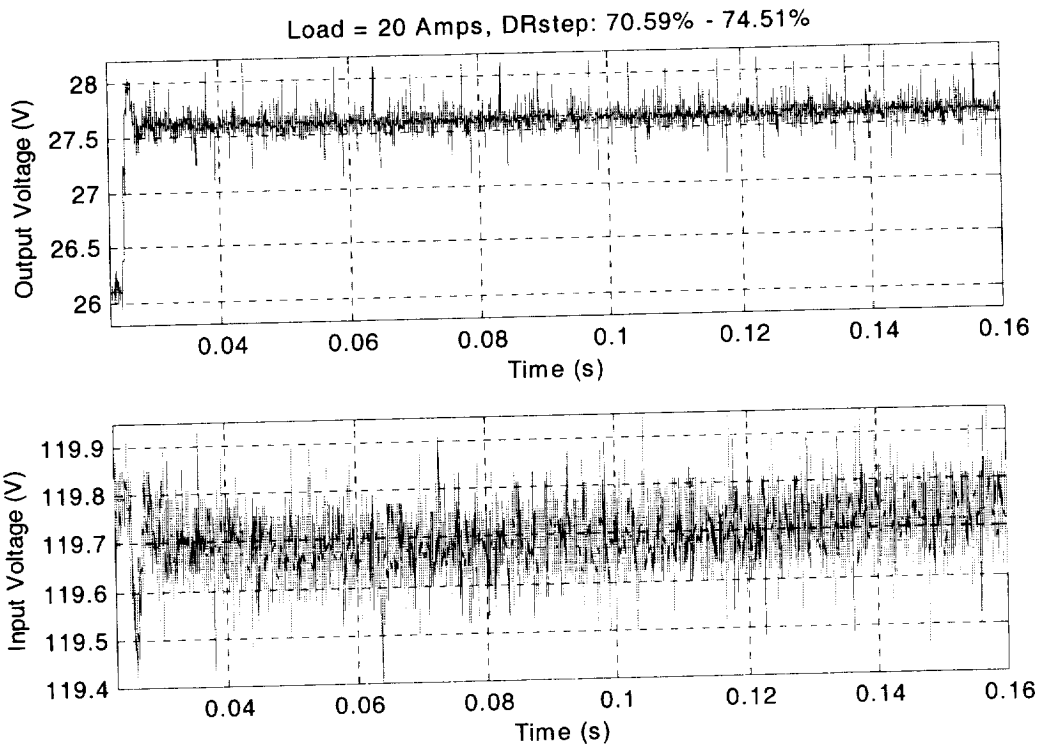


Figure 4-17a: Duty Ratio Step Input Voltage Transient at 20 Amps

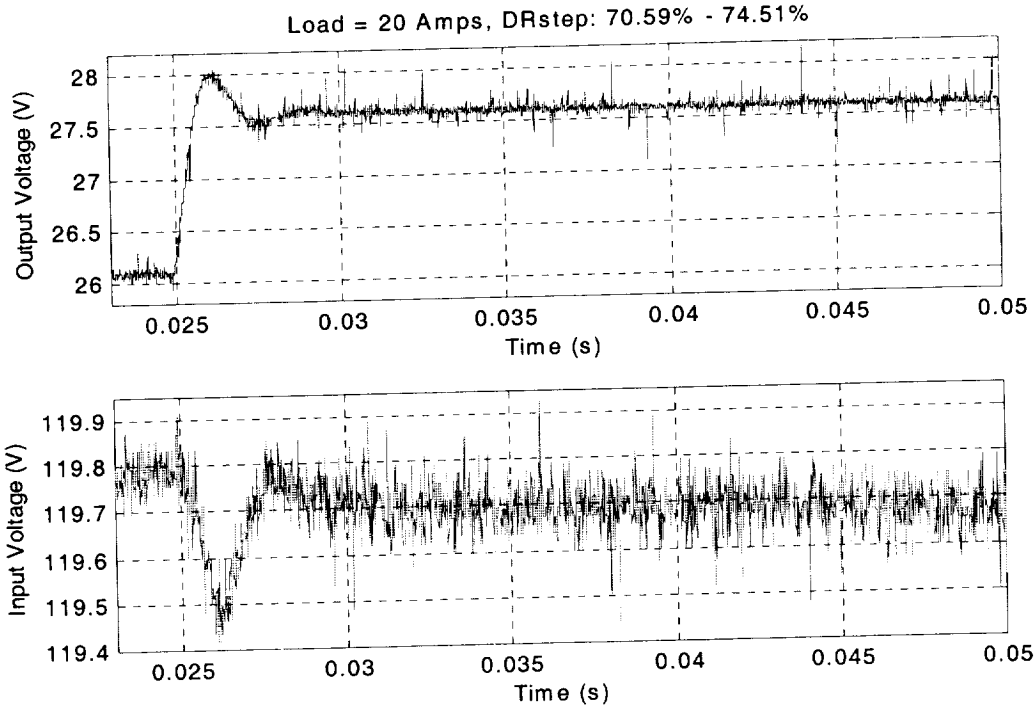


Figure 4-17b: Input Voltage Transient at Reduced Time Range

Current Load Step

As previously mentioned, the current load ramps to the final value and this can be measured as well as modeled. Figures 4-18a and 4-18b show the 4-Amp current load ramp at two different initial values—4 Amps and 20 Amps—as well as the simulation approximation for the SABER® and MATLAB® models. For the current load "step" from 4 Amps to 8 Amps, the simulation ramp time was set at 25 μs and, for the load "step" from 20 Amps to 24 Amps, the simulation ramp time was set at 17.5 μs. In SABER®, the load ramp is accomplished using the "i_pwl" piecewise-linear current source template and, in MATLAB®, it was accomplished with an IF condition in conjunction with following ramp equation (See Appendix B, section B):

$$I_0 = \left(\frac{I_{of} - I_{oi}}{t_{fnl} - t_{st}}\right)t - (I_{of} - I_{oi})\frac{t_{st}}{t_{fnl} - t_{st}} + I_{oi} \tag{4.1}$$

where I_{oi} is the initial current, I_{of} is the final current, t_{st} is the beginning of the ramp and t_{fnl} is the end of the ramp.

A more accurate approximation would be to model the load "step" with a first-order transfer function, but this requires extra time and effort that is not warranted at this time. The data was measured with a TEKTRONIX A6303 Current Probe, AM 503B Current Probe Amplifier, and an HP INFINIUM Oscilloscope. The data was then converted from mV to Amps and the step-time shifted to 0.02 seconds.

