

Electronic Theses and Dissertations, 2004-2019

2009

Unified Large And Small Signal State-space Based Modeling And Symbolic Simulation For Pwm Converters

Ehab Shoubaki
University of Central Florida

 Part of the [Electrical and Electronics Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Shoubaki, Ehab, "Unified Large And Small Signal State-space Based Modeling And Symbolic Simulation For Pwm Converters" (2009). *Electronic Theses and Dissertations, 2004-2019*. 4014.
<https://stars.library.ucf.edu/etd/4014>

**UNIFIED LARGE AND SMALL SIGNAL STATE-SPACE BASED
MODELING AND SYMBOLIC SIMULATION FOR PWM CONVERTERS**

by
EHAB H. SHOUBAKI
M.Sc. University of Central Florida, 2005

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2009

Major Professor: Issa Batarseh

© 2009 Ehab H. Shoubaki

ABSTRACT

In this Dissertation, which concentrates on discrete modeling for control purposes of DC/DC converters and simulation through symbolic techniques. A Unified Discrete State-Space Model for power converters in CCM is presented. Two main approaches to arriving at the discrete model are used. The first approach involves an impulse function approximation of the duty cycle modulation of the converter switches, and this approach results in a small signal discrete model. The Second approach is direct and does not involve any approximation of the modulation, this approach yields both a large signal nonlinear discrete model and a linear small signal model. Harmonic analysis of the converter's states at steady-state is done for steady-state waveform acquisition, which increases the accuracy of the model especially for finding the control to inductor current frequency response. Also the harmonic Analysis technique is used to both obtain the response of the converter to a load transient and to finding the optimal duty cycle response that minimizes the disturbance. Finally the Discrete model is verified for the Half-Bridge DC/DC topology for its three main control schemes (Asymmetric, Symmetric, DCS). A GUI platform in MATLAB is presented as a wrapper that utilizes the models and analysis presented in this thesis. Symbolic simulation techniques are developed in general manner for linear piecewise circuits and then through State-Space formalism specialized for DC/DC converters. A general symbolic solver programmed in JAVA that implements said techniques is presented.

ACKNOWLEDGMENTS

This author wishes to thank his advisor , Dr. Issa Batarseh , for his help and support through the rough waters of academia. Also much thanks is directed towards Dr. Jaber Abuqahouq for his involved discussions and helpful advice. Much appreciation also for my friends and family. It is a given that without the sacrifices and support of my father , Hamed Shoubaki , and my mother , Durar Shishakli , I would never have started on this path I am finishing with this work. Last and most, I thank my beautiful wife Hiba for her patience and immense support, without her assurance this work could never have seen the light of day.

TABLE OF CONTENTS

LIST OF FIGURES	ix
LIST OF TABLES	xiii
1 CHAPTER 1 INTRODUCTION.....	1
1.1 Need for automatic control in PWM converters	1
1.2 Average Continuous Modeling of PWM converters.....	3
1.3 Introduction to Digital Control and Discrete modeling of PWM converters.....	14
1.4 Need for Simulation of PWM converters.....	16
1.5 Dissertation Outline.....	16
2 CHAPTER 2 OVERVIEW AND STEADY-STATE HARMONIC ANALYSIS.....	18
2.1 General State-Space Switched Model.....	18
2.2 Small Signal perturbation.....	20
2.3 Motivation for Harmonic Analysis	22
2.4 Fourier Series	23
2.5 Theoretical acquisition of Harmonics of steady-state waveforms	26
2.6 Implementation in MATLAB.....	29
3 CHAPTER 3 HARMONIC ANALYSIS FOR THE ACQUISITION OF TRANSIENT WAVEFORMS FOR PERIODOCALLY VARYING LOAD.....	32
3.1 Motivation	32

3.2	Acquisition of Step load response from discrete Duty Cycle Signal	33
3.3	Finding Optimal discrete Duty Cycle signal for certain load pulse	43
4	CHAPTER 4 SMALL AND LARGE SIGNAL DISCRETE DYNAMIC ANALYSIS.....	49
4.1	Motivation	49
4.2	Discrete small signal State-Space Model	49
4.3	Discrete Large signal State-Space Mode	59
4.4	Discrete Small signal State-Space Model without impulse approximation	65
5	CHAPTER 5 APPLICATION OF DISCRETE MODELING METHOD TO THE HALF-BRIDGE DC/DC TOPOLOGY WITH CURRENT DOUBLER.....	74
5.1	Overview of Half-Bridge topology with three main control method.....	74
5.2	State-Space representation of Modes of the Half-Bridge DC/DC topology with current doubler	75
5.3	Application of Chapter 3 small-signal model to the Half-Bridge DC/DC topology with current doubler	79
5.4	Application of Chapter 4 Large-signal model to the Half-Bridge DC/DC topology with current doubler	88
5.5	Comparison with an experimental prototype	90
6	CHAPTER 6 IMPLEMENTATION OF DISCRETE MODEL IN A GUI FOR CONTROLLER DESIGN USING MATLAB	93
6.1	Overview of GUI platform.....	93

6.2	Design Example	101
7	CHAPTER 7 JAVA BASED SYMBOLIC CIRCUIT SOLVER (SYMCIRC).....	104
7.1	Motivation	104
7.2	Review of Spice	105
7.3	Transfer Function Based Symbolic Simulation	107
7.4	Java Implementation	111
7.5	Time-Domain and Frequency Domain Modules.....	114
7.6	Symbolic Manipulation Modules.....	119
7.7	Graphical and User Interfaces and Top level modules	120
7.8	Demonstration	122
8	CHAPTER 8 STATE-SPACE BASE SYMBOLIC SIMULATION FOR DC/DCCONVERTERS.....	128
8.1	Introduction and Motivation.....	128
8.2	State-Space Base equations for Generic Second Order converter with Two Operating Modes.....	129
8.3	Conditions for Stability and One Cycle Periodicity.....	131
8.4	Approach to dealing with Defective Transition Matrices	134
8.5	Derivation and Summary of Symbolic Simulation Equations for basic Second Order Topologies.....	136
8.6	Symbolic linearization for Acquisition of dynamic characteristics	146

8.7	Example : Application to a synchronous Buck topology implemented in Mathematica	149
9	CHAPTER 9 CONCLUSION AND FUTURE WORK.....	161
9.1	Conclusion.....	161
9.2	Future Work	163
	REFERENCES	164

LIST OF FIGURES

Figure 1-1 : Illustration of regulation through duty cycle control	2
Figure 1-2 : Switching components cell in each of the three main Topologies: (a) Buck, (b) Boost, (c) Buck-Boost.....	7
Figure 1-3 : Switching Cell Block	8
Figure 1-4 : Cell Block Waveforms in the case of DCM	10
Figure 1-5 : Cell Block continuous model for DCM	10
Figure 1-6 : Small-Signal equivalent circuit.....	11
Figure 1-7 : Simulation of actual buck circuit and average model in PSPICE.....	12
Figure 1-8 : Frequency response of Buck , (a) Magnitude , (b) Phase	13
Figure 1-9 : Comparison between Average Model vs actual circuit simulation.....	14
Figure 2-1 : Switching functions $\bar{d}_1(t)$ and $\bar{d}_2(t)$ for two switches in steady-state.....	19
Figure 2-2 : Perturbation of Switching functions $\hat{d}_1(t)$ and $\hat{d}_2(t)$ for two switches	20
Figure 2-3 : Magnitude of Fourier Coefficients of $\bar{d}_1(t)$	25
Figure 2-4 : Magnitude of Fourier Coefficients of $\bar{d}_2(t)$	25
Figure 2-5 : Harmonic Analysis implementation using MATLAB	31
Figure 3-1 : Modeling the load as input alone or with parallel resistance	34
Figure 3-2 : Output Load current assumed waveform for application of Harmonic Analysis technique.....	36
Figure 3-3 : Decomposition of duty cycle signal $d(t)$	37

Figure 3-4 : Magnitude (a) and Phase (b) of Fourier coefficients of each $d_k(t)$ (D constant).....	39
Figure 4-1 : Approximation of Duty cycle perturbations as impulse functions	50
Figure 4-2 : Model Piecewise linear regions with varying switching instants	61
Figure 5-1 : (a) Isolated Half-Bridge Converter, (b) Mode I (S1 on, S2 off), (b) Mode II (S1 off, S2 on), (c) Mode III (S1 off, S2 off).....	77
Figure 5-2 : HB DC/DC steady-state waveforms for symmetric control	81
Figure 5-3 : HB DC/DC steady-state waveforms for asymmetric control.....	82
Figure 5-4 : HB DC/DC steady-state waveforms for DCS control	83
Figure 5-5 : Simpler Simulation of the Asymmetric Control case	84
Figure 5-6 : Frequency Responce for HB DC/DC example for Symmetric , Asymmetric and DCS control respectively	85
Figure 5-7 : Matlab Script to implement Discrete Model.....	86
Figure 5-8 :Comparison between the Average Model and the Discrete Model for DCS control.	87
Figure 5-9 : Application of Large Signal model to HB DC/DC topology for Symmetric and DCS control respectively	88
Figure 5-10 : Application of Large Signal model to HB DC/DC topology for Asymmetric control	89
Figure 5-11 : Comparison of Closed loop frequency response between experiment and theory .	91
Figure 6-1 : Digital Controller Designer main window	95
Figure 6-2 : Digital Controller Designer parameters window	96
Figure 6-3 : Digital Controller Designer Time response window	97
Figure 6-4 : Digital Controller Designer parameter sweep window	98
Figure 6-5 : Digital Controller Designer Root Locus window	99

Figure 6-6 : Digital Controller Designer Steady-State waveforms window.....	100
Figure 6-7 : Parameters for Lab Prototype Half-Bridge with current doubler and Symmetric control.....	101
Figure 6-8 : Desired Bandwidth and phase margin with required compensator poles and zeros.....	102
Figure 6-9 : Compensator realization in the analog domain.....	103
Figure 6-10 : Difference Equation coefficients for Digital Controller realization.....	103
Figure 7-1 : Voltage Dependent current sources into the indefinite admittance matrix.....	108
Figure 7-2 : Outline of transforming other dependent sources into voltage dependent current source.....	109
Figure 7-3 : Pre-determinant evaluation of limit finding.....	111
Figure 7-4 : Overview of modules of SymCirc.....	113
Figure 7-5 : Roots of fourth order polynomial equation.....	116
Figure 7-6 : Newton-Raphson Algorithm.....	116
Figure 7-7 : Laguerre's Algorithm.....	117
Figure 7-8 : Graphical User Interface Front end for SymCirc.....	121
Figure 7-9 : Demonstration Circuits.....	122
Figure 7-10 : Transfer functions and time domain expressions for circuit of figure 7-9 (a).....	123
Figure 7-11: voltage waveform across C0 for circuit of figure 7-9 (b).....	124
Figure 7-12 : Current waveform across C0 for circuit of figure 7-9 (b).....	125
Figure 7-13 : Waveform of power dissipated through R1 for circuit of figure 7-9 (b).....	126
Figure 7-14 : Frequency response with respect to input voltage for circuit 7-9 (a).....	127
Figure 8-1 : Illustration of x_{ripple} , $x_{\text{ss,min}}$ and $x_{\text{ss,max}}$	131

Figure 8-2 : (a) Buck ,(b) Boost and (c) Buck-Boost with all relevant parameters and parasitic	145
Figure 8-3 : Algorithm for CAS aided acquisition of the perturbed symbolic variables.....	149
Figure 8-4 : Discrete Large signal response for (a) Output Voltage , (b) Inductor Current	151
Figure 8-5 : First 10 switching cycles simulation for $d=0.3$	153
Figure 8-6 : Switching cycles between $500T_s$ and $510T_s$	154
Figure 8-7 : One (a) Inductor current and (b) Capacitor voltage switching cycle plotted versus d between 0 to 1 in 0.1 increments	155
Figure 8-8 : Current and voltage ripple vs Frequency	156
Figure 8-9 : Inductor current ripple versus (a) inductance , and (b) Duty Cycle ratio	157
Figure 8-10 : Bode Plot of frequency response	159
Figure 8-11 : Frequency response versus variance in operating point (Duty Cycle).....	159
Figure 8-12 : Frequency response versus output filter capacitance.....	160
Figure 8-13 : Frequency response versus output filter Inductance.....	160

LIST OF TABLES

Table 4-1 : Model parameters substitution for various topologies with different control schemes	57
Table 5-1 : Model parameter substitution for each control scheme.....	75
Table 8-1 : : a_{ij} and b_{ij} parameters for the Buck Converter	142
Table 8-2 : : a_{ij} and b_{ij} parameters for the Boost Converter.....	143
Table 8-3 : : a_{ij} and b_{ij} parameters for the Buck-Boost Converter.....	144
Table 8-4 : $[c_1 \ c_2]$ for different controlled outputs and topologies	148

CHAPTER 1

INTRODUCTION

1.1 Need for automatic control in PWM converters

Since the inception of SMPS converters (Switched Mode Power Supply) it was known that operation without regulation is not adequate for cases where the load is variable, or the parameters of the converter change either intentionally or through aging. That is why automatic control is widely implemented in switching converters and open loop operation is nonexistent.

Depending on the topology used there are many different control schemes available. One of the earliest is to control the on and off times of the switches in a SMPS and to vary their values depending upon how far the system is off the desired operational range. Another technique is to control another parameter in the circuit and use that parameter to regulate the output, essentially having two control loops, an inner and an outer one. Current Mode Programming (whether peak or averaged) is the most famous example for this type of control, where there is an inner loop that controls the inductor current and then an outer loop controlling the reference to the previous loop, the output voltage being dependant on the inductor current. This control technique is popular since it simplifies the outer control loop because the power stage is essentially operating as a voltage controlled current source. A third method of automatic control is achieved through varying the switching frequency, and this method is prevalent in soft switching topologies.

Due to this immense importance for automatic control in the switched power field the question arises as to how to close the loop on the converter, that is how to design an appropriate compensator to maintain stability at the required operating point. It is apparent that the powerful tools of Linear Control Theory (frequency domain theories) and Modern Control Theory (time domain theories) cannot be used directly for DC/DC converter since they are highly nonlinear systems. The rest of this chapter will describe the different approaches for solving this problem.

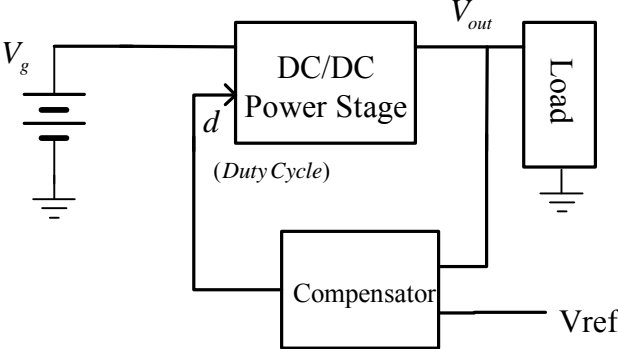


Figure 1-1 : Illustration of regulation through duty cycle control

1.2 Average Continuous Modeling of PWM converters

Since the power converter is not just nonlinear but also of piecewise switched nature, averaging techniques must be used to obtain a continuous model, which is then linearized using small signal perturbation. Middlebrook and Cuk (1) pioneered this field in the late seventies using state space averaging, and later others (2) -(3) like Vorperien introduced circuit oriented average models. Both methods differ only in form, the second being more popular since circuit oriented analysis is ubiquitous in the electrical engineering community, and engineers prefer to deal normally with circuits rather than matrices. Another reason for the preference for circuit oriented methods is the ease by which they can be implemented in popular circuit simulation software like PSPICE.

Such so called small signal analysis of the system can yield for moderately complex systems very complex and tedious symbolic expressions of the transfer functions of interest. Those expressions are called “high entropy expressions” (a term coined by Middlebrook) . There are many ways of dealing with such complex systems, one is order reduction; i.e. applying some simplifying assumptions to the system. Order reduction can be piecewise in frequency meaning that the simplifying assumptions can be different for different frequency ranges, this decreases reduction errors and as an example is used in (4). Another is empirical, and is to simply bypass the analysis stage and use a frequency response analyzer to obtain the open loop transfer function. But the main disadvantage in this case is the lack of insight into the effect of the various components and parameters on the dynamics of the system

The State-Space Average Modeling is the most general method of the two , it requires translating the DC/DC converter into State-Space representation for each switching mode, that is for each piecewise linear region in time. Since this method involves matrix algebra Modern Control techniques can be easily exploited. Although even if this method is used most prefer to derive transfer functions from the resultant model so as to use traditional control techniques. To show how this works assume the topology of interest has two modes of operation, and each mode can be described with the following State-Space equations, for Mode I,

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_1\mathbf{x} + \mathbf{B}_1\mathbf{u} \quad (1.1)$$

$$y = \mathbf{C}_1\mathbf{x} + \mathbf{D}_1\mathbf{u} \quad (1.2)$$

and Mode II,

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_2\mathbf{x} + \mathbf{B}_2\mathbf{u} \quad (1.3)$$

$$y = \mathbf{C}_2\mathbf{x} + \mathbf{D}_2\mathbf{u} \quad (1.4)$$

And the duration of Mode I is dT_s and of Mode II is $d'T_s$ when d is the Duty cycle at the particular operating point and $d'=1-d$. Then (1) proves that the power stage can be approximated by the following continuous State-Space model,

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_{\text{avg}}\mathbf{x} + \mathbf{B}_{\text{avg}}\mathbf{u} \quad (1.5)$$

$$\vec{y} = \mathbf{C}_{\text{avg}}\mathbf{x} + \mathbf{D}_{\text{avg}}\mathbf{u} \quad (1.6)$$

where,

$$\mathbf{A}_{\text{avg}} = d\mathbf{A}_1 + d'\mathbf{A}_2 \quad (1.7)$$

$$\mathbf{B}_{\text{avg}} = d\mathbf{B}_1 + d'\mathbf{B}_2 \quad (1.8)$$

$$\mathbf{C}_{\text{avg}} = d\mathbf{C}_1 + d'\mathbf{C}_2 \quad (1.9)$$

$$\mathbf{D}_{\text{avg}} = d\mathbf{D}_1 + d'\mathbf{D}_2 \quad (1.10)$$

Equations (1.5) and (1.6) represent a continuous average model, but it is still not linear with respect to the duty cycle as input. The way to achieve the linear model which by nature is small-signal is to apply small signal perturbations to all the variables in (1.5)-(1.6), and then to group the DC and small signal parts separately. Doing so gives,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}_{\text{avg}}\hat{\mathbf{x}} + \mathbf{B}_{\text{avg}}\hat{v}_g + [(\mathbf{A}_1 - \mathbf{A}_2)\mathbf{X} + (\mathbf{B}_1 - \mathbf{B}_2)V_g]\hat{d} \quad (1.11)$$

$$\hat{y} = \mathbf{C}_{\text{avg}}\hat{\mathbf{x}} + (\mathbf{C}_1 - \mathbf{C}_2)\mathbf{X}\hat{d} \quad (1.12)$$

where,

$$\mathbf{X} = -\mathbf{A}^{-1}\mathbf{B}V_g \quad (1.13)$$

$$Y = -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}V_g \quad (1.14)$$

Equations (1.11)-(1.12) represent the desired linear small signal model useful for designing an appropriate compensator. A transfer function can be extracted from it through applying the Laplace transform and it turns out to be,

$$\frac{\hat{Y}(s)}{\hat{d}(s)} = (\mathbf{C}_1 - \mathbf{C}_2)\mathbf{X} + \mathbf{C}_{\text{avg}}(s\mathbf{I} - \mathbf{A}_{\text{avg}})^{-1} [(\mathbf{A}_1 - \mathbf{A}_2)\mathbf{X} + (\mathbf{B}_1 - \mathbf{B}_2)V_g] \quad (1.15)$$

The circuit oriented Average modeling methods are numerous but all depend on averaging all currents and voltages in the converter circuit over one switching period. Vorperian [2] introduced a methodology similar to the one used for analyzing a transistor circuit where the block in the circuit containing the switching devices is examined and replaced by a closely equivalent average model consisting of dependent voltage and current sources. The blocks to be replaced are as shown in figure 1-2 for the three main switching topologies.

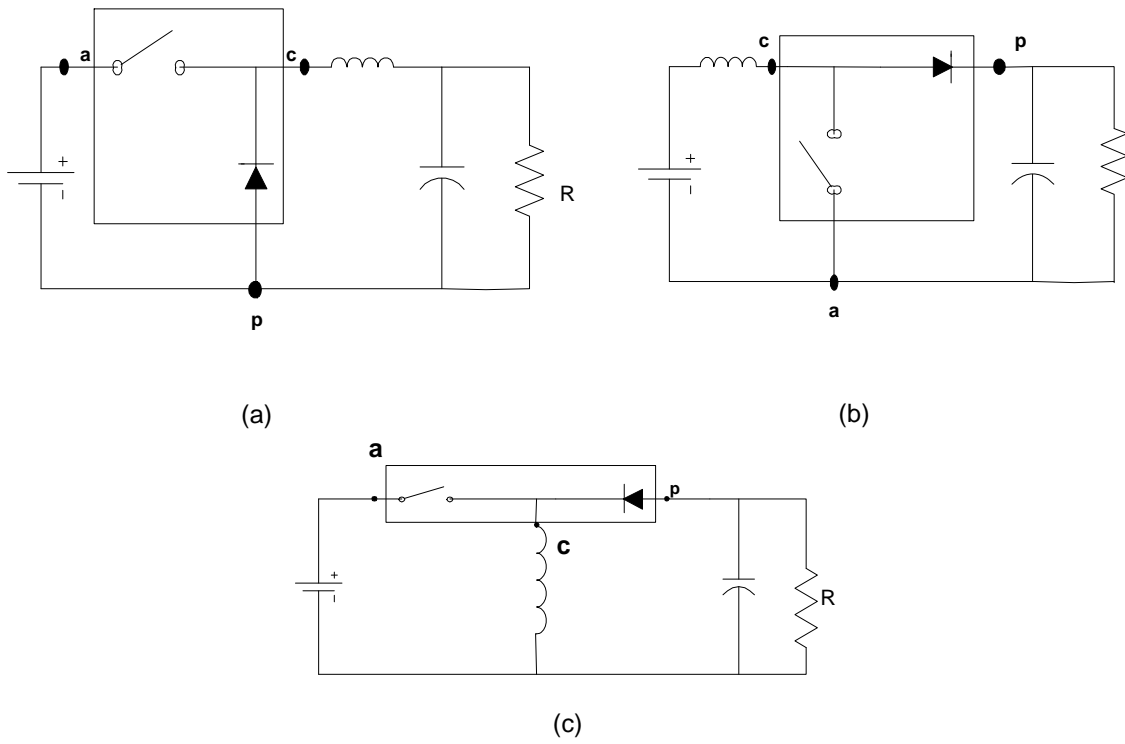


Figure 1-2 : Switching components cell in each of the three main Topologies: (a) Buck, (b) Boost, (c) Buck-Boost

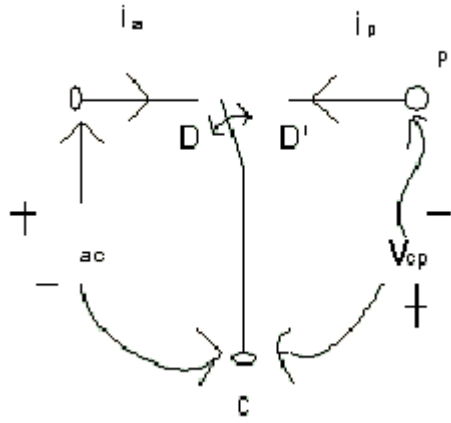


Figure 1-3 : Switching Cell Block

Figure 1-1 shows a detail of the switching cell block, it is a three terminal block where the 'c' node represents where the inductor is connected and the 'a' node is at the terminal where the active switch is, likewise the 'p' node is where the passive diode is. The heart of this modeling method is to examine the waveforms in general across the terminals of the block and try to relate their average values together. For example for the special case where the topology is operating in DCM the currents $i_a(t)$ and $i_p(t)$ look as shown in Figure 1-4, and the Continuous Model that can be deduced from the averaging of these waveforms that are shown in Figure 1-5. (2) showed that this equivalent circuit of the block has ,

$$\mu = \frac{d^2}{2LF_s} \frac{v_{cp}}{i_a} = \frac{d^2}{2LF_s} \frac{v_{ac}}{i_p} \tag{1.16}$$

so it is still a nonlinear model and linearizing it requires applying small signal perturbation which yields,

$$\hat{i}_p = g_f \hat{v}_{ac} + k_o \hat{d} - g_o \hat{v}_{cp} \quad (1.17)$$

where,

$$k_o = \frac{2I_p}{D} = \frac{2DV_{ac}^2}{2LF_s V_{cp}} \quad (1.18)$$

$$g_o = \frac{I_p}{V_{cp}} = \frac{D^2 V_{ac}^2}{2LF_s V_{cp}^2} \quad (1.19)$$

$$g_f = \frac{2I_p}{V_{ac}} = \frac{2D^2 V_{ac}}{2LF_s V_{cp}} \quad (1.20)$$

And,

$$\hat{i}_a = g_i \hat{v}_{ac} + k_i \hat{d} \quad (1.21)$$

Where,

$$g_i = \frac{I_a}{V_{ac}} = \frac{D^2}{2LF_s} \quad (1.22)$$

$$k_i = \frac{2I_a}{D} = \frac{2DV_{ac}}{2LF_s} \quad (1.23)$$

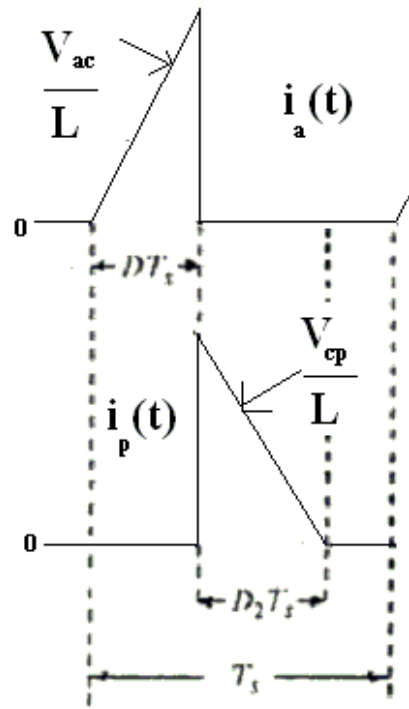


Figure 1-4 : Cell Block Waveforms in the case of DCM

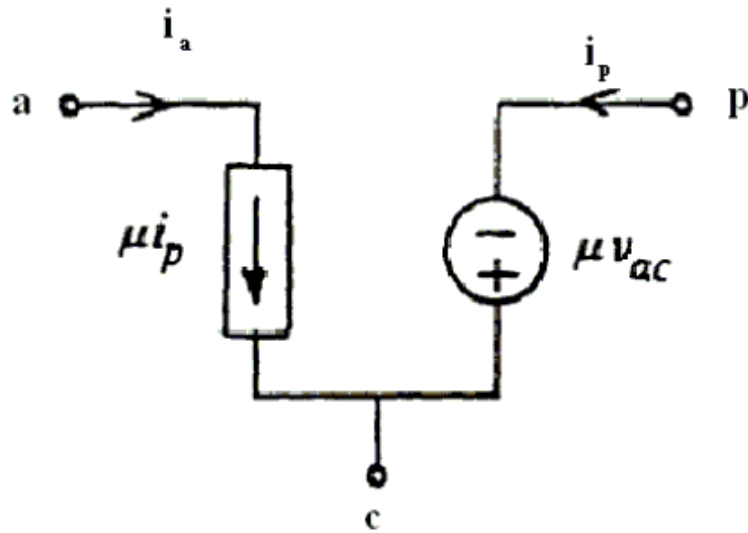


Figure 1-5 : Cell Block continuous model for DCM

And therefore the small signal equivalent circuit is as shown in figure 1-6, see (2). It was said that one advantage of a circuit oriented approach is the ease by which the model can be implemented in PSPICE, so for demonstration purpose the Buck circuit in figure 1-7 was simulated both in original switching form and with the average model. Figure 1-8 displays the control to output voltage frequency response obtained and figure 1-9 compares the average model vs actual responses.

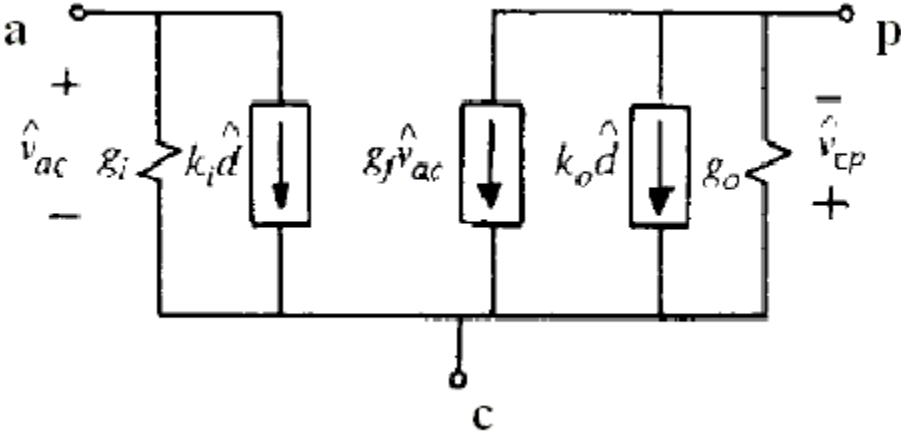


Figure 1-6 : Small-Signal equivalent circuit

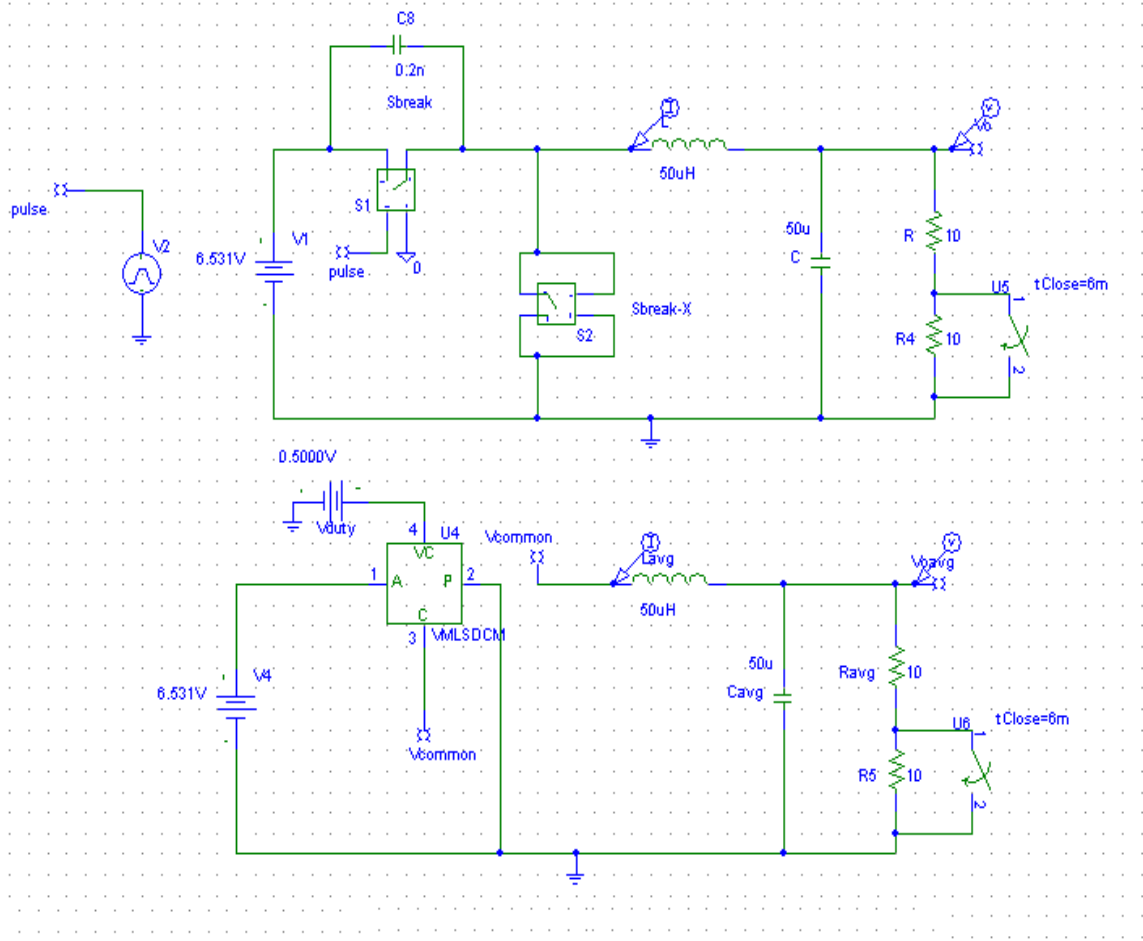
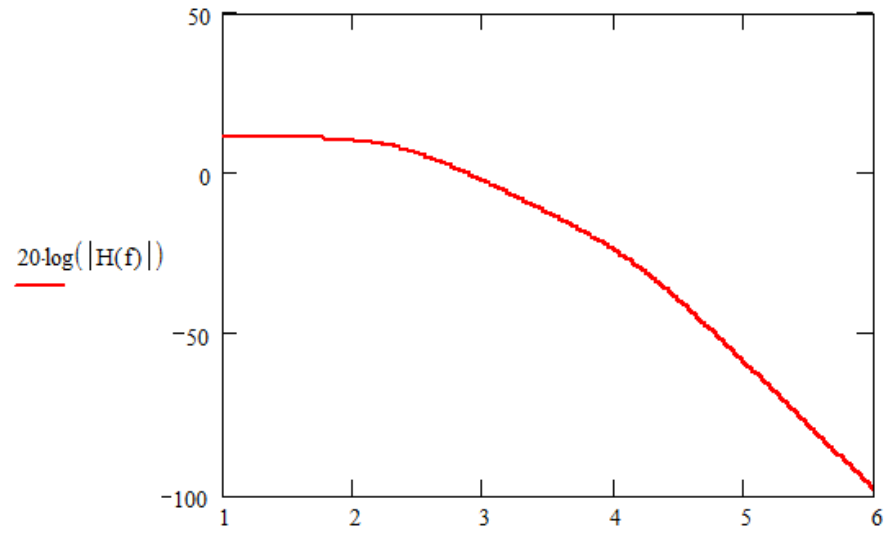
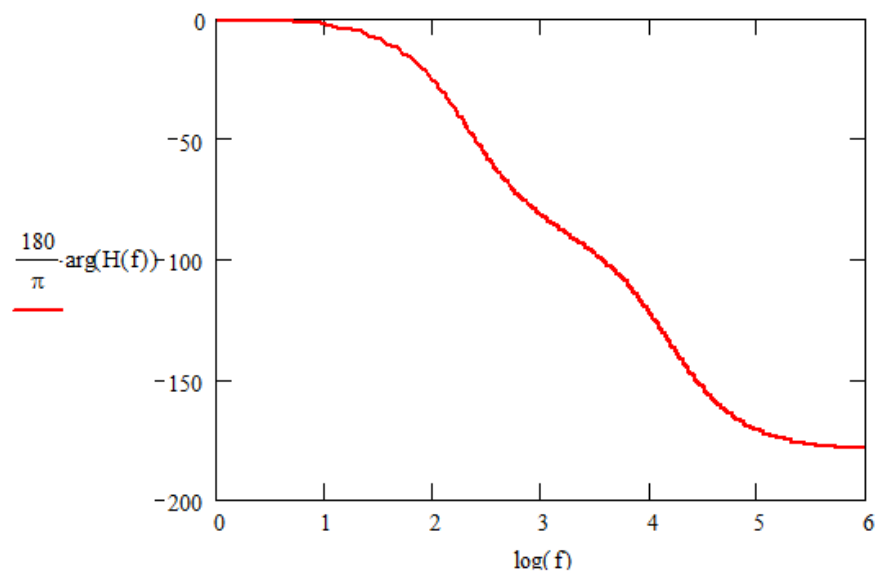


Figure 1-7 : Simulation of actual buck circuit and average model in PSPICE



(a)



(b)

Figure 1-8 : Frequency response of Buck , (a) Magnitude , (b) Phase

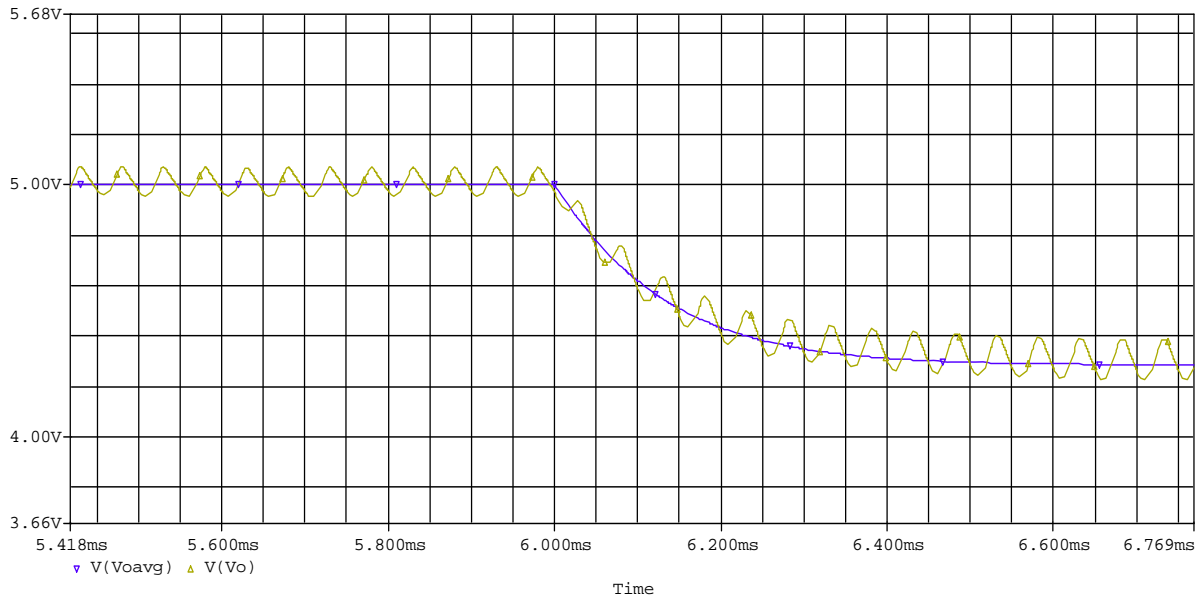


Figure 1-9 : Comparison between Average Model vs actual circuit simulation

1.3 Introduction to Digital Control and Discrete modeling of PWM converters

Conventional averaging methods for dynamic modeling yield models that are inaccurate at the upper frequency range because they ignore the sampled nature of the Duty Cycle signals that drive the power stage. Furthermore they predict frequency response content at higher than half the switching frequency, of course such information is meaningless.

One objective of this dissertation is to derive a general unified large and small discrete-space models for PWM converters in CCM so as they can be utilized for digital control of the power stage. It was shown in (5) that discrete models are more accurate in predicting the dynamics of switching converters since switching is a naturally discrete phenomena and as thus the methodology used there will be implemented in this work. The main line of attack will be the

recognition of the different modes of operation of the power stage and finding their state space matrices. This can be done since in each mode the converter is essentially a linear system. Ultimately the transient and forced time responses within each mode are derived and after relating them with each other a discrete state-space system can be deduced. Thus we arrive at the discrete model and having it in state-space form makes it versatile since it can be used to obtain pulsed filter transfer functions or simply plugged directly into a numerical package like MATLAB. Different control techniques and different topologies will result in different modes of operation thus alter the results of the model.

One of the useful utilizations of the discrete model is to obtain frequency response curves for the open loop power stage. Phase and gain margin techniques can then be used to design an appropriate compensator for stabilization of the closed loop response and furthermore the compensator can be discretized and implemented through a DSP controller. Such a methodology is an indirect digital design method since the compensator is first designed in the analog domain and then mapped into the digital domain. Having a discrete model of the Power stage in the first place makes it prudent to use a direct design method such as pole-placement where the coefficients of the digital compensator are directly obtained.

The discrepancy in the frequency response between the discrete models derived in this dissertation and the average continuous models is examined in chapter 5 for the Half-Bridge topology, and is found to be substantial at high frequencies approaching one half the switching frequency.

1.4 Need for Simulation of PWM converters

After the design phase of any technological artifact ends and before production is started adequate testing is required to insure the device will work as planned and within the desired parameters. Prototypes in such case are extremely useful, but still, for the verification of new concepts simulation is necessary before embarking on the expense of prototype building. The purpose of simulation is to predict how the system will operate, and its basis is the theoretical laws that govern its components and their interactions.

1.5 Dissertation Outline

After this introductory chapter of this dissertation chapter 2 will start with an overview of the general switched State -Space equation describing the operation of DC/DC converters. Then for the purpose of increasing the accuracy of the modeling the problem of evaluating the Steady-State waveforms is attacked using a Fourier analysis technique, and sample MATLAB code demonstrating how to implement the analysis results.

Chapter 3 and 4 commence with the task of Discrete modeling .Both chapters incorporate any general topology with at most two switches and two freewheeling modes where the phase shift between the duty cycle pulses is a variable within the models. Chapter 4 uses an impulse approximation of the duty cycle modulations to somewhat simplify the analysis that ends up with a small signal model, it also examines the large signal discrete model and introduces small-signal analysis directly to it without any impulse approximations. Finally chapter 5 applies the

developed models to the Half-Bridge DC/DC converter for all its prominent control methods (Symmetric, Asymmetric and DCS). Then Chapter 6 presents a GUI platform developed in MATLAB as an implementations of the models of this dissertation, this platforms is essentially used to aid digital controller design.

Chapter 7 is concerned with symbolic simulation and details SYMCIRC, a Java based symbolic circuit solver. Chapter 8 examines symbolic simulation using state-space techniques. Chapter 9 is the conclusion and future work section of this dissertation.

CHAPTER 2 OVERVIEW AND STEADY-STATE HARMONIC ANALYSIS

2.1 General State-Space Switched Model

Any switching power converter has switches that turn on and off in a predetermined manner to alter between different modes of operation and therefore achieve power transfer in a controlled way. A general State-Space switched equation is presented here to account for the very general case of converters that have two switches or less and two freewheeling modes or less. That is not to say other situations are not accounted for, as will be explained later. So the general switching equation, assuming the converter has two switches S1 and S2, is,

$$\frac{d\mathbf{x}}{dt} = [d_1(t)\mathbf{A}_1 + d_2(t)\mathbf{A}_2 + d_3(t)\mathbf{A}_3]\mathbf{x} + [d_1(t)\mathbf{b}_1 + d_2(t)\mathbf{b}_2 + d_3(t)\mathbf{b}_3]v_g \quad (2.1)$$

where $d_1(t)$, $d_2(t)$ and $d_3(t)$ in steady state operation are as shown in Figure 2-1, and their definition is,

$$d_1(t) = \begin{cases} 1 & \text{if S1 on} \\ 0 & \text{Otherwise} \end{cases} \quad (2.2)$$

$$d_2(t) = \begin{cases} 1 & \text{if S2 on} \\ 0 & \text{Otherwise} \end{cases} \quad (2.3)$$

And,

$$d_3(t) = 1 - d_1(t) - d_2(t) \quad (2.4)$$

In Figure 2-1 there is a phase shift α between the duty pulses for S1 and S2 , this phase shift is useful when a certain topology like the Half-Bridge has different control schemes controlled by the phase shift between the different duty cycle pulses. It will be shown in chapter 5 that this phase shift does affect the dynamics of the converter in a visible manner.

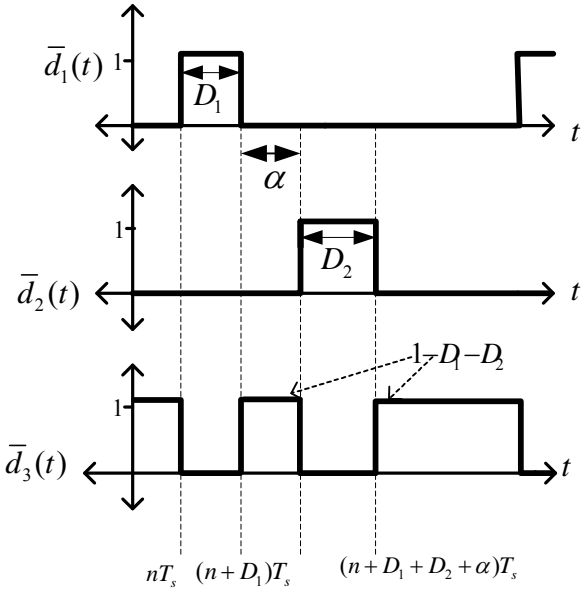


Figure 2-1 : Switching functions $\bar{d}_1(t)$ and $\bar{d}_2(t)$ for two switches in steady-state.