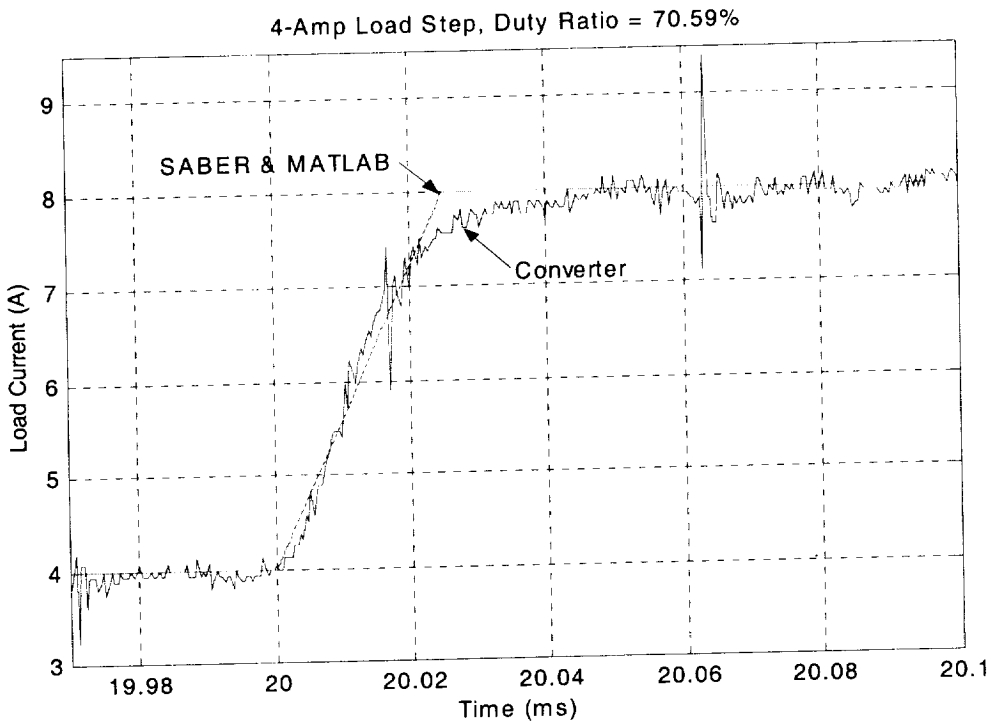


Figure 4-18a: Four-Amp Load Transient Starting at 4 Amps

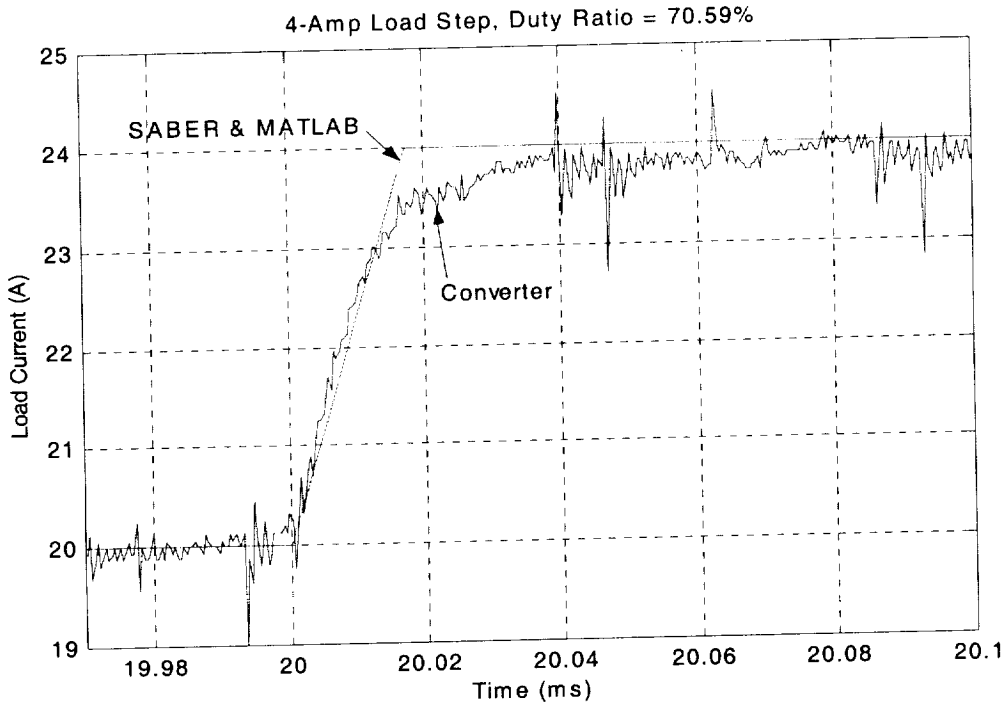


Figure 4-18b: Four-Amp Load Transient Starting at 20 Amps

The parameter results, at a duty ratio of 70.59% and input voltage of 120 V_{dc}, for the converter and three models are listed in Table 4-2 and the output voltage transients are shown in Figure 4-19 and Figure 4-20. As seen from these plots and the table, both the SABER® and MATLAB® models yielded better results for the 20 – 24 Amps load step. Both model are very accurate in terms of voltage undershoot. According to Table 4-2, the DC output impedance of the converter decreased from 0.11 Ω to 0.05 Ω—a 54.5% decrease. The SABER® model predicts a decrease from 0.066 Ω to 0.034 Ω—a 48.5% decrease—and the MATLAB® model predicts a change from 0.023 Ω to 0.033 Ω—a 43.5% *increase*. In terms of percent change, the SABER® model does an excellent job but the predicted DC output impedance is 30% - 40 % lower than actual. The improved transfer function model was tuned to the 4 – 8 Amps load step in terms of time to final value and undershoot. Note that the DC output impedance is set to 0.075 Ω.

Of course, the dynamic of the power supply plays a bigger role than before. The input voltage sag is about 0.45 V (almost twice for the duty ratio step) and it takes about 0.4s (over twice for the duty ratio step) to reach steady-state. This large voltage drop is manifested in the output voltage as shown in Figures 4-21 and 4-22. Based on collected data, the magnitude of the voltage sag seems to be independent of the initial (or final) current load.

	V_i (V)	V_f (V)	V_{os} (V)	t_{fv} (ms)	t_p (ms)	t_s (ms)
Load Step = 4 - 8 Amps						
Converter	27.140	26.690	0.200	0.22	0.68	N/A
SABER	27.140	26.875	0.321	0.18	0.87	N/A
MATLAB	27.140	27.048	0.450	0.03	0.72	N/A
Tfi	27.140	26.840	*0.356*	*0.19*	*0.66*	*N/A*
Load Step = 20 - 24 Amps						
Converter	26.090	25.885	0.325	0.03	0.50	N/A
SABER	26.090	25.953	0.369	0.04	0.70	N/A
MATLAB	26.090	25.957	0.362	0.04	0.67	N/A
Tfi	26.090	25.790	0.356	0.19	0.66	N/A

Table 4-2: Load Step Output Transient Parameter Values

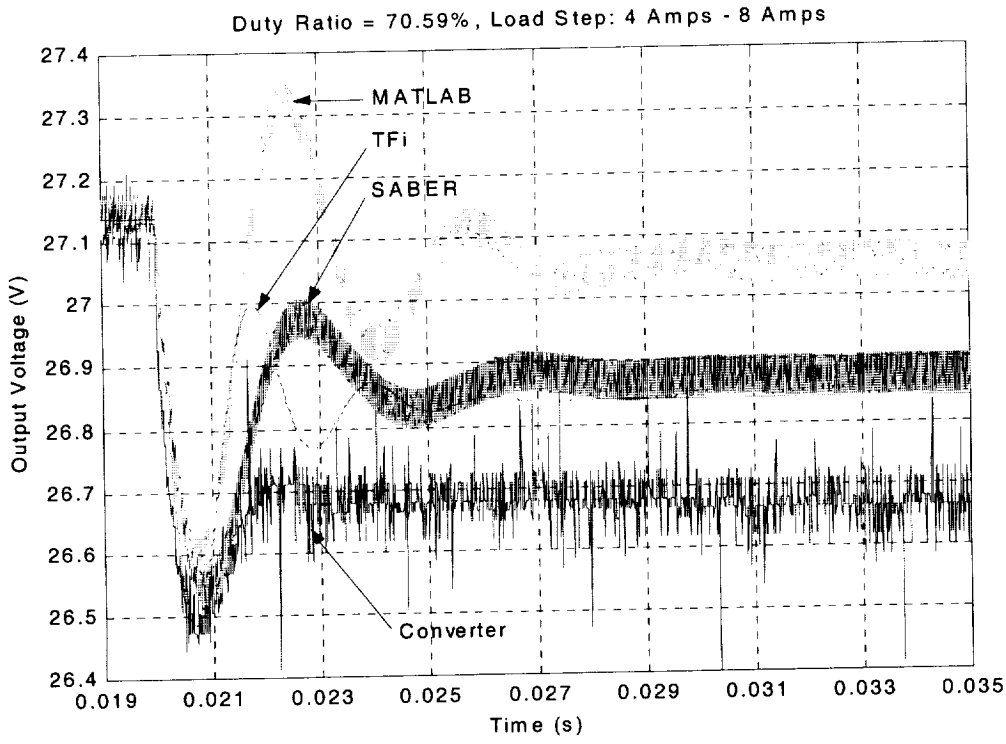


Figure 4-19: Load Step Output Voltage Transient at 70.59% D.R. (4 – 8 Amps)

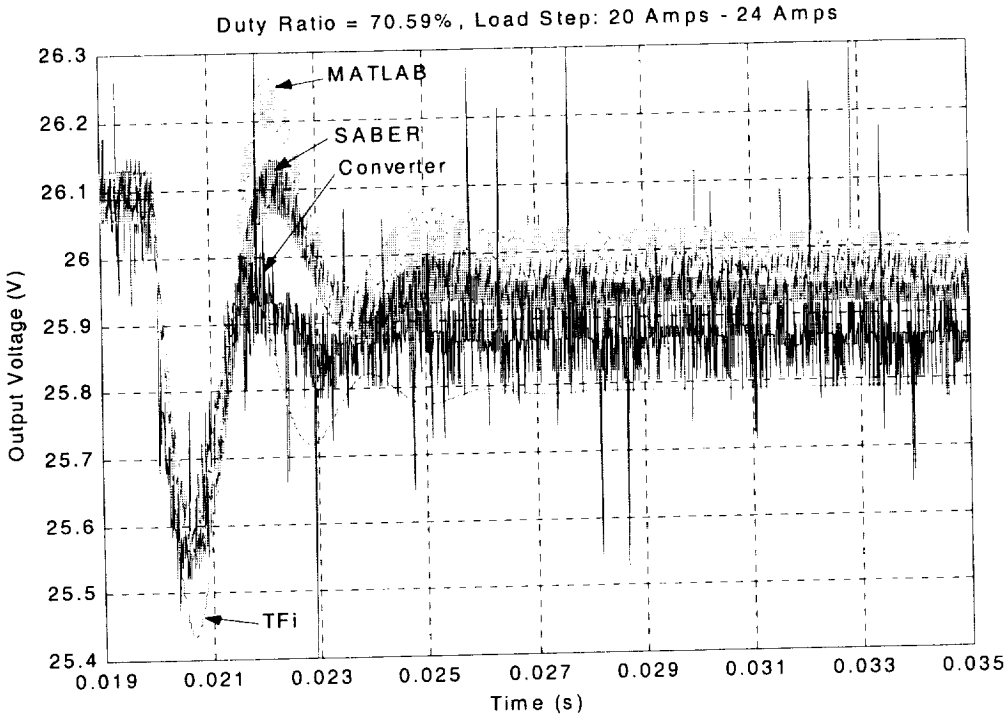


Figure 4-20: Load Step Output Voltage Transient at 70.59% D.R. (20 – 24 Amps)