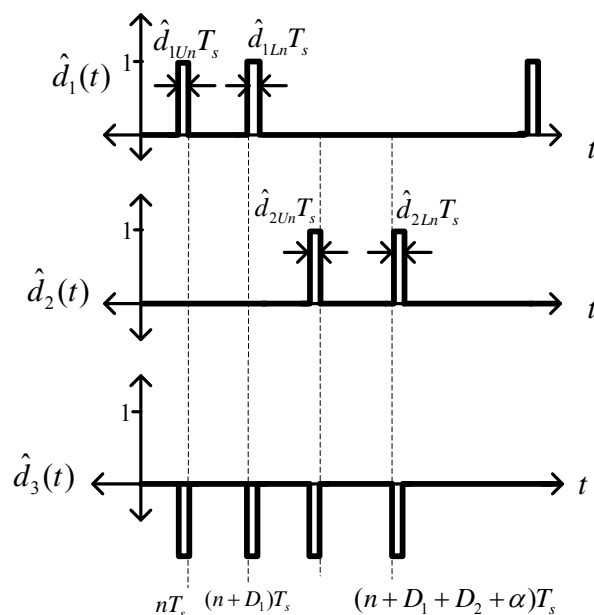


Figure 2-2 : Perturbation of Switching functions $\hat{d}_1(t)$ and $\hat{d}_2(t)$ for two switches

2.2 Small Signal perturbation

Since it is desired to obtain a dynamic model from the switched state-space form the first step whether the main interest is discrete or continuous modeling is to apply small signal perturbation about steady state conditions. The particular notation used here marks all steady state signals with a bar over the variable (\bar{a}) while small signal perturbations are marked with the hat symbol (\hat{a}). The components of each state variable, switching functions and the input voltage are,

$$x = \bar{x} + \hat{x}, d_1(t) = \bar{d}_1(t) + \hat{d}_1(t), d_2(t) = \bar{d}_2(t) + \hat{d}_2(t), v_g = V_g + \hat{v}_g$$

And when substituted into (2.1) they cause its decomposition into two equations, a steady-state one and a small-signal one. The Steady-State equation is,

$$\begin{aligned} \frac{d\bar{\mathbf{x}}}{dt} = & \left[\bar{d}_1(t)\mathbf{A}_1 + \bar{d}_2(t)\mathbf{A}_2 + (1 - \bar{d}_1(t) - \bar{d}_2(t))\mathbf{A}_3 \right] \bar{\mathbf{x}} \\ & + \left[\bar{d}_1(t)\mathbf{b}_1 + \bar{d}_2(t)\mathbf{b}_2 + (1 - \bar{d}_1(t) - \bar{d}_2(t))\mathbf{b}_3 \right] V_g \end{aligned} \quad (2.5)$$

While the small signal equation is,

$$\begin{aligned} \frac{d\hat{\mathbf{x}}}{dt} = & \left[\bar{d}_1(t)\mathbf{A}_1 + \bar{d}_2(t)\mathbf{A}_2 + (1 - \bar{d}_1(t) - \bar{d}_2(t))\mathbf{A}_3 \right] \hat{\mathbf{x}} \\ & + \left[\bar{d}_1(t)\mathbf{b}_1 + \bar{d}_2(t)\mathbf{b}_2 + (1 - \bar{d}_1(t) - \bar{d}_2(t))\mathbf{b}_3 \right] \hat{v}_g \\ & + \left[(\mathbf{b}_1 - \mathbf{b}_3)V_g - (\mathbf{A}_1 - \mathbf{A}_3)\bar{\mathbf{X}}(t) \right] \hat{d}_1(t) \\ & + \left[(\mathbf{b}_2 - \mathbf{b}_3)V_g - (\mathbf{A}_2 - \mathbf{A}_3)\bar{\mathbf{X}}(t) \right] \hat{d}_2(t) \end{aligned} \quad (2.6)$$

The small signals $\hat{d}_1(t)$ and $\hat{d}_2(t)$ are the small signal perturbations of the switching signal and are as shown in figure 2-2. So now the stage is set for the derivation of a small-signal discrete model from equation (2.6). But first operation of the converter at steady state needs to be considered. This is due to the fact that $\bar{\mathbf{X}}(t)$ is a component of equation (2.6).

2.3 Motivation for Harmonic Analysis

Throughout the literature (1)-(6) the small ripple approximation is incorporated into the analysis either for obtaining the steady-state operation point for its own sake or because the derived small-signal dynamic model requires it as an input parameter. But while the small ripple approximation is reasonable for any capacitor voltage in the converter it certainly is not for inductor currents. In the context of a continuous average model (1) errors from this assumption will not be significant since all state variables are averaged anyway, but it will make a difference for models (discrete or not) that depend on the values of the states at different instants of time in one switching cycle. The model derived in this dissertation needs the values of the states at the instants where the converter switches from one mode to another, and that is the justification for the analysis presented in later sections.

2.4 Fourier Series

Going back to equation (2.5) since $\bar{\mathbf{x}}$ is the steady state vector and is periodic with respect to time. Fourier analysis can be applied to it and it can be represented in the following fourier series form,

$$\bar{\mathbf{x}} = \sum_{n=-\infty}^{\infty} \left(\bar{\mathbf{X}}_n e^{j \frac{2\pi n}{T_s} t} \right) \quad (2.7)$$

where $\bar{\mathbf{X}}_n$ is the n^{th} complex Fourier coefficient (harmonic) of $\bar{\mathbf{x}}(t)$. And in a similar fashion other periodic functions of time in (2.5) can be decomposed as,

$$V_g = \sum_{n=-\infty}^{\infty} \left(V_{g_n} e^{j \frac{2\pi n}{T_s} t} \right) \quad (2.8)$$

where ,

$$V_{g_n} = \frac{1}{T_s} \int_0^{T_s} V_g(t) e^{-j \frac{2\pi n}{T_s} t} dt \quad (2.9)$$

Also,

$$\bar{d}_1(t) = \sum_{n=-\infty}^{\infty} \left(\bar{D}_{1n} e^{j \frac{2\pi n}{T_s} t} \right) \quad (2.10)$$

$$\bar{d}_2(t) = \sum_{n=-\infty}^{\infty} \left(\bar{D}_{2n} e^{j \frac{2\pi n}{T_s} t} \right) \quad (2.11)$$

Where,

$$\overline{D}_{1n} = \begin{cases} D_1 & , n = 0 \\ j \frac{e^{-jD_1 2\pi n} - 1}{2\pi n} & , \text{Otherwise} \end{cases} \quad (2.12)$$

$$\overline{D}_{2n} = \begin{cases} D_2 & , n = 0 \\ j \frac{e^{-j(D_1+\alpha)2\pi n} [e^{-jD_2 2\pi n} - 1]}{2\pi n} & , \text{Otherwise} \end{cases} \quad (2.13)$$

Complex harmonics in (2.12) and (2.13) are plotted in Figure 2-3 and Figure 2-4 for n=-20 till n=20 , it can be noted that higher coefficients approach zero in magnitude , but not in a fast manner.

So, substituting (2.7) - (2.13) into (2.5) , and also entering the differentiation operator into the summation on the left hand side gives,

$$\begin{aligned} \sum_{n=-\infty}^{\infty} \left(j \frac{2\pi n}{T_s} \overline{\mathbf{X}}_n e^{j \frac{2\pi n}{T_s} t} \right) = & \left[\sum_{n=-\infty}^{\infty} \left(\overline{D}_{1n} e^{j \frac{2\pi n}{T_s} t} [\mathbf{A}_1 - \mathbf{A}_3] \right) + \sum_{n=-\infty}^{\infty} \left(\overline{D}_{2n} e^{j \frac{2\pi n}{T_s} t} [\mathbf{A}_2 - \mathbf{A}_3] \right) + \mathbf{A}_3 \right] \left[\sum_{n=-\infty}^{\infty} \left(\overline{\mathbf{X}}_n e^{j \frac{2\pi n}{T_s} t} \right) \right] \\ & + \left[\sum_{n=-\infty}^{\infty} \left(\overline{D}_{1n} e^{j \frac{2\pi n}{T_s} t} [\mathbf{b}_1 - \mathbf{b}_3] \right) + \sum_{n=-\infty}^{\infty} \left(\overline{D}_{2n} e^{j \frac{2\pi n}{T_s} t} [\mathbf{b}_2 - \mathbf{b}_3] \right) + \mathbf{b}_3 \right] \left[\sum_{n=-\infty}^{\infty} \left(V_{g_n} e^{j \frac{2\pi n}{T_s} t} \right) \right] \end{aligned} \quad (2.14)$$

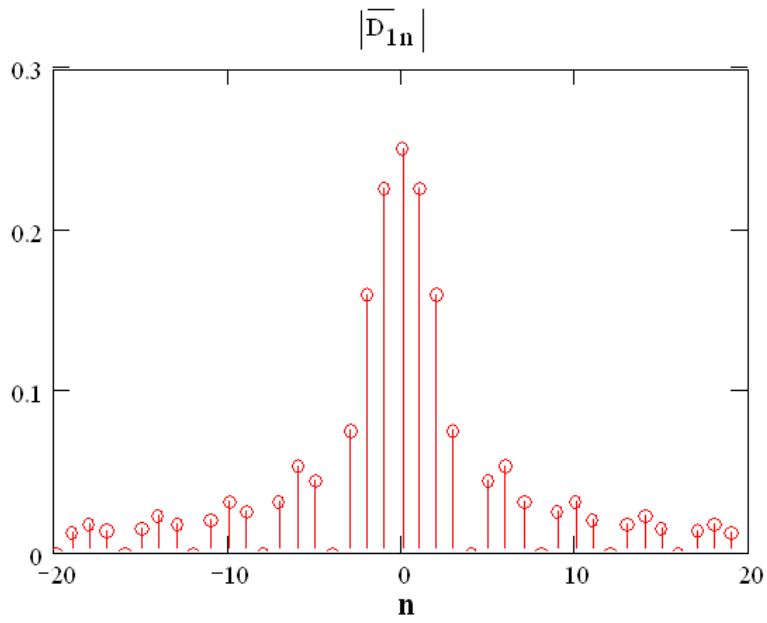


Figure 2-3 : Magnitude of Fourier Coefficients of $\overline{d}_1(t)$

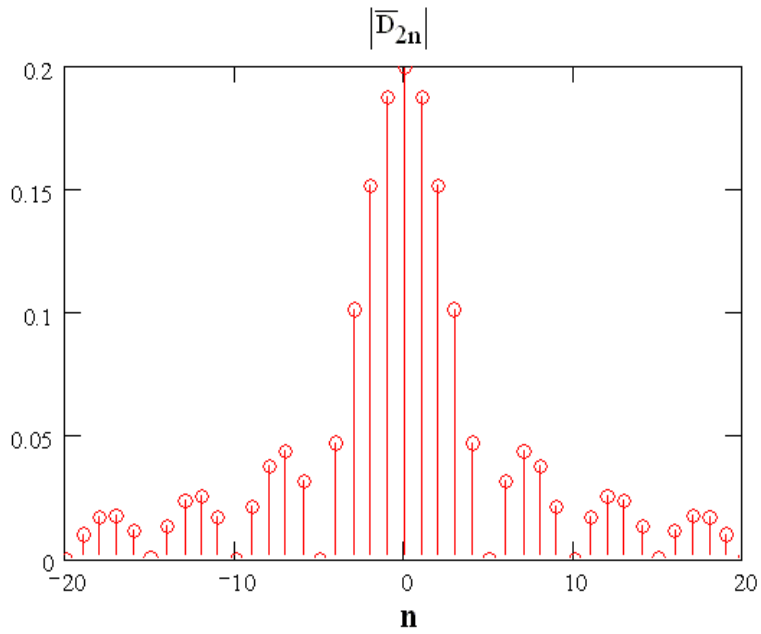


Figure 2-4 : Magnitude of Fourier Coefficients of $\overline{d}_2(t)$

After some algebraic and summation sign and index manipulations, the process leads to,

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} \left(j \frac{2\pi n}{T_s} \bar{\mathbf{X}}_n e^{j \frac{2\pi n}{T_s} t} \right) &= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \left(\bar{D}_{1n} [\mathbf{A}_1 - \mathbf{A}_3] \bar{\mathbf{X}}_m e^{j \frac{2\pi n}{T_s} t} e^{j \frac{2\pi m}{T_s} t} \right) \\
&+ \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \left(\bar{D}_{2n} [\mathbf{A}_2 - \mathbf{A}_3] \bar{\mathbf{X}}_m e^{j \frac{2\pi n}{T_s} t} e^{j \frac{2\pi m}{T_s} t} \right) \\
&+ \sum_{m=-\infty}^{\infty} \left(\mathbf{A}_3 \bar{\mathbf{X}}_m e^{j \frac{2\pi m}{T_s} t} \right) \\
&+ \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \left(\bar{D}_{1n} [\mathbf{b}_1 - \mathbf{b}_3] V_{g_m} e^{j \frac{2\pi n}{T_s} t} e^{j \frac{2\pi m}{T_s} t} \right) \\
&+ \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \left(\bar{D}_{2n} [\mathbf{b}_2 - \mathbf{b}_3] V_{g_m} e^{j \frac{2\pi n}{T_s} t} e^{j \frac{2\pi m}{T_s} t} \right) \\
&+ \sum_{m=-\infty}^{\infty} \left(\mathbf{b}_3 V_{g_m} e^{j \frac{2\pi m}{T_s} t} \right)
\end{aligned} \tag{2.15}$$

2.5 Theoretical acquisition of Harmonics of steady-state waveforms

What is derived up to now can be reduced to a simpler form by applying the following kernel (2.16) to both sides of (2.15). This particular kernel was used because it represents a dot product with a complex exponential function that is orthogonal to the exponential functions in the Fourier series only for some components on each side of (2.15), otherwise its value is zero. This greatly reduces the complexity of (2.15). The kernel is shown below,

$$K\{f(t)\} = \frac{1}{T_s} \int_0^{T_s} f(t) e^{-j \frac{2\pi k}{T_s} t} dt \tag{2.16}$$

and the orthogonality property is demonstrated by,

$$\frac{1}{T_s} \int_0^{T_s} e^{j\frac{2\pi n}{T_s}t} e^{-j\frac{2\pi k}{T_s}t} dt = \begin{cases} 1 & , n = k \\ 0 & , n \neq k \end{cases} \quad (2.17)$$

and,

$$\frac{1}{T_s} \int_0^{T_s} e^{j\frac{2\pi n}{T_s}t} e^{j\frac{2\pi m}{T_s}t} e^{-j\frac{2\pi k}{T_s}t} dt = \begin{cases} 1 & , n + m = k \\ 0 & , n + m \neq k \end{cases} \quad (2.18)$$

So, after applying the aforementioned kernel equation (2.15) reduces to,

$$\begin{aligned} j\frac{2\pi k}{T_s} \bar{\mathbf{X}}_k &= \sum_{m=-\infty}^{\infty} \bar{D}_{1k-m} [\mathbf{A}_1 - \mathbf{A}_3] \bar{\mathbf{X}}_m + \sum_{m=-\infty}^{\infty} \bar{D}_{2k-m} [\mathbf{A}_2 - \mathbf{A}_3] \bar{\mathbf{X}}_m + \mathbf{A}_3 \bar{\mathbf{X}}_k \\ &+ \sum_{m=-\infty}^{\infty} \bar{D}_{1k-m} [\mathbf{b}_1 - \mathbf{b}_3] V_{g_m} + \sum_{m=-\infty}^{\infty} \bar{D}_{2k-m} [\mathbf{b}_2 - \mathbf{b}_3] V_{g_m} + \mathbf{b}_3 V_{g_k} \end{aligned} \quad (2.19)$$

Since the input voltage V_g is almost constant in value and varies at a much slower rate than the switching frequency, the substitution $V_{g_n}=0$ for $n \neq 0$ can be applied. So again equation (2.19) reduces to,

$$\begin{aligned} \sum_{m=-\infty}^{\infty} \left[\bar{D}_{1k-m} [\mathbf{A}_1 - \mathbf{A}_3] + \bar{D}_{2k-m} [\mathbf{A}_2 - \mathbf{A}_3] \right] \bar{\mathbf{X}}_m + \left[\mathbf{A}_3 - j\frac{2\pi k}{T_s} \mathbf{I} \right] \bar{\mathbf{X}}_k \\ = - \left[\bar{D}_{1k} [\mathbf{b}_1 - \mathbf{b}_3] + \bar{D}_{2k} [\mathbf{b}_2 - \mathbf{b}_3] \right] V_{g_0} - \mathbf{b}_3 V_{g_k} \end{aligned} \quad (2.20)$$

For the purpose of simplifying the notation the following defined substitutions are introduced,

$$\gamma_{\mathbf{A},k-m} = \bar{D}_{1k-m} [\mathbf{A}_1 - \mathbf{A}_3] + \bar{D}_{2k-m} [\mathbf{A}_2 - \mathbf{A}_3] \quad (2.21)$$

$$\gamma_{\mathbf{b},k} = \bar{D}_{1k} [\mathbf{b}_1 - \mathbf{b}_3] + \bar{D}_{2k} [\mathbf{b}_2 - \mathbf{b}_3] \quad (2.22)$$

$$\mathbf{\Omega}_k = \mathbf{A}_3 - j \frac{2\pi k}{T_s} \mathbf{I} \quad (2.23)$$

So finally (2.20) reduces to,

$$\sum_{m=-\infty}^{\infty} \gamma_{A,k-m} \bar{\mathbf{X}}_m + \mathbf{\Omega}_k \bar{\mathbf{X}}_k = -\gamma_{b,k} V_{go} - \mathbf{b}_3 V_{gk} \quad (2.24)$$

Since any converter system is a physical system this means that it essentially is low pass and higher frequency harmonics will approach zero. This allows the reduction of the infinite series in (2.24) into a finite one. Assuming that it is required to acquire N harmonics then k can be swept in integer increments form -N till +N. Thus giving the following set of vector equations,

$$\begin{aligned} k = -N &\Rightarrow (\gamma_{A,o} + \mathbf{\Omega}_{-N}) \bar{\mathbf{X}}_{-N} + \gamma_{A,-1} \bar{\mathbf{X}}_{-N+1} + \gamma_{A,-2} \bar{\mathbf{X}}_{-N+2} + \dots \\ &\quad + \gamma_{A,-N} \bar{\mathbf{X}}_o + \gamma_{A,-N-1} \bar{\mathbf{X}}_1 + \dots + \gamma_{A,-2N} \bar{\mathbf{X}}_N = -\gamma_{b,-N} V_{go} \\ k = -N + 1 &\Rightarrow \gamma_{A,1} \bar{\mathbf{X}}_{-N} + (\gamma_{A,o} + \mathbf{\Omega}_{-N+1}) \bar{\mathbf{X}}_{-N+1} + \gamma_{A,-1} \bar{\mathbf{X}}_{-N+2} + \dots \\ &\quad + \gamma_{A,-N+1} \bar{\mathbf{X}}_o + \gamma_{A,-N} \bar{\mathbf{X}}_1 + \dots + \gamma_{A,-2N+1} \bar{\mathbf{X}}_N = -\gamma_{b,-N+1} V_{go} \\ k = -N + 2 &\Rightarrow \gamma_{A,2} \bar{\mathbf{X}}_{-N} + \gamma_{A,1} \bar{\mathbf{X}}_{-N+1} + (\gamma_{A,o} + \mathbf{\Omega}_{-N+2}) \bar{\mathbf{X}}_{-N+2} + \dots \\ &\quad + \gamma_{A,-N+2} \bar{\mathbf{X}}_o + \gamma_{A,-N+1} \bar{\mathbf{X}}_1 + \dots + \gamma_{A,-2N+2} \bar{\mathbf{X}}_N = -\gamma_{b,-N+2} V_{go} \\ &\dots\dots\dots \\ &\dots\dots\dots \\ k = 0 &\Rightarrow \gamma_{A,N} \bar{\mathbf{X}}_{-N} + \gamma_{A,N-1} \bar{\mathbf{X}}_{-N+1} + \gamma_{A,N-2} \bar{\mathbf{X}}_{-N+2} + \dots \\ &\quad + (\gamma_{A,o} + \mathbf{\Omega}_o) \bar{\mathbf{X}}_o + \gamma_{A,-1} \bar{\mathbf{X}}_1 + \dots + \gamma_{A,-N} \bar{\mathbf{X}}_N = (-\gamma_{b,0} - \mathbf{b}_3) V_{go} \\ &\dots\dots\dots \\ k = N &\Rightarrow \gamma_{A,2N} \bar{\mathbf{X}}_{-N} + \gamma_{A,2N-1} \bar{\mathbf{X}}_{-N+1} + \gamma_{A,2N-2} \bar{\mathbf{X}}_{-N+2} + \dots \\ &\quad + \gamma_{A,N} \bar{\mathbf{X}}_o + \gamma_{A,N-1} \bar{\mathbf{X}}_1 + \dots + (\gamma_{A,o} + \mathbf{\Omega}_N) \bar{\mathbf{X}}_N = -\gamma_{b,N} V_{go} \end{aligned} \quad (2.25)$$

Equations (2.25) can be accumulated into one system of linear equations as shown below,

$$\begin{bmatrix}
 \gamma_{A,o} + \Omega_{-N} & \gamma_{A,-1} & \gamma_{A,-2} & \cdots & \gamma_{A,-N} & \gamma_{A,-N-1} & \cdots & \gamma_{A,-2N} \\
 \gamma_{A,1} & \gamma_{A,o} + \Omega_{-N+1} & \gamma_{A,-1} & \cdots & \gamma_{A,-N+1} & \gamma_{A,-N} & \cdots & \gamma_{A,-2N+1} \\
 \gamma_{A,2} & \gamma_{A,1} & \gamma_{A,o} + \Omega_{-N+2} & \cdots & \gamma_{A,-N+2} & \gamma_{A,-N+1} & \cdots & \gamma_{A,-2N+2} \\
 \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 \gamma_{A,N} & \gamma_{A,N-1} & \gamma_{A,N-2} & \cdots & \gamma_{A,o} + \Omega_o & \gamma_{A,-1} & \cdots & \gamma_{A,-N} \\
 \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 \gamma_{A,2N} & \gamma_{A,2N-1} & \gamma_{A,2N-2} & \cdots & \gamma_{A,N} & \gamma_{A,N-1} & \cdots & \gamma_{A,o} + \Omega_N
 \end{bmatrix}
 \begin{bmatrix}
 \bar{\mathbf{X}}_{-N} \\
 \cdots \\
 \bar{\mathbf{X}}_o \\
 \cdots \\
 \bar{\mathbf{X}}_N
 \end{bmatrix}
 =
 \begin{bmatrix}
 -\gamma_{b,-N} \\
 \cdots \\
 -\gamma_{b,o} - \mathbf{b}_3 \\
 \cdots \\
 -\gamma_{b,N}
 \end{bmatrix}
 \quad (2.26)$$

All that remains is to solve the system of linear equations shown in (2.26) to obtain the Fourier complex coefficients for each state in the DC/DC converter circuit. Those coefficients can then be utilized to construct the steady-state waveforms by using (2.7). Any linear algebra computer package can be used for solving (2.7), but MATLAB was utilized for obtaining the results presented in this dissertation. The matrix on the left-hand side of (2.7) can be large depending on the number needed harmonics. For example if the converter has 6 state-variables (6th degree system) and 20 harmonics are needed then the matrix will be a 246 x 246, but still such a matrix equation will just need a fraction of a second to solve in modern computers, so this is much faster than iterative simulation packages like Spice. A note about accuracy though, (2.7) should not be solved by matrix inversion because its large size will compound round off error, rather it should be solved by Gaussian Elimination which gives machine accuracy and is available in MATLAB

2.6 Implementation in MATLAB

Sample MATLAB code is presented in this section as an example of how its scripting language can be relatively easily implemented towards solving (2.7). Results from such an implementation are plotted in chapter 5.

```

function sswaveforms=SteadyStatesim(D1,D2,alpha,Ts,A1,A2,A3,b1,b2,b3,C,Vg,N,t)
%Generates CCM steady state waveforms given
%state-space matrices for different modes.

n=-2.*N:1:2.*N;
n=n+(n==0).*realmin;
D1fc=j.*(exp(-j.*2.*pi.*n.*D1)-1)./2./pi./n;
D2fc=j.*exp(-j.*2.*pi.*n.*(D1+alpha))./2./pi./n.*(exp(-j.*2.*pi.*n.*D2)-1);

temp1=[];
btemp=[];
for k=0:(2.*N),
    temp2=[];
    for m=k:-1:(-2.*N+k),
        gamma=D1fc(2.*N+1+m).*(A1-A3)+D2fc(2.*N+1+m).*(A2-A3);
        if m==0,
            omega=A3-j.*2.*pi.*(-N+k)./Ts.*eye(size(A3));
        else
            omega=0;
        end

        temp2=[temp2 gamma+omega];
    end

    temp1=[temp1 ; temp2];
    gamma=D1fc(N+1+k).*(b1-b3)+D2fc(N+1+k).*(b2-b3);

    if k==N,
        omega=b3;
    else
        omega=0;
    end

    btemp=[btemp;-(gamma+omega).*Vg];
end

AA=temp1;
clear temp1;

bb=btemp;
clear btemp;
fcoef=AA\bb;
fcoef=reshape(fcoef,length(b1),2.*N+1);
temp=0;

n=diag(-N:1:N);
t=ones(2.*N+1,1)*t;
t=n*t;
t=exp(t.*2.*pi./Ts.*j);
sswaveforms=fcoef*t;

```

Figure 2-5 : Harmonic Analysis implementation using MATLAB

CHAPTER 3

HARMONIC ANALYSIS FOR THE ACQUISITION OF TRANSIENT WAVEFORMS FOR PERIODOCALLY VARYING LOAD

3.1 Motivation

The previous Chapter of this Dissertation dealt with the case of DC/DC converters in steady state and with steady load, for which a Harmonic analysis technique was used to acquire steady-State waveforms. An important expansion of the methodology developed there would be generalizing it so it can be applied for the acquisition of the dynamic response of DC/DC converters at step load. Obtaining waveforms in the varying load case facilitates investigating implementing algorithms that search for the optimal duty cycle sequence for specific load transients, as will be discussed in this chapter. After arriving at the optimal or close to optimal duty cycle for given switching load conditions lookup table techniques or system identification are two options for generating that optimal signal.

3.2 Acquisition of Step load response from discrete Duty Cycle Signal

Reexamining the general switching equation (3.1) first presented in the previous chapter the question presents itself of how to go through manipulating it to deal with the case of varying output power load. The kind of load for the purpose of this analysis will be limited to periodically varying step load, since the instant step presents a worst case scenario, were the actual system will have a finite slew rate. Also, the need for considering the load to be periodically varying stems from the desire of applying the harmonic analysis technique.

$$\begin{aligned} \frac{d\mathbf{x}}{dt} = & \left[d(t)\mathbf{A}_1 + (1-d(t))\mathbf{A}_2 \right] \mathbf{x} \\ & + \left[d(t)\mathbf{b}_1 + (1-d(t))\mathbf{b}_2 \right] \mathbf{u} \end{aligned} \quad (3.1)$$

The varying load will enter the switching equation through the vector \mathbf{u} which represents all inputs to the converter. Although in exposition only the load current will be made to vary in the theory the derived method can be used for the case of varying input voltage too. The above implies that the load current is modeled as an input to the converter, Figure 3-1 shows two different approaches of modeling the transient load as an input.

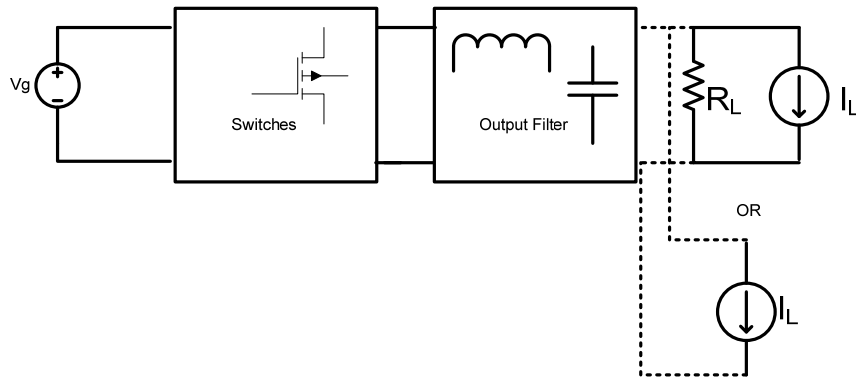


Figure 3-1 : Modeling the load as input alone or with parallel resistance

It is possible to model the load as just an input source to simplify the dynamic equations but that is not recommended, because real life loads have an output impedance. Also accounting for a load resistance helps to stabilize the model response by absorbing some of the circulating energy in the output filter, and without it the model may give an oscillating response which does not conform to the actual behavior of a real device. The point here is that the complexity introduced by a system parameter that when is varied makes the system Time Variant rather than LTI is overcome by remodeling that parameter as an input (In this case a Current Source instead of a passive impedance). The difference between the current waveforms in the response of the converter is slight, yet the insight gained by such a simplification of load behavior is immense, since time varying systems are orders of magnitude harder to analyze than Linear time invariant systems, due to the impossibility of finding a general set of basis for the solution space.

Figure 3-2 shows what waveform the source representing the output load follows. The Load is assumed to be a periodic function of period T_L that switches between two defined current values I_1 and I_2 . This introduction of periodicity in the waveform of the load allows the

application of harmonic analysis techniques, and is a good way to retrieve the periodical nature of all system signals, albeit with period T_L rather than T_s as in the previous chapter.

To keep both T_L and T_s in sync by making $T_L=M*T_s$ where M is an integer simplifies the analysis, and is not a limitation because T_L is an introduced artifact anyway, and can be assumed as large as computationally possible. As mentioned, all signals in the system are periodic, and furthermore T_L must be picked to allow the states of the converter to approach steady state before the arrival of the next load step instant. Ultimately the response of the converter states to that step from steady state and their transition back to steady state is what we are trying calculate.

Since T_L is an integer multiple of T_s it is possible to decompose the duty cycle signal $d(t)$ into M components. Each of which is periodic and repeats every T_L , as shown in Figure 3-3, and which can be written as,

$$d(t) = \sum_{k=1}^{T_L/T_s} d_k(t) \quad (3.2)$$

Where,

$$d_k(t) = \begin{cases} 1 & \text{if } (k-1)T_s < t < (k-1+D_k)T_s \\ 0 & \text{Otherwise} \end{cases} \quad (3.3)$$

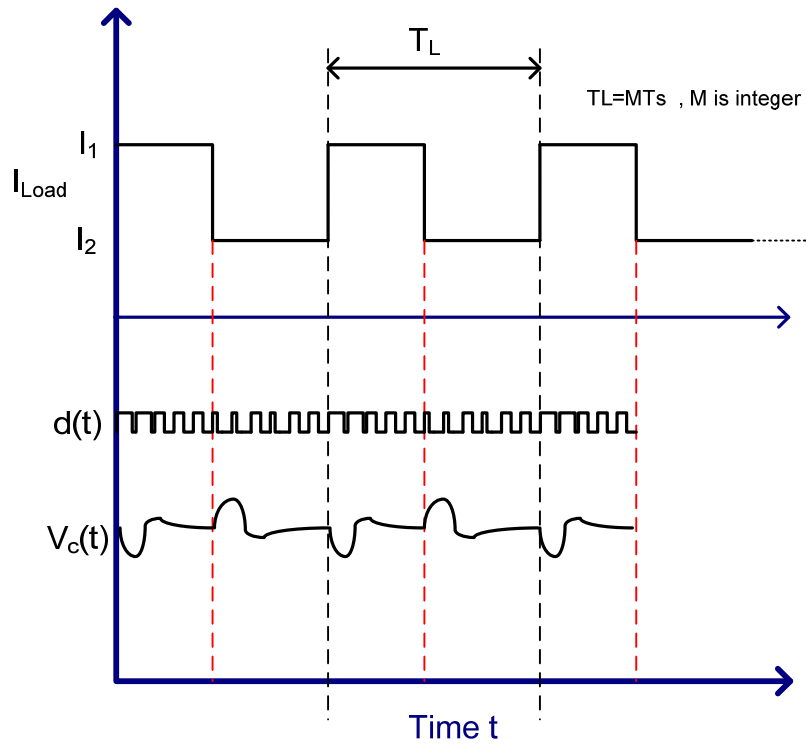


Figure 3-2 : Output Load current assumed waveform for application of Harmonic Analysis technique

As was done in the previous chapter, this periodic duty cycle function can be expressed in terms of its Fourier series coefficients, as shown,

$$d_k(t) = \sum_{n=-\infty}^{T_L/T_s} \left(\mathbf{D}_{k,n} e^{j \frac{2\pi n}{T_L} t} \right) \quad (3.4)$$

Where,

$$D_{k,n} = \begin{cases} \frac{T_s}{T_L} D_k & , n = 0 \\ \frac{e^{-j\frac{2\pi n(k-1)T_s}{T_L}} - e^{-j\frac{2\pi n(k-1+D_k)T_s}{T_L}}}{2j\pi n} & , \text{otherwise} \end{cases} \quad (3.5)$$

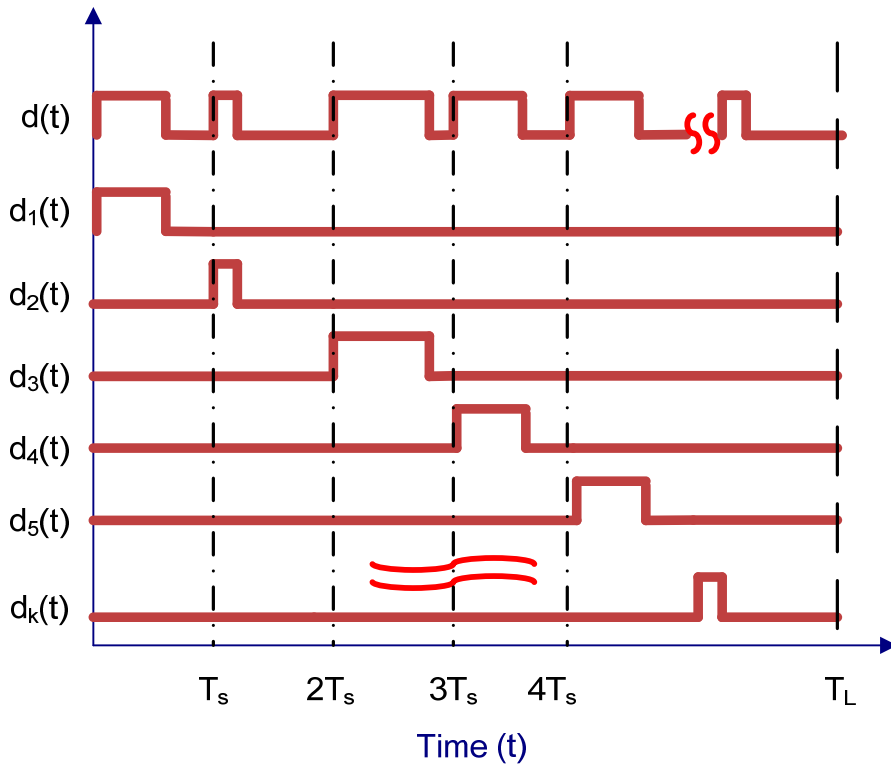


Figure 3-3 : Decomposition of duty cycle signal d(t)

It can be noted that if as an example we assume constant duty cycle D_{ss} the $d_k(t)$ components present in the Fourier domain share the same magnitude naturally but the difference lies in the linear phase slope introduced by the constant time delay in each period T_L .

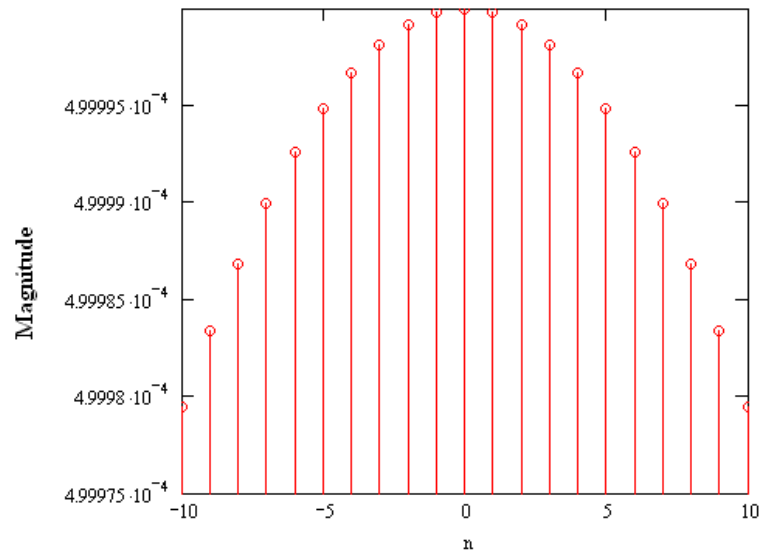
Other time signals that are expanded into their Fourier series form are the state and input vectors of the state space representation,

$$\bar{\mathbf{x}} = \sum_{n=-\infty}^{\infty} \left(\mathbf{X}_n e^{j\frac{2\pi n}{T_L} t} \right) \quad (3.6)$$

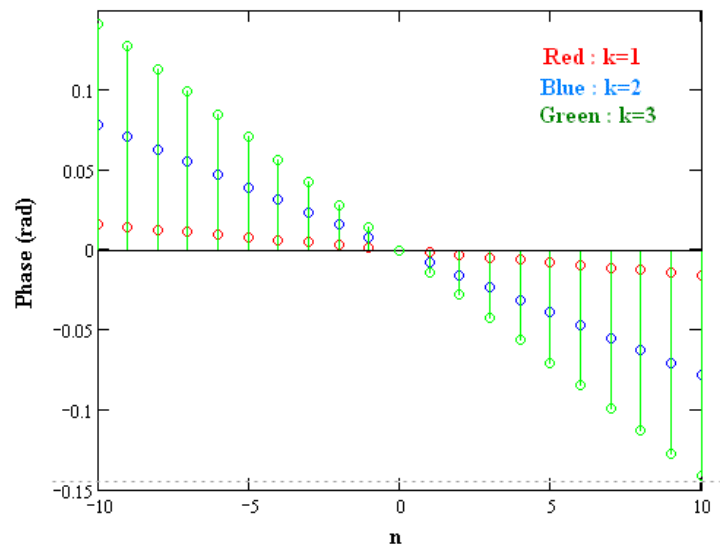
$$\bar{\mathbf{u}} = \sum_{n=-\infty}^{\infty} \left(\mathbf{U}_n e^{j\frac{2\pi n}{T_L} t} \right) \quad (3.7)$$

Equation (3.6) decomposes the state vector which is the unknown waveform we desire to solve for while equation (3.7), decomposes the vector representing the inputs to the converter which are totally determined. The coefficients U_n are found through,

$$\mathbf{U}_n = \frac{1}{T_L} \int_0^{T_L} \bar{\mathbf{u}}(t) e^{-j\frac{2\pi n}{T_L} t} dt \quad (3.8)$$



(a)



(b)

Figure 3-4 : Magnitude (a) and Phase (b) of Fourier coefficients of each $d_k(t)$ (D constant)

So substituting equations (3.2),(3.4),(3.6) and (3.7) into the general switching model (3.1) gives,

$$\begin{aligned} \sum_{n=-\infty}^{\infty} \left(j \frac{2\pi n}{T_L} \bar{\mathbf{X}}_n e^{j \frac{2\pi n}{T_L} t} \right) &= \left[\sum_{k=1}^{T_L/T_s} \sum_{m=-\infty}^{\infty} D_{k,n} e^{j \frac{2\pi m}{T_L} t} (\mathbf{A}_1 - \mathbf{A}_2) + \mathbf{A}_2 \right] \sum_{n=-\infty}^{\infty} \left(\bar{\mathbf{X}}_n e^{j \frac{2\pi n}{T_L} t} \right) \\ &+ \left[\sum_{k=1}^{T_L/T_s} \sum_{m=-\infty}^{\infty} D_{k,n} e^{j \frac{2\pi m}{T_L} t} (\mathbf{b}_1 - \mathbf{b}_2) + \mathbf{b}_2 \right] \sum_{n=-\infty}^{\infty} \left(\mathbf{U}_n e^{j \frac{2\pi n}{T_L} t} \right) \end{aligned} \quad (3.9)$$

After some algebraic and summation sign manipulation we arrive at,

$$\begin{aligned} \sum_{n=-\infty}^{\infty} \left(j \frac{2\pi n}{T_L} \bar{\mathbf{X}}_n e^{j \frac{2\pi n}{T_L} t} \right) &= \sum_{k=1}^{T_L/T_s} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left(D_{k,m} (\mathbf{A}_1 - \mathbf{A}_2) \bar{\mathbf{X}}_n e^{j \frac{2\pi m}{T_L} t} e^{j \frac{2\pi n}{T_L} t} \right) \\ &+ \sum_{n=-\infty}^{\infty} \left(\mathbf{A}_2 \bar{\mathbf{X}}_n e^{j \frac{2\pi n}{T_L} t} \right) \\ &+ \sum_{k=1}^{T_L/T_s} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left(D_{k,m} (\mathbf{b}_1 - \mathbf{b}_2) \mathbf{U}_n e^{j \frac{2\pi m}{T_L} t} e^{j \frac{2\pi n}{T_L} t} \right) \\ &+ \sum_{n=-\infty}^{\infty} \left(\mathbf{b}_2 \mathbf{U}_n e^{j \frac{2\pi n}{T_L} t} \right) \end{aligned} \quad (3.10)$$

Now we can apply the kernel operator described below to (3.10),

$$K\{f(t)\} = \frac{1}{T_L} \int_0^{T_L} f(t) e^{-j \frac{2\pi q}{T_L} t} dt \quad (3.11)$$

Where q is a dummy integer variable that can be anything. This reduces equation (3.10) it to the much more manageable form below,

$$\begin{aligned}
\left(j \frac{2\pi q}{T_L} \bar{\mathbf{X}}_q \right) &= \sum_{k=1}^{T_L/T_s} \sum_{n=-\infty}^{\infty} D_{k,q-n} (\mathbf{A}_1 - \mathbf{A}_2) \bar{\mathbf{X}}_n \\
&\quad + \mathbf{A}_2 \bar{\mathbf{X}}_q \\
&\quad + \sum_{k=1}^{T_L/T_s} \sum_{n=-\infty}^{\infty} D_{k,q-n} (\mathbf{b}_1 - \mathbf{b}_2) \mathbf{U}_n \\
&\quad + \mathbf{b}_2 \mathbf{U}_q
\end{aligned} \tag{3.12}$$

Rearranging the above equation,

$$\left(j \frac{2\pi q}{T_L} \mathbf{I} - \mathbf{A}_2 \right) \bar{\mathbf{X}}_q - \sum_{n=-\infty}^{\infty} \sum_{k=1}^{T_L/T_s} D_{k,q-n} (\mathbf{A}_1 - \mathbf{A}_2) \bar{\mathbf{X}}_n = \mathbf{b}_2 \mathbf{U}_q + \sum_{n=-\infty}^{\infty} \sum_{k=1}^{T_L/T_s} D_{k,q-n} (\mathbf{b}_1 - \mathbf{b}_2) \mathbf{U}_n \tag{3.13}$$

A number of definitions can introduced,

$$\gamma_{\mathbf{A},q-n} = \sum_{k=1}^{T_L/T_s} D_{k,q-n} (\mathbf{A}_1 - \mathbf{A}_2) \tag{3.14}$$

$$\gamma_{\mathbf{b},q-n} = \sum_{k=1}^{T_L/T_s} D_{k,q-n} (\mathbf{b}_1 - \mathbf{b}_2) \tag{3.15}$$

$$\mathbf{\Omega}_q = j \frac{2\pi q}{T_L} \mathbf{I} - \mathbf{A}_2 \tag{3.16}$$

Which yields,

$$\mathbf{\Omega}_q \bar{\mathbf{X}}_q - \sum_{n=-\infty}^{\infty} \gamma_{\mathbf{A},q-n} \bar{\mathbf{X}}_n = \mathbf{b}_2 \mathbf{U}_q + \sum_{n=-\infty}^{\infty} \gamma_{\mathbf{b},q-n} \mathbf{U}_n \tag{3.17}$$

Since q in equation (3.17) is a dummy integer, therefore the equation remains valid for any value of q , a limitless number of equations can be generated. But the purpose is to solve for the unknown coefficient vectors X_n , of which there is an infinite quantity. The trick here is to recognize that real physical signals are low pass therefore the higher frequency Fourier coefficients approach zero. This allows us to reduce the range in the Fourier series summation to a finite value N , so (3.17) becomes,

$$\mathbf{\Omega}_q \bar{X}_q - \sum_{n=-N}^N \gamma_{a,q-n} \bar{X}_n = \mathbf{b}_2 U_q + \sum_{n=-N}^N \gamma_{b,q-n} U_n \quad (3.18)$$

Since equation (3.18) has $N+1$ vectors, and each vector is of a length equal to the order of the system O . (most DC/DC converters are second order). It is apparent that there are $(N+1)*O$ unknowns and therefore q must vary between $-N$ to N to yield the necessary system of linear equations that must be solved to give X_n for all n . This step is exactly the same as described in chapter 2. After finding the Fourier coefficients the time domain signals can be obtained from equation (3.6).

3.3 Finding Optimal discrete Duty Cycle signal for certain load pulse

The previous section addressed the problem of finding the response of a converter if a specific disturbance and duty cycle control sequence is known, this section will investigate the inverse problem. That is the problem of finding a duty cycle signal that reproduces a desired response, or produces a response that is as close as possible to the desired response.

Reexamining equation (3.10), after applying the kernel this alternate form that introduces the shift in the index to the states and input rather than the duty cycle can be derived,

$$\left(j \frac{2\pi q}{T_L} \mathbf{I} - \mathbf{A}_2 \right) \bar{\mathbf{X}}_q - \mathbf{b}_2 \mathbf{U}_q = \sum_{k=1}^{T_L/T_s} \sum_{m=-\infty}^{\infty} D_{k,m} (\mathbf{A}_1 - \mathbf{A}_2) \bar{\mathbf{X}}_{q-m} + \sum_{k=1}^{T_L/T_s} \sum_{m=-\infty}^{\infty} D_{k,m} (\mathbf{b}_1 - \mathbf{b}_2) \mathbf{U}_{q-m} \quad (3.19)$$

It is noticed that the term $D_{k,m}$ in equation (3.19) is an exponential function of the individual duty cycles as shown in equation (3.5). Therefore what was obtained here is a set of nonlinear equations in the duty cycle ratios for any arbitrary response. The previous can be solved but requires a lot of processing time, and is highly inefficient to process. An alternate approach is to investigate the response of the converter in the time domain using the discrete equation, and make use of the fact that the curves of the response waveforms are almost straight lines.

The discrete equation governing the response of any generic equation is,

$$\begin{aligned}
\mathbf{x}(kT_s) &= e^{\mathbf{A}_2(1-D_k)T_s} e^{\mathbf{A}_1 D_k T_s} \mathbf{x}((k-1)T_s) \\
&+ e^{\mathbf{A}_2(1-D_k)T_s} \int_{(k-1)T_s}^{(k-1+D_k)T_s} e^{((k-1+D_k)T_s-\lambda)\mathbf{A}_1} \mathbf{B}_1 \mathbf{u}(\lambda) d\lambda \\
&+ \int_{(k-1+D_k)T_s}^{kT_s} e^{(kT_s-\lambda)\mathbf{A}_2} \mathbf{B}_2 \mathbf{u}(\lambda) d\lambda
\end{aligned} \tag{3.20}$$

This equation will be examined further later in this dissertation, but for the purpose of this chapter the Exponential transition matrices can be simplified by truncating their Taylor series expansion.

Assuming variables T_1 to T_n which are the time lengths the converter is at a certain mode (i.e. Converter is in mode 1 at $0 < t < T_1$, in mode 2 at $T_1 < t < T_1 + T_2$, in mode 3 at $T_1 + T_2 < t < T_1 + T_2 + T_3$), equation (3.20) can be rewritten as,

$$\begin{aligned}
\mathbf{x}\left(\sum_{k=0}^n T_k\right) &= e^{\mathbf{A}_n T_n} \mathbf{x}\left(\sum_{k=0}^{n-1} T_k\right) \\
&+ \int_{\sum_{k=0}^{n-1} T_k}^{\sum_{k=0}^n T_k} e^{\left(\sum_{k=0}^n T_k - \lambda\right)\mathbf{A}_n} d\lambda \mathbf{B}_n \mathbf{u}_n
\end{aligned} \tag{3.21}$$

So using,

$$\int_{t_o'}^{t'} e^{\mathbf{A}(t'-\lambda)} \mathbf{b} \mathbf{u}(\lambda) d\lambda = -\mathbf{A}^{-1} \left[\mathbf{I} - e^{\mathbf{A}(t'-t_o')} \right] \mathbf{b} \mathbf{U} \tag{3.22}$$

And,

$$e^{\mathbf{A}_n T_n} \approx \mathbf{I} + \mathbf{A}_n T_n \quad (3.23)$$

Note that equation (3.22) implies that the integration range can be shifted in time by an arbitrary value and still not affect the value of the integration, therefore that all gives,

$$\begin{aligned} \mathbf{x} \left(\sum_{k=0}^n T_k \right) &= [\mathbf{I} + \mathbf{A}_n T_n] \mathbf{x} \left(\sum_{k=0}^{n-1} T_k \right) \\ &+ \int_{\sum_{k=0}^{n-1} T_k}^{\sum_{k=0}^n T_k} \left[\mathbf{I} + \left(\sum_{k=0}^n T_k - \lambda \right) \mathbf{A}_n \right] d\lambda \mathbf{B}_n \mathbf{u}_n \end{aligned} \quad (3.24)$$

Evaluating the integral and simplifying finally gives,

$$\begin{aligned} \mathbf{x} \left(\sum_{k=0}^n T_k \right) &= [\mathbf{I} + \mathbf{A}_n T_n] \mathbf{x} \left(\sum_{k=0}^{n-1} T_k \right) \\ &+ \left[T_n \mathbf{I} + \frac{T_n^2}{2} \mathbf{A}_n \right] \mathbf{B}_n \mathbf{u}_n \end{aligned} \quad (3.25)$$

The equation relating the state as a function of all T_i 's and the initial condition can be derived by induction, and is found to be,

$$\begin{aligned}
\mathbf{x}\left(\sum_{k=0}^n T_k\right) &= [\mathbf{I} + \mathbf{A}_n T_n][\mathbf{I} + \mathbf{A}_{n-1} T_{n-1}][\mathbf{I} + \mathbf{A}_{n-2} T_{n-2}] \dots [\mathbf{I} + \mathbf{A}_1 T_1] \mathbf{x}(0) \\
&\quad + T_1 [\mathbf{I} + \mathbf{A}_n T_n] \dots [\mathbf{I} + \mathbf{A}_4 T_4][\mathbf{I} + \mathbf{A}_3 T_3][\mathbf{I} + \mathbf{A}_2 T_2] \left[\mathbf{I} + \frac{T_1}{2} \mathbf{A}_1 \right] \mathbf{B}_1 \mathbf{u}_1 \\
&\quad + T_2 [\mathbf{I} + \mathbf{A}_n T_n] \dots [\mathbf{I} + \mathbf{A}_4 T_4][\mathbf{I} + \mathbf{A}_3 T_3] \left[\mathbf{I} + \frac{T_2}{2} \mathbf{A}_2 \right] \mathbf{B}_2 \mathbf{u}_2 \\
&\quad + T_3 [\mathbf{I} + \mathbf{A}_n T_n] \dots [\mathbf{I} + \mathbf{A}_4 T_4] \left[\mathbf{I} + \frac{T_3}{2} \mathbf{A}_3 \right] \mathbf{B}_3 \mathbf{u}_3 \\
&\quad + \dots \\
&\quad + \dots \\
&\quad + T_{n-1} [\mathbf{I} + \mathbf{A}_n T_n] \left[\mathbf{I} + \frac{T_{n-1}}{2} \mathbf{A}_{n-1} \right] \mathbf{B}_{n-1} \mathbf{u}_{n-1} \\
&\quad + T_n \left[\mathbf{I} + \frac{T_n}{2} \mathbf{A}_n \right] \mathbf{B}_n \mathbf{u}_n
\end{aligned} \tag{3.26}$$

$$\begin{aligned}
\mathbf{x}\left(\sum_{k=0}^n T_k\right) &= \left(\prod_{q=n}^1 [\mathbf{I} + \mathbf{A}_q T_q] \right) \mathbf{x}(0) \\
&\quad + \sum_{m=1}^{n-1} T_m \left(\prod_{q=n}^{m+1} [\mathbf{I} + \mathbf{A}_q T_q] \right) \left[\mathbf{I} + \frac{T_m}{2} \mathbf{A}_m \right] \mathbf{B}_m \mathbf{u}_m \\
&\quad + T_n \left[\mathbf{I} + \frac{T_n}{2} \mathbf{A}_n \right] \mathbf{B}_n \mathbf{u}_n
\end{aligned} \tag{3.27}$$

While (3.27) is suitable as a starting point, it can be simplified further by the small switching period assumption through cancellation of time terms multiplied together to give,

$$\begin{aligned}
\mathbf{x} \left(\sum_{k=0}^n T_k \right) &= \left[\mathbf{I} + \mathbf{A}_n T_n + \mathbf{A}_{n-1} T_{n-1} + \mathbf{A}_{n-2} T_{n-2} + \dots + \mathbf{A}_1 T_1 \right] \mathbf{x}(0) \\
&+ T_1 \left[\mathbf{I} + \mathbf{A}_n T_n + \dots + \mathbf{A}_4 T_4 + \mathbf{A}_3 T_3 + \mathbf{A}_2 T_2 + \frac{T_1}{2} \mathbf{A}_1 \right] \mathbf{B}_1 \mathbf{u}_1 \\
&+ T_2 \left[\mathbf{I} + \mathbf{A}_n T_n + \dots + \mathbf{A}_4 T_4 + \mathbf{A}_3 T_3 + \frac{T_2}{2} \mathbf{A}_2 \right] \mathbf{B}_2 \mathbf{u}_2 \\
&+ T_3 \left[\mathbf{I} + \mathbf{A}_n T_n + \dots + \mathbf{A}_4 T_4 + \frac{T_3}{2} \mathbf{A}_3 \right] \mathbf{B}_3 \mathbf{u}_3 \\
&+ \dots \\
&+ \dots \\
&+ T_{n-1} \left[\mathbf{I} + \mathbf{A}_n T_n + \frac{T_{n-1}}{2} \mathbf{A}_{n-1} \right] \mathbf{B}_{n-1} \mathbf{u}_{n-1} \\
&+ T_n \left[\mathbf{I} + \frac{T_n}{2} \mathbf{A}_n \right] \mathbf{B}_n \mathbf{u}_n
\end{aligned} \tag{3.28}$$

$$\begin{aligned}
\mathbf{x} \left(\sum_{k=0}^n T_k \right) &= \left[\mathbf{I} + \sum_{q=n}^1 \mathbf{A}_q T_q \right] \mathbf{x}(0) \\
&+ \sum_{m=1}^{n-1} T_m \left[\mathbf{I} + \sum_{q=n}^{m+1} \mathbf{A}_q T_q + \frac{T_m}{2} \mathbf{A}_m \right] \mathbf{B}_m \mathbf{u}_m \\
&+ T_n \left[\mathbf{I} + \frac{T_n}{2} \mathbf{A}_n \right] \mathbf{B}_n \mathbf{u}_n
\end{aligned} \tag{3.29}$$

Our interested lies with the Error function, defined as,

$$\begin{aligned}
\Theta_n (T_1, T_2, T_3, \dots, T_n) &= \left\| \left(\mathbf{x} \left(\sum_{k=0}^n T_k \right) - \mathbf{x}_{ss} \right) \right\| \\
&= \left(\mathbf{x} \left(\sum_{k=0}^n T_k \right) - \mathbf{x}_{ss} \right)^T \left(\mathbf{x} \left(\sum_{k=0}^n T_k \right) - \mathbf{x}_{ss} \right)
\end{aligned} \tag{3.30}$$

And the global error function defined as,

$$\Theta(T_1, T_2, T_3, \dots, T_n) = \sum_{p=1}^n \Theta_p(T_1, T_2, T_3, \dots, T_p) \quad (3.31)$$

All that remains is the minimization of (3.31) using any numerical package like Matlab or Mathematica, noting the following constraint,

$$T_{odd} + T_{even} = T_s \quad (3.32)$$

CHAPTER 4

SMALL AND LARGE SIGNAL DISCRETE DYNAMIC ANALYSIS

4.1 Motivation

The purpose of this chapter is to achieve a small-signal linear time invariant discrete state-space model from (2.6), which although being linear is time variant. Such a model will be useful for obtaining a pulse transfer function for the converter and therefore a frequency response that inherently includes the effects of sampling. Also a large signal discrete model will be derived without any approximations other than that the input voltage is almost constant throughout one switching cycle. The small signal model can then be alternately obtained through Taylor series expansion with cancellation of higher order terms.

4.2 Discrete small signal State-Space Model

The methodology used here for attaining the model is to consider each mode separately and relate the small-signal state of the converter at the end of each mode to the state at its start. Then all the relations are telescoped until the state at a point is described by a function of a previous state exactly one switching cycle in the past and all the duty cycle impulse perturbations in between. This process will be elaborated more by the derivation presented in this section. A discrete model will be arrived at which relates $\hat{x}[(n+1+D_1)T_s]$ to $\hat{x}[(n+D_1)T_s]$, the choice of considering the state sample at $t=(n+D_1)T_s$ is arbitrary , But for cases where an analog

compensator is used for closing the loop it is preferable to consider the sampling time which corresponds with the natural sampling of the PWM modulator.

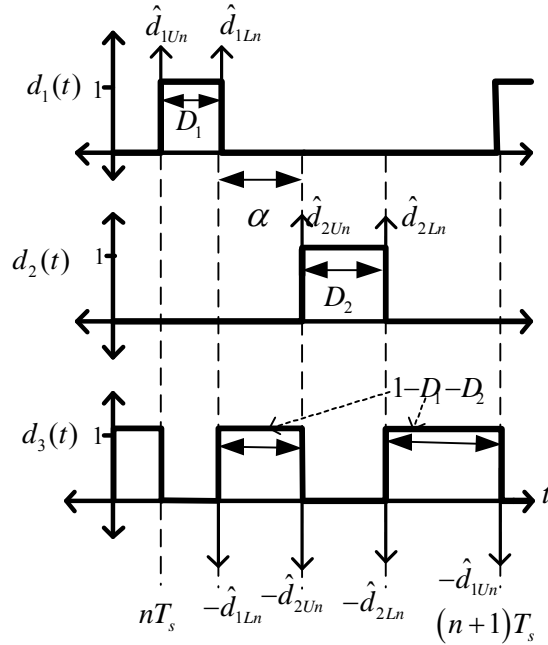


Figure 4-1 : Approximation of Duty cycle perturbations as impulse functions

The Duty cycle perturbations shown in Figure 2-2 can be approximated as weighted impulse functions with weights equal to the amount of perturbation at any edge throughout the pulses. Equations (4.1)-(4.2) describe this approximation as a train of impulses,

$$\hat{d}_1(t) \approx T_s \sum_{n=-\infty}^{\infty} \hat{d}_{1Ln} \delta[t - nT_s] + T_s \sum_{n=-\infty}^{\infty} \hat{d}_{1Un} \delta[t - (n + D_1)T_s] \quad (4.1)$$

$$\begin{aligned} \hat{d}_2(t) \approx & T_s \sum_{n=-\infty}^{\infty} \hat{d}_{2Ln} \delta[t - (n + D_1 + \alpha)T_s] \\ & + T_s \sum_{n=-\infty}^{\infty} \hat{d}_{2Un} \delta[t - (n + D_1 + \alpha + D_2)T_s] \end{aligned} \quad (4.2)$$

The variables \hat{d}_{2Ln} , \hat{d}_{2Un} , \hat{d}_{1Ln} , \hat{d}_{1Un} represent the amount of change at each boundary edge as shown in Figure 4-1. They don't necessarily have equal values, actually their values depend on the type of PWM modulator used. So describing the change in each boundary edge separately like this guarantees the accommodation into the model many diverse PWM generation circuits or even the direct output of a digital controller. Essentially the model will be a four input multiple output State-Space equation.

In (2.6), assuming that $\hat{v}_g = \hat{v}_{gn}$, that is, assuming that the input voltage is almost constant in value through one switching cycle, which is an adequate assumption since any disturbance in the input voltage has a much less frequency than the switching frequency. So, considering (2.6) for $(n+D_1)T_s \leq t \leq (n+D_1+\alpha)T_s$, it becomes,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}_3\hat{\mathbf{x}} + \mathbf{b}_3\hat{v}_{gn} + [(\mathbf{b}_1 - \mathbf{b}_3)V_g - (\mathbf{A}_1 - \mathbf{A}_3)\bar{\mathbf{X}}[t]]\hat{d}_{1Un}T_s\delta[t - (n + D_1)T_s] \quad (4.3)$$

In any region in time where a dynamic system is linear and its states governed by a general state-space equation such as,

$$\frac{d\mathbf{w}}{dt} = \mathbf{A}\mathbf{w}(t) + \mathbf{b}\mathbf{u}(t) \quad (4.4)$$

if the state of the system is known at an initial time t_0 then the solution of (4.4) is given as,

$$\mathbf{w}(t) = e^{A(t-t_o)}\mathbf{w}(t_o) + \int_{t_o}^t e^{A(t-\lambda)}\mathbf{b}\mathbf{u}(\lambda)d\lambda \quad (4.5)$$

So, returning to equation (4.3) it is apparent that $\mathbf{x}[(n+D_1+\alpha)T_s]$, which is at the upper edge of the first piecewise linear region of the converter system, is a function of $\mathbf{x}[(n+D_1)T_s]$ as stated before. So,

$$\begin{aligned} \hat{\mathbf{x}}[(n+D_1+\alpha)T_s] &= e^{\alpha A_3 T_s} \hat{\mathbf{x}}[(n+D_1)T_s] + \left[\int_{(n+D_1)T_s}^{(n+D_1+\alpha)T_s} e^{A_3[(n+D_1+\alpha)T_s-\lambda]} d\lambda \right] \mathbf{b}_3 \hat{v}_{gn} \\ &+ \left(\int_{(n+D_1)T_s}^{(n+D_1+\alpha)T_s} e^{A_3[(n+D_1+\alpha)T_s-\lambda]} \left[(\mathbf{b}_1 - \mathbf{b}_3) V_g - (\mathbf{A}_1 - \mathbf{A}_3) \bar{\mathbf{X}}[\lambda] \right] \delta[\lambda - (n+D_1)T_s] d\lambda \right) \hat{d}_{1Un} T_s \end{aligned} \quad (4.6)$$

And due to the sifting property of the impulse function,

$$\begin{aligned} \hat{\mathbf{x}}[(n+D_1+\alpha)T_s] &= e^{\alpha A_3 T_s} \hat{\mathbf{x}}[(n+D_1)T_s] + \left[\int_{(n+D_1)T_s}^{(n+D_1+\alpha)T_s} e^{A_3[(n+D_1+\alpha)T_s-\lambda]} d\lambda \right] \mathbf{b}_3 \hat{v}_{gn} \\ &+ T_s e^{\alpha A_3 T_s} \left[(\mathbf{b}_1 - \mathbf{b}_3) V_g - (\mathbf{A}_1 - \mathbf{A}_3) \bar{\mathbf{X}}[(n+D_1)T_s] \right] \hat{d}_{1Un} \end{aligned} \quad (4.7)$$

Before moving to the next linear region it is worth noting that there is a certain symmetry to the state-space form of the model that suggests the following definitions which ease the size and tediousness of the notation to a large extent. These definitions are,

$$\Psi_1 = e^{\alpha A_3 T_s} \quad (4.8)$$

$$\Psi_2 = e^{D_2 A_2 T_s} \quad (4.9)$$

$$\Psi_3 = e^{(1-D_1-D_2-\alpha)A_3T_s} \quad (4.10)$$

$$\Psi_4 = e^{D_1A_1T_s} \quad (4.11)$$

and,

$$\Delta_n(\lambda) = (\mathbf{b}_n - \mathbf{b}_3)V_g + (\mathbf{A}_n - \mathbf{A}_3)\bar{\mathbf{X}}(\lambda) \quad (4.12)$$

Also,

$$\mathbf{S}_1 = \int_{(n+D_1)T_s}^{(n+D_1+\alpha)T_s} e^{A_3[(n+D_1+\alpha)T_s-\lambda]} d\lambda \quad (4.13)$$

$$\mathbf{S}_2 = \int_{(n+D_1+\alpha)T_s}^{(n+D_1+\alpha+D_2)T_s} e^{A_2[(n+D_1+\alpha+D_2)T_s-\lambda]} d\lambda \quad (4.14)$$

$$\mathbf{S}_3 = \int_{(n+D_1+\alpha+D_2)T_s}^{(n+1)T_s} e^{A_3[(n+1)T_s-\lambda]} d\lambda \quad (4.15)$$

$$\mathbf{S}_4 = \int_{(n+1)T_s}^{(n+1+D_1)T_s} e^{A_1[(n+1+D_1)T_s-\lambda]} d\lambda \quad (4.16)$$

For the special case where the A matrices in each particular mode of operation are invertible, Equations (4.13)-(4.16) can be written as,

$$\mathbf{S}_1 = [\Psi_1 - \mathbf{I}]A_3^{-1} \quad (4.17)$$

$$\mathbf{S}_2 = [\Psi_2 - \mathbf{I}]A_2^{-1} \quad (4.18)$$

$$\mathbf{S}_3 = [\mathbf{\Psi}_3 - \mathbf{I}] \mathbf{A}_3^{-1} \quad (4.19)$$

$$\mathbf{S}_4 = [\mathbf{\Psi}_4 - \mathbf{I}] \mathbf{A}_1^{-1} \quad (4.20)$$

and finally the following is defined,

$$\mathbf{\Psi}_{n:m} = \prod_{k=0}^{n-m} \mathbf{\Psi}_{n-k} \quad (4.21)$$

$$\mathbf{S}_{n:m} = \prod_{k=0}^{n-m} \mathbf{S}_{n-k} \quad (4.22)$$

Now, continuing the derivation using the definitions (4.8)-(4.22) and substituting them into (4.7) gives,

$$\hat{\mathbf{x}}[(n + D_1 + \alpha)T_s] = \mathbf{\Psi}_1 \hat{\mathbf{x}}[(n + D_1)T_s] + \mathbf{S}_1 \mathbf{b}_3 \hat{v}_{gn} + T_s \mathbf{\Psi}_1 \mathbf{\Delta}_1 [(n + D_1)T_s] \hat{d}_{1Un} \quad (4.23)$$

For the next linear region, $(n + D_1 + \alpha)T_s \leq t \leq (n + D_1 + \alpha + D_2)T_s$,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}_2 \hat{\mathbf{x}} + \mathbf{b}_2 \hat{v}_g + \mathbf{\Delta}_2(t) \hat{d}_{2Ln} T_s \delta[t - (n + D_1 + \alpha)T_s] \quad (4.24)$$

so,

$$\hat{\mathbf{x}}[(n + D_1 + \alpha + D_2)T_s] = \mathbf{\Psi}_2 \hat{\mathbf{x}}[(n + D_1 + \alpha)T_s] + \mathbf{S}_2 \mathbf{b}_2 v_{gn} + T_s \mathbf{\Psi}_2 \mathbf{\Delta}_2 [(n + D_1 + \alpha)T_s] \hat{d}_{2Ln} \quad (4.25)$$

Substituting (4.23) into (4.25) gives,

$$\begin{aligned}\hat{\mathbf{x}}[(n+D_1+\alpha+D_2)T_s] &= \Psi_{2:1}\hat{\mathbf{x}}[(n+D_1)T_s] + (\Psi_2\mathbf{S}_1\mathbf{b}_3 + \mathbf{S}_2\mathbf{b}_2)\hat{v}_{gn} \\ &\quad + T_s\Psi_{2:1}\Delta_1[(n+D_1)T_s]\hat{d}_{1Un} \\ &\quad + T_s\Psi_2\Delta_2[(n+D_1+\alpha)T_s]\hat{d}_{2Ln}\end{aligned}\quad (4.26)$$

Then in the interval $(n+D_1+\alpha+D_2)T_s \leq t \leq (n+1)T_s$,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}_3\hat{\mathbf{x}} + \mathbf{b}_3\hat{v}_g + \Delta_2(t)\hat{d}_{2Un}T_s\delta[t - (n+D_1+\alpha+D_2)T_s]\quad (4.27)$$

So,

$$\hat{\mathbf{x}}[(n+1)T_s] = \Psi_3\hat{\mathbf{x}}[(n+D_1+\alpha+D_2)T_s] + \mathbf{S}_3\mathbf{b}_3\hat{v}_{gn} + T_s\Psi_3\Delta_2[(n+D_1+\alpha+D_2)T_s]\hat{d}_{2Un}\quad (4.28)$$

Again, Substituting (4.26) into (4.28) gives,

$$\begin{aligned}\hat{\mathbf{x}}[(n+1)T_s] &= \Psi_{3:1}\hat{\mathbf{x}}[(n+D_1)T_s] + (\Psi_{3:2}\mathbf{S}_1\mathbf{b}_3 + \Psi_3\mathbf{S}_2\mathbf{b}_2 + \mathbf{S}_3\mathbf{b}_3)\hat{v}_{gn} \\ &\quad + T_s\Psi_{3:1}\Delta_1[(n+D_1)T_s]\hat{d}_{1Un} \\ &\quad + T_s\Psi_{3:2}\Delta_2[(n+D_1+\alpha)T_s]\hat{d}_{2Ln} \\ &\quad + T_s\Psi_3\Delta_2[(n+D_1+\alpha+D_2)T_s]\hat{d}_{2Un}\end{aligned}\quad (4.29)$$

Finally the final linear region is encountered, when $(n+1)T_s \leq t \leq (n+1+D_1)T_s$,

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{A}_1\hat{\mathbf{x}} + \mathbf{b}_1\hat{v}_{gn} + \Delta_1(t)T_s\hat{d}_{1Ln}\delta[t - nT_s]\quad (4.30)$$

so,

$$\hat{\mathbf{x}}[(n+1+D_1)T_s] = \mathbf{\Psi}_4 \hat{\mathbf{x}}[(n+1)T_s] + \mathbf{S}_4 \mathbf{b}_1 \hat{v}_{gn} + T_s \mathbf{\Psi}_4 \mathbf{\Delta}_1 [(n+1)T_s] \hat{d}_{1Ln} \quad (4.31)$$

Then substituting (4.29) into (4.31) yields:

$$\begin{aligned} \hat{\mathbf{x}}[(n+1+D_1)T_s] = & \mathbf{\Psi}_{4:1} \hat{\mathbf{x}}[(n+D_1)T_s] + (\mathbf{\Psi}_{4:2} \mathbf{S}_1 \mathbf{b}_3 + \mathbf{\Psi}_{4:3} \mathbf{S}_2 \mathbf{b}_2 + \mathbf{\Psi}_4 \mathbf{S}_3 \mathbf{b}_3 + \mathbf{S}_4 \mathbf{b}_1) \hat{v}_{gn} \\ & + T_s \mathbf{\Psi}_{4:1} \mathbf{\Delta}_1 [(n+D_1)T_s] \hat{d}_{1Un} \\ & + T_s \mathbf{\Psi}_{4:2} \mathbf{\Delta}_2 [(n+D_1+\alpha)T_s] \hat{d}_{2Ln} \\ & + T_s \mathbf{\Psi}_{4:3} \mathbf{\Delta}_2 [(n+D_1+\alpha+D_2)T_s] \hat{d}_{2Un} \\ & + T_s \mathbf{\Psi}_4 \mathbf{\Delta}_1 [(n+1)T_s] \hat{d}_{1Ln} \end{aligned} \quad (4.32)$$

Equation (4.32) is the general form of the small signal discrete model in state-space form, and it can be tailored for the different control schemes for various topologies by adjusting the values of α , D_1 and D_2 as necessary. For example it can even be used to model the three main topologies (Buck , Boost and Buckboost) or any topology that is controlled by two or less switches and has a freewheeling mode. Of course different topologies can share the same α , D_1 and D_2 parameters depending on how their switches are operated, yet the A and b for each mode should be different.

Table 4-1 : Model parameters substitution for various topologies with different control schemes

Buck	$\alpha = 0$	$D_2 = 1 - D_1$
Boost	$\alpha = 0$	$D_2 = 1 - D_1$
BuckBoost	$\alpha = 0$	$D_2 = 1 - D_1$
Symmetric	$\alpha = \frac{1 - D_1 - D_2}{2}$	$D_2 = D_1$
Asymmetric	$\alpha = 0$	$D_2 = 1 - D_1$
DCS	$\alpha = 0$	$D_2 = D_1$

Equation (4.32) is in the form of a discrete state equation that can be implemented within a linear system software package. Various useful information for stability analysis and compensation can be obtained through the model, such as the frequency response curves, root locus plots or Nyquist diagrams. Moreover, having the discrete model in state space form allows the use of modern control methods such as pole-placement.

It is worth noticing that (4.32) has four small duty cycle modulations as input signals. Normally in a completely digital closed loop system the digital compensator directly controls the modulations in the duty cycle ratios, or its output may be connected to an analog PWM generator. This PWM generator, whether analog or digital is modeled with small signal approximation as a linear operator as shown in (4.33),

$$\begin{bmatrix} \hat{d}_{1un} \\ \hat{d}_{2ln} \\ \hat{d}_{2un} \\ \hat{d}_{1ln} \end{bmatrix} = Mc[n] \quad (4.33)$$

where $M=[m_1 \ m_2 \ m_3 \ m_4]^T$ and $c[n]$ is the output of the discrete compensator or the input of the pulse width modulator. Substituting (4.33) into (4.32) gives a more useful and complete model for pulse transfer function acquisition. Considering what was stated above, and selecting the output voltage of the converter to be the output of the model, for the case where the modulator input to output voltage pulse transfer function is required, the complete representation of the discrete model is,

$$\hat{\mathbf{x}}[n+1] = \mathbf{A}_d \hat{\mathbf{x}}[n] + \mathbf{b}_d c[n] \quad (4.34)$$

$$\hat{y}[n] = \mathbf{C}_d \hat{\mathbf{x}}[n] \quad (4.35)$$

Since the interest is stability analysis it is assumed that $\hat{v}_{gn} = 0$, so,

$$\mathbf{A}_d = \Psi_{4:1} \quad (4.36)$$

and,

$$\mathbf{b}_d = T_s \begin{bmatrix} \Psi_{4:1} \Delta_1 [(n+D_1)T_s] \\ \Psi_{4:2} \Delta_2 [(n+D_1+\alpha)T_s] \\ \Psi_{4:3} \Delta_2 [(n+D_1+\alpha+D_2)T_s] \\ \Psi_4 \Delta_1 [(n+1)T_s] \end{bmatrix} M \quad (4.37)$$

C_d normally depends on the particular topology, and is the C matrix at or just right before the moment of sampling.

Note that $\hat{y}[n]$ represents the perturbation in the converter output voltage. Equations (4.34)-(4.35) are the general state-space representation of a discrete-time system from which finally the desired pulse transfer function is derived in (4.38),

$$\frac{v_o(z)}{c(z)} = C_d [zI - A_d]^{-1} b_d \quad (4.38)$$

4.3 Discrete Large signal State-Space Mode

In the previous sections a small signal State-Space dynamic model was developed, and it utilized approximating the perturbations of the duty cycle functions controlling the converter switches as impulse functions. The main assumption of course is that those perturbations are small in magnitude and the converter states remain close to the steady state operating point. Due to this approximation the large signal dynamics are lost, that is large signal response could not be obtained.

As was done in chapter 3 the basis of the Discrete model is the state space switching model described in equation (2.1). But now the relation between edges of each piecewise linear region is derived before the application of small signal perturbation. Figure 4-2 shows those linear regions between switching instants, and emphasizes that the time of switching between one mode to the next can vary by any amount within some constraints. These constraints are,

$$d_{1un} < (\alpha - d_{2ln}) < (\alpha + D_2 + d_{2un}) < (1 - D_1 - d_{1ln})$$

Since the desired model will be used for large signal discrete simulation it is desirable to choose a sampling instant within the switching cycle that is as close as possible to the average magnitudes of the state variables within that period. Choosing an instant of switching as a beginning point will result in a model that tracks the peaks of the state waveforms so for this dissertation the midpoint of the first duty cycle switching is chosen to be the sampling instant.

So, to begin the derivation process the interval $(n+D_1/2)T_s \leq t \leq (n+D_1+d_{1u})T_s$ is considered where from (2.1),

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_1\mathbf{x} + \mathbf{b}_1\mathbf{u} \quad (4.39)$$

$$\begin{aligned} \mathbf{x}\left[\left(n+D_1+d_{1un}\right)T_s\right] &= e^{\mathbf{A}_1\left(\frac{D_1+d_{1un}}{2}\right)T_s} \mathbf{x}\left[\left(n+\frac{D_1}{2}\right)T_s\right] \\ &+ \left(\int_{\left(n+\frac{D_1}{2}\right)T_s}^{\left(n+D_1+d_{1un}\right)T_s} e^{\mathbf{A}_1\left[\left(n+D_1+d_{1un}\right)T_s-\lambda\right]} d\lambda \right) \mathbf{b}_1\mathbf{u}\left[nT_s\right] \end{aligned} \quad (4.40)$$

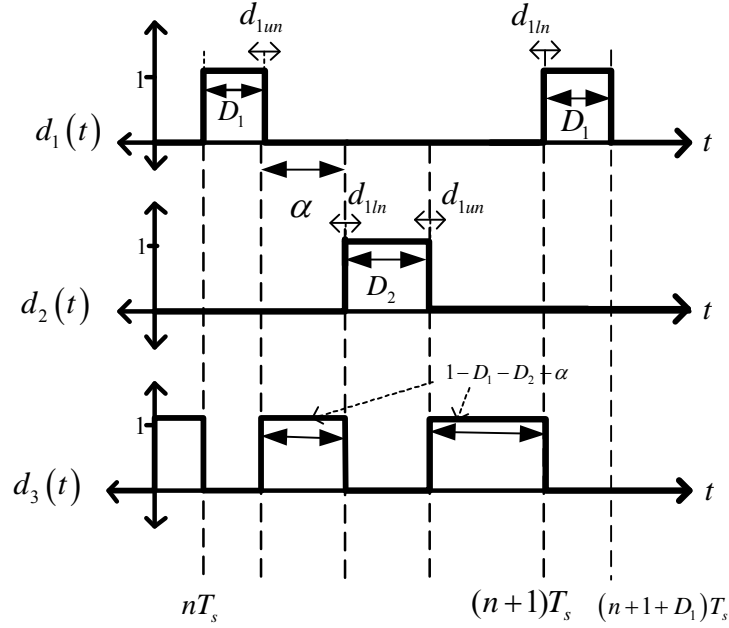


Figure 4-2 : Model Piecewise linear regions with varying switching instants

Now , the next interval to consider is $(n+D_1+d_{1u})T_s \leq t \leq (n+D_1+\alpha-d_{2l})T_s$ where,

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_3 \mathbf{x} + \mathbf{b}_3 \mathbf{u} \quad (4.41)$$

$$\begin{aligned} \mathbf{x}[(n+D_1+\alpha-d_{2l})T_s] &= e^{\mathbf{A}_3(\alpha-d_{2l}-d_{1u})T_s} \mathbf{x}[(n+D_1+d_{1u})T_s] \\ &+ \left(\int_{(n+D_1+d_{1u})T_s}^{(n+D_1+\alpha-d_{2l})T_s} e^{\mathbf{A}_3[(n+D_1+\alpha-d_{2l})T_s-\lambda]} d\lambda \right) \mathbf{b}_3 \mathbf{u}[nT_s] \end{aligned} \quad (4.42)$$

Substituting (4.40) into (4.42) yields ,

$$\begin{aligned}
\mathbf{x}\left[\left(n+D_1+\alpha-d_{2ln}\right)T_s\right] &= e^{A_3(\alpha-d_{2ln}-d_{1ln})T_s} e^{A_1\left(\frac{D_1+d_{1ln}}{2}\right)T_s} \mathbf{x}\left[\left(n+\frac{D_1}{2}\right)T_s\right] \\
&+ \left[e^{A_3(\alpha-d_{2ln}-d_{1ln})T_s} \left(\int_{\left(n+\frac{D_1}{2}\right)T_s}^{(n+D_1+d_{1ln})T_s} e^{A_1[(n+D_1+d_{1ln})T_s-\lambda]} d\lambda \right) \mathbf{b}_1 \right. \\
&\quad \left. + \left(\int_{(n+D_1+d_{1ln})T_s}^{(n+D_1+\alpha-d_{2ln})T_s} e^{A_3[(n+D_1+\alpha-d_{2ln})T_s-\lambda]} d\lambda \right) \mathbf{b}_3 \right] \mathbf{u}[nT_s]
\end{aligned} \tag{4.43}$$

It seems that the form of the discrete difference equation will be of a cumbersome nature from a notational perspective. Yet the same symmetry first encountered in chapter 3 still holds , and the notation can be immensely simplified through making some appropriate definitions as follows,

$$\mathbf{\Psi}_1 = e^{A_1\left(\frac{D_1+d_{1ln}}{2}\right)T_s} \tag{4.44}$$

$$\mathbf{\Psi}_2 = e^{A_3(\alpha-d_{2ln}-d_{1ln})T_s} \tag{4.45}$$

$$\mathbf{\Psi}_3 = e^{A_2(D_2+d_{2ln}+d_{2un})T_s} \tag{4.46}$$

$$\mathbf{\Psi}_4 = e^{A_3(1-D_1-\alpha-D_2-d_{2un}-d_{1ln})T_s} \tag{4.47}$$

$$\mathbf{\Psi}_5 = e^{A_1\left(\frac{D_1+d_{1ln}}{2}\right)T_s} \tag{4.48}$$

also,

$$\mathbf{S}_1 = \int_{\left(n+\frac{D_1}{2}\right)T_s}^{(n+D_1+d_{1un})T_s} e^{\mathbf{A}_1[(n+D_1+d_{1un})T_s-\lambda]} d\lambda \quad (4.49)$$

$$\mathbf{S}_2 = \int_{(n+D_1+d_{1un})T_s}^{(n+D_1+\alpha-d_{2ln})T_s} e^{\mathbf{A}_3[(n+D_1+\alpha-d_{2ln})T_s-\lambda]} d\lambda \quad (4.50)$$

$$\mathbf{S}_3 = \int_{(n+D_1+\alpha-d_{2ln})T_s}^{(n+D_1+\alpha+D_2+d_{2un})T_s} e^{\mathbf{A}_2[(n+D_1+\alpha+D_2+d_{2un})T_s-\lambda]} d\lambda \quad (4.51)$$

$$\mathbf{S}_4 = \int_{(n+D_1+\alpha+D_2+d_{2un})T_s}^{(n+1-d_{1ln})T_s} e^{\mathbf{A}_3[(n+1-d_{1ln})T_s-\lambda]} d\lambda \quad (4.52)$$

$$\mathbf{S}_5 = \int_{(n+1-d_{1ln})T_s}^{\left(n+1+\frac{D_1}{2}\right)T_s} e^{\mathbf{A}_1\left[\left(n+1+\frac{D_1}{2}\right)T_s-\lambda\right]} d\lambda \quad (4.53)$$

The integrals in \mathbf{S}_1 till \mathbf{S}_5 can be evaluated using a numerical integration method such as Simpsons Rule, or the matrix exponential within each integral could be expanded into its Taylor Series and integrated term by term. So in that case,

$$\mathbf{S}_1 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_1^k}{(k+1)!} \left[\left(\frac{D_1}{2} + d_{1un} \right) T_s \right]^{k+1} \quad (4.54)$$

$$\mathbf{S}_2 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_3^k}{(k+1)!} \left[(\alpha - d_{2ln} - d_{1un}) T_s \right]^{k+1} \quad (4.55)$$

$$\mathbf{S}_3 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_2^k}{(k+1)!} \left[(D_2 + d_{2ln} + d_{2un}) T_s \right]^{k+1} \quad (4.56)$$

$$\mathbf{S}_4 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_3^k}{(k+1)!} \left[(1 - D_1 - \alpha - D_2 - d_{2un} - d_{1ln}) T_s \right]^{k+1} \quad (4.57)$$

$$\mathbf{S}_5 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_1^k}{(k+1)!} \left[\left(\frac{D_1}{2} + d_{1ln} \right) T_s \right]^{k+1} \quad (4.58)$$

For the case where the switching period is very small (which is almost always true for switching DC/DC power converters) the terms for $k > 0$ in (4.54) to (4.58) can be eliminated. The most accurate path though is to use an integral numerical method since the speed of convergence of the presented power series depends also on the norms of the \mathbf{A} matrices which is a parameter that varies with the circuit components values. Also the results (4.54) to (4.58) verify that \mathbf{S}_1 till \mathbf{S}_5 are not dependant on n when all the duty cycle perturbations are zero and the converter is in steady state, so n can be set to zero in that case.

So, continuing the same procedure outlined before for the region $(n + D_1 + \alpha - d_{2l})T_s \leq t \leq (n + D_1 + \alpha + D_2 + d_{2u})T_s$ and every following region until the interval where $(n + 1 - d_{1l})T_s \leq t \leq (n + D_1/2)T_s$ is reached, and without repeating the same algebraic manipulation in chapter 2 the followed result is induced,

$$\begin{aligned} \mathbf{x} \left[(n + 1 + D_1) T_s \right] &= \Psi_{5:1} \mathbf{x} \left[(n + D_1) T_s \right] \\ &+ (\Psi_{5:2} \mathbf{S}_1 \mathbf{b}_1 + \Psi_{5:3} \mathbf{S}_2 \mathbf{b}_3 + \Psi_{5:4} \mathbf{S}_3 \mathbf{b}_2 + \Psi_5 \mathbf{S}_4 \mathbf{b}_3 + \mathbf{S}_5 \mathbf{b}_1) \mathbf{u} \left[n T_s \right] \end{aligned} \quad (4.59)$$

Equation (4.59) represents the Large Signal model , it should be noted that for any fixed switching duty cycles the converter behaves as a discrete linear system when the states are a function of the system input (input voltage for example , or perhaps load current). The nonlinearity appears only when the effect of modulating the switching cycles is considered. It is apparent that all the $\Psi_n, \Psi_{n:m}$ and S_n matrices are functions of the duty cycle modulations in a nonlinear manner. The achieved model can be easily implemented using computer linear algebra software such as MATLAB.

4.4 Discrete Small signal State-Space Model without impulse approximation

As was mentioned in section 3, all the $\Psi_n, \Psi_{n:m}$ and S_n matrices are functions of the duty cycle modulations in a nonlinear fashion So the first step for linearizing the Discrete model would be to reduce those matrices to a form that is linear with respect to the cycle modulations. This form comes through the small signal assumption which leads to the desired small signal dynamic model for any steady-state operating point. But before continuing the steady state operating point needs to be found , so going back to (4.21) and assuming that,

$$d_{1u} = d_{2l} = d_{2u} = d_{1l} = 0 \quad (4.60)$$

and noting that the converter reaches steady-state after a certain N when for all $n > N$,

$$\mathbf{x}[n+1] = \mathbf{x}[n] = \bar{\mathbf{X}} \quad (4.61)$$

of course the input also should be constant otherwise a constant steady state won't be reached , the constant input is denoted for all n as,

$$\mathbf{u}[n] = \bar{\mathbf{U}} \quad (4.62)$$

Now , substituting (4.23) and (4.24) into (4.21) and solving gives,

$$\bar{\mathbf{X}} = (\mathbf{I} - \bar{\Psi}_{5:1})^{-1} (\bar{\Psi}_{5:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{5:3} \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{5:4} \bar{\mathbf{S}}_3 \mathbf{b}_2 + \bar{\Psi}_5 \bar{\mathbf{S}}_4 \mathbf{b}_3 + \bar{\mathbf{S}}_5 \mathbf{b}_1) \bar{\mathbf{U}} \quad (4.63)$$

where any matrix that has a bar on it indicates that (4.22) is satisfied. So from this point forward any steady state component of a matrix is represented with the same matrix name but with a bar on it. On the other hand and matrix with a hat on it signifies a matrix with a small signal simplification.