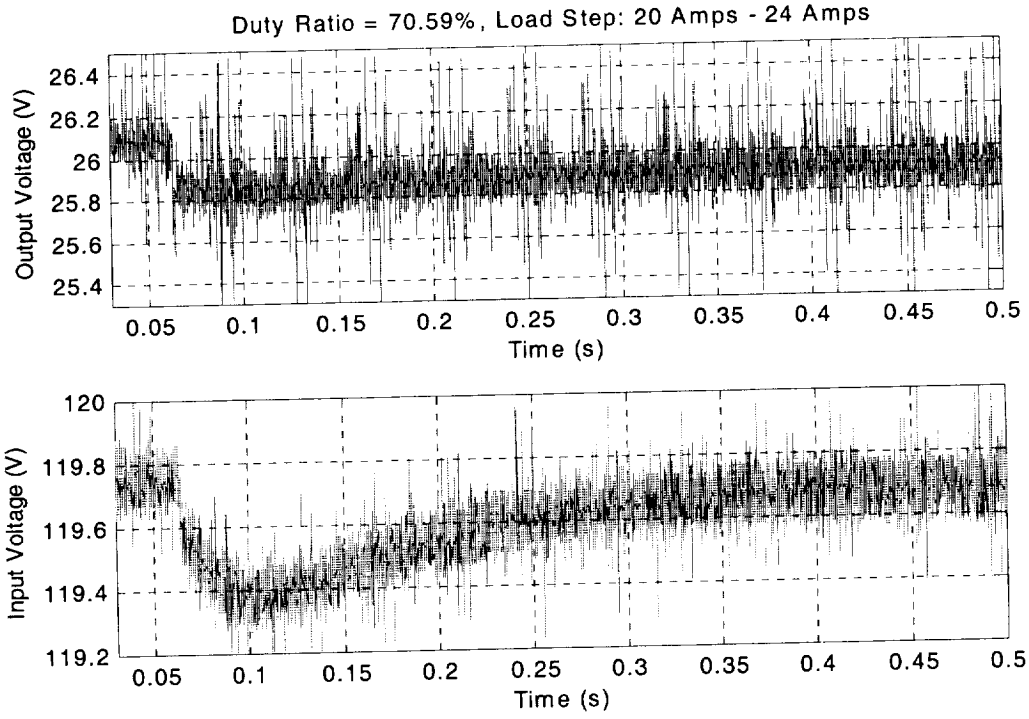


Figure 4-21: Load Step Input Voltage Transient at 70.59% D.R. (20 – 24 Amps)

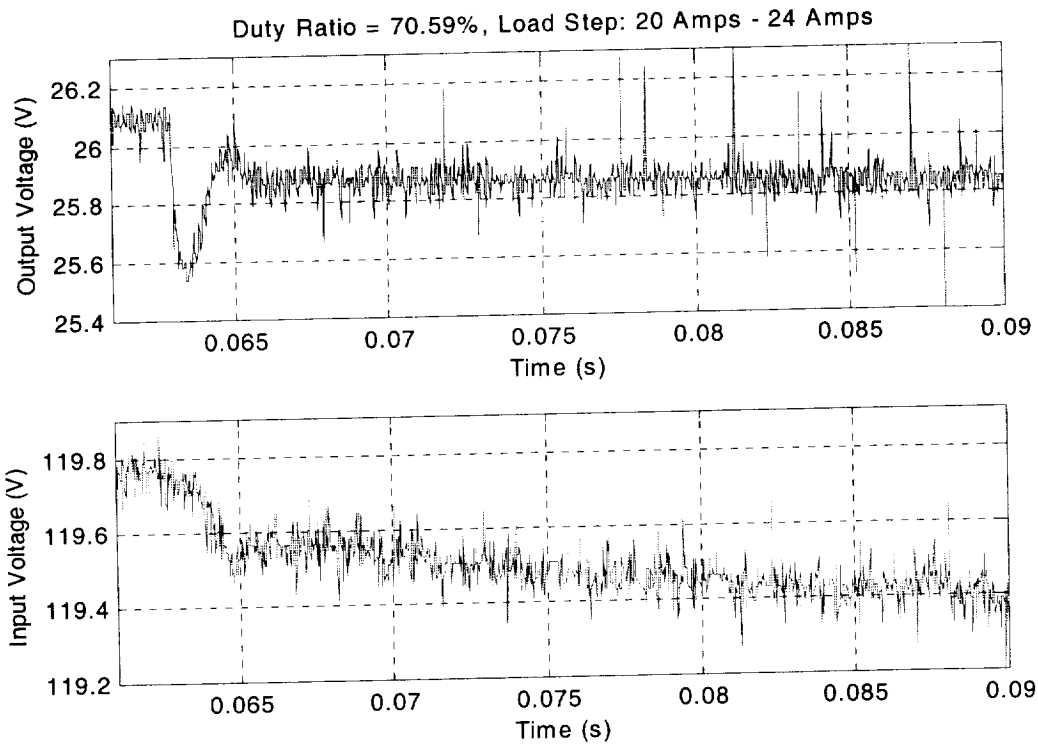


Figure 4-22: Load Step Input Voltage Transient at Reduced Time Range

Input Voltage Step

The third and last transient dynamic to validate is that of an input voltage step but, before proceeding, the output of the power supply must be verified for a step command from 120 V to 126 V. The response is shown in Figure 4-23 with a first-order approximation—equation 4.2.

$$\frac{1}{0.0215s+1} \tag{4.2}$$

Note that the converter response is not first-order as it slightly overshoots.

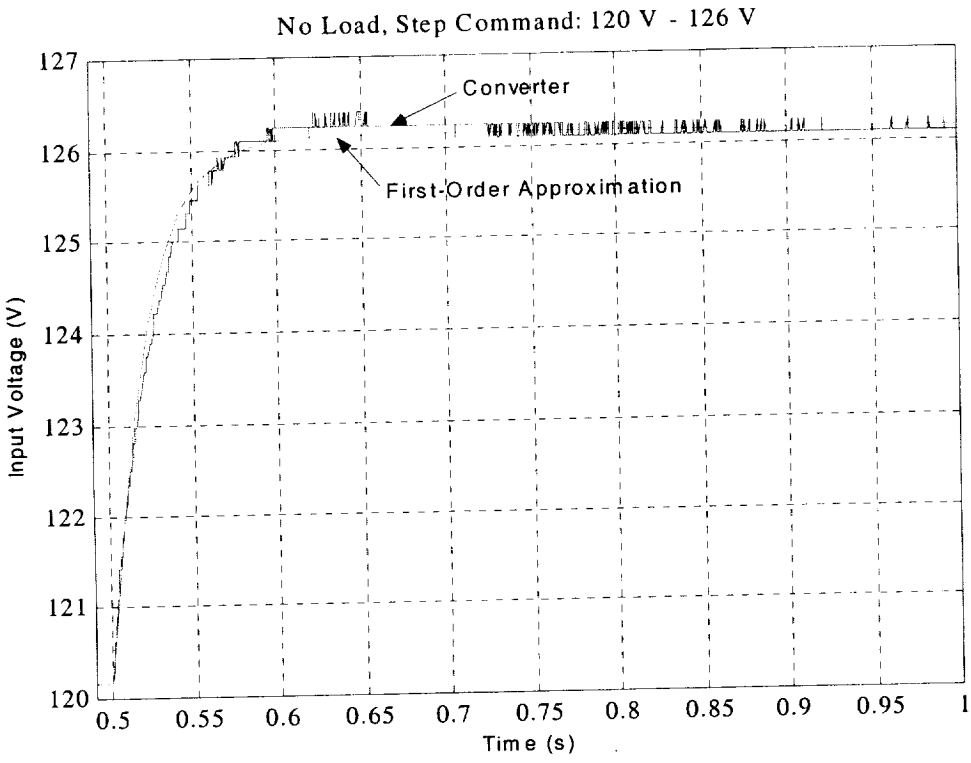


Figure 4-23: Power Supply Response to Step Command with No Load

The response of the converter to an input as in Figure 4-23 is shown in Figure 4-24. This response can be modeled by the following transfer function:

$$\frac{0.333*(1.651e-2s+1)}{(\frac{1}{2\pi*6.366})^2 s^2+2*\frac{0.8}{2\pi*6.366}s+1} \tag{4.3}$$

and is shown as TF in the figure below. The real-axis zero increases the overshoot and reduces the rise time while not affecting the nature of the response, that is, underdamped, damped, and so on.[25] The closer the zero is to the real-part of the poles the greater the effect on the response.

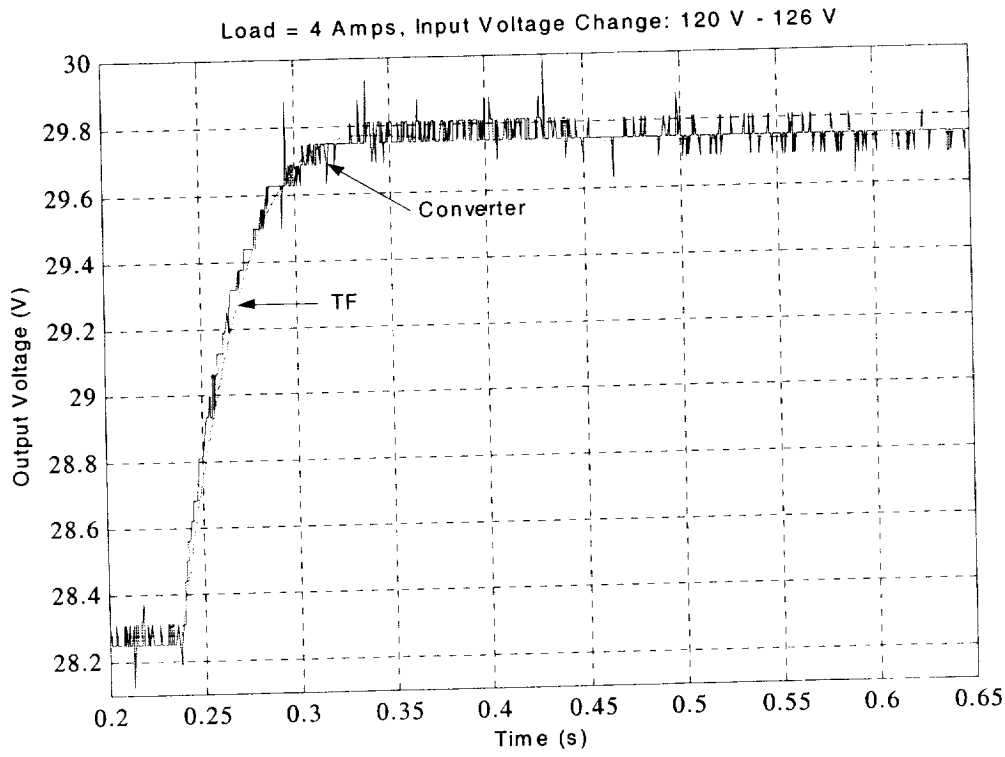


Figure 4-24: Supply Change Output Voltage Transient at 72.94% D.R. (4 Amps)