For the small signal analysis a certain simplification of the binomial expansion where one of the variables has small values will be used. The general form of the binomial expansion is,

$$(a + \hat{b})^n = a^n + \binom{n}{1} a^{n-1} \hat{b} + \binom{n}{2} a^{n-2} \hat{b}^2 + \dots + \binom{n}{n-1} a \hat{b}^{n-1} + \hat{b}^n \quad (4.64)$$

where,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (4.65)$$

and since \hat{b} is small when compared to a all high order powers can be eliminated to give,

$$(a + \hat{b})^n \approx a^n + n a^{n-1} \hat{b} \quad (4.66)$$

The line of attack will be to use the approximation in (4.28) and the Taylor series truncation to express the matrices in the discrete model as linear functions of the duty cycle modulations, so each matrix presented in (4.21) is investigated and manipulated to reach the desired result. Then, starting with \mathbf{S}_1 and manipulating it algebraically to get,

$$\mathbf{S}_1 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_1^k}{(k+1)!} \left[\frac{D_1}{2} T_s + \hat{d}_{1un} T_s \right]^{k+1} \quad (4.67)$$

then by applying the binomial approximation and taking the summation sign inside , then doing the appropriate cancellations ,

$$\hat{\mathbf{S}}_1 = \sum_{k=0}^{\infty} \frac{\mathbf{A}_1^k}{(k+1)!} \left(\frac{D_1}{2} T_s \right)^{k+1} + \sum_{k=0}^{\infty} \frac{\mathbf{A}_1^k}{k!} \left(\frac{D_1}{2} T_s \right)^k \hat{d}_{1un} T_s \quad (4.68)$$

so,

$$\hat{\mathbf{S}}_1 = \bar{\mathbf{S}}_1 + \bar{\Psi}_1 T_s \hat{d}_{1un} \quad (4.69)$$

Repeating the same process with the other \mathbf{S} matrices gives,

$$\hat{\mathbf{S}}_2 = \bar{\mathbf{S}}_2 - \bar{\Psi}_2 T_s \hat{d}_{2ln} - \bar{\Psi}_2 T_s \hat{d}_{1un} \quad (4.70)$$

$$\hat{\mathbf{S}}_3 = \bar{\mathbf{S}}_3 + \bar{\Psi}_3 T_s \hat{d}_{2ln} + \bar{\Psi}_3 T_s \hat{d}_{2un} \quad (4.71)$$

$$\hat{\mathbf{S}}_4 = \bar{\mathbf{S}}_4 - \bar{\Psi}_4 T_s \hat{d}_{2un} - \bar{\Psi}_4 T_s \hat{d}_{1ln} \quad (4.72)$$

$$\hat{\mathbf{S}}_5 = \bar{\mathbf{S}}_5 + \bar{\Psi}_5 T_s \hat{d}_{1ln} \quad (4.73)$$

Moving on to the Ψ matrices their Taylor Series expansions can be used to arrive at their small signal forms, for example for Ψ_1 ,

$$\Psi_1 = e^{\mathbf{A}_1 \left(\frac{D_1}{2} + d_{1un} \right) T_s} = e^{\mathbf{A}_1 \frac{D_1}{2} T_s} e^{\mathbf{A}_1 d_{1un} T_s} \quad (4.74)$$

So,

$$\Psi_1 = \bar{\Psi}_1 \sum_{k=0}^{\infty} \frac{\mathbf{A}_1^k}{k!} (d_{1un} T_s)^k \quad (4.75)$$

and if d_{1un} is small all terms where $k > 1$ can be ignored so what remains would be,

$$\hat{\Psi}_1 = \bar{\Psi}_1 \left[\mathbf{I} + \mathbf{A}_1 T_s \hat{d}_{1un} \right] \quad (4.76)$$

and in the same manner,

$$\hat{\Psi}_2 = \bar{\Psi}_2 \left[\mathbf{I} - \mathbf{A}_3 T_s \hat{d}_{2ln} - \mathbf{A}_3 T_s \hat{d}_{1un} \right] \quad (4.77)$$

$$\hat{\Psi}_3 = \bar{\Psi}_3 \left[\mathbf{I} + \mathbf{A}_2 T_s \hat{d}_{2ln} + \mathbf{A}_2 T_s \hat{d}_{2un} \right] \quad (4.78)$$

$$\hat{\Psi}_4 = \bar{\Psi}_4 \left[\mathbf{I} - \mathbf{A}_3 T_s \hat{d}_{2un} - \mathbf{A}_3 T_s \hat{d}_{1ln} \right] \quad (4.79)$$

$$\hat{\Psi}_5 = \bar{\Psi}_5 \left[\mathbf{I} + \mathbf{A}_1 T_s \hat{d}_{1ln} \right] \quad (4.80)$$

If the $\Psi_{5:1}$ matrix is examined in (4.21) after substituting the small signal forms in (4.38) to (4.44) into it the following algebraic manipulations can be done ,

$$\hat{\Psi}_{5:1} = \hat{\Psi}_5 \hat{\Psi}_4 \hat{\Psi}_3 \hat{\Psi}_2 \hat{\Psi}_1 \quad (4.81)$$

but,

$$\hat{\Psi}_5 \hat{\Psi}_4 = \left[\bar{\Psi}_5 + \bar{\Psi}_5 \mathbf{A}_1 T_s \hat{d}_{1ln} \right] \left[\bar{\Psi}_4 - \bar{\Psi}_4 \mathbf{A}_3 T_s \hat{d}_{2un} - \bar{\Psi}_4 \mathbf{A}_3 T_s \hat{d}_{1ln} \right]$$

So,

$$\hat{\Psi}_5 \hat{\Psi}_4 = \bar{\Psi}_{5:4} - \bar{\Psi}_{5:4} \mathbf{A}_3 T_s \hat{d}_{2un} + \left(\bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_4 - \bar{\Psi}_{5:4} \mathbf{A}_3 \right) T_s \hat{d}_{1ln} \quad (4.82)$$

and in a similar manner,

$$\begin{aligned} \hat{\Psi}_5 \hat{\Psi}_4 \hat{\Psi}_3 &= \bar{\Psi}_{5:3} + \bar{\Psi}_{5:3} \mathbf{A}_2 T_s \hat{d}_{2ln} + \left(\bar{\Psi}_{5:3} \mathbf{A}_2 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_3 \right) T_s \hat{d}_{2un} \\ &\quad + \left(\bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:3} - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_3 \right) T_s \hat{d}_{1ln} \end{aligned} \quad (4.83)$$

$$\begin{aligned} \hat{\Psi}_5 \hat{\Psi}_4 \hat{\Psi}_3 \hat{\Psi}_2 &= \bar{\Psi}_{5:2} \\ &\quad - \bar{\Psi}_{5:2} \mathbf{A}_3 T_s \hat{d}_{1un} \\ &\quad + \left(\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_2 - \bar{\Psi}_{5:2} \mathbf{A}_3 \right) T_s \hat{d}_{2ln} \\ &\quad + \left(\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_2 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:2} \right) T_s \hat{d}_{2un} \\ &\quad + \left(\bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:2} - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:2} \right) T_s \hat{d}_{1ln} \end{aligned} \quad (4.84)$$

Finally,

$$\begin{aligned}
\hat{\Psi}_5 \hat{\Psi}_4 \hat{\Psi}_3 \hat{\Psi}_2 \hat{\Psi}_1 &= \bar{\Psi}_{5:1} \\
&+ \left(\bar{\Psi}_{5:1} \mathbf{A}_1 - \bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\Psi}_1 \right) T_s \hat{d}_{1un} \\
&+ \left(\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_{2:1} - \bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\Psi}_1 \right) T_s \hat{d}_{2ln} \\
&+ \left(\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_{2:1} - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:1} \right) T_s \hat{d}_{2un} \\
&+ \left(\bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:1} - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:1} \right) T_s \hat{d}_{1ln}
\end{aligned} \tag{4.85}$$

Now attention is turned to the $(\Psi_{5:2} \mathbf{S}_1 \mathbf{b}_1 + \Psi_{5:3} \mathbf{S}_2 \mathbf{b}_3 + \Psi_{5:4} \mathbf{S}_3 \mathbf{b}_2 + \Psi_5 \mathbf{S}_4 \mathbf{b}_3 + \mathbf{S}_5 \mathbf{b}_1)$ matrix in (4.59), which can be attacked component by component, but first to simplify the analysis the matrix is rearranged as,

$$\left(\Psi_5 \left(\Psi_4 \left(\Psi_3 \left(\Psi_2 \mathbf{S}_1 \mathbf{b}_1 + \mathbf{S}_2 \mathbf{b}_3 \right) + \mathbf{S}_3 \mathbf{b}_2 \right) + \mathbf{S}_4 \mathbf{b}_3 \right) + \mathbf{S}_5 \mathbf{b}_1 \right)$$

So starting with,

$$\hat{\Psi}_2 \hat{\mathbf{S}}_1 \mathbf{b}_1 + \hat{\mathbf{S}}_2 \mathbf{b}_3 \approx \left[\bar{\Psi}_2 - \bar{\Psi}_2 \mathbf{A}_3 T_s \hat{d}_{2ln} - \bar{\Psi}_2 \mathbf{A}_3 T_s \hat{d}_{1un} \right] \left(\bar{\mathbf{S}}_1 + \bar{\Psi}_1 T_s \hat{d}_{1un} \right) \mathbf{b}_1 + \left(\bar{\mathbf{S}}_2 - \bar{\Psi}_2 T_s \hat{d}_{2ln} - \bar{\Psi}_2 T_s \hat{d}_{1un} \right) \mathbf{b}_3$$

gives,

$$\begin{aligned}
\hat{\Psi}_2 \hat{\mathbf{S}}_1 \mathbf{b}_1 + \hat{\mathbf{S}}_2 \mathbf{b}_3 &\approx \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\mathbf{S}}_2 \mathbf{b}_3 \\
&+ \left(\bar{\Psi}_{2:1} \mathbf{b}_1 - \bar{\Psi}_2 \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_2 \mathbf{b}_3 \right) T_s \hat{d}_{1un} \\
&+ \left(-\bar{\Psi}_2 \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_2 \mathbf{b}_3 \right) T_s \hat{d}_{2ln}
\end{aligned} \tag{4.86}$$

and by continuing along this path,

$$\begin{aligned}
\hat{\Psi}_3 \left(\hat{\Psi}_2 \hat{\mathbf{S}}_1 \mathbf{b}_1 + \hat{\mathbf{S}}_2 \mathbf{b}_3 \right) + \hat{\mathbf{S}}_3 \mathbf{b}_2 &= \bar{\Psi}_{3:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_3 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\mathbf{S}}_3 \mathbf{b}_2 \\
&+ \left(\bar{\Psi}_{3:1} \mathbf{b}_1 - \bar{\Psi}_{3:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{3:2} \mathbf{b}_3 \right) T_s \hat{d}_{1un} \\
&+ \left(-\bar{\Psi}_{3:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{3:2} \mathbf{b}_3 + \bar{\Psi}_3 \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_3 \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_3 \mathbf{b}_2 \right) T_s \hat{d}_{2ln} \\
&+ \left(\bar{\Psi}_3 \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_3 \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_3 \mathbf{b}_2 \right) T_s \hat{d}_{2un}
\end{aligned} \tag{4.87}$$

and ,

$$\begin{aligned}
\hat{\Psi}_4 \left(\hat{\Psi}_3 \left(\hat{\Psi}_2 \hat{\mathbf{S}}_1 \mathbf{b}_1 + \hat{\mathbf{S}}_2 \mathbf{b}_3 \right) + \hat{\mathbf{S}}_3 \mathbf{b}_2 \right) &= \\
&\bar{\Psi}_{4:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{4:3} \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_4 \bar{\mathbf{S}}_3 \mathbf{b}_2 \\
&+ \left(\bar{\Psi}_{4:1} \mathbf{b}_1 - \bar{\Psi}_{4:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{4:2} \mathbf{b}_3 \right) T_s \hat{d}_{1un} \\
&+ \left(\begin{array}{l} -\bar{\Psi}_{4:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{4:2} \mathbf{b}_3 + \bar{\Psi}_{4:3} \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 \\ + \bar{\Psi}_{4:3} \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{4:3} \mathbf{b}_2 \end{array} \right) T_s \hat{d}_{2ln} \\
&+ \left(\begin{array}{l} \bar{\Psi}_{4:3} \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{4:3} \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{4:3} \mathbf{b}_2 \\ -\bar{\Psi}_4 \mathbf{A}_3 \bar{\Psi}_{3:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_4 \mathbf{A}_3 \bar{\Psi}_3 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ -\bar{\Psi}_4 \mathbf{A}_3 \bar{\mathbf{S}}_3 \mathbf{b}_2 \end{array} \right) T_s \hat{d}_{2un}
\end{aligned} \tag{4.88}$$

So finally,

$$\begin{aligned}
\hat{\Psi}_5 \left(\hat{\Psi}_4 \left(\hat{\Psi}_3 \left(\hat{\Psi}_2 \hat{\mathbf{S}}_1 \mathbf{b}_1 + \hat{\mathbf{S}}_2 \mathbf{b}_3 \right) + \hat{\mathbf{S}}_3 \mathbf{b}_2 \right) + \hat{\mathbf{S}}_4 \mathbf{b}_3 \right) + \hat{\mathbf{S}}_5 \mathbf{b}_1 = \\
\bar{\Psi}_{5:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{5:3} \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{5:4} \bar{\mathbf{S}}_3 \mathbf{b}_2 + \bar{\Psi}_5 \bar{\mathbf{S}}_4 \mathbf{b}_3 + \bar{\mathbf{S}}_5 \mathbf{b}_1 \\
+ \left(\bar{\Psi}_{5:1} \mathbf{b}_1 - \bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{5:2} \mathbf{b}_3 \right) T_s \hat{d}_{1un} \\
+ \left(\begin{array}{l} -\bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{5:2} \mathbf{b}_3 + \bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 \\ + \bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{5:3} \mathbf{b}_2 \end{array} \right) T_s \hat{d}_{2ln} \\
+ \left(\begin{array}{l} \bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ + \bar{\Psi}_{5:3} \mathbf{b}_2 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_3 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\mathbf{S}}_3 \mathbf{b}_2 - \bar{\Psi}_{5:4} \mathbf{b}_3 \end{array} \right) T_s \hat{d}_{2un} \\
+ \left(\begin{array}{l} \bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:3} \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ + \bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_4 \bar{\mathbf{S}}_3 \mathbf{b}_2 + \bar{\Psi}_5 \mathbf{A}_1 \bar{\mathbf{S}}_4 \mathbf{b}_3 + \bar{\Psi}_5 \mathbf{b}_1 \\ - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_3 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\mathbf{S}}_3 \mathbf{b}_2 - \bar{\Psi}_{5:4} \mathbf{b}_3 \end{array} \right) T_s \hat{d}_{1ln}
\end{aligned} \tag{4.89}$$

After finding the small signal forms for the two matrices in (4.59) its time now to apply small signal perturbation to that equation and cancel out small signal products, so,

$$\begin{aligned}
\bar{\mathbf{X}} + \hat{\mathbf{x}}[n] = \hat{\Psi}_{5:1} \left(\bar{\mathbf{X}} + \hat{\mathbf{x}}[n] \right) \\
+ \left(\hat{\Psi}_{5:2} \hat{\mathbf{S}}_1 \mathbf{b}_1 + \hat{\Psi}_{5:3} \hat{\mathbf{S}}_2 \mathbf{b}_3 + \hat{\Psi}_{5:4} \hat{\mathbf{S}}_3 \mathbf{b}_2 + \hat{\Psi}_5 \hat{\mathbf{S}}_4 \mathbf{b}_3 + \hat{\mathbf{S}}_5 \mathbf{b}_1 \right) \left(\bar{\mathbf{U}} + \hat{\mathbf{u}}[n] \right)
\end{aligned} \tag{4.90}$$

notice that $\bar{\mathbf{X}}[n+1] = \bar{\mathbf{X}}[n] = \bar{\mathbf{X}}$ is constant at steady state and the same can be said about the large signal component of the input to the model, substituting (4.39) and (4.43) into (4.44) and doing all the cancellations and removing all constant components gives the final form of the small signal discrete model,

$$\begin{aligned}
\hat{\mathbf{x}}[n+1] &= \bar{\Psi}_{5:1} \hat{\mathbf{x}}[n] \\
&+ \left(\bar{\Psi}_{5:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{5:3} \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{5:4} \bar{\mathbf{S}}_3 \mathbf{b}_2 + \bar{\Psi}_5 \bar{\mathbf{S}}_4 \mathbf{b}_3 + \bar{\mathbf{S}}_5 \mathbf{b}_1 \right) \hat{\mathbf{u}}[n] \\
&+ \left[\left(\bar{\Psi}_{5:1} \mathbf{A}_1 - \bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\Psi}_1 \right) \bar{\mathbf{X}} + \begin{pmatrix} \bar{\Psi}_{5:1} \mathbf{b}_1 - \bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 \\ -\bar{\Psi}_{5:2} \mathbf{b}_3 \end{pmatrix} \bar{\mathbf{U}} \right] T_s \hat{d}_{1in} \\
&+ \left[\left(\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_{2:1} - \bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\Psi}_1 \right) \bar{\mathbf{X}} + \begin{pmatrix} -\bar{\Psi}_{5:2} \mathbf{A}_3 \bar{\mathbf{S}}_1 \mathbf{b}_1 - \bar{\Psi}_{5:2} \mathbf{b}_3 \\ +\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 \\ +\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 + \bar{\Psi}_{5:3} \mathbf{b}_2 \end{pmatrix} \bar{\mathbf{U}} \right] T_s \hat{d}_{2in} \\
&+ \left[\left(\bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_{2:1} - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:1} \right) \bar{\mathbf{X}} + \begin{pmatrix} \bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\Psi}_2 \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_{5:3} \mathbf{A}_2 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ +\bar{\Psi}_{5:3} \mathbf{b}_2 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 \\ -\bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_3 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ -\bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\mathbf{S}}_3 \mathbf{b}_2 - \bar{\Psi}_{5:4} \mathbf{b}_3 \end{pmatrix} \bar{\mathbf{U}} \right] T_s \hat{d}_{2in} \\
&+ \left[\left(\bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:1} - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:1} \right) \bar{\mathbf{X}} + \begin{pmatrix} \bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 + \bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_{4:3} \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ +\bar{\Psi}_5 \mathbf{A}_1 \bar{\Psi}_4 \bar{\mathbf{S}}_3 \mathbf{b}_2 + \bar{\Psi}_5 \mathbf{A}_1 \bar{\mathbf{S}}_4 \mathbf{b}_3 \\ +\bar{\Psi}_5 \mathbf{b}_1 - \bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_{3:2} \bar{\mathbf{S}}_1 \mathbf{b}_1 \\ -\bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\Psi}_3 \bar{\mathbf{S}}_2 \mathbf{b}_3 \\ -\bar{\Psi}_{5:4} \mathbf{A}_3 \bar{\mathbf{S}}_3 \mathbf{b}_2 - \bar{\Psi}_{5:4} \mathbf{b}_3 \end{pmatrix} \bar{\mathbf{U}} \right] T_s \hat{d}_{1in}
\end{aligned} \tag{4.91}$$

Although the small signal model here looks more complex than the one developed in the previous sections, there is a certain symmetry to it that can be exploited when implementing it in a simulation or linear algebra package. With the arrival at the small signal discrete model with no impulse function approximation this chapter is concluded.

CHAPTER 5

APPLICATION OF DISCRETE MODELING METHOD TO THE HALF-BRIDGE DC/DC TOPOLOGY WITH CURRENT DOUBLER

5.1 Overview of Half-Bridge topology with three main control method

Established control techniques for the Half-Bridge DC/DC power converter are the symmetric and asymmetric (complimentary). While asymmetric control achieves ZVS conditions for converter switches its disadvantages are unbalanced component stresses and the existence of a nonzero DC magnetizing current in the isolation transformer, which increases the risk of transformer saturation and generally has a detrimental effect on the dynamics of the converter. A new control method called DCS PWM (Duty-Cycle Shifted) was proposed in (7) to achieve ZVS for at least one of the Half-Bridge switches, this control method is derived from symmetric control and entails switching S2 almost instantly after S1 turns off (see figure 5-1a). Of course meeting ZVS conditions is desirable because it allows the converter to operate at higher switching frequencies with greater efficiencies. Thus arises the need to accurately close the loop on this topology to meet design specifications, but as was stated in the introduction to this dissertation conventional averaging methods for dynamic modeling yield models that are inaccurate at the upper frequency range because they ignore the sampled nature of the Duty Cycle signals that drive the power stage. Furthermore they predict frequency response content at higher than half the switching frequency.

5.2 State-Space representation of Modes of the Half-Bridge DC/DC topology with current doubler

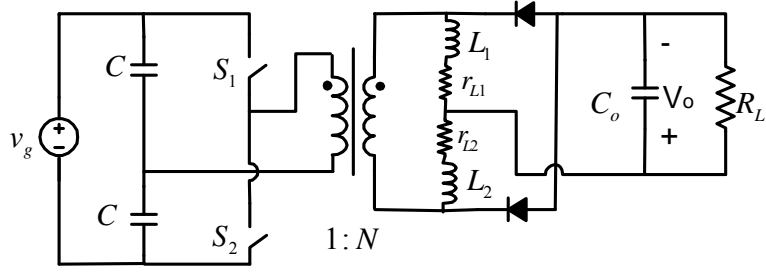
Figure 5-1 shows the typical Half-Bridge topology and its three main modes of operation , Mode III being the freewheeling mode. All parasitics for the output filter and half-bridge capacitors are included. The system can be represented in piecewise state-space form as shown in the chapter 2 of this thesis and the parameters of the discrete model for the different control schemes are shown in Table 5-1, which shows the appropriate substitution of the parameters in equation (3.32)

Table 5-1 : Model parameter substitution for each control scheme

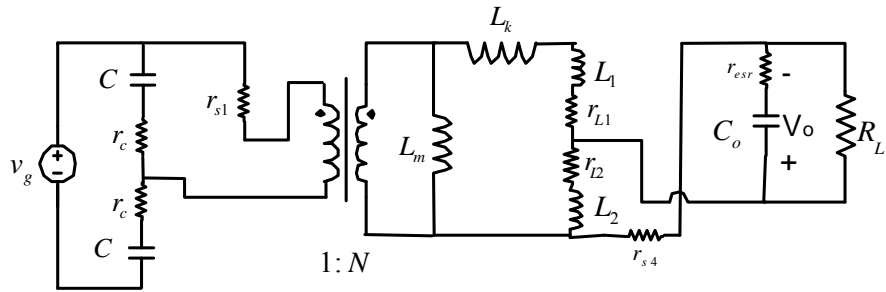
Symmetric	$\alpha = \frac{1-D_1-D_2}{2}$	$D_2 = D_1$
Asymmetric	$\alpha = 0$	$D_2 = 1-D_1$
DCS	$\alpha = 0$	$D_2 = D_1$

So the first step now for applying the small signal model in chapter 3 is to find the A and b matrices for all the modes of operations , not forgetting of course to define the states. This is done in (5.1) till (5.7) shown below,

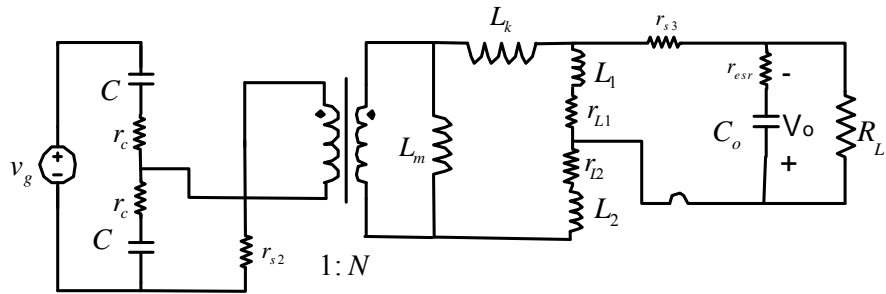
$$\hat{\mathbf{x}}^T = \left[\hat{v}_{c1} \quad \hat{v}_{c2} \quad \hat{i}_{Lm} \quad \hat{i}_{L1} \quad \hat{i}_{L2} \quad \hat{v}_{co} \right] \quad (5.1)$$



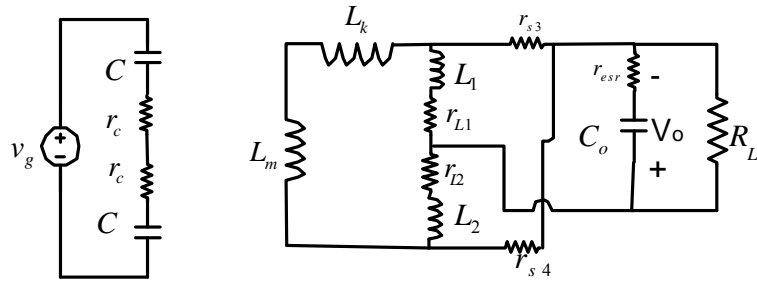
(a)



(b)



(c)



(d)

Figure 5-1 : (a) Isolated Half-Bridge Converter, (b) Mode I (S1 on, S2 off), (c) Mode II (S1 off, S2 on), (d) Mode III (S1 off, S2 off)

$$\mathbf{b}_1^T = \begin{bmatrix} \frac{1}{2Cr_c} & \frac{1}{2Cr_c} & \frac{N}{2L_m} & \frac{N}{2(L_k + L_1)} & 0 & 0 \end{bmatrix} \quad (5.2)$$

$$\mathbf{b}_2^T = \begin{bmatrix} \frac{1}{2Cr_c} & \frac{1}{2Cr_c} & \frac{-N}{2L_m} & 0 & \frac{-N}{2(L_k + L_2)} & 0 \end{bmatrix} \quad (5.3)$$

$$\mathbf{b}_3^T = \begin{bmatrix} \frac{1}{2Cr_c} & \frac{1}{2Cr_c} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.4)$$

and,

$$\mathbf{A}_1 = \begin{bmatrix} \frac{-1}{2Cr_c} & \frac{-1}{2Cr_c} & \frac{-N}{2C} & \frac{-N}{2C} & 0 & 0 \\ \frac{-1}{2Cr_c} & \frac{-1}{2Cr_c} & \frac{N}{2C} & \frac{N}{2C} & 0 & 0 \\ \frac{N}{L_m} & \frac{-N}{L_m} & \frac{-\alpha_1}{2L_m} & \frac{-\alpha_1}{2L_m} & 0 & 0 \\ \frac{N}{2(L_k + L_1)} & \frac{-N}{2(L_k + L_1)} & \frac{-\alpha_1}{2(L_k + L_1)} & \frac{-\left(r_{L1} + \beta_4 + \frac{\alpha_1}{2}\right)}{L_k + L_1} & \frac{\beta_4}{L_k + L_1} & \frac{r_{esr} \| R_L}{r_{esr} (L_k + L_1)} \\ 0 & 0 & 0 & \frac{\beta_4}{L_2} & \frac{-(r_{L2} + \beta_4)}{L_2} & \frac{-(r_{esr} \| R_L)}{r_{esr} L_2} \\ 0 & 0 & 0 & \frac{-(r_{esr} \| R_L)}{r_{esr} C_o} & \frac{r_{esr} \| R_L}{r_{esr} C_o} & \frac{-1}{C_o (r_{esr} + R_L)} \end{bmatrix} \quad (5.5)$$

$$\mathbf{A}_2 = \begin{bmatrix} \frac{-1}{2Cr_c} & \frac{-1}{2Cr_c} & \frac{-N}{2C} & 0 & \frac{-N}{2C} & 0 \\ \frac{-1}{2Cr_c} & \frac{-1}{2Cr_c} & \frac{N}{2C} & 0 & \frac{N}{2C} & 0 \\ \frac{N}{L_m} & \frac{-N}{L_m} & \frac{-\alpha_2}{2L_m} & 0 & \frac{-\alpha_1}{2L_m} & 0 \\ 0 & 0 & 0 & \frac{-(r_{L1} + \beta_3)}{L_1} & \frac{\beta_3}{L_1} & \frac{r_{esr} \| R_L}{r_{esr} L_1} \\ \frac{N}{2(L_k + L_2)} & \frac{-N}{2(L_k + L_2)} & \frac{-\alpha_2}{2(L_k + L_2)} & \frac{\beta_3}{L_k + L_2} & \frac{-\left(r_{L2} + \beta_3 + \frac{\alpha_2}{2}\right)}{L_k + L_2} & \frac{-(r_{esr} \| R_L)}{r_{esr} (L_k + L_2)} \\ 0 & 0 & 0 & \frac{-(r_{esr} \| R_L)}{r_{esr} C_o} & \frac{r_{esr} \| R_L}{r_{esr} C_o} & \frac{-1}{C_o (r_{esr} + R_L)} \end{bmatrix} \quad (5.6)$$

$$\mathbf{A}_3 = \begin{bmatrix} \frac{-1}{2Cr_c} & \frac{-1}{2Cr_c} & 0 & 0 & 0 & 0 \\ \frac{-1}{2Cr_c} & \frac{-1}{2Cr_c} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-(r_{s3} + r_{s4})}{L_m + L_k} & \frac{-r_{s3}}{L_m + L_k} & \frac{-r_{s4}}{L_m + L_k} & 0 \\ 0 & 0 & \frac{-r_{s3}}{L_1} & \frac{-(r_{L1} + \beta_3)}{L_1} & \frac{r_{esr} \parallel R_L}{L_1} & \frac{r_{esr} \parallel R_L}{r_{esr} L_1} \\ 0 & 0 & \frac{-r_{s4}}{L_2} & \frac{r_{esr} \parallel R_L}{L_2} & \frac{-(r_{L2} + \beta_4)}{L_2} & \frac{-(r_{esr} \parallel R_L)}{r_{esr} L_2} \\ 0 & 0 & 0 & \frac{-(r_{esr} \parallel R_L)}{r_{esr} C_o} & \frac{r_{esr} \parallel R_L}{r_{esr} C_o} & \frac{-1}{C_o (r_{esr} + R_L)} \end{bmatrix} \quad (5.7)$$

where,

$$\alpha_n = N^2(2r_{sn} + r_c) \quad (5.8)$$

$$\beta_n = r_{sn} + (r_{esr} \parallel R_L) \quad (5.9)$$

All constants in the previous definitions are referenced in figure 5-1

5.3 Application of Chapter 3 small-signal model to the Half-Bridge DC/DC topology with current doubler

To demonstrate the model developed in this thesis in chapter 3, parameters will be taken from a prototype Half-Bridge converter with current doubler that was built in the lab. It had the following components and parameters,

$$V_{in}=48 \text{ v} , V_o=1 \text{ v} , I_o=20 \text{ A} , F_s=400 \text{ KHz}$$

$$L_1=L_2= 180\text{nH} , C_o=500\mu\text{F} , r_{esr}=0.66\text{m}\Omega , L_m=36 \mu\text{H}$$

$$r_{L1}= r_{L2}=5\text{e-}3 , L_k=3.125\text{nH} , N=1/6 , r_c=10\text{m}\Omega$$

$$r_{s1}=25 \text{ m}\Omega , r_{s2}=25 \text{ m}\Omega , r_{s3}=3 \text{ m}\Omega , r_{s4}=3 \text{ m}\Omega;$$

The first step in applying the model is getting the steady-state waveforms, which can be done using MATLAB as shown in the sample code in section (2.7) . Figures (5-2) till (5-4) shows the results from running this code. The results of the harmonic analysis are verified for the asymmetric case by the simplorer simulation shown in Figure 5-5. After finding the waveforms they are utilized in evaluating the $\Delta_1[(n+D_1)T_s]$, $\Delta_2[(n+D_1+\alpha)T_s]$, $\Delta_2[(n+D_1+\alpha+D_2)T_s]$ and $\Delta_1[(n+1)T_s]$ matrices (or vectors depending on the number of inputs) in (2.32). Figure 5-7 gives an example of how MATLAB can programmed with the discrete model. The resulting frequency response of the model is shown in figure 5-6. Figure 5-8 compare the frequency response obtained from the discrete model with the one obtained from the classical average model.

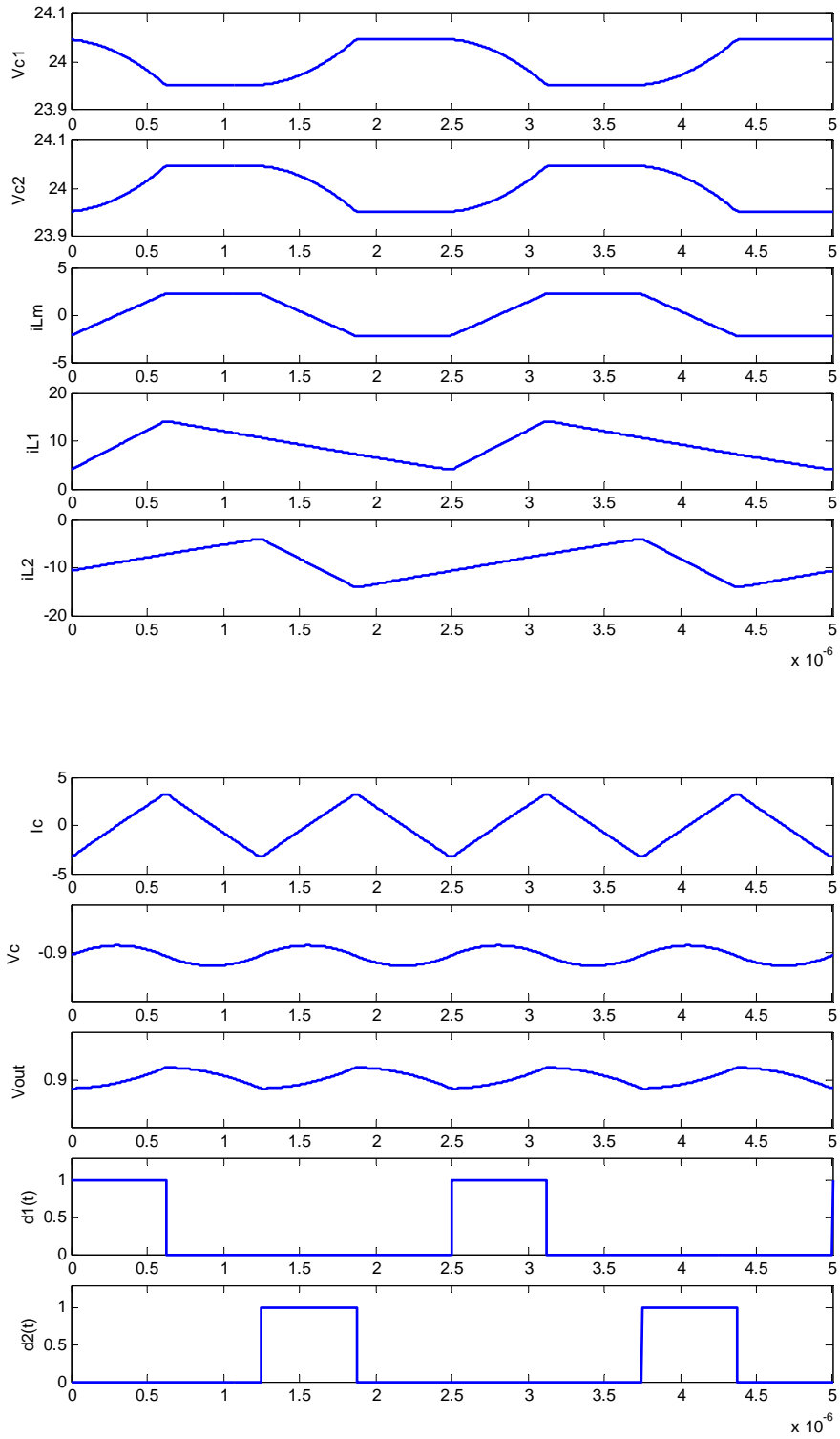


Figure 5-2 : HB DC/DC steady-state waveforms for symmetric control

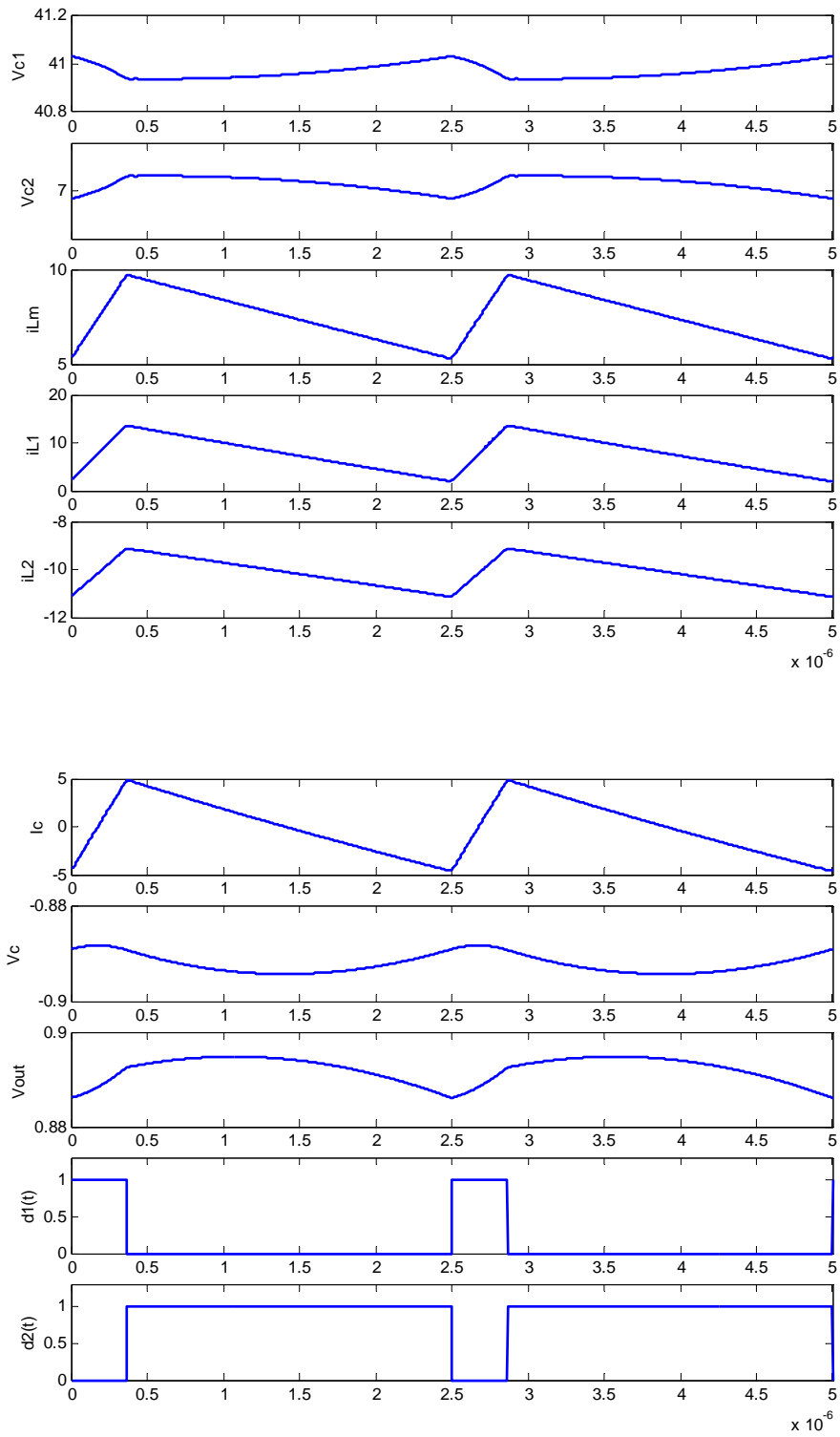


Figure 5-3 : HB DC/DC steady-state waveforms for asymmetric control

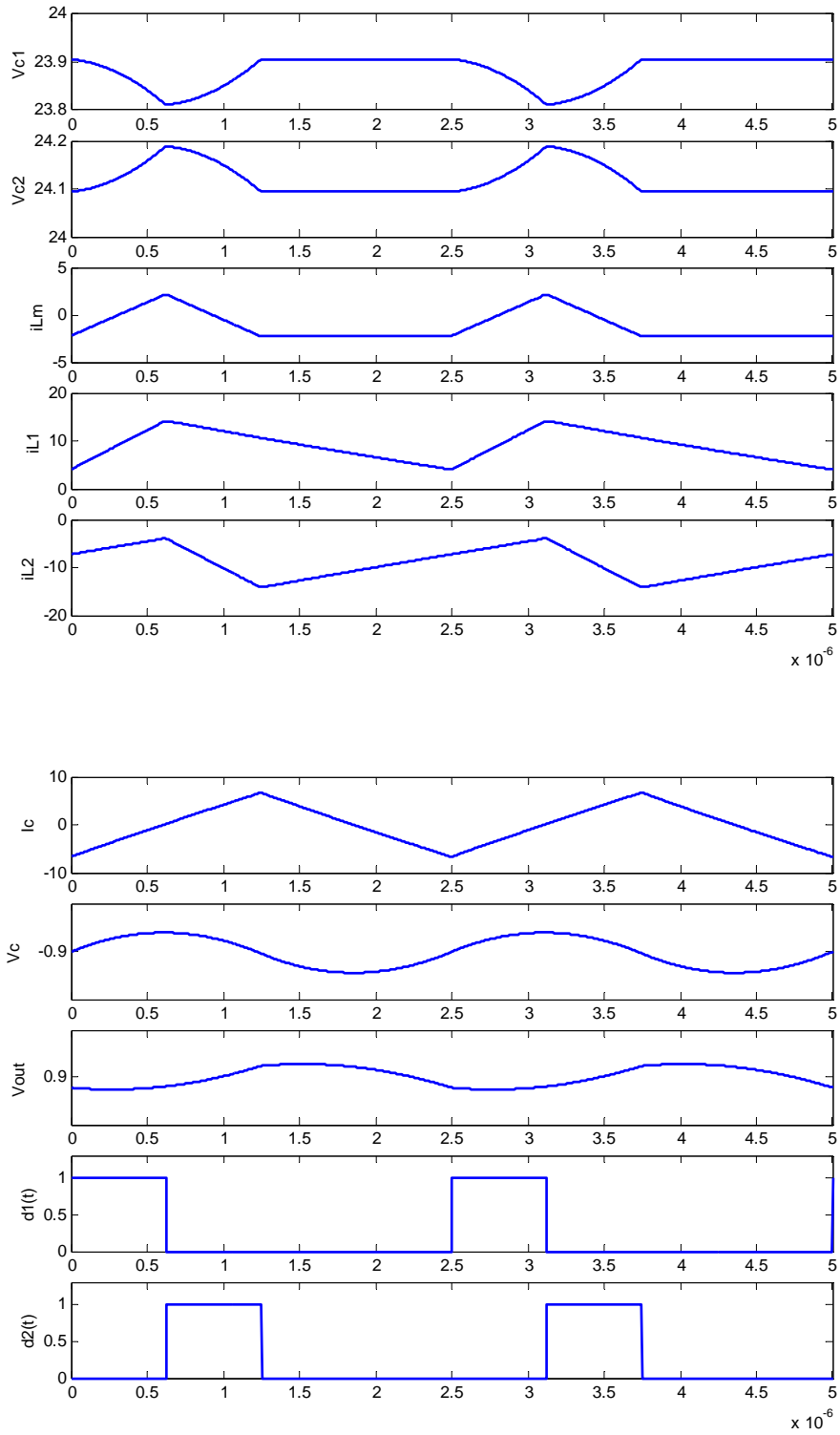


Figure 5-4 : HB DC/DC steady-state waveforms for DCS control

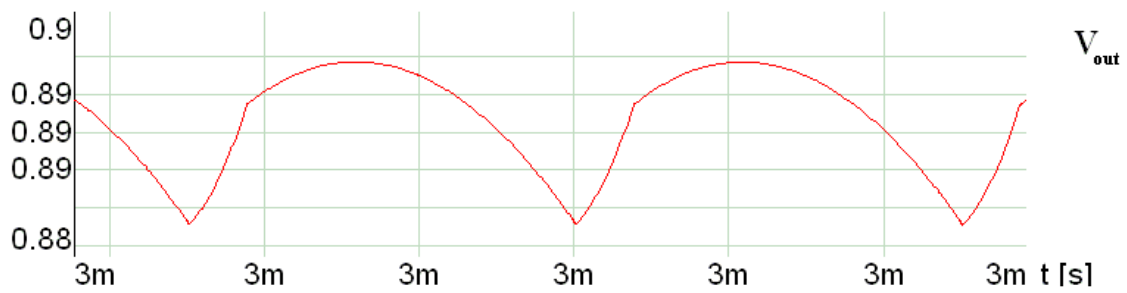
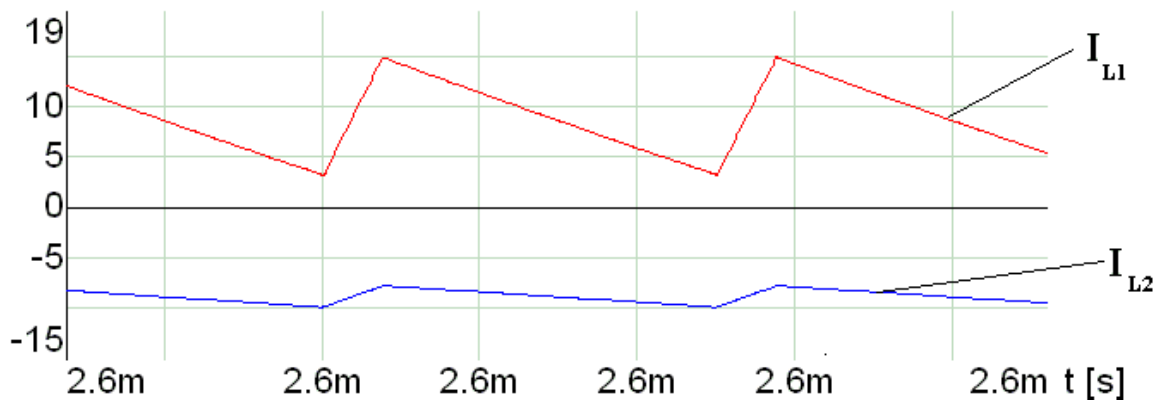
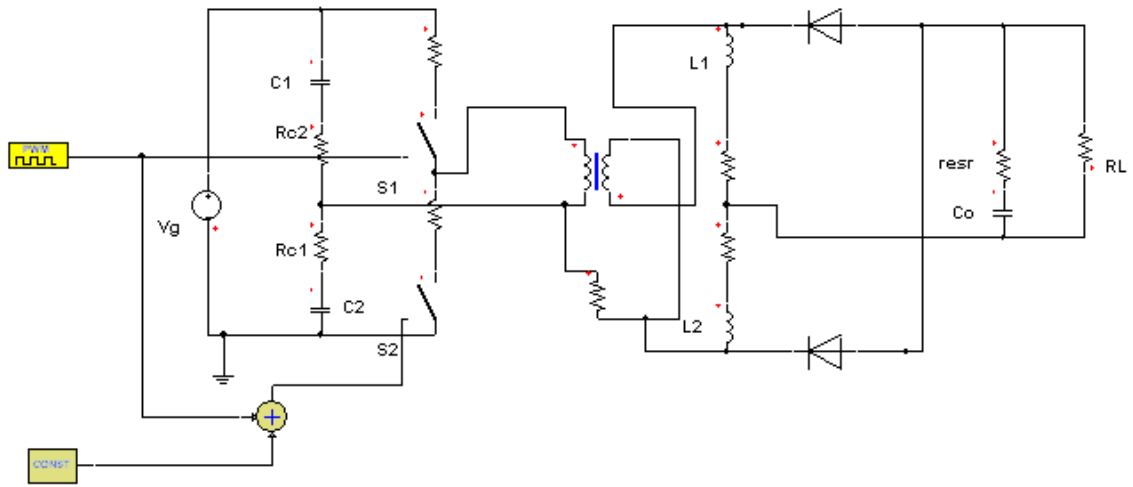


Figure 5-5 : Simplorer Simulation of the Asymmetric Control case

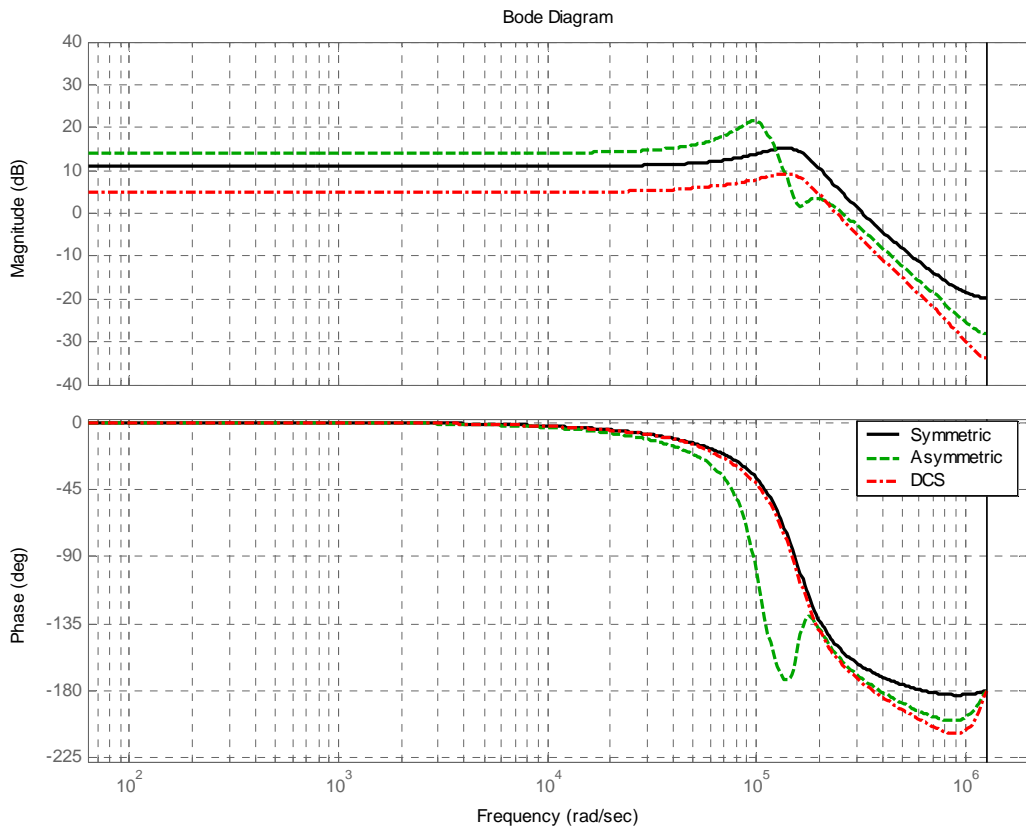


Figure 5-6 : Frequency Response for HB DC/DC example for Symmetric , Asymmetric and DCS control respectively

%Converter parameters and component values

```
a=1./64; Vg=48; Vo=1; Io=20; Fs=400e3; Ts=1./Fs; L1=180e-9; L2=180e-9; rL1=5e-3;
rL2=5e-3; Co=500e-6; resr=0.66e-3; Lm=36e-6*a; Lk=200e-9*a; N=1/6; RL=Vo/Io; resrpRL=RL.*resr./(RL+resr);
rc=10e-3; C=5e-6; rs1=25e-3; rs2=25e-3; rs3=3e-3; rs4=3e-3;
```

```
A1=[-1./(2.*C.*rc) -1./(2.*C.*rc) -N./(2.*C) -N./(2.*C) 0 0]; A1=[A1; -1./(2.*C.*rc) -1./(2.*C.*rc) N./(2.*C) N./(2.*C) 0 0];
A1=[A1; N./(2.*Lm) -N./(2.*Lm) -(N.^2).*(2.*rs1+rc)./2./Lm -(N.^2).*(2.*rs1+rc)./2./Lm 0 0];
A1=[A1; N./2./(Lk+L1) -N./2./(Lk+L1) -(N.^2).*(2.*rs1+rc)./2./(Lk+L1) - (rL1 + rs4 + resrpRL + (N.^2) .* (2.*rs1+rc)
./2) ./ (Lk + L1) (rs4+resrpRL)./(Lk+L1) resrpRL./resr./(Lk+L1)];
A1=[A1; 0 0 0 (rs4+resrpRL)./L2 -(rL2+rs4+resrpRL)./L2 -resrpRL./resr./L2 ]; A1=[A1; 0 0 0 -resrpRL./resr./Co
resrpRL./resr./Co -1./Co./(resr+RL)];
```

```
b1=[1./(2.*C.*rc) ; 1./(2.*C.*rc) ; N./(2.*Lm) ; N./2./(Lk+L1) ; 0 ; 0];
```

```
A2=[-1./(2.*C.*rc) -1./(2.*C.*rc)..-N./(2.*C) 0 -N./(2.*C) 0 0]; A2=[A2; -1./(2.*C.*rc) -1./(2.*C.*rc) N./(2.*C) 0
N./(2.*C) 0 0];
A2=[A2; N./(2.*Lm) -N./(2.*Lm) -(N.^2).*(2.*rs2+rc)./2./Lm 0 -(N.^2).*(2.*rs2+rc)./2./Lm 0 0];
A2=[A2; 0 0 0 -(rL1+rs3+resrpRL)./L1 (rs3+resrpRL)./L1 resrpRL./resr./L1 ];
A2=[A2; N./2./(Lk+L2) -N./2./(Lk+L2) -(N.^2).*(2.*rs2+rc)./2./(Lk+L2) (rs3+resrpRL)./(Lk+L2) -(rL2 + rs3 + resrpRL
+ (N.^2) .* (2.*rs2+rc) ./2) ./ (Lk+L2) -resrpRL./resr./(Lk+L2)];
A2=[A2; 0 0 0 -resrpRL./resr./Co resrpRL./resr./Co -1./Co./(resr+RL)];
```

```
b2=[1./(2.*C.*rc) ; 1./(2.*C.*rc) ; -N./(2.*Lm) ; 0 ; -N./2./(Lk+L2) ; 0];
```

```
A3=[-1./(2.*C.*rc) -1./(2.*C.*rc) 0 0 0 0]; A3=[A3; -1./(2.*C.*rc) -1./(2.*C.*rc) 0 0 0 0];
A3=[A3; 0 0 -(rs3+rs4)./(Lm+Lk) -rs3./(Lm+Lk) -rs4./(Lm+Lk) 0];
A3=[A3; 0 0 -rs3./L1 -(rL1+rs3+resrpRL)./L1 resrpRL./L1 resrpRL./resr./L1];
A3=[A3; 0 0 -rs4./L2 resrpRL./L2 -(rL2+rs4+resrpRL)./L2 -resrpRL./resr./L2];
A3=[A3; 0 0 0 -resrpRL./resr./Co resrpRL./resr./Co -1./Co./(resr+RL)];
```

```
b3=[1./(2.*C.*rc) ; 1./(2.*C.*rc) ; 0 ; 0 ; 0 ; 0];
```

%Sym

```
D1=2.*Vo./N./Vg; al=0; D2=D1;
t=D1.*Ts;
```

```
[x1,sscoef]=SteadyStatesim(D1,D2,al,Ts,A1,A2,A3,b1,b2,b3,1,Vg,20,t); t=(D1+al).*Ts;
[x2,sscoef]=SteadyStatesim(D1,D2,al,Ts,A1,A2,A3,b1,b2,b3,1,Vg,20,t); t=(D1+al+D2).*Ts;
[x3,sscoef]=SteadyStatesim(D1,D2,al,Ts,A1,A2,A3,b1,b2,b3,1,Vg,20,t); t=Ts;
[x4,sscoef]=SteadyStatesim(D1,D2,al,Ts,A1,A2,A3,b1,b2,b3,1,Vg,20,t);
x1=real(x1); x2=real(x2); x3=real(x3); x4=real(x4);
delta1=(b1-b3)*Vg-(A1-A3)*x1; delta2=(b2-b3)*Vg-(A2-A3)*x2;
delta3=(b2-b3)*Vg-(A2-A3)*x3; delta4=(b1-b3)*Vg-(A1-A3)*x4;
lambda1=expm(al.*A3.*Ts); lambda2=expm(D2.*A2.*Ts);
lambda3=expm((1-D1-D2-al).*A3.*Ts); lambda4=expm(D1.*A1.*Ts);
lambda41=lambda4*lambda3*lambda2*lambda1;
lambda42=lambda4*lambda3*lambda2;
lambda43=lambda4*lambda3; Ad=lambda41;
bd=Ts.*(lambda41*delta1 lambda42*delta2 lambda43*delta3 lambda4*delta4)*[0.5 ; 0 ; 0 ; 0.5]; dd=0;
HBf=ss(Ad,bd,cd,dd,Ts);
```

Figure 5-7 : Matlab Script to implement Discrete Model

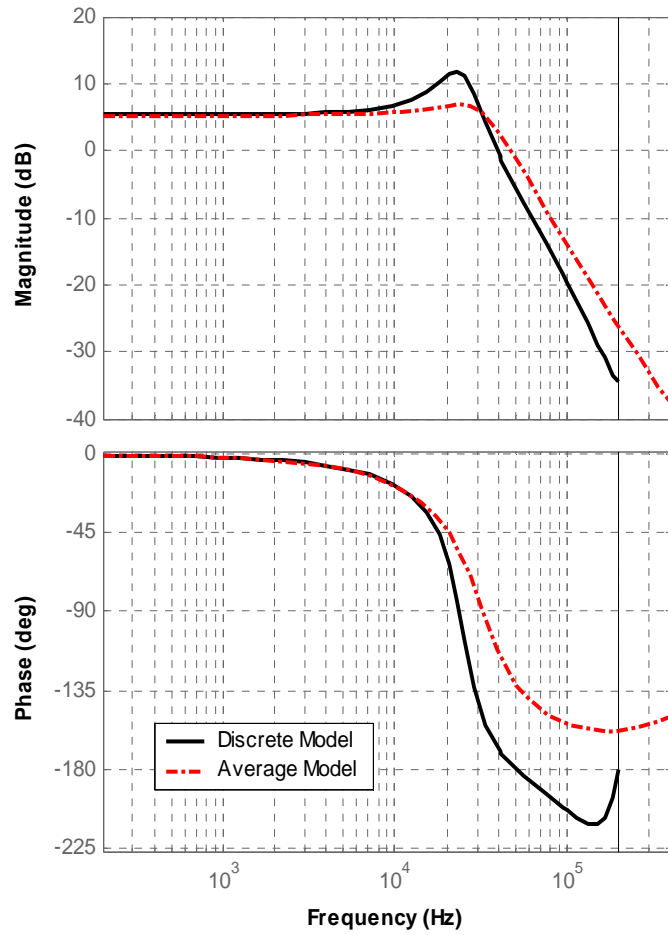


Figure 5-8 :Comparison between the Average Model and the Discrete Model for DCS control

5.4 Application of Chapter 4 Large-signal model to the Half-Bridge DC/DC topology with current doubler

The Large signal model represented by equation (4.21) is applied to the Half Bridge topology with parameters introduced in section 3. If a pulsating input is applied to it (for demonstration purposes) then the output of the model for DCS and Symmetric control is as shown in figure (5-8), while figure (5-9) is for Asymmetric control.

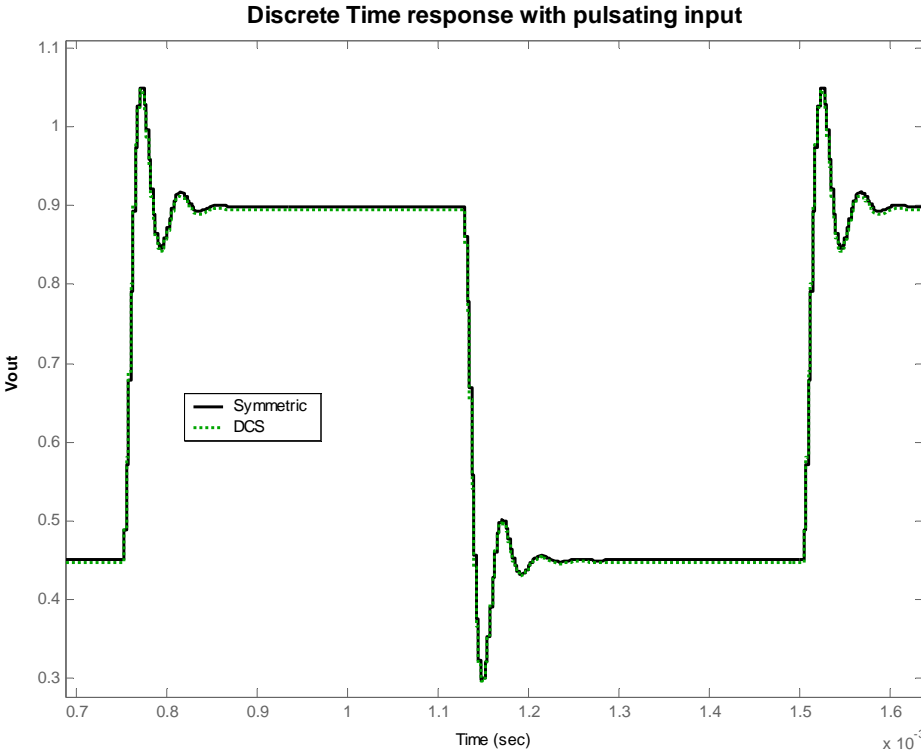


Figure 5-9 : Application of Large Signal model to HB DC/DC topology for Symmetric and DCS control respectively

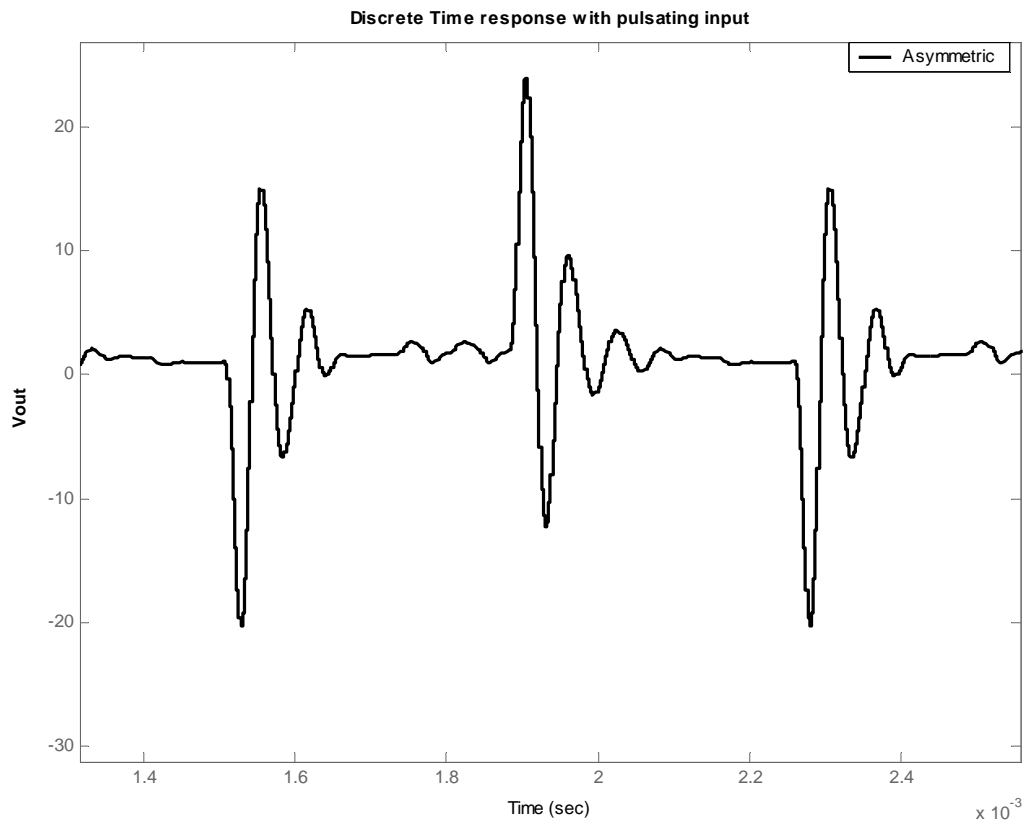


Figure 5-10 : Application of Large Signal model to HB DC/DC topology for Asymmetric control

It is apparent that Asymmetric control is the least stable control method for the Half-Bridge Topology as seen in figure (5-9) and as can be deduced from the smaller gain and phase margins in figure (5-6)

5.5 Comparison with an experimental prototype

An experimental lab prototype of a DCS controlled half-bridge Topology was used to compare with the derived models in chapter 3 and chapter 4. That pulse transfer function of the Discrete model of the converter in this case turns out to be,

$$\frac{\hat{v}_o(z)}{c(z)} = \frac{0.1542z^2 - 0.05593z - 0.08207}{z^3 - 2.633z^2 + 2.416z - 0.7744} \quad (5.10)$$

The frequency response is simply obtained by substituting $z = e^{j\omega T_s}$. And from that frequency response (the one shown in figure 5-6 for DCS) one can design an appropriate compensator to improve the dynamics of the converter and maintain stability in closed loop. This compensator is found to be,

$$\frac{c(z)}{\hat{v}_o(z)} = \frac{2.065z^3 - 0.9179z^2 - 1.97z + 1.013}{z^3 - 0.1969z^2 - 0.6427z - 0.1604} \quad (5.11)$$

The built laboratory prototype with the compensator in (6.11) had its closed loop frequency response (power stage with compensator) examined with a frequency analyzer for verification purposes. The results of the comparison between the experimental and theoretical frequency responses are plotted in Figure 6-10 and show a very high correspondence from 1 KHz till $F_s/2$.

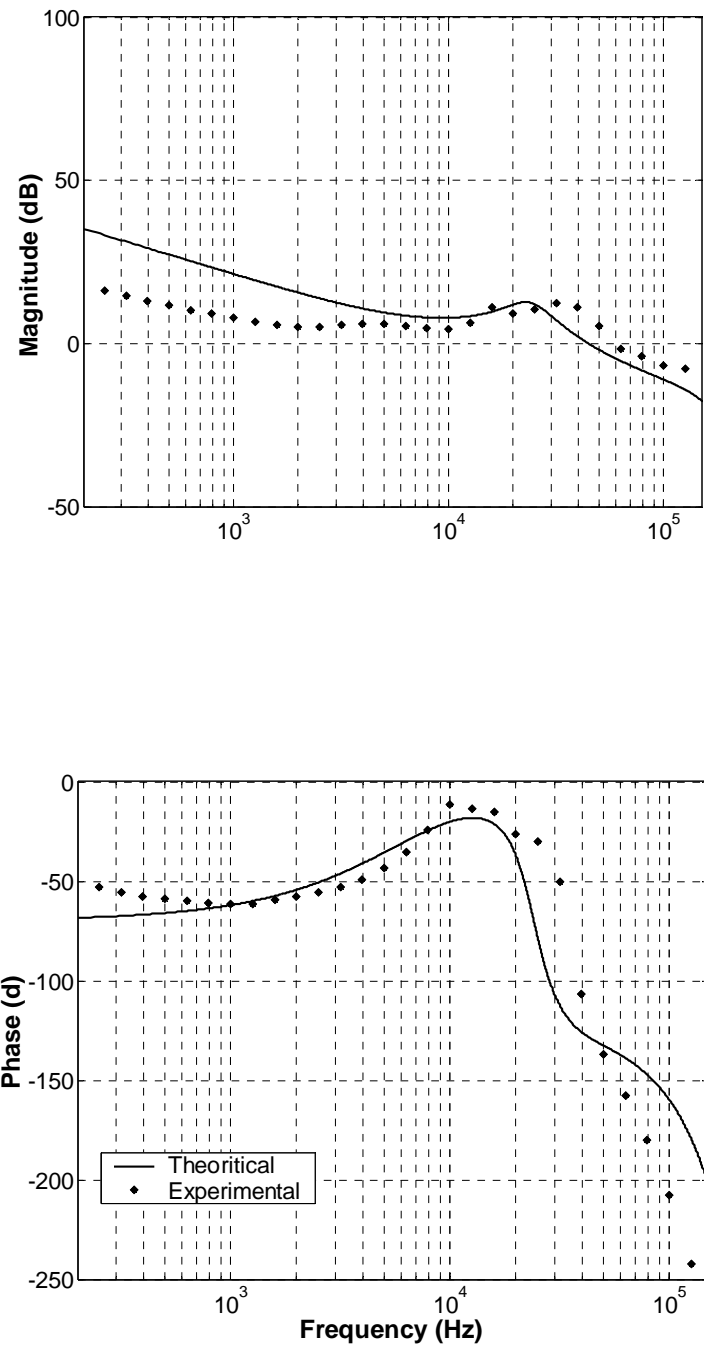


Figure 5-11 : Comparison of Closed loop frequency response between experiment and theory

It is worth noting that the Discrete model which was presented in state-space form is more adequate for direct design in the case where a digital controller is implemented, and it surpasses the continuous average models in accuracy at the higher frequency range. This last property is crucial for converters operating at high switching frequencies where modeling errors affect the robustness of the closed loop. Additionally the main reason discrete models are a more natural choice for switching converters is due to the fact that they are driven by discrete duty ratio modulations, so the sample and hold effect are implicitly included. As seen on the plots in (5-6) the measured phase at the higher frequencies has a large deviation from analog model predictions, which means if such a system was to be compensated depending on the continuous model, there would have been a high risk of instability, or at best lower transient performance than expected.

CHAPTER 6

IMPLEMENTATION OF DISCRETE MODEL IN A GUI FOR CONTROLLER DESIGN USING MATLAB

6.1 Overview of GUI platform

The purpose of this chapter is to review a GUI platform developed in MATLAB to aid controller design. Essentially this platform is a GUI wrapper for the discrete models developed in this thesis. Figures 6-1 till 6-6 show the main windows of the platform.

Starting with the main window shown in Figure 6-1, the various numbered components are,

1. The topology popup list : many topologies with different control schemes are available here.
2. View Parameters Button : opens the window shown in Figure 6-2 where the converter circuit parameter can be entered.
3. Compensator Frame : this frame contains sliders and edit boxes that are used to specify the dynamics of the compensator used in terms of its poles or zero. A compensator with up to twelve poles and zeros with an integrator can be entered. Also, the switching frequency and the compensator DC gain can be entered here.
4. Magnitude of Frequency Response.
5. Phase of Frequency Response.
6. Bode Plot Frame: contains checkboxes that when chosen causes the platform to plot the open loop, plant, compensator or closed loop frequency responses. More than one frequency response can be plotted at the same time.

7. Difference Equation Coefficients Button : this button gives the coefficients of the digital compensator equivalent to the compensator specified in the compensator frame (number 3)
8. Transfer Functions Button: this button gives the various continuous and discrete transfer functions of the compensator at hand.

Other functions can be accessed through the main window menu. For example there are menu items for saving and loading parameters and compensator data. Also the tools menu provides access to the time response window shown in Figure 6-3 , this window gives the closed loop response under step load conditions were the slew rate of the load step can be specified. Another tools menu item is the parameter sweep that opens the window of Figure 6-4, in this window any parameter of the circuit can be swept across a defined range to observe the parameters effect on the frequency response. The root locus item produces the root locus of the closed loop of the system, and finally, the Steady-State ripple item calculates the steady-state waveforms utilizing the analysis described in chapter 2.

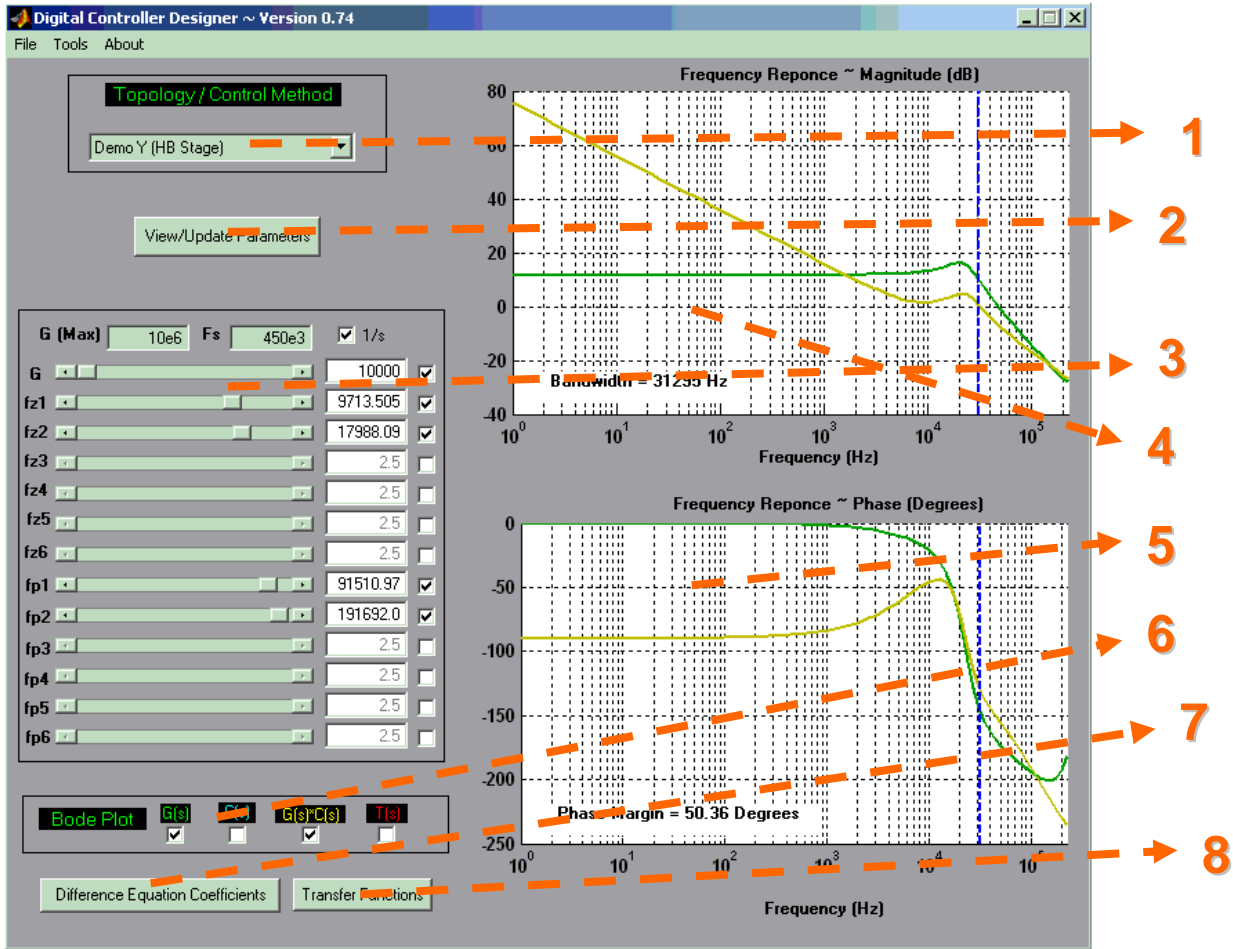


Figure 6-1 : Digital Controller Designer main window

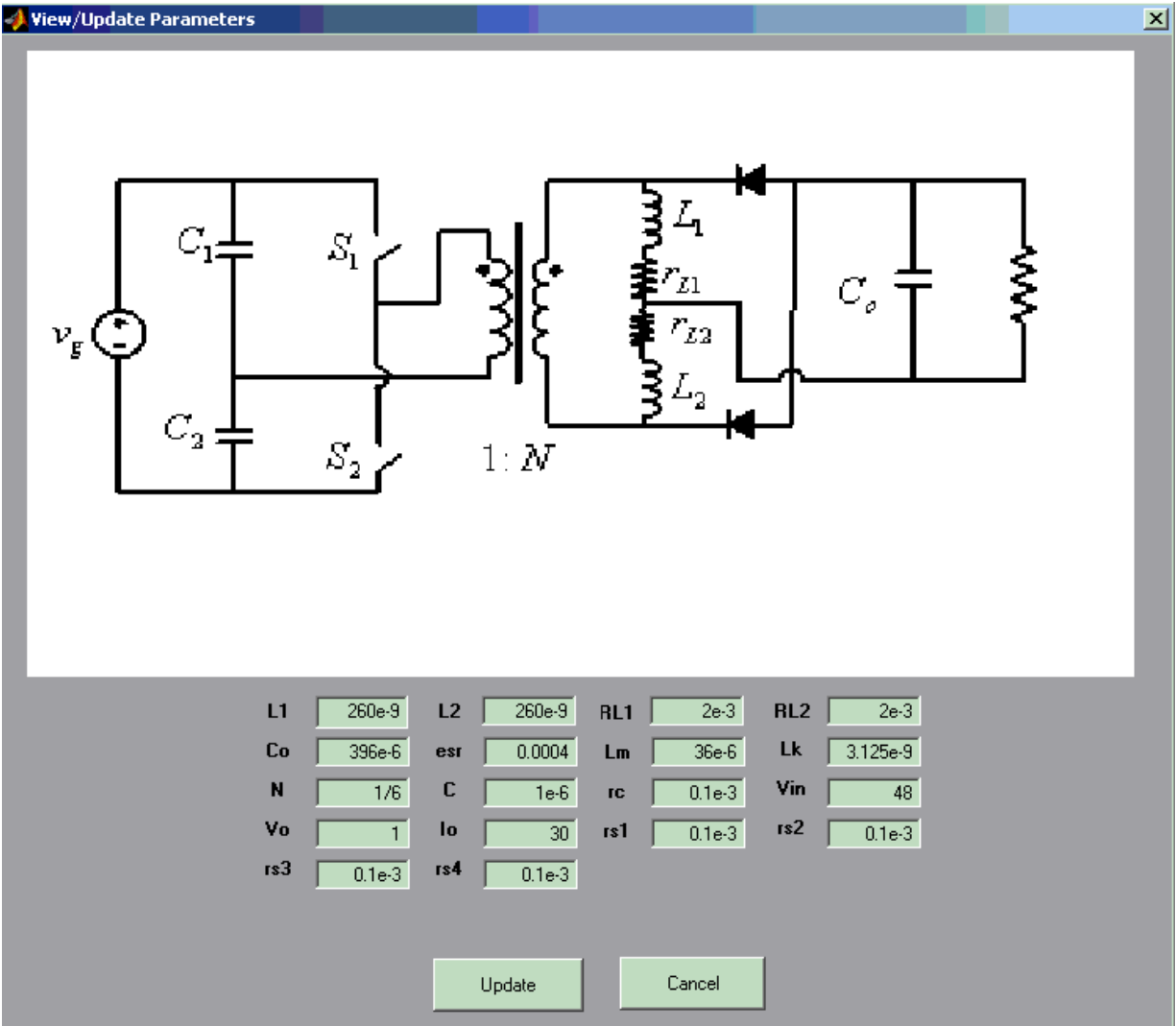


Figure 6-2 : Digital Controller Designer parameters window

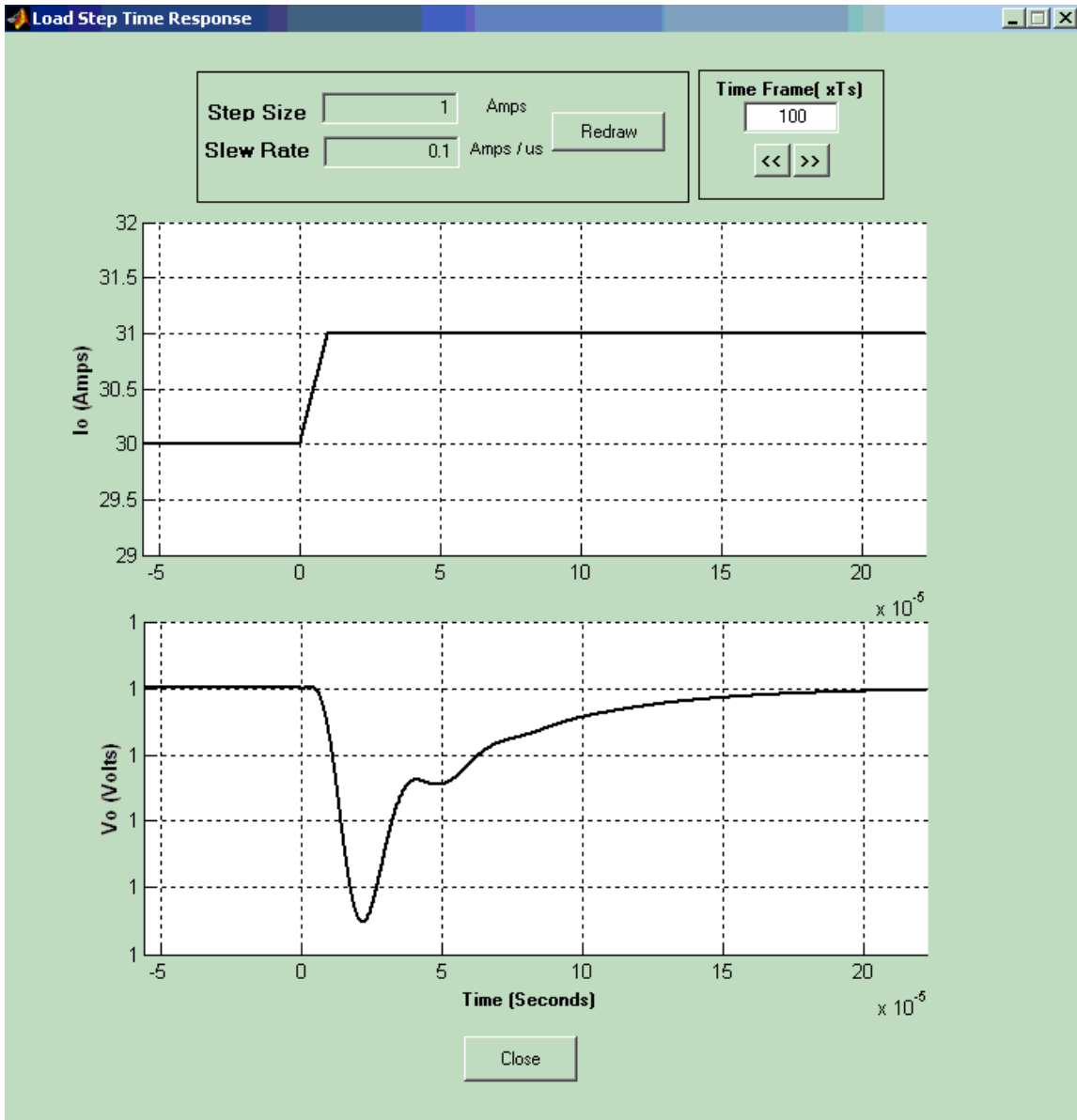


Figure 6-3 : Digital Controller Designer Time response window

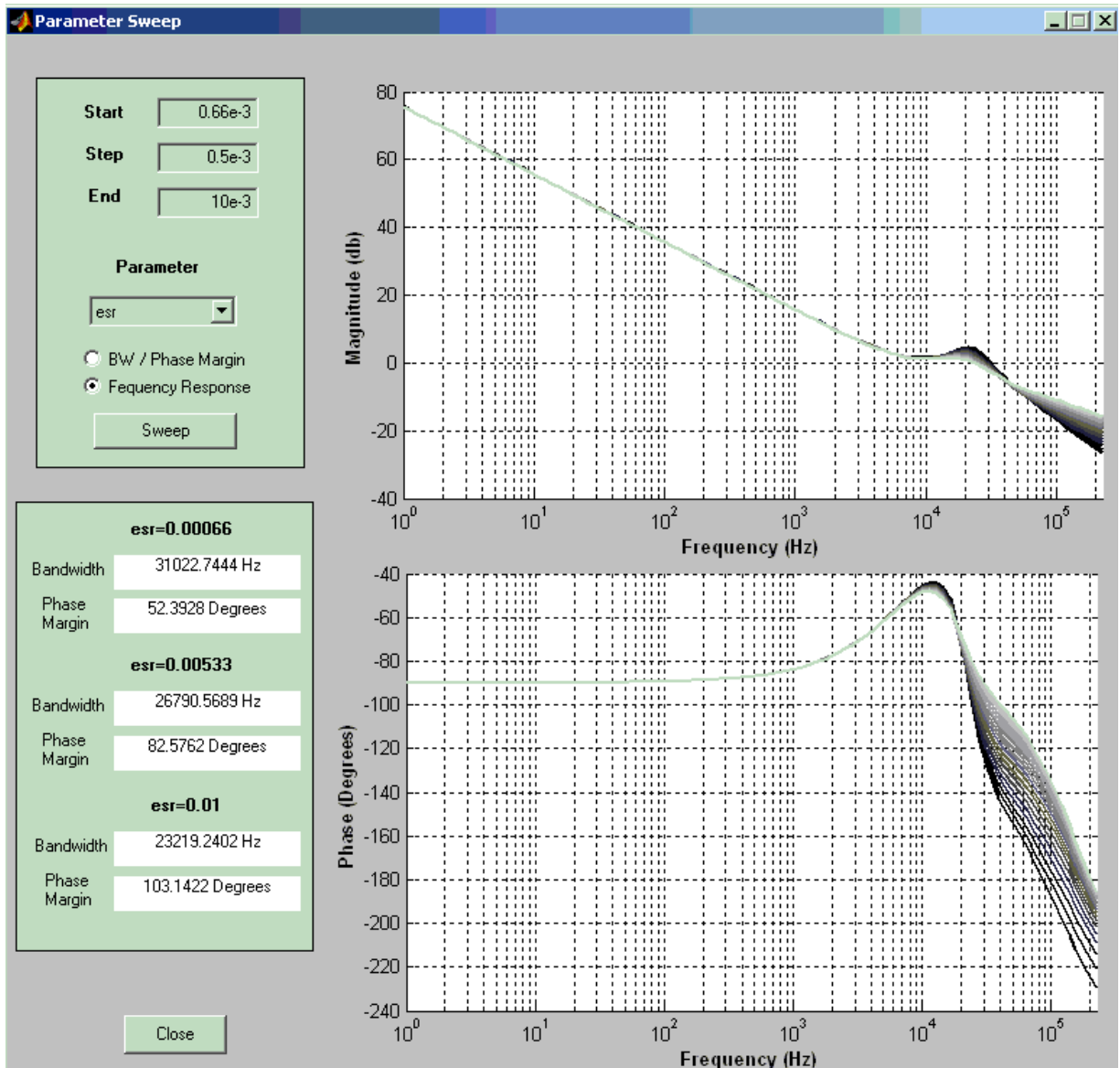


Figure 6-4 : Digital Controller Designer parameter sweep window

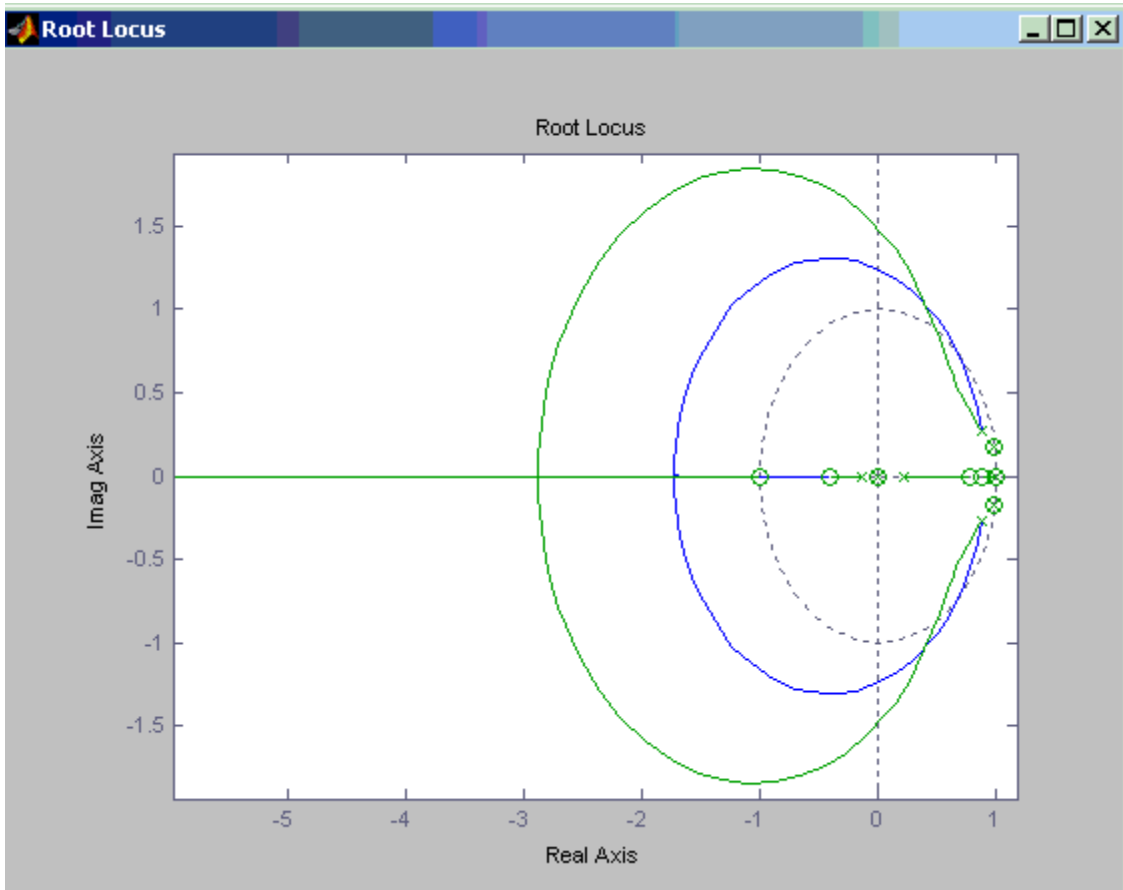


Figure 6-5 : Digital Controller Designer Root Locus window

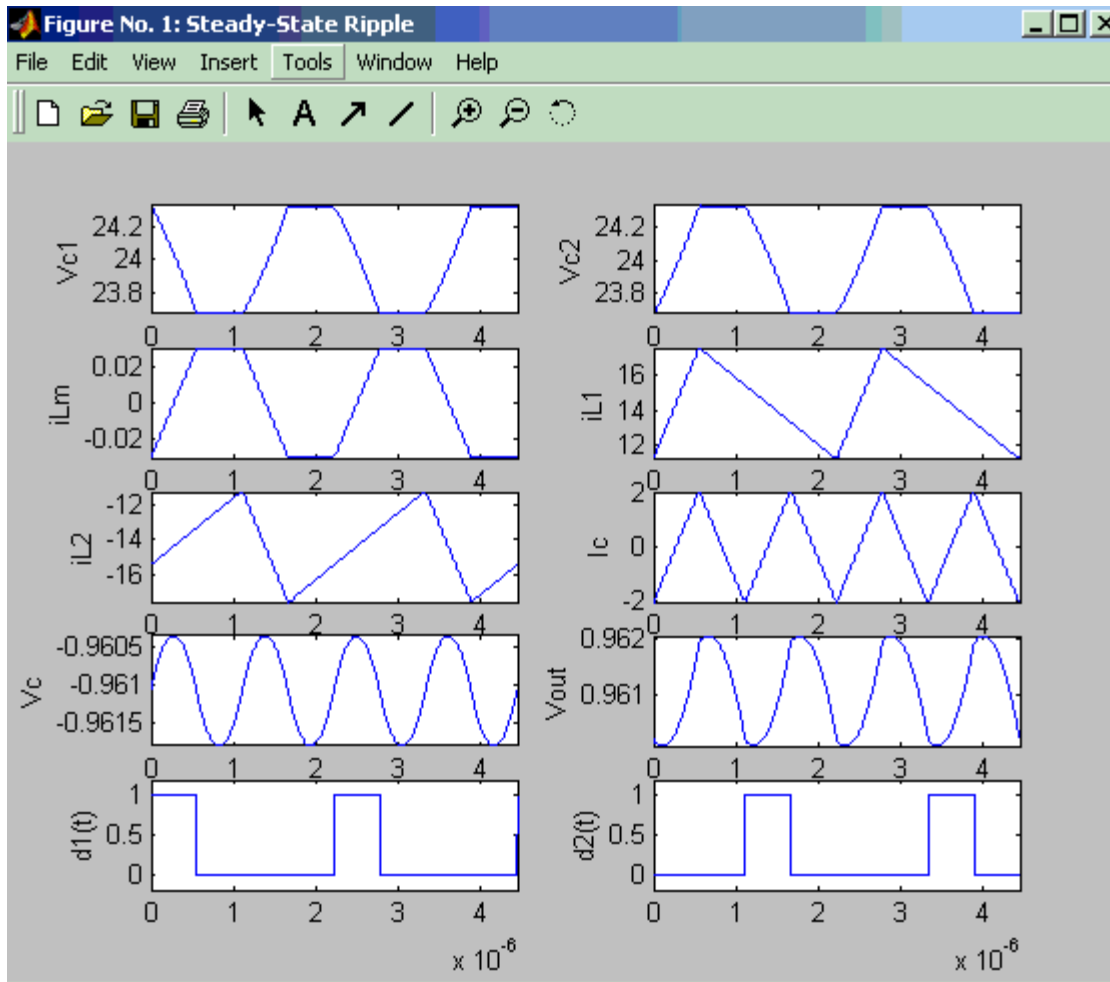


Figure 6-6 : Digital Controller Designer Steady-State waveforms window

6.2 Design Example

As an example of utilizing this GUI platform parameters from a lab prototype were entered into the parameters window those parameters are as shown below,

L1	260e-9	L2	260e-9	RL1	2e-3	RL2	2e-3
Co	396e-6	esr	0.0004	Lm	36e-6	Lk	3.125e-9
N	1/6	C	1e-6	rc	0.1e-3	Vin	48
Vo	1	Io	30	rs1	0.1e-3	rs2	0.1e-3
rs3	0.1e-3	rs4	0.1e-3				

Figure 6-7 : Parameters for Lab Prototype Half-Bridge with current doubler and Symmetric control

The next step in the controller design process is to check two poles and two zeros with DC gain and an integrator in the compensator frame of the main window. This specific compensator is the general PID that is most generally used in many cases for compensator realization. The sliders of the poles and zeros are then adjusted until the desired bandwidth and phase margins are achieved, as shown in Figure 6-8. The achieved compensator parameters can then be used to design an analog realization as shown in Figure 6-9 , or they can be used to obtain the parameters of a linear discrete compensator (a difference equation) through the use of the Difference Coefficients Button in the main window as shown in Figure 6-10.