The real interest lies, however, in the converter response to step disturbances for the purpose of controller design. Now, the system thus far modeled—C(s)/U(s)—is shown in Figure 4-25 where $G_{st}(s)$ is the desired step response. Equating (4.3) to the product of $G_{st}(s)$ and $1/(0.0215s + 1)$ and solving for $G_{st}(s)$ yields (4.4),

$$\frac{0.333*(1.651e-2s+1)(0.0215s+1)}{(\frac{1}{2\pi*6.366})^2 s^2 + 2*\frac{0.8}{2\pi*6.366}s+1} \tag{4.4}$$

Unfortunately, this transfer function does not yield the response to a step disturbance as compared to the response predicted by the SABER® model. The logical alternative is then to derive a transfer function to model the response predicted by the SABER® model as best as possible since the SABER® model, like the MATLAB® model, is nonlinear. The end result, called the Voltage Change TF, is shown in Figure 3.6 and the response shown in

Figure 4-26 and Figure 4-27. The Voltage Change TF was tuned to the 32-Amp load response.

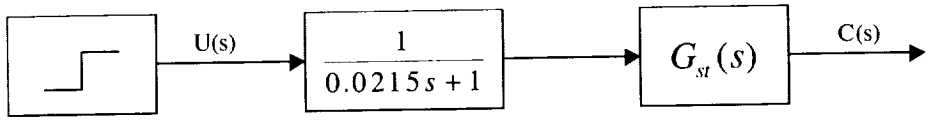


Figure 4-25: Block Diagram of Converter Setup

The parameter results for the three models at a duty ratio of 72.94% and for an input (supply) voltage step from 120 V to 126 V are shown in Table 4-3. Note that the response is load dependent and nonlinear. The fixed step integration algorithm used in the MATLAB model is once again the primary reason for the discrepancy seen between the MATLAB® model and the SABER® model. SABER® simulation with and without the input capacitors but with variable integration steps show negligible or no difference in the transient response. This is also the case for duty ratio step or load step. The main reason why the input EMI capacitors were not used (leakage inductance also) was so that reasonable fixed integration steps could be used.

	V_i (V)	V_f (V)	V_{os} (V)	t_{fv} (ms)	t_p (ms)
Load = 4 Amps					
SABER	28.090	29.548	1.012	1.42	2.22
MATLAB	28.090	29.534	1.586	1.28	2.18
TFi	28.090	29.548	1.149	0.94	1.80
Load = 32 Amps					
SABER	26.830	28.265	1.195	1.15	1.88
MATLAB	26.830	28.210	1.465	1.09	1.85
TFi	26.830	28.287	1.149	0.94	1.80

Table 4-3: Supply Step Output Voltage Transient Parameter Values

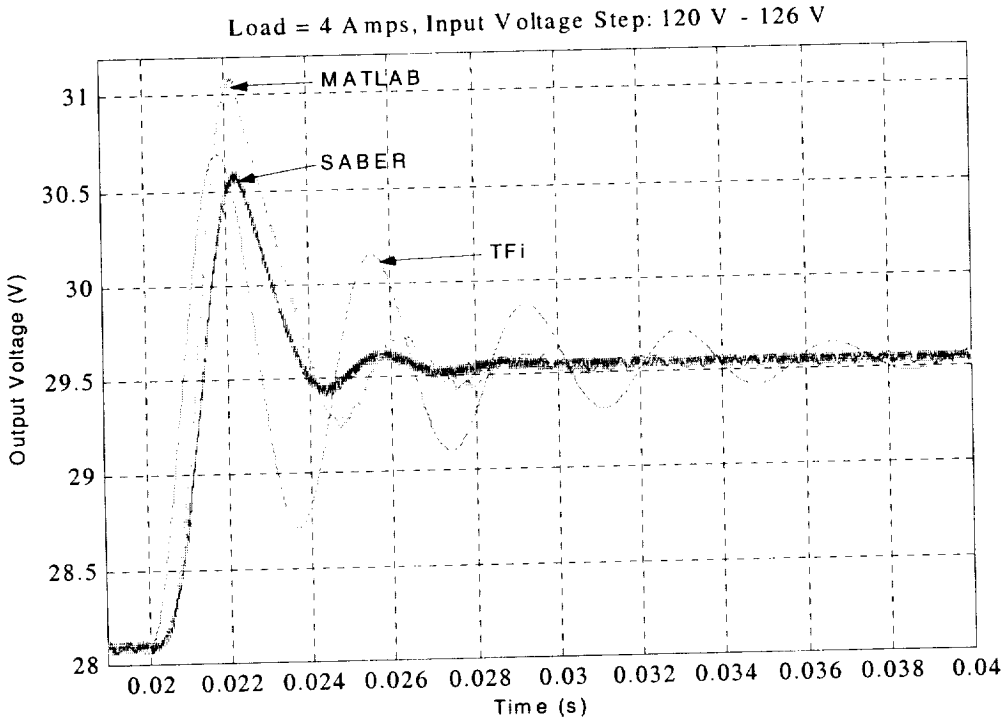


Figure 4-26: Supply Step Output Voltage Transient at 72.94% D.R. (4 Amps)

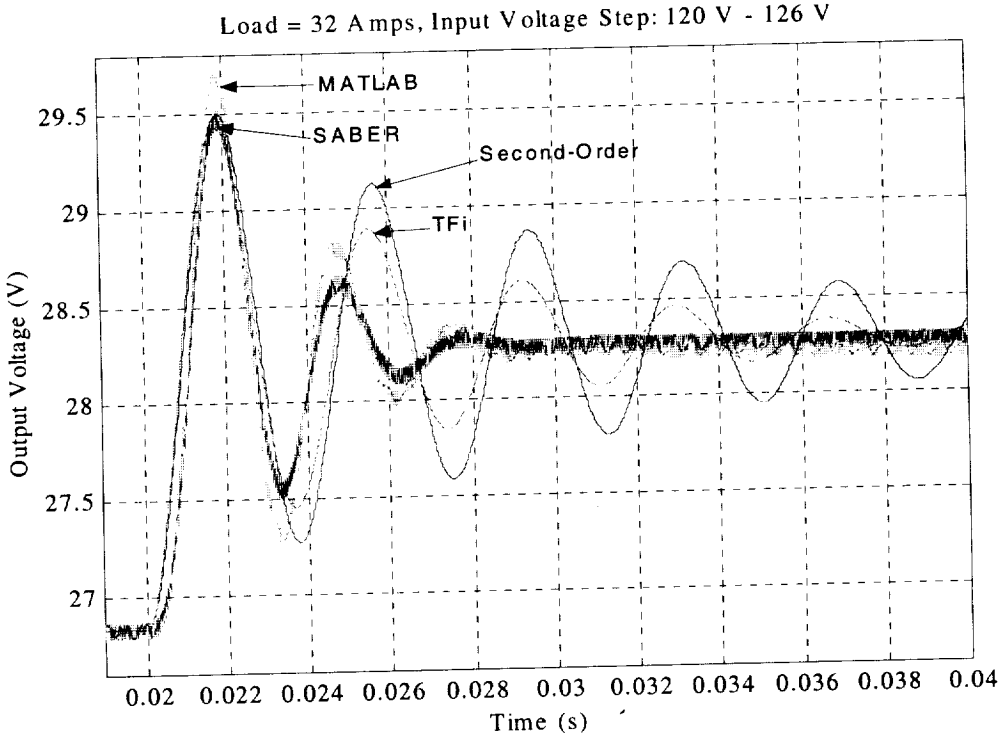


Figure 4-27: Supply Step Output Voltage Transient at 72.94% D.R. (32 Amps)

It must be pointed out that the SABER® model results in Figure 4-26 and Figure 4-27 was obtained with a fixed inductor template. The inductances in the circuit were calculated for the corresponding load. The SABER® model could not simulate with the polynomial inductance templates for input voltage step only. The probable reason is that the inductances were allowed to reach low values (See Figure 2-5 and Figure 2-6) resulting in very large currents through the inductors and terminating the simulation.

4.4 State-Space Average Model Response

Since the state-space averaging technique is not the main focus of this work, although it serves as a benchmark, a quick comparison is made to the MATLAB® (switched) model using the SABER® model as a reference. The steady-state response is compared first then the transient response. The input voltage is set to 120 V_{dc}.