With this design example this chapter is concluded.

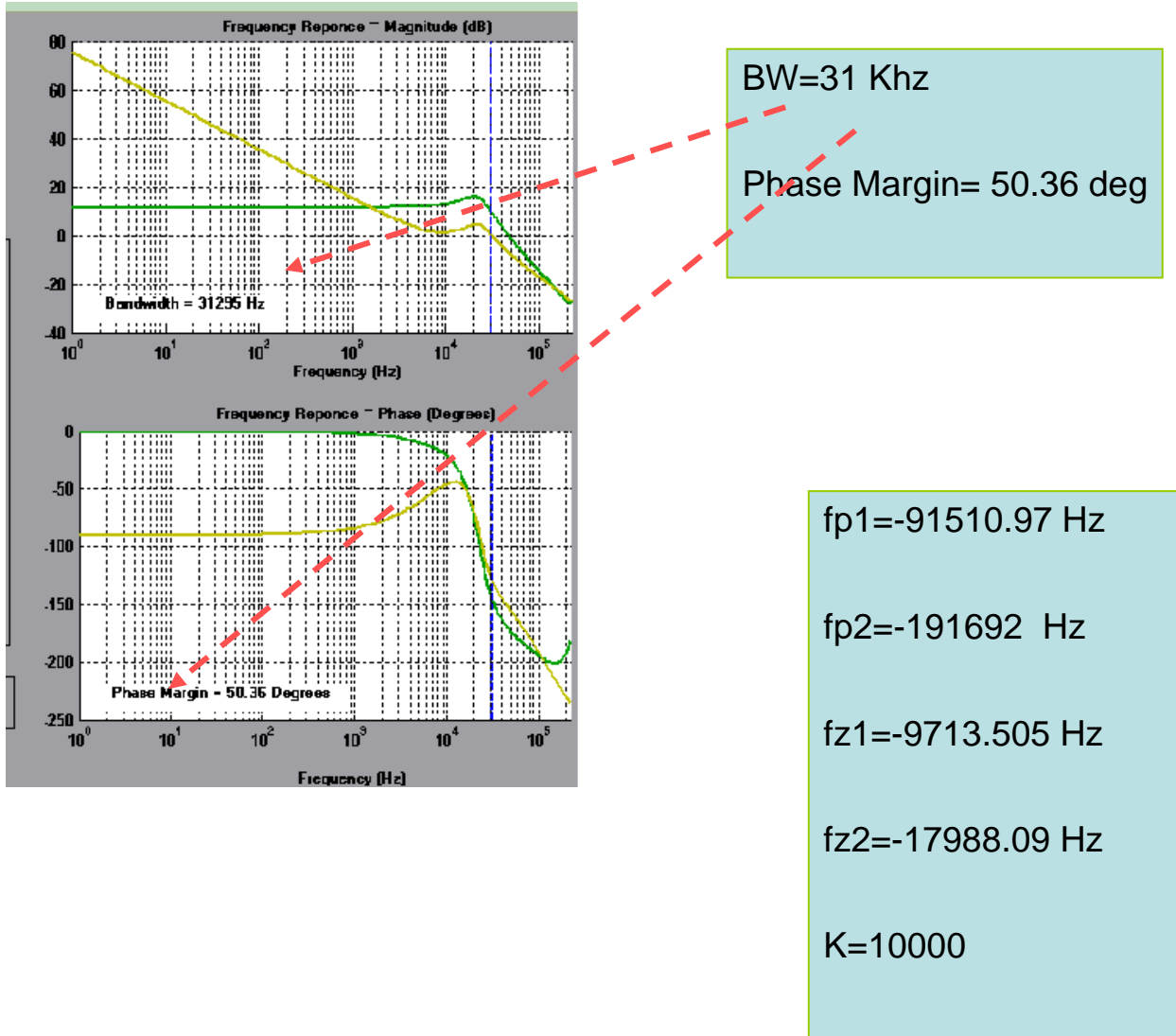


Figure 6-8 : Desired Bandwidth and phase margin with required compensator poles and zeros

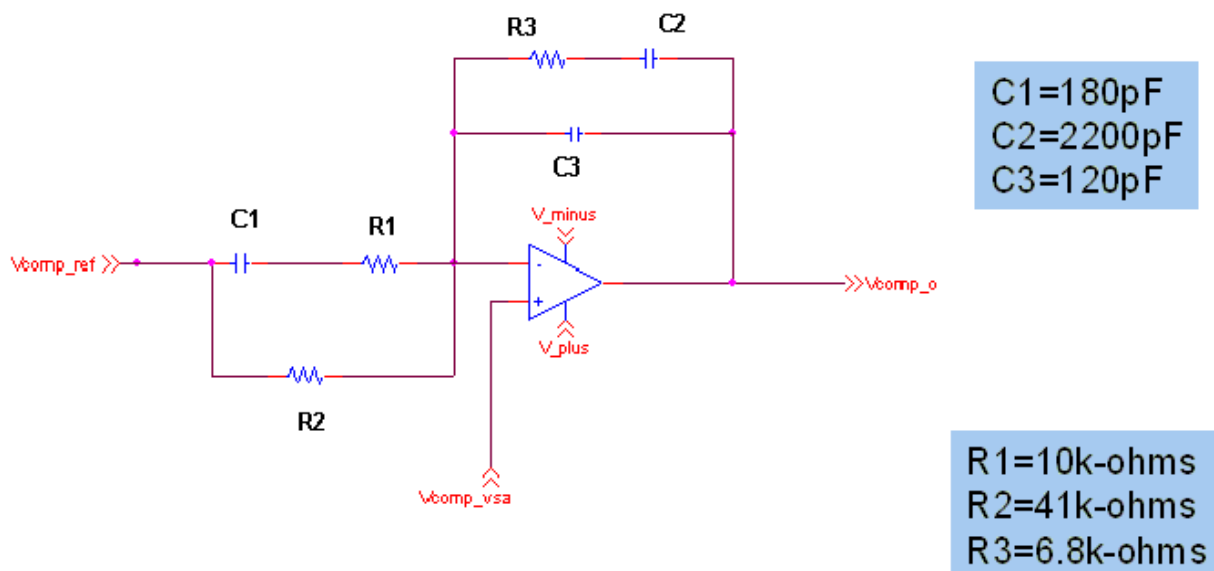


Figure 6-9 : Compensator realization in the analog domain

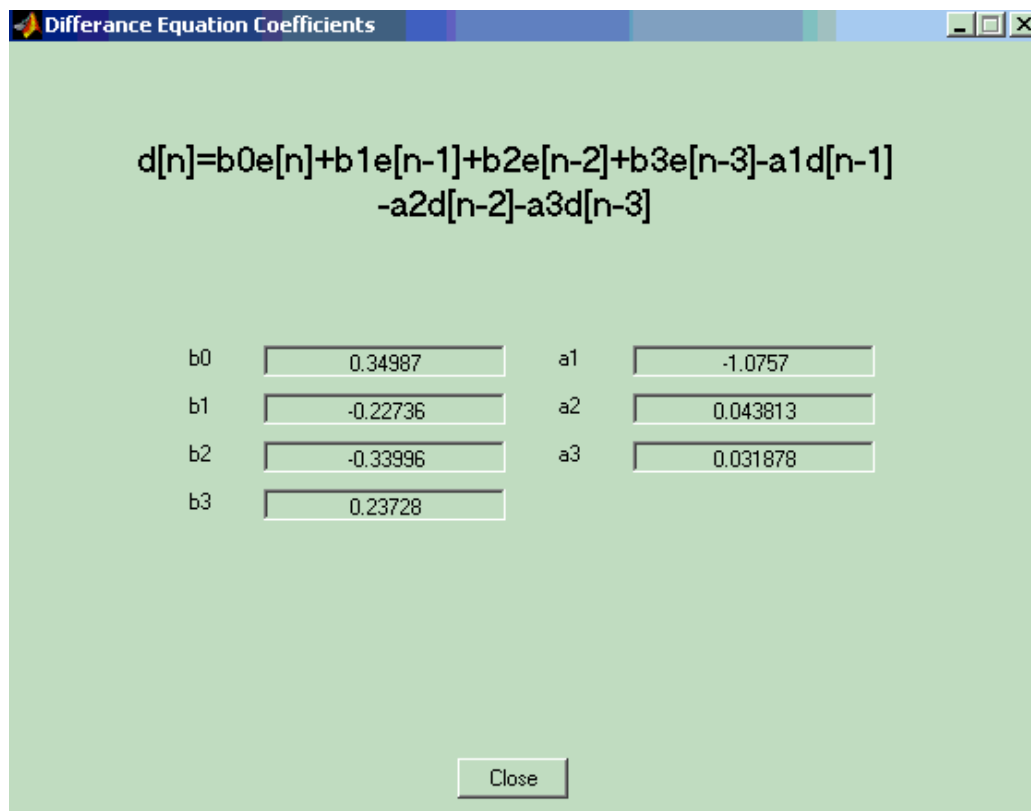


Figure 6-10 : Difference Equation coefficients for Digital Controller realization

CHAPTER 7

JAVA BASED SYMBOLIC CIRCUIT SOLVER (SYMCIRC)

7.1 Motivation

The goal of the work presented here is to develop a versatile symbolic based linear circuit with switches solver. The solver works by accepting a Netlist, and the element one wants to find the voltage across or current on, as input parameters. Then it either produces the plot or the time domain expression of the output. Frequency domain plots or Symbolic Transfer Functions are also produced. The solver gets its input from a Web-based GUI circuit drawer developed at UCF[]. Since an early design criteria was to allow mobility of access to the developed tools it was decided to write all components in the Java programming language due to its extensive support of web applications.

Typical simulation tools Electrical Engineers encounter are numerical in nature, that is when presented with an input circuit they iteratively solve it across a set of small time steps. The result is represented as a data set of output versus time, which can be plotted for further inspection. Such results don't help one understand the ultimate nature of circuits as Linear Time Invariant systems with a finite dimensional basis in the solution space. PSpice is one example of such an approach and is the most widely used.

The SymCirc Symbolic Solver provides all simulation results as Time domain expressions composed of the basis functions that exclusively include exponentials, sines, cosines and/or t raised to any power.

One big motivation for going the symbolic simulation route, other than the gained insight, is the fact that for circuits with switching components convergence problems are avoided, since elements are represented as symbols rather than numbers. Also subsequent simulation of the same circuit with different parameter values is much faster, due to the fact that the system of equations representing the circuits need not be solved more than once.

7.2 Review of Spice

SPICE is one of the ubiquitous numerical simulation packages for electronic circuits. It stands for “Simulation Program with Integrated Circuit Emphasis” and was developed by Larry Nagel and Donald Pederson at Berkeley. Its predecessor was called CANCER: “Computer Analysis of Nonlinear Circuits Excluding Radiation”, since it was the first simulator developed for civilian application that don’t need radiation hardness verification. The first incarnation of the program was SPICE1 which was coded in Fortran and used regular nodal analysis and fixed time-step transient analysis. Later the improved SPICE2 was introduced which used modified nodal analysis and variable time-step transient analysis, allowing more sample points where the waveforms rapidly change. Finally SPICE3 was written based on the Fortran code in C by Thomas Quarles, who also added X window plotting.

The Berkeley code is the base of all subsequent implementations of SPICE (like PSPICE, HSPICE, ISPICE, XSPICE, etc.). Most electronic manufacturers built in house simulation products based on that code. One popular product is the implementation provided by ORCAD with assorted tools.

The main theoretical backbone of SPICE is the utilization of the Gauss-Siedel method for solving the system of equations arising from multiple node circuits. If nonlinear elements exist a modified Newton-Raphson algorithm is used . Dynamic circuits are solved by time-stepping while considering capacitors and inductors as constant voltage and current sources respectively at each time point. For example the capacitor voltage can be stepped as follows,

$$v_{n+1} = \frac{i_n \Delta t}{C} + v_n \quad (7.1)$$

While the inductor current is stepped using,

$$i_{n+1} = \frac{v_n}{L} \Delta t + i_n \quad (7.2)$$

The resulting simulation data points can then be plotted for further investigation. SPICE is a very powerful tool for nonlinear circuits , but suffers from poor performance for circuits with rapidly switching elements, such as SMPS converters. Such poor performance manifests as slow simulation speed around switching points and convergence problems that more often than not derail the completion of the simulation. So it is worth investigation Symbolic simulation for the subtype of circuits that exhibit piecewise linearity with switching.

7.3 Transfer Function Based Symbolic Simulation

Considering any circuit as being a system with multiple inputs and outputs, and input being any dependent source and an output being the voltage or current of any circuit element, it is possible to obtain transfer functions generally partially using a methodology outlined in (8). The stepping stone to symbolic simulation are the set of transfer functions between all inputs and nodes of interest. A way to begin is to first obtain what is called the “Indefinite Admittance Matrix”, which is like a regular admittance matrix but with the ground assumed to be outside the circuit, and each node is considered a port. The Indefinite admittance matrix is constructed by populating every row i and column j with admittance terms y_{ij} as follows,

$$y_{ij} = \begin{cases} \text{Sum of admittances connected to node } i & , \text{if } i = j \\ \text{Negative sum of admittances between } i \text{ and } j & , \text{if } i \neq j \end{cases} \quad (7.3)$$

That takes care of all passive elements. Active elements represented by dependent sources are populated into the matrix through different disparaging methods. For the sake of elegance and ease of code implementation a unified approach was developed which populates all dependent sources by transforming them to Voltage dependent current sources. Those are populated as shown in figure 7-1. The transformation of other dependent sources into a Voltage dependent source is outlined in figure 7-2.

As can be seen the unified approach depends on the introduction of an infinitesimal resistance X_0 that can be later made to approach zero.

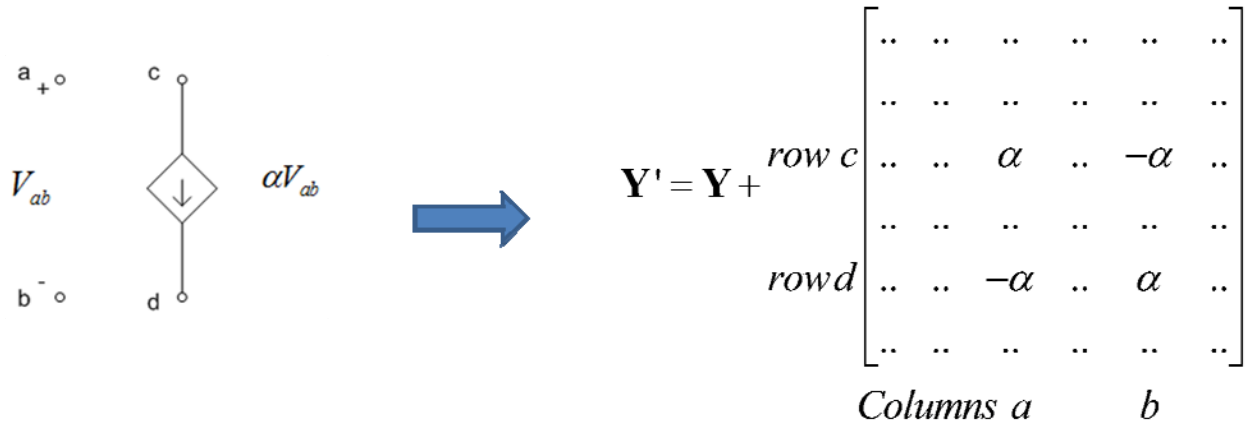


Figure 7-1 : Voltage Dependent current sources into the indefinite admittance matrix

Another novel methodology for populating the admittance matrix used in this work is to consider the independent sources to be admittances, therefore their symbols are added as prescribed by equation (7.1). How this exactly works will be explained later in this chapter.

To obtain the transfer function from the indefinite admittance matrix, two constructs must be defined, the first order and second order cofactors defined respectively as follows,

$$Y_{ij} = (-1)^{i+j} \det \mathbf{Y}_{ij} \quad (7.4)$$

$$Y_{pq,rs} = \text{sgn}(p-r) \text{sgn}(q-s) (-1)^{p+q+r+s} \det \mathbf{Y}_{pq,rs} \quad (7.5)$$

Where \mathbf{Y}_{ij} represent the admittance matrix with rows i and column j removed, and $\mathbf{Y}_{pq,rs}$ represents same matrix with rows p and r and columns q and s removed.

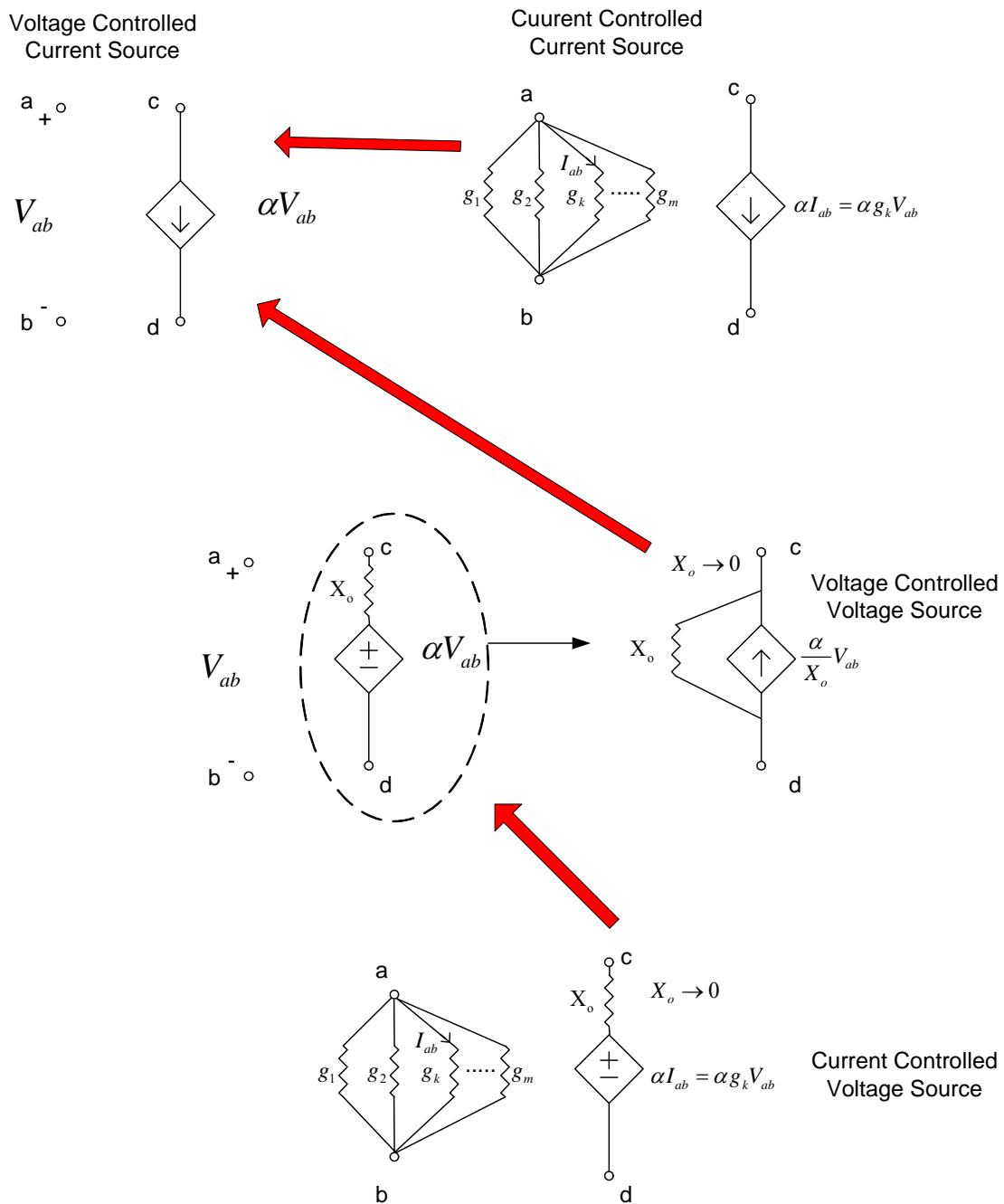


Figure 7-2 : Outline of transforming other dependent sources into voltage dependent current source

Taking into account definitions (7.4) and (7.5) the transfer function representing the voltage gain between arbitrary output at nodes p,q and input at nodes r,s is given by,

$$\frac{V_{pq}}{V_{rs}} = \frac{Y_{rp,sq}}{Y_{rr,ss}} \quad (7.6)$$

And the transfer impedance representing the voltage at nodes p,q versus an input current injected into node s and extracted from node r is given by,

$$\frac{V_{pq}}{I_{rs}} = \frac{Y_{rp,sq}}{Y_{uv}} \quad (7.7)$$

Calculating the determinants of matrices that contain infinitesimal elements X_0 is simplified by substituting an arbitrary matrix with another matrix as shown in figure 7-3. This follows from the determinant property that states that multiplying any row by an element changes the determinant by the same proportion. Infinitesimal elements are then cancelled out between the numerators and denominators of (7.6) and (7.7).

The end product of applying the techniques of this chapter to matrices with symbolic elements is symbolic transfer functions that can be used to initiate the simulation as will be shown.

$$\begin{vmatrix} m_{11} & m_{12} & m_{13} & m_{14} & \dots & \dots & m_{1n} \\ m_{21} & m_{22} + \frac{1}{X_o} & m_{23} & m_{24} & \dots & \dots & m_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ m_{k1} & m_{k2} - \frac{1}{X_o} & m_{k3} & m_{k4} & m_{k5} + \frac{1}{X_o} & \dots & m_{kn} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ m_{n1} & \dots & \dots & \dots & \dots & \dots & \dots \end{vmatrix} \Rightarrow \left(\frac{1}{X_o} \right)^R \begin{vmatrix} m_{11} & m_{12} & m_{13} & m_{14} & \dots & \dots & m_{1n} \\ 0 & 1 & 0 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & -1 & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ m_{n1} & \dots & \dots & \dots & \dots & \dots & \dots \end{vmatrix}$$

$R \rightarrow$ Number of rows X_o appears in $X_o \rightarrow 0$

Figure 7-3 : Pre-determinant evaluation of limit finding

7.4 Java Implementation

The object oriented language Java was chosen to implement the symbolic simulator, since the object paradigm allows for more modular and easily reusable code. This allows the concentration on each part of the design separately. Java also allows for easy web deployment for distributed use. The main results obtained in this chapter were coded into a versatile Symbolic simulation engine that processes circuits inputted as netlists for simulation. The engine is a standalone component that can be interfaced with any user interface front end can only needs to provide the engine with a netlist, whose source might be a graphical schematic or just a typed in description of the circuit.

The symbolic solver engine can be decomposed into three main components, as follows,

- The Netlist Parser module: a module that converts a Netlist into a corresponding symbolic indefinite admittance matrix, while storing the values for the elements and inputs and other parameters like switch states for further use.
- Symbolic module: a module that implements the capability of constructing symbolic variables, aggregating them into expressions (multivariate polynomial or rational), and manipulating them in different ways (simplification, substitution, finding limits, etc.). Those symbolic objects can also be combined into Symbolic Matrix Objects on which symbolic determinant calculation, cofactoring and population operations can be performed. This module is a critical part of SymCirc with which the tenets of the Indefinite Admittance Matrix approach for circuit analysis as described in this chapter are implemented.
- Time Domain Module: a module that picks up a rational function of s with numerical coefficients that represents the Laplace transform of the output and through a process of root extraction, residue evaluation and applying the inverse laplace transform arrives at the time domain expressions of the signal at the output.
- Switching Module: The part of SymCirc that collects information about the switches within the circuit and their states with respect to time.

Figure 7-4 shows an overview of the modules of the SymCirc Engine detailing their submodules and interactions.

7.5 Time-Domain and Frequency Domain Modules

The time and frequency domain modules of SymCirc consist of the Java implemented classes : Complex , Polynomial , RationalFunction and LTResponse. The purpose of this set of modules is to represent any transfer function in the s domain with numerical parameters , and have the capability of finding the inverse Laplace Transform of those transfer functions to get the Time domain transform , and vice versa.

The rudimentary Complex class is responsible for the representation and numerical manipulation of Complex numbers, since JAVA does not have native Complex number support (like most C like high level languages). The other rudimentary class is the Polynomial Class which represents univariate polynomials with complex coefficients , and allows for basic polynomial algebra like adding and subtracting and multiplying polynomials together to get new polynomials. Other functionality that is important for this application is the ability to find the roots of the polynomial and also deflate the polynomial of one of it's roots (Divide that polynomial by $x-r$ where r is a roots to get a polynomial of lesser degree).This functionality is important for implementing the inverse Laplace transform.

The problem of finding the roots of a polynomial is solvable to varying degrees of difficulty for all polynomials up to degree 4. For example the quadratic formula for second degree polynomials is well known and figure 7-5 shows how to get the roots of the fourth order polynomial equation. As can be seen the formula for the forth order equation is rather long and cumbersome and may cause inaccurate results when programmed into a computer due to limited machine accuracy. Actually, depending on the specific coefficients numerical methods might be

more accurate and have the prospect of getting as close to the actual roots as desired. But that is not the only obstacle, for Abel's impossibility theorem states that there is no general solution in radicals (roots and powers added, subtracted, multiplied and divided together) in terms of the equation coefficients to equations of degree five or higher. A closed form of the roots at degree five or higher is impossible to find.

The aforementioned fact makes it impossible to get a general symbolic expression describing the time domain response of an arbitrary circuit with more than four energy storage elements, therefore we must settle with substituting the numerical values of the parameters before attempting to find the time domain response. This is different from numerical analysis due to the fact that the time response is for all time t rather than a number of sampled points and is an exact expression. Also the Frequency domain transfer function is still fully symbolic.

The standard algorithm for finding a root of a polynomial equation is the Newton-Raphson method outlined in figure 8-6. It is not specific to polynomials and has quadratic convergence but is not guaranteed to converge. A better algorithm and the one implemented in SymCirc is Laguerre's Method outlined in figure 7-7. It's main advantage over Newton's method is that it is guaranteed to converge no matter what the initial guess is (although guarantee is empirical), and also it converges to a complex root even if the initial guess is real. It converges cubically therefore is faster than Newton's, but is still simple to implement compared to more sophisticated methods like Jenkins-Traub.

$$x^4 + a x^3 + b x^2 + c x + d = 0$$



$$\begin{aligned}
 &-\frac{a}{4} - \frac{1}{2} \sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{f}{3\sqrt[3]{2}} + \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} - \frac{1}{2} \sqrt{\frac{a^2}{2} - \frac{4b}{3} - \frac{f}{3\sqrt[3]{2}} - g - \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} \\
 &-\frac{a}{4} - \frac{1}{2} \sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{f}{3\sqrt[3]{2}} + \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} + \frac{1}{2} \sqrt{\frac{a^2}{2} - \frac{4b}{3} - \frac{f}{3\sqrt[3]{2}} - g - \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} \\
 &-\frac{a}{4} + \frac{1}{2} \sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{f}{3\sqrt[3]{2}} + \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} - \frac{1}{2} \sqrt{\frac{a^2}{2} - \frac{4b}{3} - \frac{f}{3\sqrt[3]{2}} + g - \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} \\
 &-\frac{a}{4} + \frac{1}{2} \sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{f}{3\sqrt[3]{2}} + \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}} + \frac{1}{2} \sqrt{\frac{a^2}{2} - \frac{4b}{3} - \frac{f}{3\sqrt[3]{2}} + g - \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}}
 \end{aligned}$$

Intermediate Parameters



$$\begin{aligned}
 e &= \sqrt{(2b^3 - 9acb - 72db + 27c^2 + 27a^2d)^2 - 4(b^2 - 3ac + 12d)^3} \\
 f &= \sqrt[3]{2b^3 - 9acb - 72db + 27c^2 + 27a^2d + e} \\
 g &= \frac{-a^3 + 4ba - 8c}{4 \sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{f}{3\sqrt[3]{2}} + \frac{\sqrt[3]{2}(b^2 - 3ac + 12d)}{3f}}}
 \end{aligned}$$

Figure 7-5 : Roots of fourth order polynomial equation

Choose initial guess x_0

Repeat while $|x_{k+1} - x_k| > \text{Tolerance}$ For $k \in 1, 2, 3, \dots$

$$x_{k+1} = x_k - \frac{P(x_k)}{P'(x_{k+1})}$$

Figure 7-6 : Newton-Raphson Algorithm

Choose initial guess x_0

Repeat while $|x_{k+1} - x_k| > \text{Tolerance}$ For $k \in 1, 2, 3, \dots$

$$\begin{array}{|l}
 G = \frac{p'(x_k)}{p(x_k)} \\
 H = G^2 - \frac{p''(x_k)}{p(x_k)} \\
 \text{if } \left| G + \sqrt{(n-1)(nH - G^2)} \right| > \left| G - \sqrt{(n-1)(nH - G^2)} \right| \\
 \left| a = \frac{n}{G + \sqrt{(n-1)(nH - G^2)}} \right. \\
 \text{else} \\
 \left. a = \frac{n}{G - \sqrt{(n-1)(nH - G^2)}} \right. \\
 x_{k+1} = x_k - a
 \end{array}$$

Figure 7-7 : Laguerre's Algorithm

After one root is found the polynomial is deflated and the remainder is applied to the root finding algorithm again until all roots are found. Of course the number of roots is equal to the degree of the polynomial.

All this functionality built into the Polynomial class comes into play in the next class in the hierarchy called "RationalPolynomial". Basically it represents a rational function with the numerator and denominator being polynomials, therefore naturally it is built using two polynomial objects . Similar to the Polynomial class , any set of rational polynomial objects can be added ,subtracted , multiplied or divided with each other to obtain new rational polynomials. Furthermore the rational polynomials are always reduced to simplest form with common factors

between numerator and denominator removed. The paramount functionality included in this class is the ability to calculate the residues of any rational function. It is known that any transfer function $T(s)$ can be written as,

$$T(s) = \frac{A_{s1}}{s - s_1} + \frac{A_{s2}}{s - s_2} + \frac{A_{s3}}{s - s_3} + \dots + \frac{A_{sN}}{s - s_N} \quad (7.8)$$

Where the complex constants A_{s1} to A_{sN} are called the residues of the rational functions, with one residue associated with each root of the denominator. The residues are calculated as follows,

$$Res_{s=s_k} T(s) = \frac{1}{(m-1)!} \lim_{s \rightarrow s_k} \left\{ \frac{d^{m-1}}{ds^{m-1}} \left[(s - s_k)^m T(s) \right] \right\} \quad (7.9)$$

Where m is the multiplicity of the specific root s_k , since the existence of repeated roots is possible. Then, those after the rational polynomial class calculates those residues and has knowledge of all the roots and their multiplicity the Inverse Laplace of the transfer function represented by the rational polynomial can be easily found by inverting each term in equation (8.8) and adding them together. Each term in 8.8 corresponds to,

$$\frac{A_{s_k}}{s - s_k} \Leftrightarrow A_{s_k} t^{m-1} e^{s_k t} \quad (7.10)$$

Where again m is the multiplicity of the particular root s_k .

The result of invoking the Inverse Laplace function on a rational polynomial object gives an LTResponse object, which is basically an object that represents the Transient response expression and allows it to be displayed in Latex or text format. It also allows for time

expression manipulation and evaluation at certain time points. Finally a Laplace Transform functionality is included to find the Laplace transform of any arbitrary time expression consisting on an aggregate of sines, cosines and exponential functions. This allows the performance of simulations with inputs represented by those functions.

7.6 Symbolic Manipulation Modules

The previous section dealt with the modules relating to manipulating transfer functions with numerical coefficients and time expressions to obtain the response of a particular circuit. But to obtain the transfer functions it is necessary to implement the methodology of section 3 of this chapter. The methodology of section three is normally evaluated at specific numerical values for the parameters, yet what is required here is to apply it to General symbolic objects.

The classes responsible for modeling those symbolic objects are “sop”, “rsop” and “SopMatrix” classes. There is no need to program a completely general algebraic symbol manipulator for SymCirc since the application of the operators outlined in section three can always result in a rational form with both numerator and denominator being sums of products of algebraic symbols. The “sop” class is the base construct and represent as its namesake implies a sum of products. All the basic arithmetic binary operators can be applied to two sop’s and furthermore they can be expanded to any power. They can also be displayed in latex format sorted by symbol power. The “rsop” class is built upon the “sop” class and just joins two sop’s to form a rational sum of products. Finally the “SopMatrix” class represents a matrix of rsop

objects, and includes imports functionality for implementing section three such as symbolic determinant finding and cofactor finding.

The symbolic modules are the focal point in the hierarchal design of SymCirc , they communicate with both the Netlist parser that has information about the circuit from the user interface, and with the lower level time domain modules. The end result of a particular simulation is a symbolic transfer function represented by an rsop object, which is then converted into a rational polynomial object by substituting the given parameter values.

7.7 Graphical and User Interfaces and Top level modules

All the SymCirc engine needs to properly work is a Netlist representation of the circuit that must be modeled, the format of that Netlist has only to be compatible with the SPICE standard. For convenience a graphical circuit drawer can be added to SymCirc to speed up and ease circuit construction. A graphical circuit drawer developed in UCF was used for this purpose and a screenshot shown in figure 8-5. This GUI communicates with the top level of SymCirc by passing the Netlist or simulation results and receiving simulation expressions. The top level modules include the parser , switchset module and source module.

The parser parses through the provided Netlist for the purpose of populating the indefinite admittance matrix. When it encounters sources and switches it stores them in the “source” and “switchset” objects respectively. The “source” object’s main purpose is to store initial and final conditions of the capacitors and inductors of the circuit at switching boundaries,

due to the fact that said boundary values are modeled as added sources to the circuit. The “switchset” stores the properties of all the switches in the circuit , their initial states and their switching time, communicating that information the mid-level modules of SymSirc.

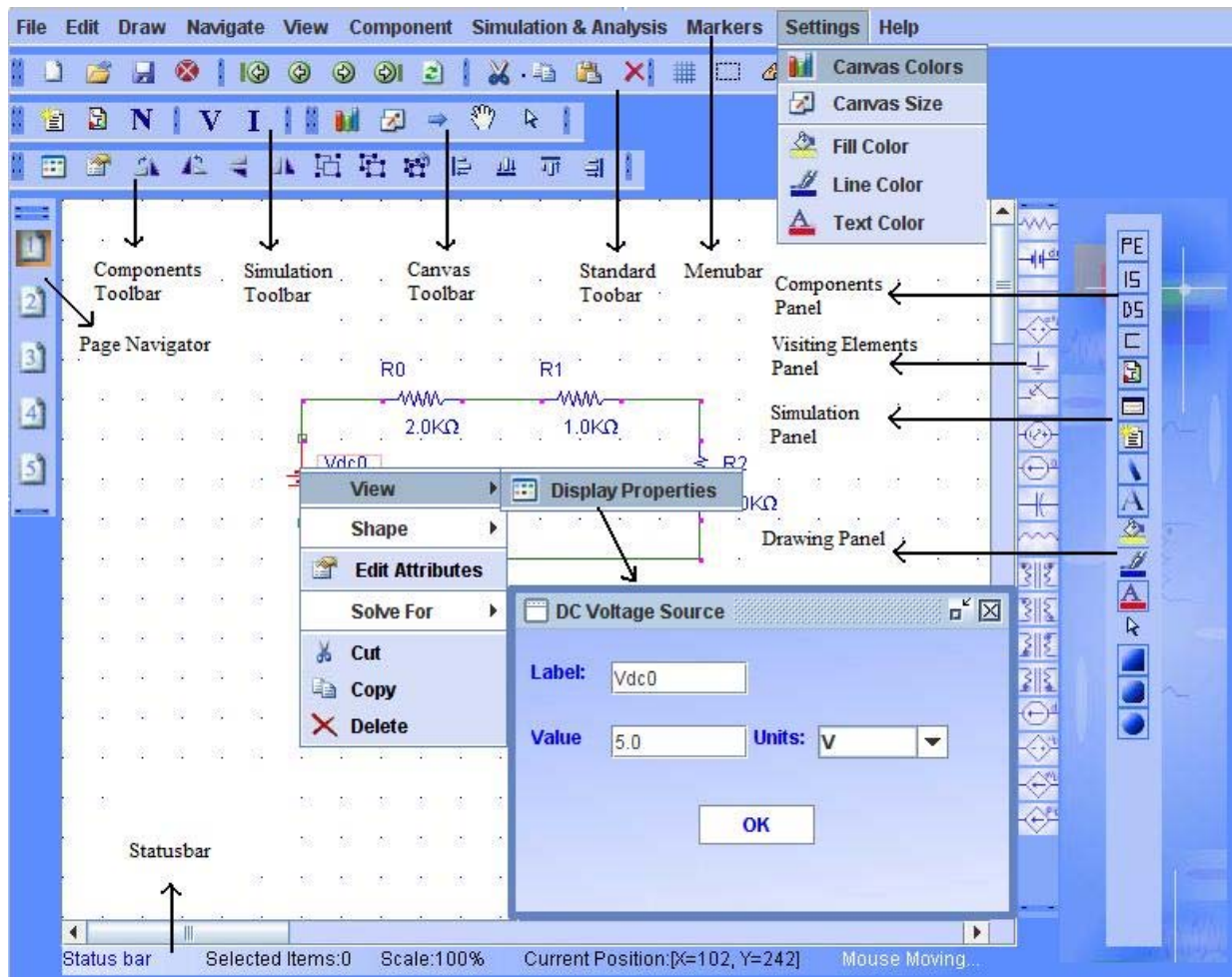
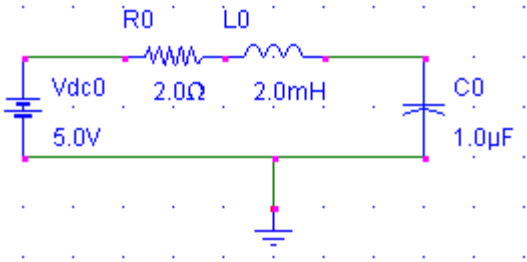


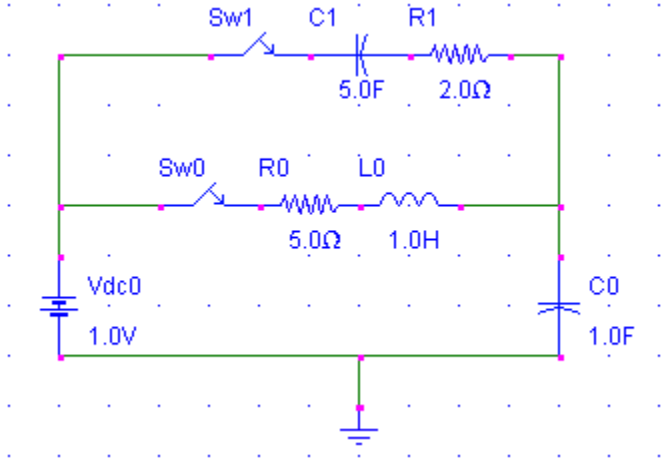
Figure 7-8 : Graphical User Interface Front end for SymCirc

7.8 Demonstration

For demonstration purposes the circuits in figure 8-9 were entered into SymCirc, and various simulation results were obtained and are shown in figures 7-10 to 7-14



(a)



(b)

Figure 7-9 : Demonstration Circuits

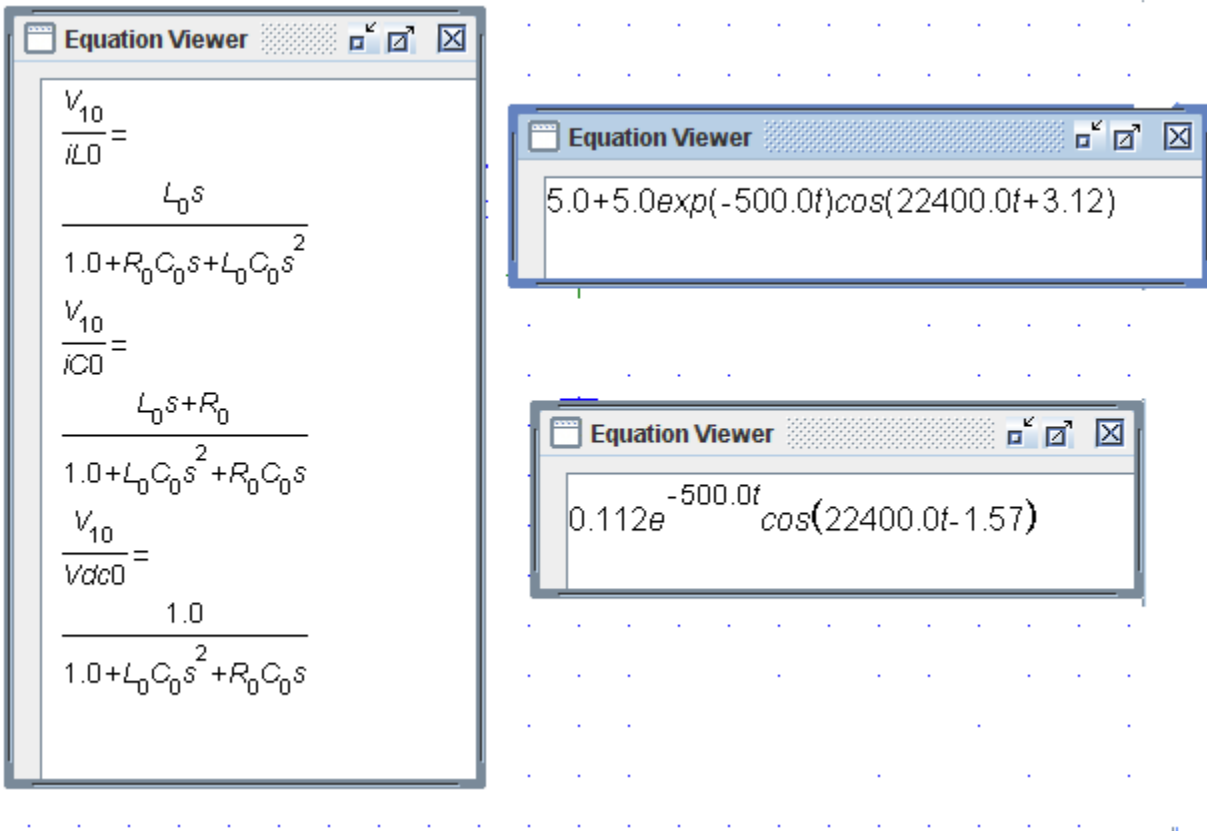


Figure 7-10 : Transfer functions and time domain expressions for circuit of figure 7-9 (a)

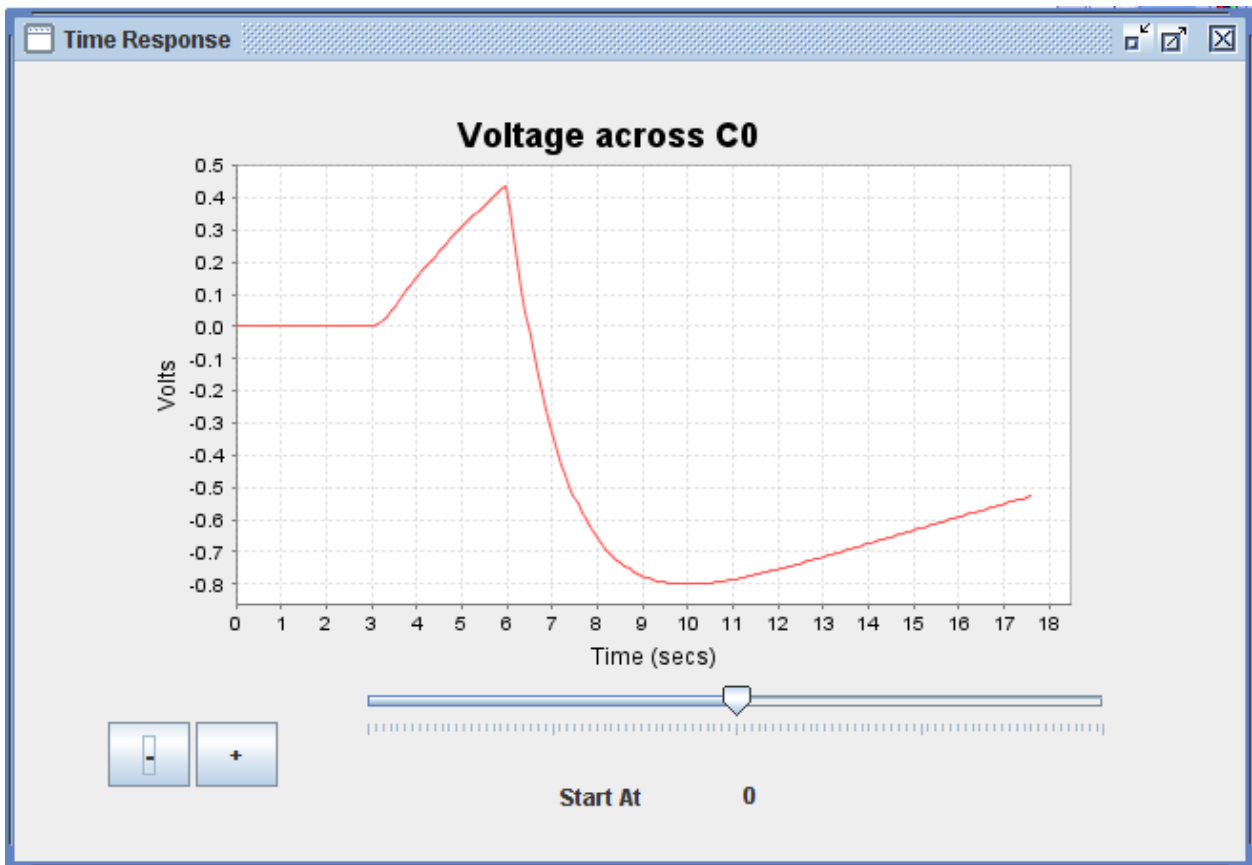


Figure 7-11: voltage waveform across C0 for circuit of figure 7-9 (b)

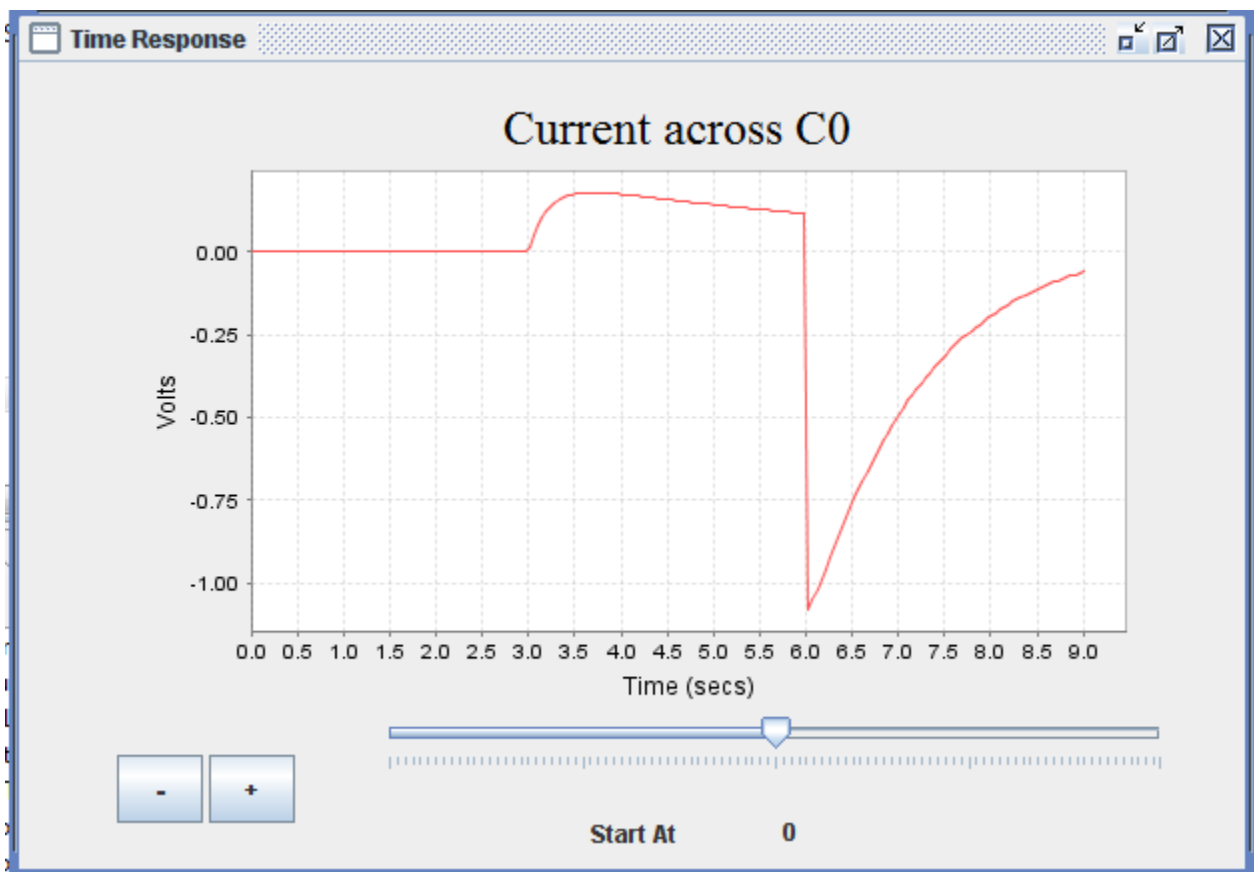


Figure 7-12 : Current waveform across C0 for circuit of figure 7-9 (b)

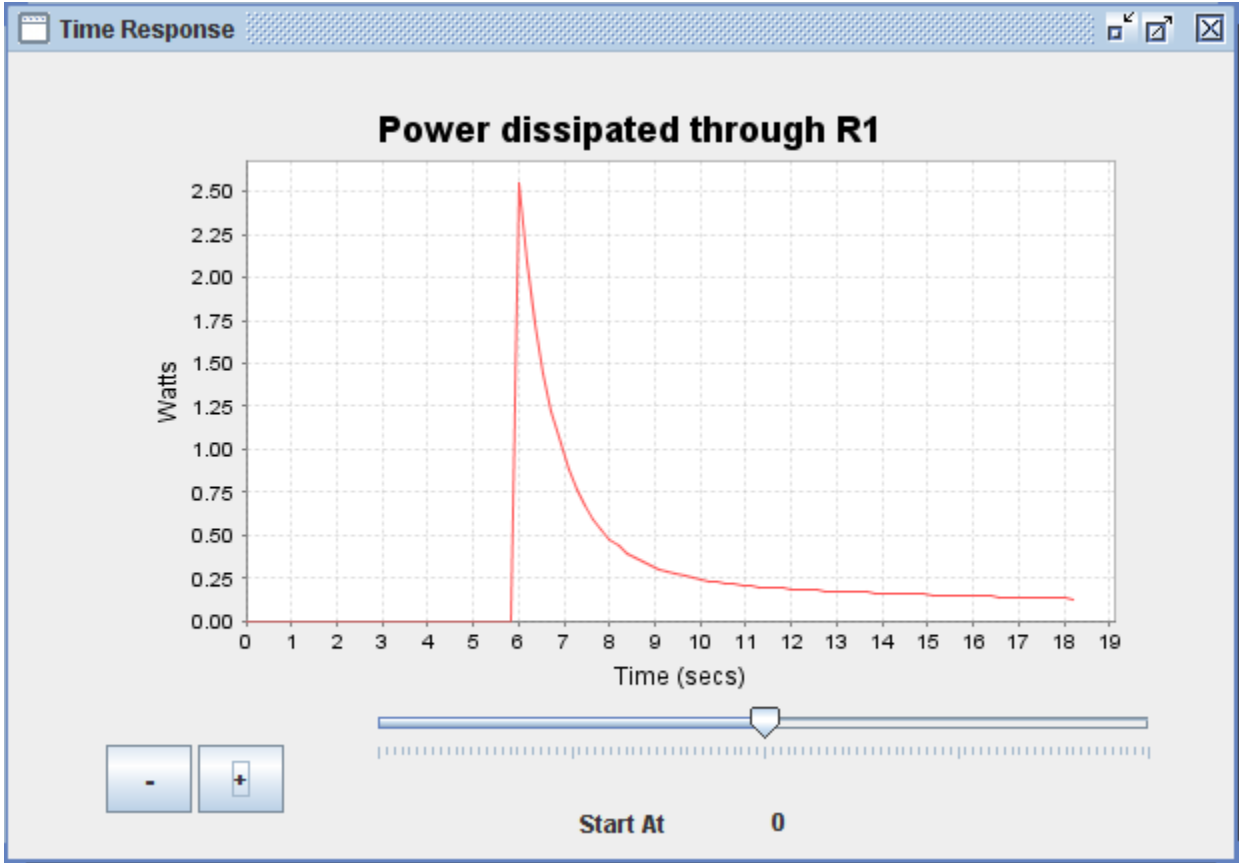


Figure 7-13 : Waveform of power dissipated through R1 for circuit of figure 7-9 (b)

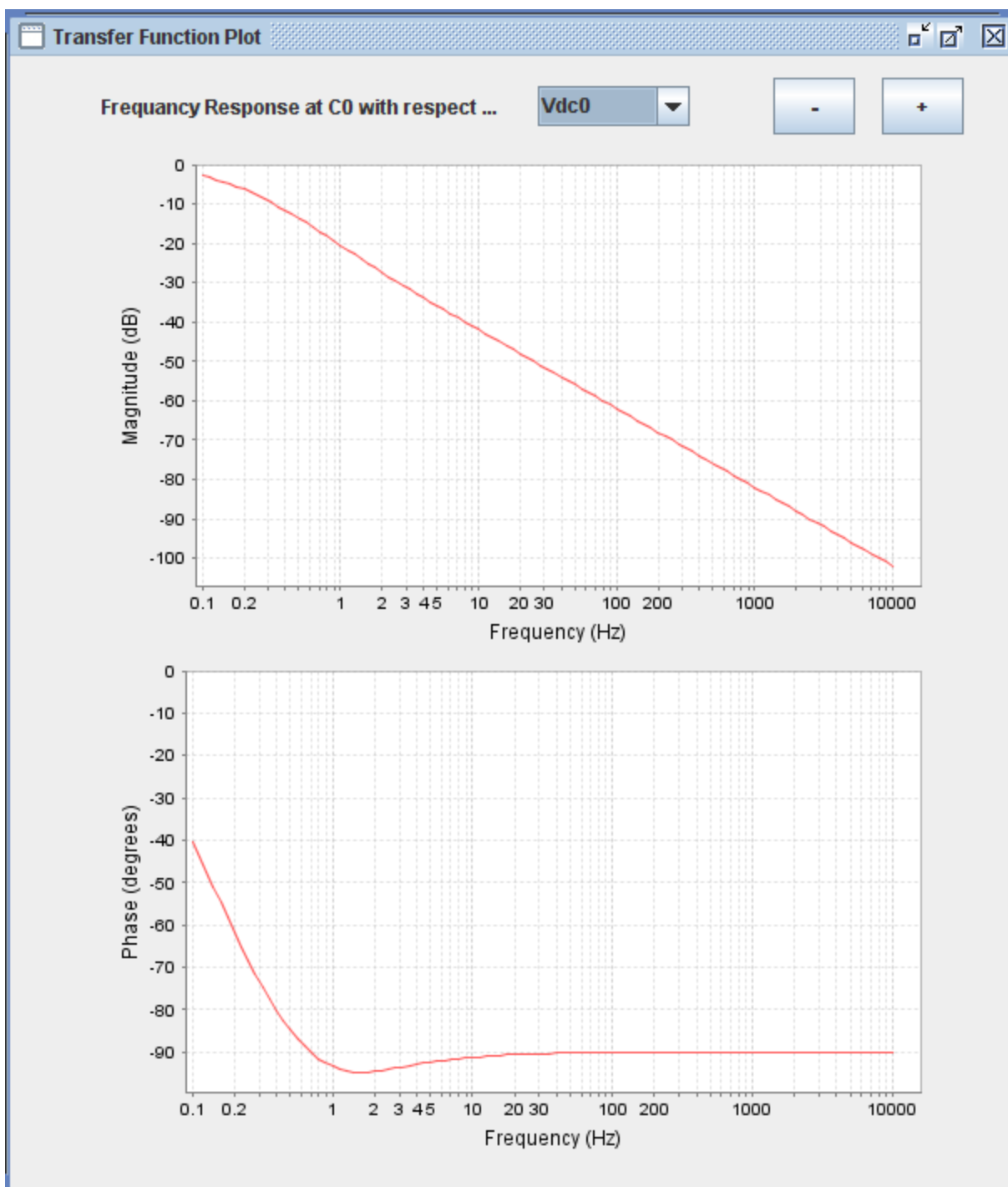


Figure 7-14 : Frequency response with respect to input voltage for circuit 7-9 (a)

CHAPTER 8

STATE-SPACE BASE SYMBOLIC SIMULATION FOR DC/DCCONVETERS

8.1 Introduction and Motivation

As stated in chapter 8 iterative numerical based simulation of DC/DC converters tends to be slow and prone to convergence difficulties, so a Symbolic approach is worth investigation. The idea is to use one of numerous CAS (Computer Algebra Systems) platforms to acquire symbolic formulas that describe the response of the converters as a function of all relevant parameters. While such symbolic closed forms are long winded and complex and very cumbersome to attempt to derive by hand, said modern CAS platforms makes deriving them quite attainable without reverting to first order simplifications. The benefits of having those formulas lies in immensely increasing the speed of simulation, and avoiding numerical issues related to ill-conditioned systems. Also beneficial is the ease with which the effect of changing specific parameters (say, the inductance, or the capacitor's esr) on the system response can be studied and plotted as will be shown within this chapter. This chapter will also investigate obtaining the dynamic characteristics through the resulting symbolic simulation, which can be then used to close the loop. The Mathematica platform was the CAS chosen for demonstration and for all implementation code.

8.2 State-Space Base equations for Generic Second Order converter with Two Operating Modes

It is natural to use the state space form as a starting point since it presents a concise form of the differential equations governing the behavior of the converter, a form which many modern CAS platforms can directly manipulate. In a manner similar to what was done in chapter 4 for obtaining the discrete large signal model the discrete model of a generic Two mode converter can be shown to be,

$$\mathbf{x}[(n+1)T] = e^{\mathbf{A}_2(1-D)T} e^{\mathbf{A}_1 d_n T} \mathbf{x}[nT] + \left(\begin{array}{c} e^{\mathbf{A}_2(1-D)T} \left[\int_{nT}^{(n+d_n)T} e^{\mathbf{A}_1[(n+d_n)T-\lambda]} d\lambda \right] \mathbf{B}_1 \\ + \left[\int_{(n+d_n)T}^{(n+1)T} e^{\mathbf{A}_2[(n+1)T-\lambda]} d\lambda \right] \mathbf{B}_2 \end{array} \right) \mathbf{U} \quad (8.1)$$

Where $\mathbf{x}[nT]$ and $\mathbf{x}[(n+1)T]$ are vectors representing the states of the converter (inductor current and capacitor voltage) at time s nT and $(n+1)T$ where T is the switching period ($1/F_s$). \mathbf{A}_k and \mathbf{B}_k represent the A and B matrices of the system at mode k . Finally \mathbf{U} is the vector representing the input, note that it is possible to have multiple inputs (e.g. and input voltage and a step current at the output modeled as an input).

While one may be tempted to plug equation (8.1) directly into the Mathematica, that will not properly work since the resulting symbolic result will be huge and consume a lot of memory, so it will be practically impossible to deal with. Also as will be explored later the symbolic form of the result is dependent on the specific numerical values of the converter parameters.

Based on (9.1) it can be noted that at steady state $\mathbf{x}[nT] = \mathbf{x}[(n+1)T]$ since the response should be stable and periodic, so this gives the steady state inductor current and capacitor voltage to be,

$$\mathbf{x}_{ss,\min} = \left(\mathbf{I} - e^{\mathbf{A}_2(1-d_{ss})T} e^{\mathbf{A}_1 d_{ss} T} \right)^{-1} \left(\begin{array}{c} e^{\mathbf{A}_2(1-d_{ss})T} \left[\int_{nT}^{(n+d_{ss})T} e^{\mathbf{A}_1[(n+d_{ss})T-\lambda]} d\lambda \right] \mathbf{B}_1 \\ + \left[\int_{(n+d_{ss})T}^{(n+1)T} e^{\mathbf{A}_2[(n+1)T-\lambda]} d\lambda \right] \mathbf{B}_2 \end{array} \right) \mathbf{U} \quad (8.2)$$

Aforementioned equation represents the minimum values of the rippled response at steady state as shown in figure 8-1. The maximum values of the rippled response is given by,

$$\mathbf{x}_{ss,\max} = e^{\mathbf{A}_1 d_{ss} T} \mathbf{x}_{ss,\min} + \left[\int_0^{d_{ss} T} e^{\mathbf{A}_1(d_{ss} T - \lambda)} d\lambda \right] \mathbf{B}_1 \mathbf{U} \quad (8.3)$$

Therefore the magnitude of the ripple is,

$$\mathbf{x}_{ripple} = \mathbf{x}_{ss,\max} - \mathbf{x}_{ss,\min} \quad (8.4)$$

Finally, the response at any continuous time t in between the sampled points of the discrete model is given for $t \in \text{Mode I}$ by,

$$\mathbf{x}(t) = e^{\mathbf{A}_1 t} \mathbf{x}[nT] + \left[\int_0^t e^{\mathbf{A}_1(t-\lambda)} d\lambda \right] \mathbf{B}_1 \mathbf{U} \quad (8.5)$$

And for $t \in \text{Mode II}$ by,

$$\mathbf{x}(t) = e^{\mathbf{A}_2(t-(n+d_n)T)} \mathbf{x}((n+d_n)T) + \left[\int_{(n+d_n)T}^t e^{\mathbf{A}_2(t-\lambda)} d\lambda \right] \mathbf{B}_2 \mathbf{U} \quad (8.6)$$

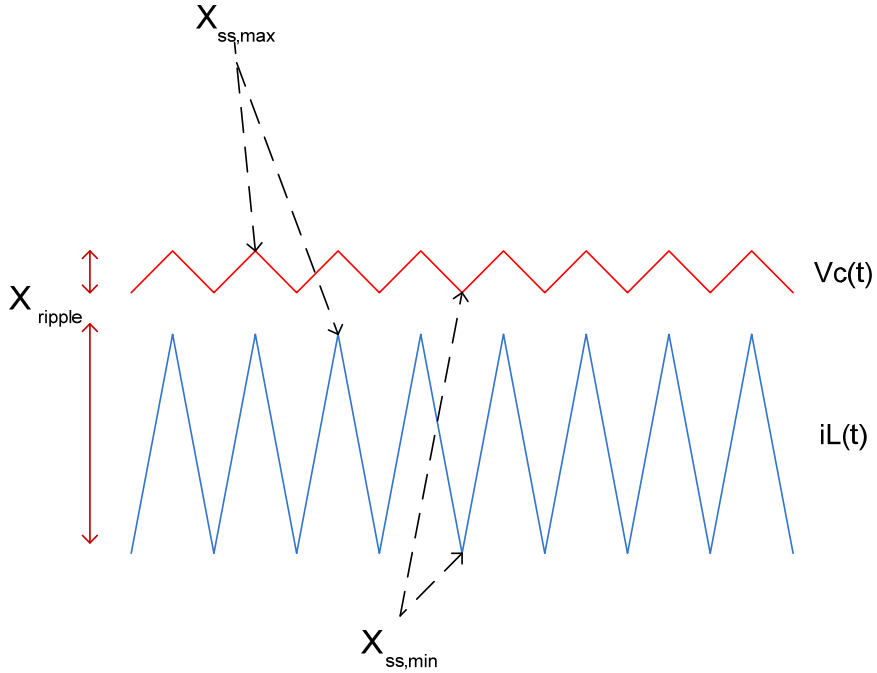


Figure 8-1 : Illustration of x_{ripple} , $x_{\text{ss,min}}$ and $x_{\text{ss,max}}$

8.3 Conditions for Stability and One Cycle Periodicity

In the previous section the Steady-State point was found by assuming stability and one cycle periodicity as n approaches infinity. But can those conditions be assured ? The first step is to examine the n^{th} state assuming that $x[0]$ is an initial state. For the first few switching cycle iterations the following set of equations are obtained,

$$\mathbf{x}[T] = \mathbf{A}_d \mathbf{x}[0] + \mathbf{B}_d \mathbf{u}[0]$$

$$\mathbf{x}[2T] = \mathbf{A}_d^2 \mathbf{x}[0] + \mathbf{A}_d \mathbf{B}_d \mathbf{u}[0] + \mathbf{B}_d \mathbf{u}[T]$$

$$\mathbf{x}[3T] = \mathbf{A}_d^3 \mathbf{x}[0] + \mathbf{A}_d^2 \mathbf{B}_d \mathbf{u}[0] + \mathbf{A}_d \mathbf{B}_d \mathbf{u}[T] + \mathbf{B}_d \mathbf{u}[2T]$$

So , from induction , the n^{th} state should be,

$$\mathbf{x}[n] = \mathbf{A}_d^n \mathbf{x}[0] + \sum_{j=0}^{n-1} \mathbf{A}_d^{n-1-j} \mathbf{B}_d \mathbf{u}[j] \quad (8.7)$$

Where ,

$$\mathbf{A}_d = e^{\mathbf{A}_2(1-d_n)T} e^{\mathbf{A}_1 d_n T} \quad (8.8)$$

$$\mathbf{B}_d = e^{\mathbf{A}_2(1-D)T} \left[\int_{nT}^{(n+d_n)T} e^{\mathbf{A}_1[(n+d_n)T-\lambda]} d\lambda \right] \mathbf{B}_1 + \left[\int_{(n+d_n)T}^{(n+1)T} e^{\mathbf{A}_2[(n+1)T-\lambda]} d\lambda \right] \mathbf{B}_2 \quad (8.9)$$

And the input $\mathbf{u}[nT]$ is assumed to be constant through each switching cycle.

Now, if for given parameter values the Eigenvalues (λ_1, λ_2) of the \mathbf{A}_d are determined the necessary and sufficient conditions for stability and periodicity are $|\lambda_1| < 1$ and $|\lambda_2| < 1$. This can be shown by examining the part of (8.7) responsible for the natural response . First the \mathbf{A}_d matrix is diagonalised then raised to power n ,

$$\mathbf{A}^n = \mathbf{V}_A \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix} \mathbf{V}_A^{-1} \quad (8.10)$$

It is apparent that ,

$$\lim_{\substack{n \rightarrow \infty \\ |\lambda_1|, |\lambda_2| < 1}} \mathbf{A}_d^n = \mathbf{0} \quad (8.11)$$

Which is expected since the natural response is supposed to die out and give way to the forced response. The case when $\lambda_1 = \lambda_2 = 1$ causes the natural response to keep cycling,

$$\lim_{\substack{n \rightarrow \infty \\ |\lambda_1|, |\lambda_2| = 1}} \mathbf{A}_d^n = \mathbf{V}_A \begin{bmatrix} e^{jn\angle\lambda_1} & 0 \\ 0 & e^{jn\theta\angle\lambda_2} \end{bmatrix} \mathbf{V}_A^{-1} \quad (8.12)$$

That happen when one of the eigenvalues is 1 also. So care must me taken to when designing the converter to insure the eigenvalues are less than one. Similarly, if the portion responsible for the forced response is examined assuming constant input,

$$\sum_{j=0}^{n-1} \mathbf{A}_d^{n-1-j} \mathbf{B}_d \mathbf{u}[j] \approx \left[\sum_{j=0}^{n-1} \mathbf{A}_d^{n-1-j} \right] \mathbf{B} \mathbf{U} \quad (8.13)$$

Diagonalising ,

$$\begin{aligned} \left[\sum_{j=0}^{n-1} \mathbf{A}^{n-1-j} \right] \mathbf{B} \mathbf{U} &= \left[\mathbf{V}_A \sum_{j=0}^{n-1} \begin{bmatrix} \lambda_1^{n-1-j} & 0 \\ 0 & \lambda_2^{n-1-j} \end{bmatrix} \mathbf{V}_A^{-1} \right] \mathbf{B} \mathbf{U} \\ &= \left[\mathbf{V}_A \begin{bmatrix} \sum_{k=0}^{n-1} \lambda_1^k & 0 \\ 0 & \sum_{k=0}^{n-1} \lambda_2^k \end{bmatrix} \mathbf{V}_A^{-1} \right] \mathbf{B} \mathbf{U} \end{aligned} \quad (8.14)$$

But the summations in (9.14) are geometric series of the form $\sum_{k=1}^n y^k$, which are known to

converge as n approaches infinity only if $|y| < 1$. So ,

$$\lim_{n \rightarrow \infty} \left[\mathbf{V}_A \begin{bmatrix} \sum_{k=0}^{n-1} \lambda_1^k & 0 \\ 0 & \sum_{k=0}^{n-1} \lambda_2^k \end{bmatrix} \mathbf{V}_A^{-1} \right] \mathbf{B} \mathbf{U} = \left[\mathbf{V}_A \begin{bmatrix} \frac{1}{1-\lambda_1} & 0 \\ 0 & \frac{1}{1-\lambda_2} \end{bmatrix} \mathbf{V}_A^{-1} \right] \mathbf{B} \mathbf{U} \quad (8.15)$$

only if $|\lambda_1| < 1$ and $|\lambda_2| < 1$.

In the majority of cases designing DC/DC converter using simplified first order models yields systems with eigenvalues within the stable region, unless some parasitic value is so large it disrupts the tenets of the first order model.

8.4 Approach to dealing with Defective Transition Matrices

Examining a component of the expression (8.1), say $e^{A_1 d_n T}$, when the symbolic evaluation of that term is attempted the following is obtained ,

$$e^{A_1 d_n T} = \begin{bmatrix} \frac{e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n \lambda_1}{2F_s}] (a_{11} - a_{14}) + \text{Cosh}[\frac{d_n \lambda_1}{2F_s}] \lambda_1)}{\lambda_1} & \frac{(e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} - e^{\frac{d_n(a_{11}+a_{14}+\lambda_1)}{2F_s}}) a_{12}}{\lambda_1} \\ -\frac{(e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} - e^{\frac{d_n(a_{11}+a_{14}+\lambda_1)}{2F_s}}) a_{13}}{\lambda_1} & \frac{e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n \lambda_1}{2F_s}] (-a_{11} + a_{14}) + \text{Cosh}[\frac{d_n \lambda_1}{2F_s}] \lambda_1)}{\lambda_1} \end{bmatrix} \quad (8.16)$$

Where,

$$\lambda_1 = \sqrt{4a_{12}a_{13} + (a_{11} - a_{14})^2} \quad (8.17)$$

And,

$$A_1 = \begin{bmatrix} a_{11} & a_{12} \\ a_{13} & a_{14} \end{bmatrix} \quad (8.18)$$

Note that λ_1 is taken to be only the part of the eigenvalue under the square root, this is a convention that will be used throughout this chapter and leads to lessening the amount of symbols that must be defined. Before moving to evaluating the other components and

multiplying everything together the difficulty of the case where $\lambda_1 = 0$ must be acknowledged and dealt with. That case means the A matrix is a defective matrix , which complicates symbolically evaluating the symbolic response , since it is clear that the components of (8.16) are indeterminate. The solution of this issue is to use the limit finding capabilities of CAS systems to get the symbolic limit of the transition matrix as λ_1 approaches zero. For the example above the limit turns to be,

$$\lim_{\lambda_1 \rightarrow 0} e^{A_1 d_n T} = \begin{bmatrix} \frac{e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} (2Fs+a_{11}d_n-a_{14}d_n)}{2Fs} & \frac{e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} a_{12}d_n}{Fs} \\ \frac{e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} a_{13}d_n}{Fs} & \frac{e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} (2Fs-a_{11}d_n+a_{14}d_n)}{2Fs} \end{bmatrix} \quad (8.19)$$

And now (8.19) can be used instead of (8.16) whenever $\lambda_1 = 0$. This is actually an advantage to the numerical case where encountering a defective matrix case can cause the problem to become ill-conditioned , especially if very close eigenvalues are misidentified as a pair.

It should be noted that defective matrices (or even singular !) sometimes occur when one of the parasitic parameters is idealized. In that case finding the limit symbolically is superior to having to deal with the numerical model.

The above discussion clearly shows that the symbolic response does not manifest in one form , but rather is dependent on the actual numerical values of the eigenvalues of each linear mode. Therefore what needs to be acquired is a family of symbolic formula's that cover the defective case of the eigenvalues in any mode approaching each other. The coming section will commence with the detailed derivation of those families of symbolic formulas.

8.5 Derivation and Summary of Symbolic Simulation Equations for basic Second Order Topologies

All second order topologies with two modes of operation share a linear piecewise State-Space representation with Transition matrices A_1 and A_2 and load matrices B_1 and B_2 where the notation is chosen so that the “1” index represents mode I and “2” index represents Mode “2” . The symbolic variables representing the components of those matrices are defined to be,

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} a_{11} & a_{12} \\ a_{13} & a_{14} \end{bmatrix} \\ \mathbf{A}_2 &= \begin{bmatrix} a_{21} & a_{22} \\ a_{23} & a_{24} \end{bmatrix} \\ \mathbf{B}_1 &= \begin{bmatrix} b_{11} & b_{12} \\ b_{13} & b_{14} \end{bmatrix} \\ \mathbf{B}_2 &= \begin{bmatrix} b_{21} & b_{22} \\ b_{23} & b_{24} \end{bmatrix} \end{aligned} \quad (8.20)$$

In (8.20) the symbols a_{ij} and b_{ij} are functions of the particular Topology be it buck , boost or buck-boost and the actual values of the circuit elements constituting those Topologies. Now equation (8.1) can be written in this form,

$$\mathbf{x}[(n+1)T] = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{bmatrix} \mathbf{x}[nT] + \begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix} \mathbf{U} \quad (8.21)$$

Which implies that,

$$\begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{bmatrix} = e^{\mathbf{A}_2(1-d_n)T} e^{\mathbf{A}_1 d_n T} \quad (8.22)$$

$$\begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix} = \begin{pmatrix} e^{\mathbf{A}_2(1-D)T} \left[\int_{nT}^{(n+d_n)T} e^{\mathbf{A}_1[(n+d_n)T-\lambda]} d\lambda \right] \mathbf{B}_1 \\ + \left[\int_{(n+d_n)T}^{(n+1)T} e^{\mathbf{A}_2[(n+1)T-\lambda]} d\lambda \right] \mathbf{B}_2 \end{pmatrix} \quad (8.23)$$

Equations (8.22) and (8.23) represent the crux of the methodology of this chapter, and obtaining any symbolic formula from them relating α_1 to α_4 and β_1 to β_2 to all other parameters is so long winded it is practically impossible to do it by hand. At this point, an automated CAS system like Mathematica comes to the rescue.

First, the partial eigenvalues turn out to be,

$$\lambda_1 = \sqrt{4a_{12}a_{13} + (a_{11} - a_{14})^2} \quad (8.24)$$

$$\lambda_2 = \sqrt{4a_{22}a_{23} + (a_{21} - a_{24})^2} \quad (8.25)$$

Next, if $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$,

$$\begin{aligned} \alpha_1 = & \frac{1}{2\lambda_1\lambda_2} e^{-\frac{(d_n-1)(a_{21}+a_{24}+\lambda_2)}{2Fs}} \left(2 \left(e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} - e^{\frac{d_n(a_{11}+a_{14}+\lambda_1)}{2Fs}} \right) \left(-1 + e^{\frac{(d_n-1)\lambda_2}{Fs}} \right) a_{13} a_{22} + \right. \\ & e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} \left(\sinh\left(\frac{d_n\lambda_1}{2Fs}\right) (a_{11} - a_{14}) + \cosh\left(\frac{d_n\lambda_1}{2Fs}\right) \lambda_1 \right) \left(1 - e^{\frac{(d_n-1)\lambda_2}{Fs}} \right) a_{21} - a_{24} + \lambda_2 + \\ & \left. e^{\frac{(d_n-1)\lambda_2}{Fs}} (a_{24} + \lambda_2) \right) \end{aligned} \quad (8.26)$$

$$\begin{aligned}
\alpha_2 = & \frac{1}{2\lambda_1\lambda_2} e^{-\frac{(d_n-1)(a_{21}+a_{24}+\lambda_2)}{2Fs}} \left(-2e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} \left(-1 + e^{\frac{(d_n-1)\lambda_2}{Fs}} \right) a_{22} \left(\sinh\left(\frac{d_n\lambda_1}{2Fs}\right) (a_{14} - a_{11}) + \right. \right. \\
& \left. \left. \cosh\left(\frac{d_n\lambda_1}{2Fs}\right) \lambda_1 \right) - \left(e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} - e^{\frac{d_n(a_{11}+a_{14}+\lambda_1)}{2Fs}} \right) a_{12} \left(a_{21} - a_{24} + \lambda_2 + e^{\frac{(d_n-1)\lambda_2}{Fs}} (-a_{21} + \right. \right. \\
& \left. \left. a_{24} + \lambda_2) \right) \right) \quad (8.27)
\end{aligned}$$

$$\begin{aligned}
\alpha_3 = & \frac{1}{2\lambda_1\lambda_2} e^{-\frac{(d_n-1)(a_{21}+a_{24}+\lambda_2)}{2Fs}} \left(-2e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} \left(-1 + e^{\frac{(d_n-1)\lambda_2}{Fs}} \right) a_{23} \left(\sinh\left(\frac{d_n\lambda_1}{2Fs}\right) (a_{11} - a_{14}) + \right. \right. \\
& \left. \left. \cosh\left(\frac{d_n\lambda_1}{2Fs}\right) \lambda_1 \right) - \left(e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} - e^{\frac{d_n(a_{11}+a_{14}+\lambda_1)}{2Fs}} \right) a_{13} \left(-a_{21} + a_{24} + \lambda_2 + e^{\frac{(d_n-1)\lambda_2}{Fs}} (a_{21} - \right. \right. \\
& \left. \left. a_{24} + \lambda_2) \right) \right) \quad (8.28)
\end{aligned}$$

$$\begin{aligned}
\alpha_4 = & \frac{1}{2\lambda_1\lambda_2} e^{-\frac{(d_n-1)(a_{21}+a_{24}+\lambda_2)}{2Fs}} \left(2 \left(e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} - e^{\frac{d_n(a_{11}+a_{14}+\lambda_1)}{2Fs}} \right) (-1 + e^{\frac{(d_n-1)\lambda_2}{Fs}}) a_{12} a_{23} + \right. \\
& \left. e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} \left(\sinh\left(\frac{d_n\lambda_1}{2Fs}\right) (a_{14} - a_{11}) + \cosh\left(\frac{d_n\lambda_1}{2Fs}\right) \lambda_1 \right) (-a_{21} + a_{24} + \lambda_2 + e^{\frac{(d_n-1)\lambda_2}{Fs}} (a_{21} - a_{24} + \right. \\
& \left. \lambda_2) \right) \quad (8.29)
\end{aligned}$$

$$\begin{aligned}
\beta_1 = & \frac{1}{4(-a_{12}a_{13}+a_{11}a_{14})\lambda_1} \left(2e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} b_{21} \left(-(-1 + e^{\frac{d_n\lambda_1}{Fs}}) (2a_{12}a_{13} + a_{14}(-a_{11} + a_{14})) + \right. \right. \\
& \left. \left. a_{14}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} a_{14}\lambda_1 + e^{\frac{d_n\lambda_1}{Fs}} a_{14}\lambda_1 \right) + 4a_{12}b_{23} \left(\lambda_1 + e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} (\text{Sinh}[\frac{d_n\lambda_1}{2Fs}]) (a_{11} + \right. \right. \\
& \left. \left. a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2Fs}](\lambda_1) \right) + \frac{1}{\lambda_2} e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2Fs}} b_{11} \left(-4(-1 + e^{\frac{(-1+d_n)\lambda_2}{Fs}}) a_{13} a_{22} (\lambda_1 + \right. \\
& \left. e^{\frac{(a_{11}+a_{14})d_n}{2Fs}} (\text{Sinh}[\frac{d_n\lambda_1}{2Fs}]) (a_{11} + a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2Fs}](\lambda_1) \right) + e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2Fs}} \left(-(-1 + \right.
\end{aligned}$$

$$\begin{aligned}
& e^{\frac{d_n \lambda_1}{F_s}} (2a_{12}a_{13} + a_{14}(-a_{11} + a_{14})) + a_{14}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{14}\lambda_1 + e^{\frac{d_n \lambda_1}{F_s}} a_{14}\lambda_1) (-(-1 + \\
& e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{21} - a_{24} + \lambda_2 + e^{\frac{(-1+d_n)\lambda_2}{F_s}} (a_{24} + \lambda_2))) + \\
& \frac{1}{\lambda_2} 2e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{13} (-e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{22} (-(-1 + e^{\frac{d_n \lambda_1}{F_s}}) (a_{11}^2 + \\
& 2a_{12}a_{13} - a_{11}a_{14}) + a_{11}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{11}\lambda_1 + e^{\frac{d_n \lambda_1}{F_s}} a_{11}\lambda_1) + a_{12}(\lambda_1 + \\
& e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n \lambda_1}{2F_s}](a_{11} + a_{14}) - \text{Cosh}[\frac{d_n \lambda_1}{2F_s}]\lambda_1)) (-(-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{21} - a_{24} + \lambda_2 + \\
& e^{\frac{(-1+d_n)\lambda_2}{F_s}} (a_{24} + \lambda_2)))) \tag{8.30}
\end{aligned}$$

$$\begin{aligned}
\beta_2 &= \frac{1}{4(-a_{12}a_{13}+a_{11}a_{14})\lambda_1} (2e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} b_{22} (-(-1 + e^{\frac{d_n \lambda_1}{F_s}}) (2a_{12}a_{13} + a_{14}(-a_{11} + a_{14})) + \\
& a_{14}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{14}\lambda_1 + e^{\frac{d_n \lambda_1}{F_s}} a_{14}\lambda_1) + 4a_{12}b_{24}(\lambda_1 + e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n \lambda_1}{2F_s}](a_{11} + \\
& a_{14}) - \text{Cosh}[\frac{d_n \lambda_1}{2F_s}]\lambda_1)) + \frac{1}{\lambda_2} e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{12} (-4(-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{13}a_{22}(\lambda_1 + \\
& e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n \lambda_1}{2F_s}](a_{11} + a_{14}) - \text{Cosh}[\frac{d_n \lambda_1}{2F_s}]\lambda_1)) + e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-(-1 + \\
& e^{\frac{d_n \lambda_1}{F_s}}) (2a_{12}a_{13} + a_{14}(-a_{11} + a_{14})) + a_{14}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{14}\lambda_1 + e^{\frac{d_n \lambda_1}{F_s}} a_{14}\lambda_1) (-(-1 + \\
& e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{21} - a_{24} + \lambda_2 + e^{\frac{(-1+d_n)\lambda_2}{F_s}} (a_{24} + \lambda_2))) + \\
& \frac{1}{\lambda_2} 2e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{14} (-e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{22} (-(-1 + e^{\frac{d_n \lambda_1}{F_s}}) (a_{11}^2 + \\
& 2a_{12}a_{13} - a_{11}a_{14}) + a_{11}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{11}\lambda_1 + e^{\frac{d_n \lambda_1}{F_s}} a_{11}\lambda_1) + a_{12}(\lambda_1 + \\
& e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n \lambda_1}{2F_s}](a_{11} + a_{14}) - \text{Cosh}[\frac{d_n \lambda_1}{2F_s}]\lambda_1)) (-(-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}}) a_{21} - a_{24} + \lambda_2 + \\
& e^{\frac{(-1+d_n)\lambda_2}{F_s}} (a_{24} + \lambda_2)))) \tag{8.31}
\end{aligned}$$

$$\begin{aligned}
\beta_3 = & \frac{1}{4(-a_{12}a_{13}+a_{11}a_{14})\lambda_1} (2e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} b_{23}(-(-1 + e^{\frac{d_n\lambda_1}{F_s}})(a_{11}^2 + 2a_{12}a_{13} - a_{11}a_{14}) + \\
& a_{11}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{11}\lambda_1 + e^{\frac{d_n\lambda_1}{F_s}} a_{11}\lambda_1) + 4a_{13}b_{21}(\lambda_1 + e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n\lambda_1}{2F_s}](a_{11} + \\
& a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2F_s}]\lambda_1)) + \frac{1}{\lambda_2} e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{13}(-4(-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{12}a_{23}(\lambda_1 + \\
& e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n\lambda_1}{2F_s}](a_{11} + a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2F_s}]\lambda_1)) + e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-(-1 + e^{\frac{d_n\lambda_1}{F_s}})(a_{11}^2 + \\
& 2a_{12}a_{13} - a_{11}a_{14}) + a_{11}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{11}\lambda_1 + e^{\frac{d_n\lambda_1}{F_s}} a_{11}\lambda_1)((-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{21} + \\
& a_{24} + \lambda_2 + e^{\frac{(-1+d_n)\lambda_2}{F_s}} (-a_{24} + \lambda_2))) + \frac{1}{\lambda_2} 2e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{11}(-e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-1 + \\
& e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{23}(-(-1 + e^{\frac{d_n\lambda_1}{F_s}})(2a_{12}a_{13} + a_{14}(-a_{11} + a_{14}))) + a_{14}\lambda_1 - \\
& 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{14}\lambda_1 + e^{\frac{d_n\lambda_1}{F_s}} a_{14}\lambda_1) + a_{13}(\lambda_1 + e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n\lambda_1}{2F_s}](a_{11} + a_{14}) - \\
& \text{Cosh}[\frac{d_n\lambda_1}{2F_s}]\lambda_1))((-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{21} + a_{24} + \lambda_2 + e^{\frac{(-1+d_n)\lambda_2}{F_s}} (-a_{24} + \lambda_2)))) \quad (8.32)
\end{aligned}$$

$$\begin{aligned}
\beta_4 = & \frac{1}{4(-a_{12}a_{13}+a_{11}a_{14})\lambda_1} (2e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} b_{24}(-(-1 + e^{\frac{d_n\lambda_1}{F_s}})(a_{11}^2 + 2a_{12}a_{13} - a_{11}a_{14}) + \\
& a_{11}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{11}\lambda_1 + e^{\frac{d_n\lambda_1}{F_s}} a_{11}\lambda_1) + 4a_{13}b_{22}(\lambda_1 + e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n\lambda_1}{2F_s}](a_{11} + \\
& a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2F_s}]\lambda_1)) + \frac{1}{\lambda_2} e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{14}(-4(-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{12}a_{23}(\lambda_1 + \\
& e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n\lambda_1}{2F_s}](a_{11} + a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2F_s}]\lambda_1)) + e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-(-1 + e^{\frac{d_n\lambda_1}{F_s}})(a_{11}^2 + \\
& 2a_{12}a_{13} - a_{11}a_{14}) + a_{11}\lambda_1 - 2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{11}\lambda_1 + e^{\frac{d_n\lambda_1}{F_s}} a_{11}\lambda_1)((-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{21} + \\
& a_{24} + \lambda_2 + e^{\frac{(-1+d_n)\lambda_2}{F_s}} (-a_{24} + \lambda_2))) + \frac{1}{\lambda_2} 2e^{-\frac{(-1+d_n)(a_{21}+a_{24}+\lambda_2)}{2F_s}} b_{12}(-e^{\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} (-1 + \\
& e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{23}(-(-1 + e^{\frac{d_n\lambda_1}{F_s}})(2a_{12}a_{13} + a_{14}(-a_{11} + a_{14}))) + a_{14}\lambda_1 -
\end{aligned}$$

$$2e^{-\frac{d_n(a_{11}+a_{14}-\lambda_1)}{2F_s}} a_{14}\lambda_1 + e^{\frac{d_n\lambda_1}{F_s}} a_{14}\lambda_1) + a_{13}(\lambda_1 + e^{\frac{(a_{11}+a_{14})d_n}{2F_s}} (\text{Sinh}[\frac{d_n\lambda_1}{2F_s}](a_{11} + a_{14}) - \text{Cosh}[\frac{d_n\lambda_1}{2F_s}]\lambda_1))((-1 + e^{\frac{(-1+d_n)\lambda_2}{F_s}})a_{21} + a_{24} + \lambda_2 + e^{\frac{(-1+d_n)\lambda_2}{F_s}}(-a_{24} + \lambda_2)))) \quad (8.33)$$

The values of the α_i 's and the β_i 's for the other cases where,

1. $\lambda_1 = 0$ and $\lambda_2 \neq 0$
2. $\lambda_1 \neq 0$ and $\lambda_2 = 0$
3. $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$

can be obtained from (8.26)-(8.33) through keeping any λ_i that is equal to zero as a symbol, and then finding the algebraic limit of the resultant expression as λ_i approaches zero. This can be straightforwardly done with any CAS package. In case it is desired to implement the model in C or any other non-CAS programming environment, all the expressions of the limits for all cases can be statically computed only once in a CAS system and then plugged in into such programming environment. Equations for other three cases are not listed here due to their length and because they add no substance to the matter.