Steady-State

Figure 4-28 and Figure 4-29 show the steady-state comparison for a 4-amp load and a 20-amp load, respectively. For a 4-amp load, the steady-state response of the average model, the MATLAB® model and the SABER model are in close agreement; that predicted by the improved transfer function model is higher. For a 20-amp load, the average model's steady-state response is closer to the SABER® model's response. The steady-state response of the MATLAB® model is slightly higher. It is possible that the simulation had not reached steady-state since a short simulation time was necessary due to the long CPU execution time resulting from the fixed integration step.

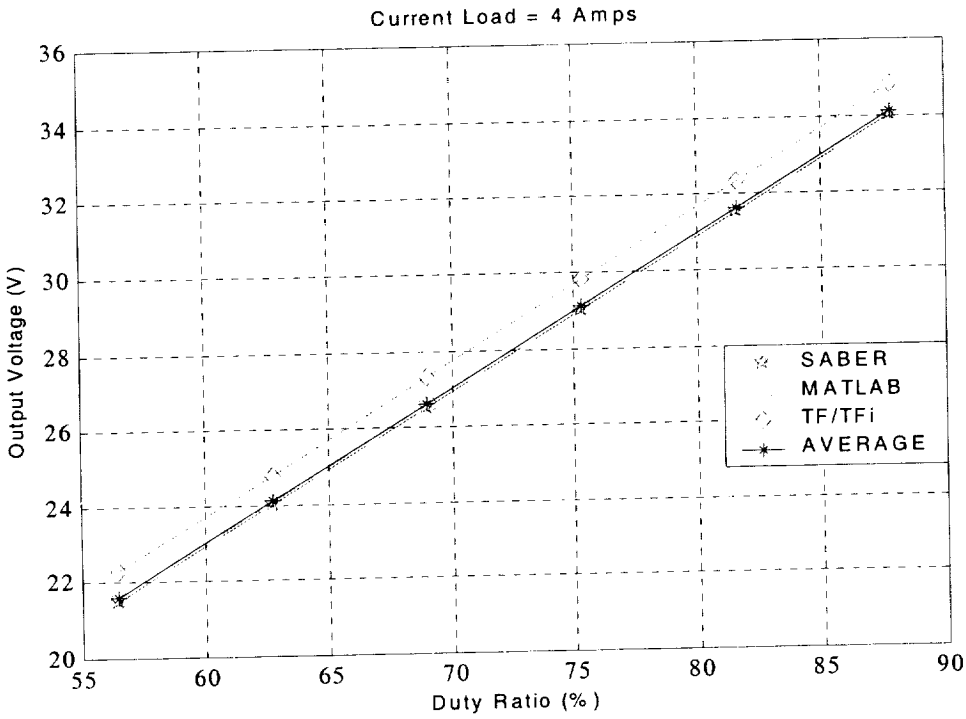


Figure 4-28: Average Model Steady-State Response at 4 Amps

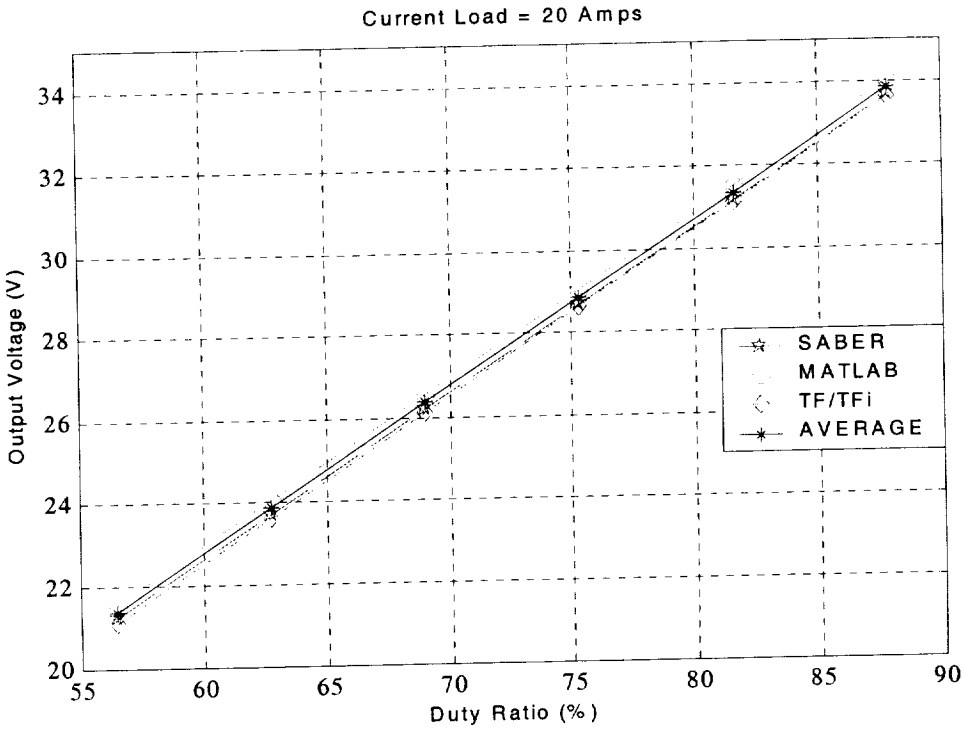


Figure 4-29: Average Model Steady-State Response at 20 Amps

Development of Intelligent PMAD Systems

By Jack Zeller, Dr Zhiqiang Gao, et al

SAE Paper

Abstract

Research studies are being conducted to develop digital control architectures for DC-DC power converters. These "smart" converters will serve as building blocks for intelligent PMAD systems. The "intelligence" objective is focused on PMAD/converter reliability, fault-tolerance, health monitoring, and improved system performance and stability. The study activities to be discussed include: 1) performance results of digital nonlinear control strategies, 2) methodology for efficiently conditioning, isolating, and digitizing converter sensed information, 3) progress on control of multi-module converters, 4) how best to use state-of-the-art digital circuit technology for implementing rugged, compact, easily tuned converter controllers, and 5) evaluation of networking approaches for providing effective, distributed, intelligence to PMAD controls. Planned future research efforts will also be discussed.

Introduction

There exists a need to improve the reliability and operational behavior of complete PMAD systems. To satisfy this need a system control strategy must be developed which achieves "intelligently integrated" PMAD systems comprised of many "smart" DC-DC power converters units. By moving from analog to digital control of power converters, we create an ideal control structure for achieving the smartness needed for fault tolerance, health-monitoring, and the ability to communicate with a PMAD system controller. Intelligent integration can be achieved by communicating each converter unit's digital performance data to a well-designed, possibly hierarchical, digital PMAD system controller.

Toward this goal, the Advanced Engineering Laboratory (AERL) at Cleveland state University (CSU) has, over the past three years, been carrying out R&D activities. This grant work is sponsored by the NASA Glenn Research Center (GRC). The chart of Figure 1 is intended to show the progression of the development objectives which the AERL is pursuing. Beginning with the mandate to develop a power converter digital control architecture, we're providing the groundwork for achieving "smart "power converter controls. To satisfy a wide range of converter output power needs with standardized low power converter modules, work is being done on the control integration needed for a "multi-module" converter unit possessing optimized efficiency as well as fault-tolerance. By developing the methodology for the transfer of performance information from Individual PMAD system elements to each other or to a PMAD supervisory controller, a structure will be in place for developing the control strategies needed to intelligently integrate complete PMAD systems.

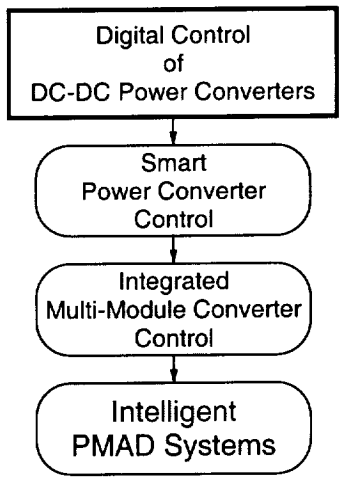


Figure 1- Progression of Development Objectives

DIGITAL CONTROL OF A 1-KW DC-DC SWITCHING POWER CONVERTER

TOMISLAV J. STIMAC

Bachelor of Electrical Engineering

Cleveland State University

June, 1999

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE OF ELECTRICAL ENGINEERING

Cleveland State University

December, 2000

To my Mom and Dad and my sister Maggie

DIGITAL CONTROL OF A DC-DC SWITCHING POWER CONVERTER

TOMISLAV J. STIMAC

ABSTRACT

A digital control algorithm is proposed to control a 1-kW DC-DC switching power converter. The design and simulation of the controller has been completed and is presented using Simulink 3 and the DSPACE rapid prototyping system. The final digital control algorithm was implemented and tested using the ED408043-1 Westinghouse DC-DC switching power converter and the results of the tests will be discussed.

PAGE

0Proceedings of IECEC'01
36th Intersociety Energy Conversion Engineering Conference
July 29–August 2, 2001, Savannah, Georgia

IECEC2001-AT-64

NONLINEAR DIGITAL CONTROL IMPLEMENTATION FOR A DC-TO-DC POWER CONVERTER

Jack Zeller, Minshao Zhu, Tom Stimac and Zhiqiang Gao

The Advanced Engineering Research Laboratory
ECE Department, Cleveland State University
Cleveland, OH, USA 44115

ABSTRACT

Closed loop control studies of a DSP-based H-bridge power converter are discussed. The experimental test facility and the analytical development tools being used are described. Open loop modeling results for the NASA-provided power converter test unit are summarized. The performance benefits of nonlinear control algorithms, readily implemented in DSP software, are discussed. Technology issues, specific to the CSU digital control structure are identified and their ongoing development studies are discussed.

INTRODUCTION

Cleveland State University is involved with research to study how the application of direct digital control to spacecraft power converters could enhance their performance and reliability as well as that of complete power management and distribution (PMAD) systems [1]. The work is being conducted by a team of faculty members and students (both graduate and undergraduate) from CSU's Electrical and Computer Engineering (ECE) department. This paper is intended to provide an overview of the entire research activity while focusing on early closed loop control performance results that have been obtained using nonlinear controller algorithms.

To provide background information for readers, the paper begins with a discussion of the objectives for conducting this specific research. This is followed by a brief description of the experimental facility being used to conduct the research. A more detailed description can be found in [2]. Next there will be a description of how the facility was used to experimentally determine linear model representations of the power converter provided by NASA for this program. These linear models [3] serve as the basis for analytically studying a variety of closed loop voltage regulation control strategies. These strategies involve nonlinear control laws that depend upon a digital controller's computational capabilities for their implementation. The main section of the paper will present initial performance results obtained with these digital control strategies. Both simulation and hardware test results will be included and discussed. The final section will: 1) discuss power converter digital control technology areas which warrant further study, and 2) describe DSP control hardware development activities intended to provide tools for broader PMAD control investigations.

RESEARCH OBJECTIVES

Reliable, efficient, well-regulated DC-to-DC power conversion equipment is critical for mission success on most space platforms. As platforms, especially manned spacecraft, become more sophisticated, reliable operation of complete power management and distribution (PMAD) systems becomes a must. As a result there is much interest in determining how and in what areas a more intelligent and robust control structure might be of value. Replacing the present analog control solution with a digital computer based control is one approach toward satisfying this need.

Much work in digital control of DC-to-DC power converters has already been accomplished and documented ([10]-[16]). Either microcontroller-based or DSP-based approaches have been used to realize sophisticated and/or flexible control algorithms, such as PID, Fuzzy Logic, Adaptive Fuzzy, and Feedforward Control. The versatility provided by software programmable digital controllers is well suited to the increasing control performance and reliability demands being placed on new space borne power converters and PMAD systems. Applying previous experience by CSU researchers on highly nonlinear control strategies [5] is the focus of the work to be reported in this paper. One objective of the CSU research will be to evaluate the closed loop performance benefits that

these new nonlinear algorithms can bring to DC-to-DC power converters. In addition our DSP-based research will be conducted so as to evaluate a multitude of control opportunities that can only be accomplished digitally. One example will be the ability to use variable PWM frequency as a method of improving low power converter efficiency. Early results of our studies as well as details of the multitude of ongoing efforts will be discussed in the following sections.

CSU RESEARCH FACILITY

In order to provide an effective research environment, CSU's ECE department allocated one of its laboratories to this project to function as a combined laboratory and office in which to conduct this research. This facility has been designated as the Advanced Engineering Research Laboratory (AERL). In order to conduct realistic experimental research, NASA provided to CSU a Westinghouse-designed 1 KW "brassboard" power converter. This SMPS unit was designed to accept an input voltage between 100 and 160 volts DC and provide a regulated and isolated output DC voltage of 28 volts for loads up to 36 Amps. Galvanic voltage isolation was obtained with a stepdown (3:1) transformer whose primary winding was pulse-width-modulated (PWM) with an H-bridge switching configuration of power MOSFET transistors. The lower voltage secondary winding was rectified and filtered to provide the 28 volt DC output. Pulse-width-modulation (PWM) of the switching devices was used to accomplish closed loop voltage regulation. The analog PWM generation circuitry and analog controller circuitry were removed, since the intent of the research is to accomplish these two functions digitally.

It was decided that a DSP-based digital system would be used rather than a microcontroller approach. Equipment needed to support this approach was put into place and configured to realize a versatile research environment. The DSP development system selected was dSpace Inc.'s [4] rapid-prototyping development system. This system is equipped with a high-performance TI DSP chip, A/D conversion capability as well as digital I/O circuitry. To expedite the development and evaluation of digital control strategies, Mathwork's Matlab/Simulink/Real-Time Workshop toolbox software was selected. Simulink provides the ability to model and accurately simulate the transient performance of dynamic processes to arrive at a set of acceptable closed loop control strategies. Mathworks' Real-Time Workshop will convert a controller, modeled in Simulink, into 'C" code which will run on dSpace's DSP processor to control actual experimental hardware (in this case the 1 KW Westinghouse power converter). This is termed hardware-in-the-loop simulation. The control laws can also be programmed in native "C" code. Then dSpace's compiler and libraries will be used to generate the code for the TI DSP chip on dSpace's processor board. This second approach has been found to generate faster operating real-time control code.

A decision was made early in the program to generate the two-phase PWM signals needed by the H-bridge outside of the DSP by using a programmable CPLD chip. This will off-load a potentially heavy computational burden from the DSP controller. A block diagram of this experimental configuration is shown in Figure 1.

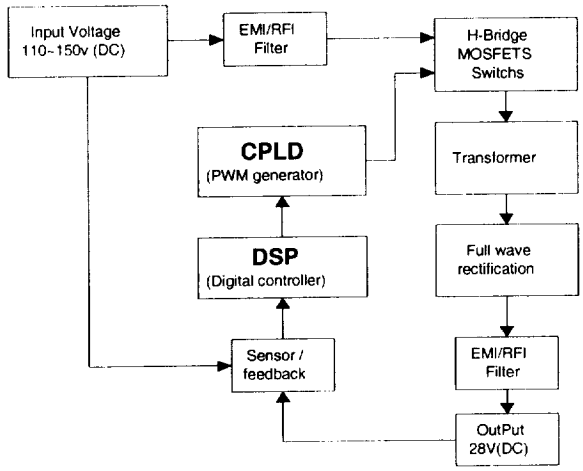


Figure 1: Experimental Facility Block Diagram Description

To complete the experimental research facility, appropriate test equipment was acquired. This included: power supplies, signal generators, digital voltmeters, digital oscilloscopes, and an electronic load bank. The photograph in Figure 2 shows how this array of equipment is configured in our facility. A more detailed description of all of this equipment is included in [2].

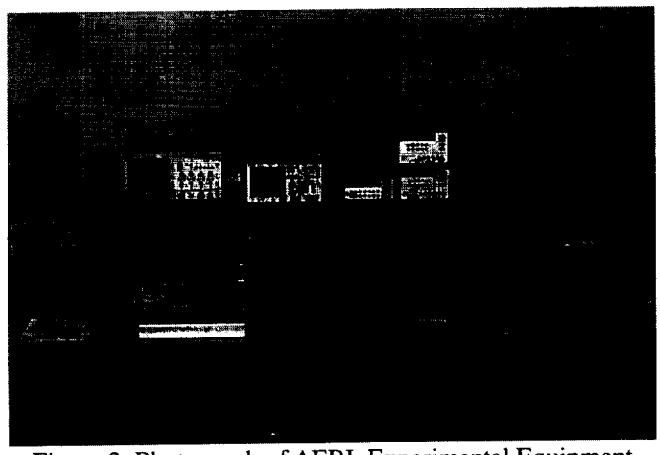


Figure 2. Photograph of AERL Experimental Equipment

Linear Model Development

To expedite the analytical control development studies, a linear (transfer function) model of the power converter process was developed. Obtaining the data for this model was the first research activity which used the AERL experimental hardware-in-the-loop configuration. A methodology for using a CPLD device to generate the PWM signals needed to drive the switching converter's gate circuitry was developed. The DSP's algorithmic logic needed to accept a variable pulse width control input and compute the outputs for the CPLD's input

registers was configured for evaluation in Simulink. The RTW toolbox was used to convert the simulated algorithmic logic into dSpace's DSP "C" code equivalent [6].

It should be noted that the initial design of the algorithm chose an eight (8)bit quantization level for each phase of the CPLD's PWM output. Thus the 28 volt DC output could only be resolved to 0.156 volts, (at 120 volts of input). This quantization has proven to be a performance limitation to the control studies and improvements are being evaluated. A brief discussion of the early results of these improvements will be presented later in this paper.

Using the hardware-in-the-loop experimental configuration of Figure 1 and the just-described PWM generation software, linear model data was obtained. By varying the input pulse count, the converter's open loop steady-state performance under varying output current (I_l) load levels and for a range of input supply voltages (V_{in}) was determined. The detailed results of this steady-state mapping can be found in [3]. Using those results, an equation was determined [3] which analytically defines the converter's steady-state output voltage over a range of conditions.

$$V_o = \frac{V_{in}}{(3*256)}(PulseCount) - 0.8 - (0.075 * I_L) \quad (1)$$

In (1) the division of the input voltage by 3 accounts for the 3:1 turns ratio of the isolation step-down transformer. The 256 factor is the maximum pulse count due to the eight bit quantization used in the initial design. The 0.8 volts accounts for the rectifier's diode drop while the 0.075 is the approximate output impedance of the converter under load. Eq.(1) can be rearranged to yield a pulse count value which would be needed to produce a particular output voltage knowing the input DC voltage and the load current. This relationship is defined as (2) below: (2)

$$PulseCount = [V_o + 0.8 + 0.075 * I_L] * (3 * 256) / V_{in}$$

The next experimental modeling activity of [3] was to determine the transient behavior of the converter process when subjected to disturbance inputs in: 1) pulse count, 2) load current, and 3) input DC supply voltage. Step inputs in each of these three parameters were used to produce time response data. Curve fit approximation's to this data were used to determine linear transfer function models. The details of the testing activity are included in [3]. The result of this activity was the open loop process transfer function block diagram of Figure 3.

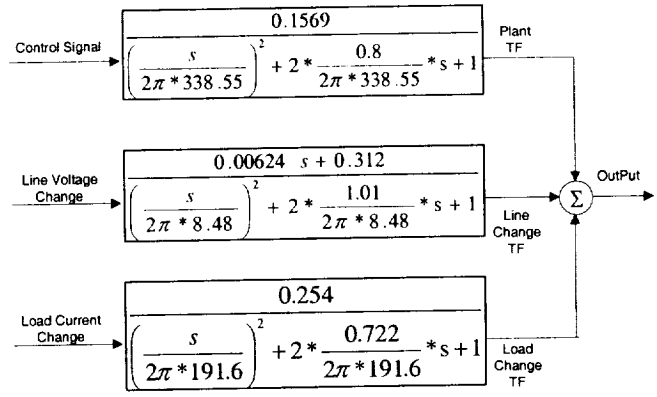


Figure 3 Linear Model Block Diagram

CLOSED LOOP PERFORMANCE RESULTS

Simulink Setup -The results of the simulation studies were obtained using a detailed Simulink model of the digitally controlled converter. The simulation includes the open loop converter model of Figure 3. A comparison of a traditional linear PID control and a nonlinear control (NPID) was performed. Figure 4 shows this model and includes blocks for the two control laws as well as a soft-start feature. Figures 4a-4d are block descriptions of the Simulink subsystems of Figure 4.

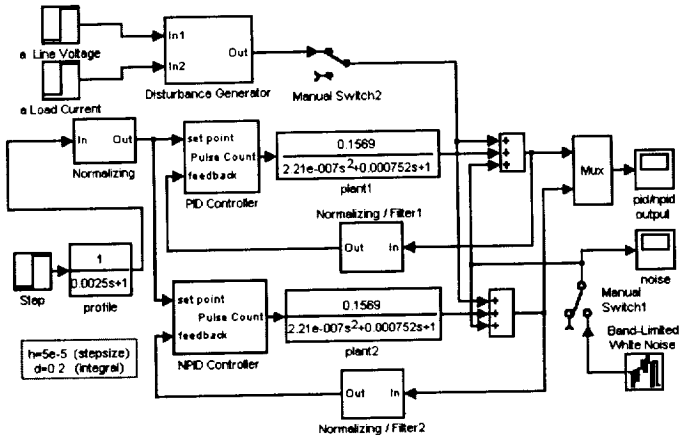


Figure 4 Simulink Simulation Block Diagram

We use a zero-order hold with a sampling period of 50us. The quantizer is used to mimic the dSpace's 12 bitA/D converter. It is set 0.0048828125.

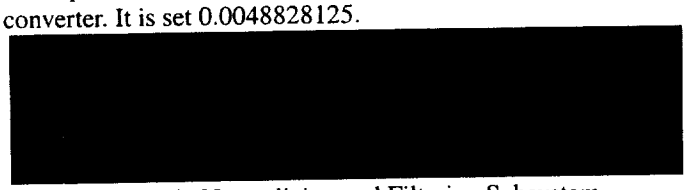


Figure 4a Normalizing and Filtering Subsystem

After comparing the setpoint and feedback signals, the control algorithm is executed and a control signal is produced. It is then converted to a pulse count and sent to the PWM

generator to create real PWM control signals for the switching MOSFET's gate drivers.

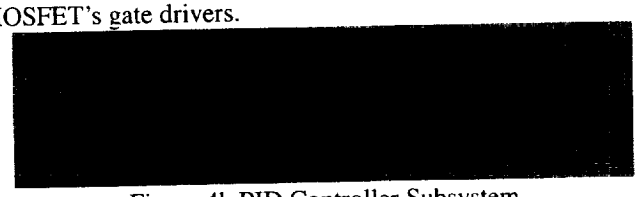


Figure 4b PID Controller Subsystem

The disturbance block in Figure 4c is used to simulate the effects of the Line voltage change and Load current change on the output voltage. It allows us to observe the disturbance rejection performance for each controller. It comes from Figure 3.

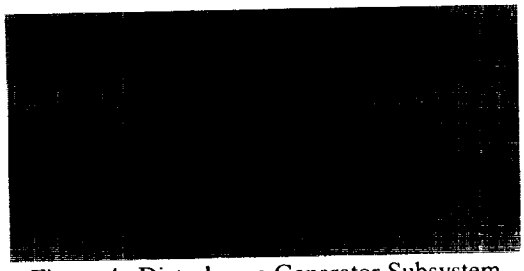


Figure 4c Disturbance Generator Subsystem

The Conversion to pulse count block is shown in Figure 4d, where the saturation limits are set at 0 and 1, respectively. This is because our PWM generator range is 0 - 240 pulse counts. The quantization level in the Quantizer is set at 1.

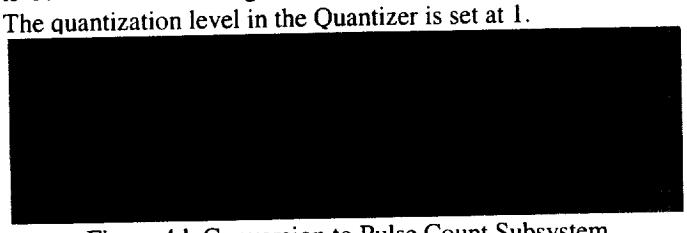


Figure 4d Conversion to Pulse Count Subsystem

NONLINEAR PROPORTIONAL AND INTEGRAL CONTROL

The initial results of the digitally controlled converter using standard Proportional-integral (PI) controller, which is primarily an integral control, have been reported in [2]. Recently, a set of high performance Nonlinear PID control algorithms have been reported [5] and some of them are used here as shown in Figure 4e, which shows the details of the Nonlinear PI block in Figure 4.

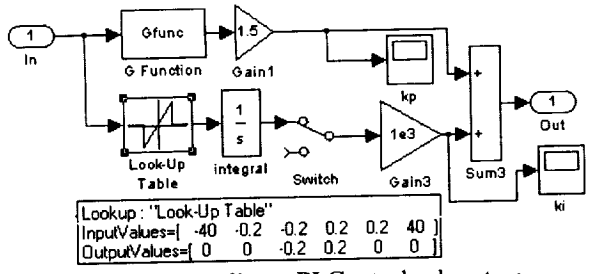


Figure 4e Nonlinear PI Control subsystem

The G-function is Figure 4e is a nonlinear gain function shown in Figure 5, where the green line is normal linear gain and the blue line represents the nonlinear G function. The design philosophy is fully explained in [5]. Here, the intuition is that the gain should be higher when the error is smaller, which makes the controller "more stiff". That is the proportional control is made more sensitive to the small errors. This will also reduce the reliance on the integral control to eliminate steady state errors. Note that the instability is often caused by the 90 degree phase lag in the integral control.

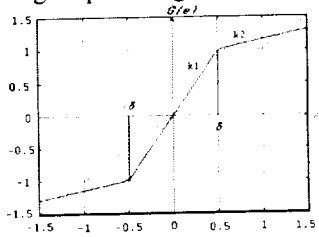


Figure 5 G-Function

It is mathematically expressed as:

$$G(e) = \begin{cases} k_2 * e + (k_1 - k_2) * \delta * \mathrm{sgn}(e) & |e| > \delta \\ k_1 * e & |e| \le \delta \end{cases} \quad (3)$$

Although the use of this nonlinear gain provides good disturbance rejection and stability robustness, it may make the controller too sensitive to noise. Therefore, a compromise is made between the nonlinear proportional control and a limited nonlinear integral control. In particular, the integral term is reformulated as

$$k_i * \int G_i(e)dt \qquad G_i(e) = \begin{cases} 0 & |e| > \delta_i \\ e & |e| \le \delta_i \end{cases} \quad (4)$$

That is, the integrator only integrates when the error is "small", typically when the output is within 10% of the set point.. This design strategy allows the control to effectively avoid undesirable overshoots and the integrator wind-up during large disturbances.

Simulation : Transient Results – Figure 5 show a comparison of the transient performance simulation results obtained for a well-tuned linear PID versus a nonlinear PI control. Figure 5(a) contains results for the application of a 20 Amp load while Figure 5(b) shows results for PWM pulse Count (control variable) respectively. The blue curves are for the PID and the green traces are for the nonlinear PI. The nonlinear controller shows a much smaller deviation from steady-state than the linear PID. Also the nonlinear algorithm is faster.

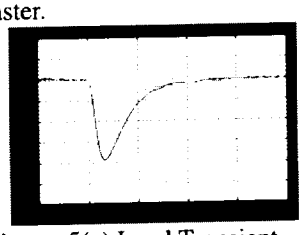


Figure 5(a).Load Transient

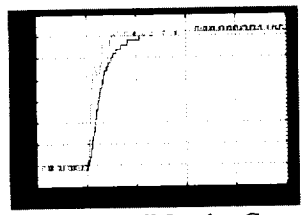


Figure 5(b)PWM pulse Count

Experimental Controller Setup –The two control algorithms were then coded in native "C" code , compiled and down loaded to the DSP system. This code could then operate the converter hardware. After extensive experimentation and tuning activity, transient performance comparison results were obtained. The use of dSpace's Control Desk software helped expedite this tuning activity. Figure 6 is a sample of what the computer screen looks like when Control Desk is employed. The designer has a great deal of critical parameter information available at a glance along with the captured transient data. Even though the actual values may not be readable in the paper, the figure is included to show the capability of the Control Desk software for enhancing productivity.

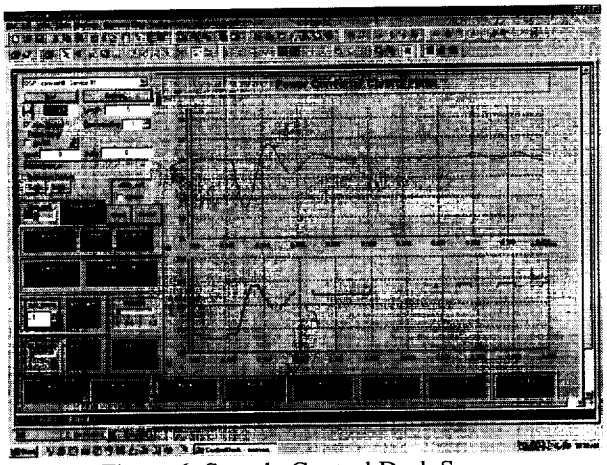


Figure 6 Sample Control Desk Screen

Experimental Transient Results – The transients caused by a sudden change in the load current were captured as was done during the simulation studies.

We used the Control Desk to assist the controller tuning and transient response monitoring. In the following figures, the top trace is Output Voltage, the lower trace is CPLD PWM pulse Count (control signal).

In the hardware test, the load current was changed from 3A to 20A. The lowest load is set 3A so that the inductor in the converter is in continuous conduction.

1) Linear PI (LPI) Controller results

The parameter for Linear PI Controller setting are :

$$K_p = 0.2, K_i = 423$$

And the response is shown in Figure 7, which indicates a 15.2ms recovery time and 3.4V peak-to-peak voltage variation.

2) Two-slope Nonlinear PI (NPI) Controller results

According to (3) and (4), the parameters for the NPI controller setting are set as

$$K_1 = 0.256,\ K_2 = 0.024, \delta = 0.4$$

$$K_i = 400, \delta_i = 0.8$$

and the response is shown in Figure.8, which yields a 5.7 ms recovery time and a 3.25V peak-to-peak voltage variation. Comparing to Figure 7 the NPI transient response performs almost 2 times better on the recovery time.

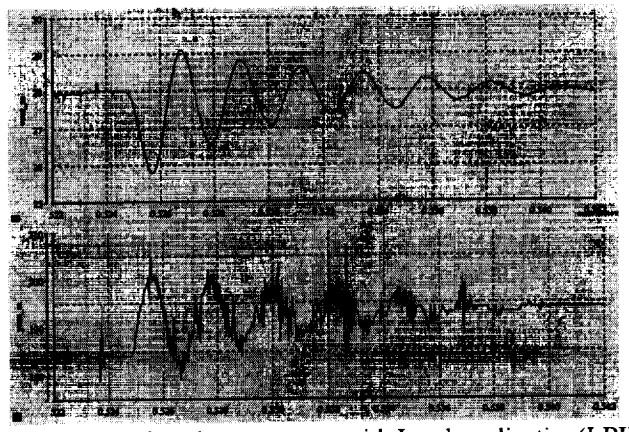


Figure 7 Transient response with Load application(LPI)

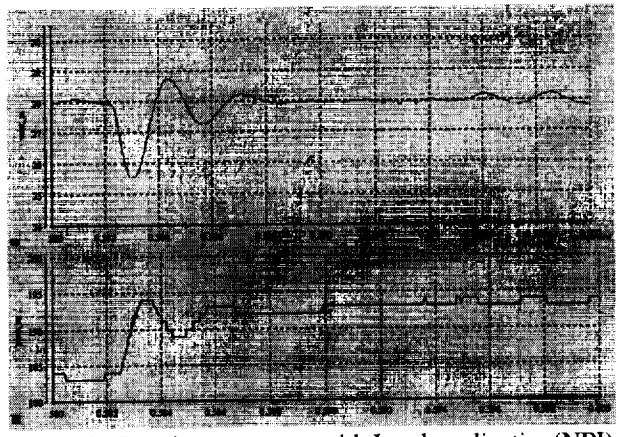


Figure 8 Transient response with Load application(NPI)

The above results show the benefits of using the NPI controller as:

1.Much cleaner control output

2.Much less ringing during load application.

3.Much faster load application recovery time during load application.

ONGOING AND FUTURE RESEARCH PURSUITS

As the AERL team undertook R & D activity to replace the traditional analog SMPS controller with a direct digital solution, a number of technology and system issues became evident. Several of these issues will now be briefly addressed in terms of each one's ongoing design and development activities.

Signal Conditioning – Critical voltage and current variables, which define the converter's performance, must be measured accurately, isolated, and conditioned for sampling by the digital controller. One important aspect of the signal conditioning is the selection of adequate anti-aliasing filters to remove (to filter out) unneeded high frequency information in the measurements. To accommodate these requirements the signal conditioning circuitry was breadboarded for the initial

experimental studies. For future control studies, a ruggedized printed circuit version of this circuitry is being designed.

PWM Generation - As stated earlier, CSU's approach to PWM generation is to generate the pulse width gate driving signals with a programmable logic device (CPLD). The present performance limiting eight-bit PWM CPLD will soon be replaced with an alternative CPLD design which provides higher resolution (finer quantization) and will have the ability to vary the PWM frequency directly through commands from the DSP software. Closed loop control testing of the higher resolution CPLD is now underway. Performance studies using the variable frequency feature will start soon. Results will be reported at a later time.

Control Mode Selection-As was shown in the results section, the new nonlinear control strategies show benefits over linear, more traditional, control modes. It must be noted at this point, however, that the AERL team has not yet implemented a current-mode inner loop. Because we generate the PWM signals digitally, a strategy for effectively using sensed transformer primary current in an inner current loop control has not yet been determined. Resolving this control design issue is a major priority.

DSP Control Development Platform –The dSpace rapid-prototype development equipment has played an invaluable role in our controls research. However, at CSU we are designing an easier-to-use DSP development platform to study converter control in a broader PMAD system context. A major feature of this platform design is the inclusion two high-speed IEEE-1394 (Firewire) data communication ports.

SUMMARY

A research program on direct digital control of power converters has been described. Analytical and experimental results for a new nonlinear control strategy are discussed and compared against traditional linear control modes. The results encourage continued study into nonlinear approaches to converter voltage regulation. Finally some of the technology issues related to digital converter control are identified and efforts for their resolution discussed.

ACKNOWLEDGMENTS

The work discussed in this paper is sponsored by the NASA Glenn Research Center under grant #NCC3-699 and by Cleveland State University. The authors would like to thank other members of the AERL team including Charles Alexander, Dave Gerdeman, Dave Wladyka, Ivan Jercic, Greg Tollis, Marcelo Gonzalez, and John Sustersic for their active participation in the development activities.

REFERENCES

1. Robert M. Button," intelligent systems for power management and distribution", NASA Glenn Research Center, Cleveland, OH 44135
2. John Sustersic," Design and Implementation of a Digital Controller for DC-DC Power Converters", 2000 SAE Power Systems Conference, San Diego, CA, Oct. 31, 2000
3. Stimac, Tom," Digital Control of a 1-kW DC-DC switching Power converter", Master's Thesis, Department of Electrical and Computer Engineering, Cleveland State University, December 2000
4. dSPACE Inc. 22260 Haggerty Road - Suite 120 Northville, MI 48167 Tel.: (248) 344-0096 Fax: (248) 344-2060.
5. Zhiqiang Gao, "Linear to Nonlinear Control Means: A practice Progression", To appear in ISA Transactions.
6. David M. Wladyka, " Investigation of C Code Execution on dSpace ", AERL Report, Cleveland State University, Cleveland,OH,44115
7. R.R. Boudreaux, R.M. Nelms, and John Y. Hung, "Simulation and Modeling of a DC-DC Converter Controller by an 8-bit Microcontroller", 0-7803-3704-2/97 IEEE
8. W.C. So, C.K. Tse and Y.S. Lee, "A Fuzzy Controller for DC-DC Converters", 0-7803-1859-5/94 IEEE
9. Keyue Ma Smedley and Slobodan Cuk, "One-Cycle Control of Switching Converters", 0-7803-0090-4/91 IEEE
10. T. Gupta, R.R. Boudreaux, R.M. Nelms, and John Y. Hung, "Implementation of a Fuzzy Controller for DC-DC Converters Using and Inexpensive 8-bit Microcontroller", IEEE Transactions on Industrial Electronics, Jan 1997.
11. R. Vinsant, J. DiFiore, and R. Clarke, "Digital Control Converts Power Supply into Intelligent Power System Peripheral," Ninth International High Frequency Power Conversion Conference, April 1994, pp. 2-6.
12. R.R. Boudreaux, R.M. Nelms, and John Y. Hung, "Digital Control of DC-DC Converters: Microcontroller Implementation Issues," Combined Proceedings of HFP Power Conversion & Advanced Power Electronics Technology, Powersystems World'96. September 1996, pp. 168-180.
13. Barry Arbetter, Dragan Maksimovic," DC-DC Converter with Fast Transient Response and High Efficiency for Low-Voltage Microprocessor Loads", Department of Electrical and Computer Engineering at the University of Colorado, 1997.
14. Barry Arbetter, Dragan Maksimovic, " Feedforward Pulse Width Modulators for Switching Power Converters", IEEE Transanctions on Power Electronics, Volume. 12, No. 2, March 1997.
15. David J. Caldwell, "Digital Power Flexibility: Applications and Advantages", HFPC '99 Proceedings, November 1999
16. Sun-Hoe Huh,Gwi-Tae Park ,"An Adaptive Fuzzy Controller for Power Converters", University of Hong Kong, 1998.