Tables 8-1 to 8-3 shows what the a_{ij} 's and b_{ij} 's are equivalent to for the case of the three main second order topologies, the buck, the boost and the buck-boost(Shown in figure 8-2, with all relevant parameters and parasitics).

It can also be noted that the λ_i 's need not exactly be zero to use the limited expression, it is sufficient for them to be close to zero. This makes this methodology superior to numerical evaluation at the difficult situation where double roots appear to be distinct and complex conjugate because of limited machine accuracy.

Table 8-1 : : a_{ij} and b_{ij} parameters for the Buck Converter

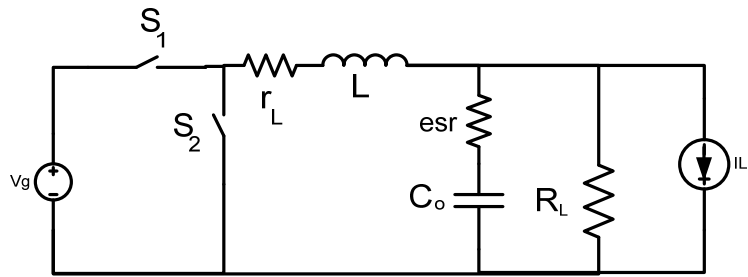
$\begin{bmatrix} a_{11} & a_{12} \\ a_{13} & a_{14} \end{bmatrix}$	$\begin{bmatrix} \frac{-(\text{esr} \cdot R_L + r_L \cdot \text{esr} + r_L \cdot R_L + r_{s1} \cdot \text{esr} + r_{s1} \cdot R_L)}{L \cdot (\text{esr} + R_L)} & \frac{-R_L}{L \cdot (\text{esr} + R_L)} \\ \frac{R_L}{C_o \cdot (\text{esr} + R_L)} & \frac{-1}{C_o \cdot (\text{esr} + R_L)} \end{bmatrix}$
$\begin{bmatrix} a_{21} & a_{22} \\ a_{23} & a_{24} \end{bmatrix}$	$\begin{bmatrix} \frac{-(\text{esr} \cdot R_L + r_L \cdot \text{esr} + r_L \cdot R_L + r_{s2} \cdot \text{esr} + r_{s2} \cdot R_L)}{L \cdot (\text{esr} + R_L)} & \frac{-R_L}{L \cdot (\text{esr} + R_L)} \\ \frac{R_L}{C_o \cdot (\text{esr} + R_L)} & \frac{-1}{C_o \cdot (\text{esr} + R_L)} \end{bmatrix}$
$\begin{bmatrix} b_{11} & b_{12} \\ b_{13} & b_{14} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{L} & \text{esr} \cdot \frac{R_L}{L \cdot (\text{esr} + R_L)} \\ 0 & \frac{-R_L}{C_o \cdot (\text{esr} + R_L)} \end{bmatrix}$
$\begin{bmatrix} b_{21} & b_{22} \\ b_{23} & b_{24} \end{bmatrix}$	$\begin{bmatrix} 0 & \text{esr} \cdot \frac{R_L}{L \cdot (\text{esr} + R_L)} \\ 0 & \frac{-R_L}{C_o \cdot (\text{esr} + R_L)} \end{bmatrix}$

Table 8-2 : a_{ij} and b_{ij} parameters for the Boost Converter

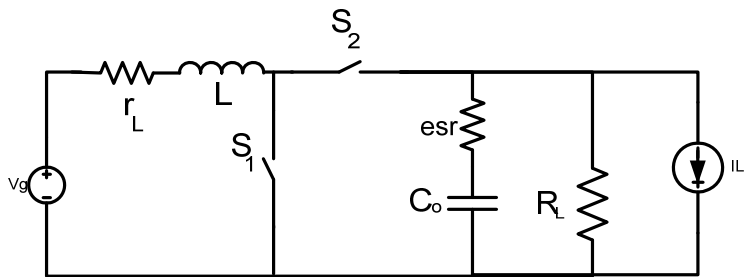
$\begin{bmatrix} a_{11} & a_{12} \\ a_{13} & a_{14} \end{bmatrix}$	$\begin{bmatrix} -\frac{r_L + r_{s1}}{L} & 0 \\ 0 & -\frac{1}{C_o \cdot (R_L + \text{esr})} \end{bmatrix}$
$\begin{bmatrix} a_{21} & a_{22} \\ a_{23} & a_{24} \end{bmatrix}$	$\begin{bmatrix} \frac{\text{esr}^2}{L \cdot R_L + L \cdot \text{esr}} & -\frac{\text{esr} + r_L + r_{s2}}{L} & -\frac{R_L}{L \cdot (R_L + \text{esr})} \\ \frac{R_L}{C_o \cdot (R_L + \text{esr})} & & -\frac{1}{C_o \cdot (R_L + \text{esr})} \end{bmatrix}$
$\begin{bmatrix} b_{11} & b_{12} \\ b_{13} & b_{14} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{R_L}{C_o \cdot (R_L + \text{esr})} \end{bmatrix}$
$\begin{bmatrix} b_{21} & b_{22} \\ b_{23} & b_{24} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{L} & \frac{R_L \cdot \text{esr}}{L \cdot (R_L + \text{esr})} \\ 0 & -\frac{R_L}{C_o \cdot (R_L + \text{esr})} \end{bmatrix}$

Table 8-3 : : a_{ij} and b_{ij} parameters for the Buck-Boost Converter

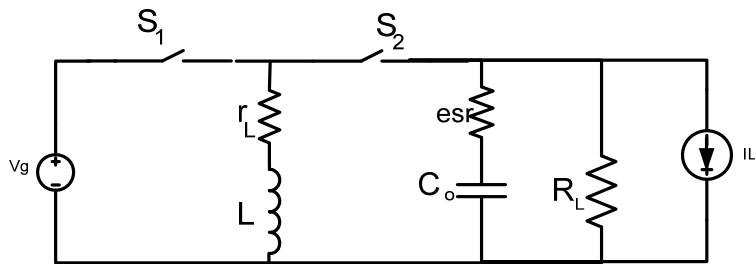
		Buck-Boost
$\begin{bmatrix} a_{11} & a_{12} \\ a_{13} & a_{14} \end{bmatrix}$	$\begin{bmatrix} -\frac{r_L + r_{s1}}{L} & 0 \\ 0 & -\frac{1}{C_o \cdot (R_L + esr)} \end{bmatrix}$	
$\begin{bmatrix} a_{21} & a_{22} \\ a_{23} & a_{24} \end{bmatrix}$	$\begin{bmatrix} \frac{esr^2}{L \cdot R_L + L \cdot esr} & -\frac{esr + r_L + r_{s2}}{L} & \frac{R_L}{L \cdot (R_L + esr)} \\ -\frac{R_L}{C_o \cdot (R_L + esr)} & & -\frac{1}{C_o \cdot (R_L + esr)} \end{bmatrix}$	
$\begin{bmatrix} b_{11} & b_{12} \\ b_{13} & b_{14} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{R_L}{C_o \cdot (R_L + esr)} \end{bmatrix}$	
$\begin{bmatrix} b_{21} & b_{22} \\ b_{23} & b_{24} \end{bmatrix}$	$\begin{bmatrix} 0 & -\frac{R_L \cdot esr}{L \cdot (R_L + esr)} \\ 0 & -\frac{R_L}{C_o \cdot (R_L + esr)} \end{bmatrix}$	



(a)



(b)



(c)

Figure 8-2 : (a) Buck ,(b) Boost and (c) Buck-Boost with all relevant parameters and parasitic

8.6 Symbolic linearization for Acquisition of dynamic characteristics

Now comes the problem of deriving the dynamic characteristics through the derived symbolic formulas of the previous section. Naturally, some sort of linearization is necessary. This can be done through the classical technique of decomposing each symbolic variable that is a function of the state and the duty cycle into steady-state and small signal components. As follows, by manipulating (8.21),

$$\begin{aligned} \mathbf{x}_{ss,\min} + \tilde{\mathbf{x}}[(n+1)T] = & \left(\begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} + \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \right) (\mathbf{x}_{ss,\min} + \tilde{\mathbf{x}}[nT]) \\ & + \left(\begin{bmatrix} \bar{\beta}_1 & \bar{\beta}_2 \\ \bar{\beta}_3 & \bar{\beta}_4 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 \\ \tilde{\beta}_3 & \tilde{\beta}_4 \end{bmatrix} \right) \mathbf{U} \end{aligned} \quad (8.34)$$

Separating the small signal components from the large signal ones gives,

$$\begin{aligned} \mathbf{x}_{ss,\min} + \tilde{\mathbf{x}}[(n+1)T] = & \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} \mathbf{x}_{ss,\min} + \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} \tilde{\mathbf{x}}[nT] \\ & + \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \mathbf{x}_{ss,\min} + \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \tilde{\mathbf{x}}[nT] \\ & + \begin{bmatrix} \bar{\beta}_1 & \bar{\beta}_2 \\ \bar{\beta}_3 & \bar{\beta}_4 \end{bmatrix} \mathbf{U} + \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 \\ \tilde{\beta}_3 & \tilde{\beta}_4 \end{bmatrix} \mathbf{U} \end{aligned} \quad (8.35)$$

Cancelling products of small perturbation variables finally gives,

$$\tilde{\mathbf{x}}[(n+1)T] = \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} \tilde{\mathbf{x}}[nT] + \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \mathbf{x}_{ss,\min} + \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 \\ \tilde{\beta}_3 & \tilde{\beta}_4 \end{bmatrix} \mathbf{U} \quad (8.35)$$

To emphasize the fact that each small-signal symbol is a linear function of the small change of the duty cycle, the convention of factoring out the duty cycle variable will be followed, giving the alternative form,

$$\tilde{\mathbf{x}}[(n+1)T] = \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} \tilde{\mathbf{x}}[nT] + \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \tilde{d}[nT] \mathbf{x}_{ss,\min} + \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 \\ \tilde{\beta}_3 & \tilde{\beta}_4 \end{bmatrix} \tilde{d}[nT] \mathbf{U} \quad (8.36)$$

The above recurrence equation represents a linear discrete system with constant coefficients. Therefore it can be used to investigate the dynamic properties of the system, and to insure a stable closed loop design through various Linear Control Theory techniques. One popular technique is the Bode plot that represents the frequency response of the open loop system. It is obtained by first applying the Hilbert transform to (8.36),

$$z\mathbf{X}[z] = \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} \mathbf{X}[z] + \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \mathbf{x}_{ss,\min} d[z] + \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 \\ \tilde{\beta}_3 & \tilde{\beta}_4 \end{bmatrix} \mathbf{U}d[z] \quad (8.37)$$

If the converter is operating in voltage mode control, then there is no interest in both states and only one variable (the output voltage in VM) must be regulated. In that case the variable of interest is obtained by multiplying $\mathbf{X}[z]$ with a special Matrix \mathbf{C} , and then after rearranging (8.37) becomes,

$$\frac{\mathbf{Y}[z]}{d[z]} = [c_1 \quad c_2] \left[z\mathbf{I} - \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 \\ \bar{\alpha}_3 & \bar{\alpha}_4 \end{bmatrix} \right]^{-1} \left[\begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 \\ \tilde{\alpha}_3 & \tilde{\alpha}_4 \end{bmatrix} \mathbf{x}_{ss,\min} + \begin{bmatrix} \tilde{\beta}_1 & \tilde{\beta}_2 \\ \tilde{\beta}_3 & \tilde{\beta}_4 \end{bmatrix} \mathbf{U} \right] \quad (8.38)$$

Table (8-4) shows $[c_1 \ c_2]$ for all 3 major topologies and for the regulated output being the Output voltage. It is apparent that the form of the C matrix for current mode is trivial, and always equal to $[1 \ 0]$.

Table 8-4 : $[c_1 \ c_2]$ for different controlled outputs and topologies

	$\mathbf{Y}[z] = \mathbf{V}_o[z]$
Buck	$\begin{bmatrix} \frac{R_L esr}{R_L + esr} & \frac{R_L}{R_L + esr} \end{bmatrix}$
Boost	$\begin{bmatrix} 0 & \frac{R_L}{R_L + esr} \end{bmatrix}$
Buck-Boost	$\begin{bmatrix} 0 & \frac{R_L}{R_L + esr} \end{bmatrix}$

It is not necessary to find the static forms of the $\tilde{\alpha}_i$'s and $\tilde{\beta}_i$'s and then store them internally to optimize simulation speed, since they need only be computed once, unlike the case of the α_i 's and β_i 's which are functions of the particular cycle and the particular duty modulation within that cycle. Of course the $\bar{\alpha}_i$'s and $\bar{\beta}_i$'s are found by simply computing α_i 's and β_i 's at the steady state operating point (with steady state duty cycle of, say, D_{ss}). An algorithm that can be implemented in a general CAS platform is outlined in figure 8-3.

```

For i = 1..4
For  $\gamma = \{\alpha, \beta\}$ 
 $\tilde{\gamma}_i \leftarrow \gamma_i|_{d_n=D_{ss}+\tilde{d}_n} - \gamma_i|_{d_n=D_{ss}}$ 
 $\tilde{\gamma}_i \leftarrow \text{Series Expansion of } \tilde{\gamma}_i \text{ at } \tilde{d}_n = 0$ 
 $\tilde{\gamma}_i \leftarrow \text{Ignore Higher Order Terms of Series } \tilde{\gamma}_i$ 
 $\tilde{\gamma}_i \leftarrow \frac{\tilde{\gamma}_i}{\tilde{d}_n}$ 

```

Figure 8-3 : Algorithm for CAS aided acquisition of the perturbed symbolic variables

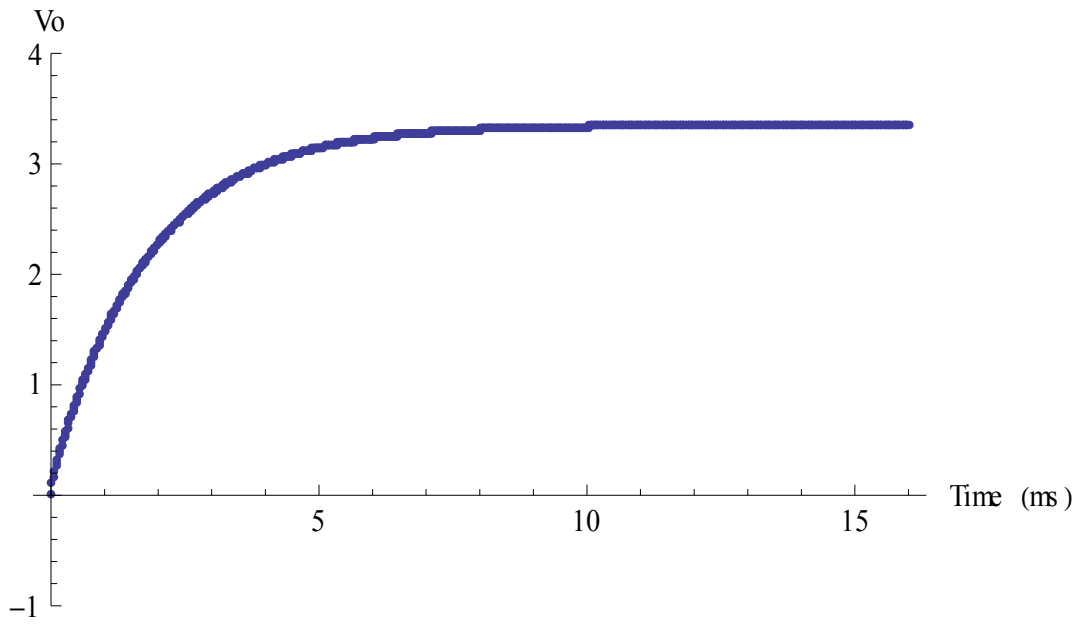
8.7 Example : Application to a synchronous Buck topology implemented in Mathematica

This section goes through an example of the technique explained in this chapter using the industry standard Computer Algebra System “Mathematica”. It is one of the oldest and most powerful CAS platforms available in the market with superior symbolic manipulation and data visualization capabilities, both those properties aid symbolic simulation immensely. A Synchronous Buck converter is chosen with the following parameters(refer to figure 8-2 (a)),

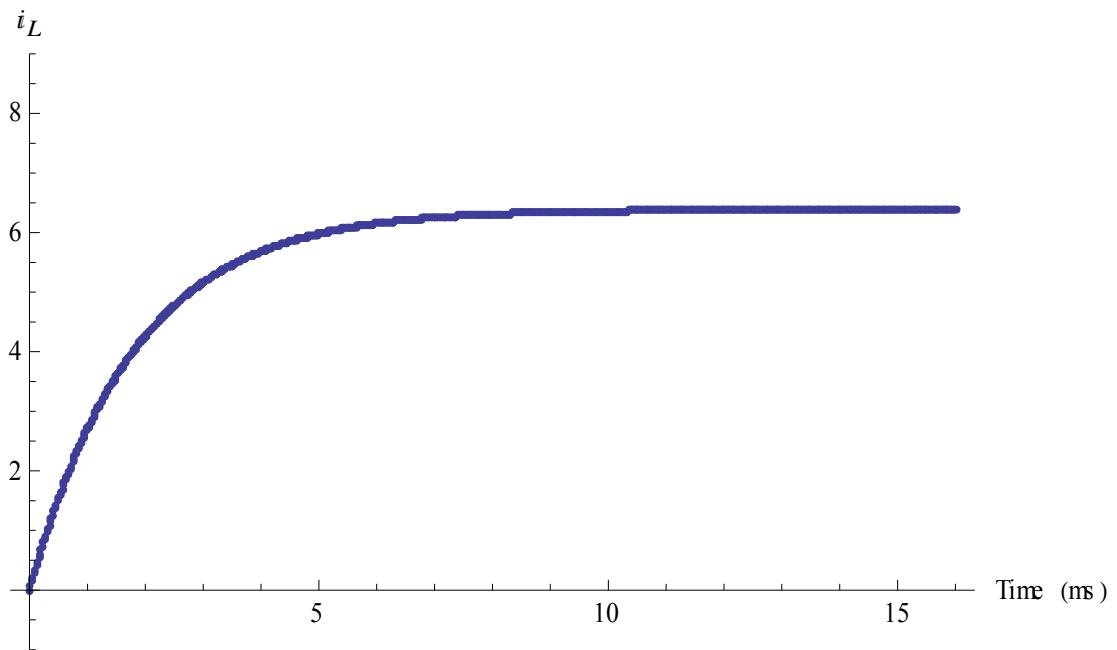
$$\begin{aligned}
V_g &= 12V \quad , \quad V_o = 3.3V \quad , \quad R_L = 0.5\Omega \quad , \quad r_L = 20\mu\Omega \\
esr &= 33m\Omega \quad , \quad r_{s1} = 10m\Omega \quad , \quad r_{s2} = 10m\Omega \quad , \quad F_s = 50\text{ KHz} \\
i_o &= 0 \quad , \quad L = 917mH \quad , \quad C_o = 40\mu F
\end{aligned}$$

The buck is represented within Mathematica as a list, which is manipulated through the programs written in a functional programming paradigm, similar to Lisp. After implementing the main portion of the simulation code guided by the equations presented in this chapter any of

the parameters of the converter can be manipulated at will and the effect of varying them can be easily observed. In the meantime the processing time is a fraction of the time needed for numerical simulation. For example, obtaining the large signal discrete response for a certain duty cycle signal through symbolic simulation for 800 switching cycles takes one tenth of the time needed to obtain same results from the numerical method. A tenfold improvement in simulation time is rather remarkable and arises due to the fact that the time consuming step of solving for the eigenvalues and eigenvectors is done only once symbolically, then used and reused for each particular topology and design. The results of said discrete large signal symbolic simulation are shown in figure 8-4.



(a)



(b)

Figure 8-4 : Discrete Large signal response for (a) Output Voltage , (b) Inductor Current

The switching response as described in (8.5) and (8.6) can be built upon the result of the discrete simulation. Moreover it is in a symbolic form as a function of time which allows arbitrary precision in plotting it without having to redo the simulation. For example assuming a steady duty cycle of 0.3 the switching expressions for the first ten switching cycles is found to be,

$$i_L(t) =$$

$$\left\{ \begin{array}{ll} 23.9986 + 0.00298635 e^{-49445.2t} - 24.0015 e^{-551.571t} & 0 \leq t < 6. \times 10^{-6} \\ 5.55112 \times 10^{-17} e^{-24998.4t} \cosh(24446.8t) + 8.32667 \times 10^{-17} e^{-49445.2t} - 1.77636 \times 10^{-14} e^{-551.571t} & t \geq 0.0001 \\ 0.0784908 e^{-24998.4t} \cosh(24446.8t) + 0.000160366 e^{-49445.2t} - 0.000119594 e^{-551.571t} + 0.0808743 e^{-24998.4t} \sinh(24446.8t) & 6. \times 10^{-6} \leq t < 0.00002 \\ 0.0776936 e^{-24998.4t} \cosh(24446.8t) + 0.00937124 e^{-49445.2t} - 24.2683 e^{-551.571t} + 0.0824426 e^{-24998.4t} \sinh(24446.8t) + 23.9986 & 0.00002 \leq t < 0.000026 \\ 0.154343 e^{-24998.4t} \cosh(24446.8t) + 0.00258768 e^{-49445.2t} - 0.000725805 e^{-551.571t} + 0.167126 e^{-24998.4t} \sinh(24446.8t) & 0.000026 \leq t < 0.00004 \\ 0.148553 e^{-24998.4t} \cosh(24446.8t) + 0.0304937 e^{-49445.2t} - 24.5382 e^{-551.571t} + 0.173985 e^{-24998.4t} \sinh(24446.8t) + 23.9986 & 0.00004 \leq t < 0.000046 \\ 0.217189 e^{-24998.4t} \cosh(24446.8t) + 0.0144205 e^{-49445.2t} - 0.00152126 e^{-551.571t} + 0.268546 e^{-24998.4t} \sinh(24446.8t) & 0.000046 \leq t < 0.00006 \\ 0.192423 e^{-24998.4t} \cosh(24446.8t) + 0.0978007 e^{-49445.2t} - 24.8113 e^{-551.571t} + 0.294503 e^{-24998.4t} \sinh(24446.8t) + 23.9986 & 0.00006 \leq t < 0.000066 \\ 0.234343 e^{-24998.4t} \cosh(24446.8t) + 0.0603419 e^{-49445.2t} - 0.0023942 e^{-551.571t} + 0.417619 e^{-24998.4t} \sinh(24446.8t) & 0.000066 \leq t < 0.00008 \\ 0.144556 e^{-24998.4t} \cosh(24446.8t) + 0.306721 e^{-49445.2t} - 25.0874 e^{-551.571t} + 0.50865 e^{-24998.4t} \sinh(24446.8t) + 23.9986 & 0.00008 \leq t < 0.000086 \\ 0.103566 e^{-24998.4t} \cosh(24446.8t) + 0.221311 e^{-49445.2t} - 0.00330263 e^{-551.571t} + 0.716517 e^{-24998.4t} \sinh(24446.8t) & 0.000086 \leq t < 0.0001 \end{array} \right.$$

$$v_c(t) =$$

$$\left\{ \begin{array}{ll} -11.9993 e^{-24998.4t} \cosh(24446.8t) - 12.27 e^{-24998.4t} \sinh(24446.8t) + 11.9993 & 0 \leq t < 6. \times 10^{-6} \\ 1.77636 \times 10^{-15} e^{-24998.4t} \cosh(24446.8t) + 4.16334 \times 10^{-17} e^{-551.571t} + 1.24345 \times 10^{-14} e^{-24998.4t} \sinh(24446.8t) & t \geq 0.0001 \\ 1.21937 e^{-24998.4t} \cosh(24446.8t) - 8.25176 e^{-49445.2t} + 0.165123 e^{-551.571t} - 26.5866 e^{-24998.4t} \sinh(24446.8t) + 11.9993 & 0.00008 \leq t < 0.000086 \\ -8.11094 e^{-24998.4t} \cosh(24446.8t) - 2.31353 e^{-49445.2t} + 0.123089 e^{-551.571t} - 16.9771 e^{-24998.4t} \sinh(24446.8t) + 11.9993 & 0.000086 \leq t < 0.000066 \\ -11.0238 e^{-24998.4t} \cosh(24446.8t) - 0.576379 e^{-49445.2t} + 0.0815341 e^{-551.571t} - 13.7882 e^{-24998.4t} \sinh(24446.8t) + 11.9993 & 0.00004 \leq t < 0.000046 \\ -11.8447 e^{-24998.4t} \cosh(24446.8t) - 0.10763 e^{-49445.2t} + 0.0404806 e^{-551.571t} - 12.6943 e^{-24998.4t} \sinh(24446.8t) + 11.9993 & 0.00002 \leq t < 0.000026 \\ 10.0299 e^{-24998.4t} \cosh(24446.8t) - 13.8918 e^{-49445.2t} + 0.207308 e^{-551.571t} - 10.0332 e^{-24998.4t} \sinh(24446.8t) & 0.000086 \leq t < 0.0001 \\ 2.73395 e^{-24998.4t} \cosh(24446.8t) - 4.15374 e^{-49445.2t} + 0.164809 e^{-551.571t} - 2.73637 e^{-24998.4t} \sinh(24446.8t) & 0.000066 \leq t < 0.00008 \\ 0.652878 e^{-24998.4t} \cosh(24446.8t) - 1.16394 e^{-49445.2t} + 0.122788 e^{-551.571t} - 0.654416 e^{-24998.4t} \sinh(24446.8t) & 0.000046 \leq t < 0.00006 \\ 0.116927 e^{-24998.4t} \cosh(24446.8t) - 0.289725 e^{-49445.2t} + 0.0812637 e^{-551.571t} - 0.11766 e^{-24998.4t} \sinh(24446.8t) & 0.000026 \leq t < 0.00004 \\ 0.00720857 e^{-24998.4t} \cosh(24446.8t) - 0.0540202 e^{-49445.2t} + 0.0402857 e^{-551.571t} - 0.0073295 e^{-24998.4t} \sinh(24446.8t) & 6. \times 10^{-6} \leq t < 0.00002 \end{array} \right.$$

(8.39)

The range of switching cycles to be obtained can be adjusted so as to zoom in on a desired time window of the switched response, the graphs of the above equations are shown in figure 8-5. A plot of a time window closer to steady state between the 500th and the 510th switching cycles is shown in figure (8-6).

The presented methodology allows for the freedom of examining the effect on the response of the variance of any parameter, as seen in figure (8-7) where one switching cycle at steady state is plotted for all possible duty ratios in 0.1 increments.

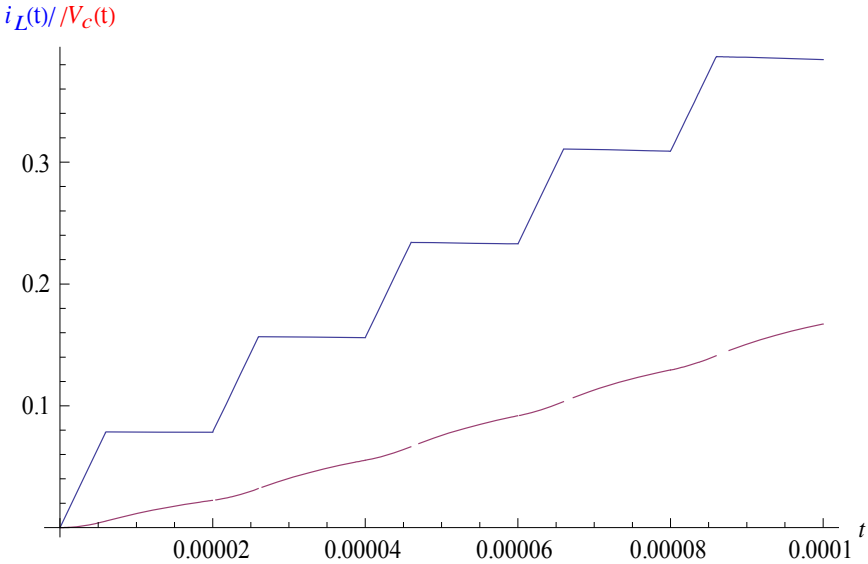


Figure 8-5 : First 10 switching cycles simulation for d=0.3

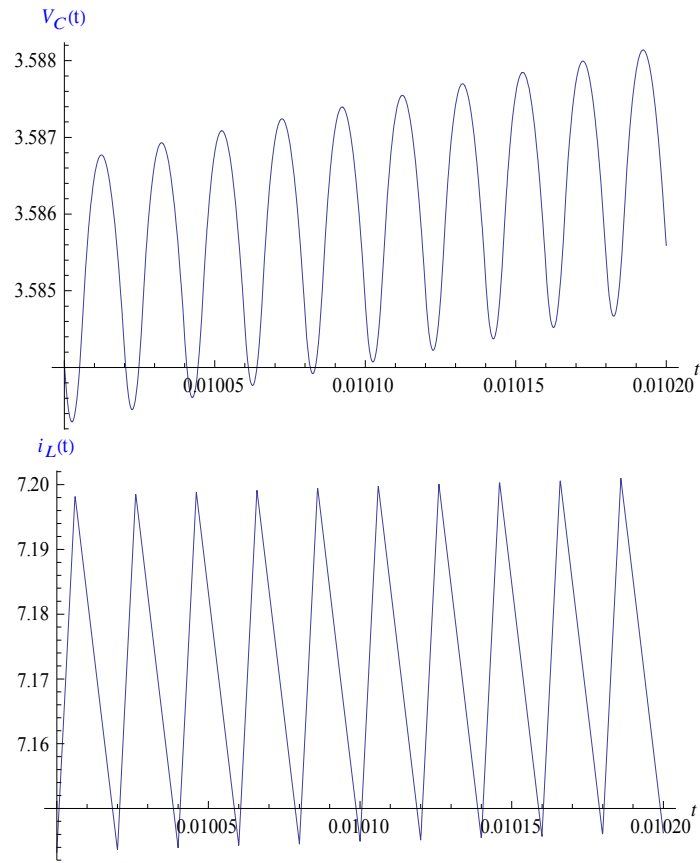
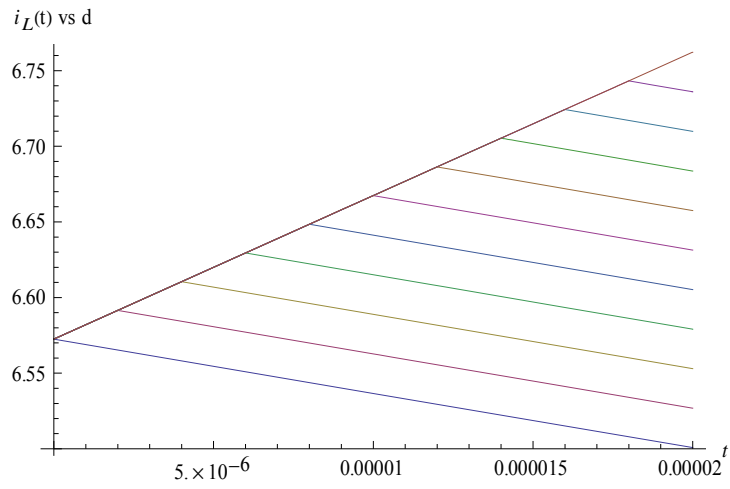
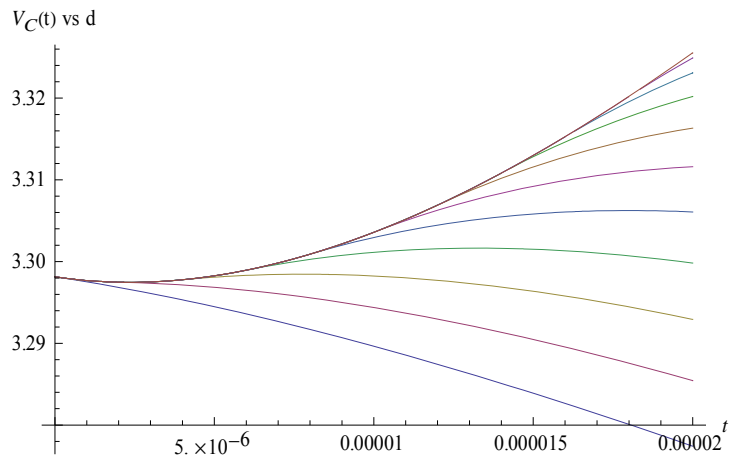


Figure 8-6 : Switching cycles between $500T_s$ and $510T_s$

Figure 8-8 shows the ripple in the inductor current and capacitor voltage waveforms as a function of switching frequency. As expected increasing the frequency reduces the ripple. Figure 8-9 shows the inductor current ripple versus two other parameters, the inductance L and the duty cycle of the operating point.



(a)



(b)

Figure 8-7 : One (a) Inductor current and (b) Capacitor voltage switching cycle plotted versus d between 0 to 1 in 0.1 increments

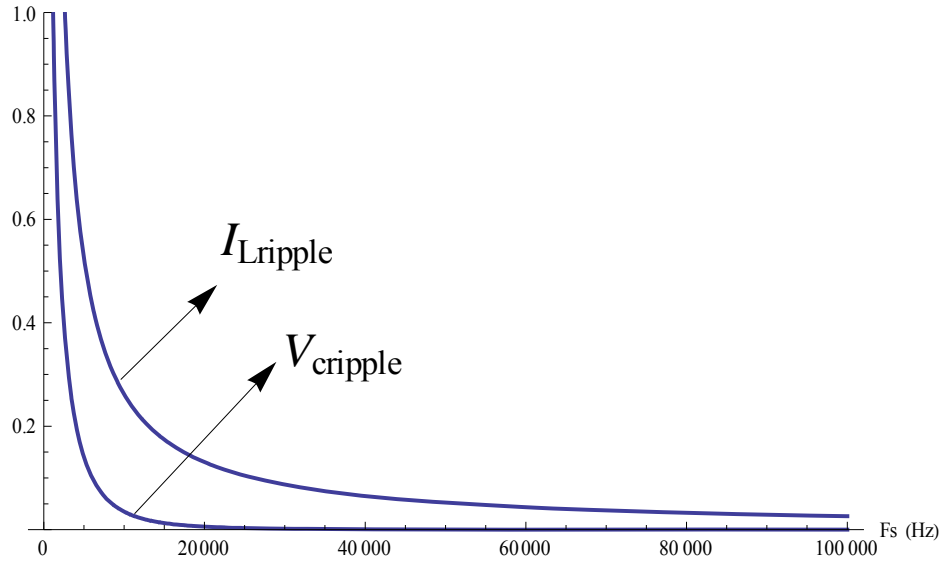
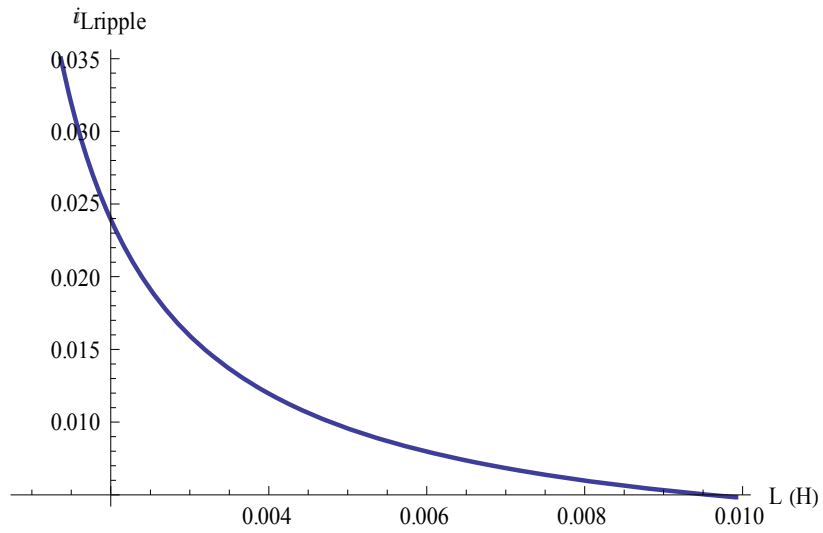


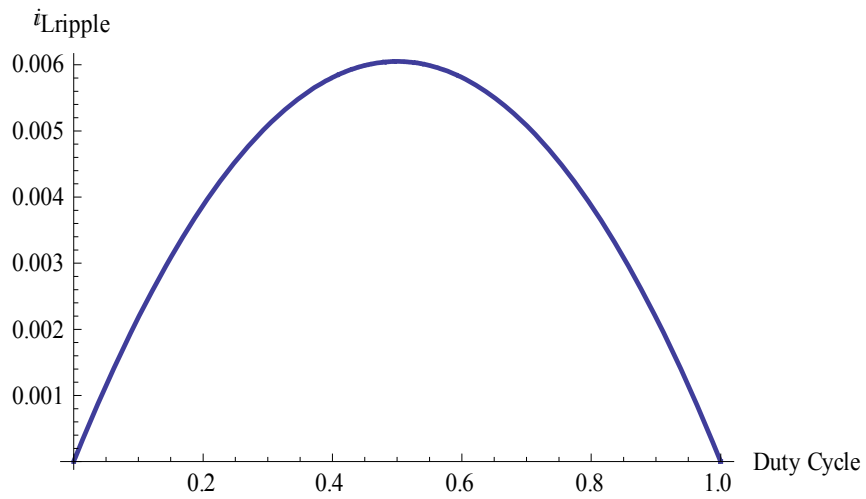
Figure 8-8 : Current and voltage ripple vs Frequency

Regarding the dynamic characteristics of the converter, the transfer function at any operating point can be quickly obtained and the frequency response plotted. For this particular converter the transfer function turns out to be,

$$H(z) = \frac{0.01523728148701839 + 0.0674437479203295z}{0.36790321473096216 - 1.3610133764710792z + z^2} \quad (8.40)$$



(a)



(b)

Figure 8-9 : Inductor current ripple versus (a) inductance , and (b) Duty Cycle ratio

So the Magnitude and phase frequency response expressions are found to be,

$$\begin{aligned}
 & \left| H \left(e^{\frac{j2\pi f}{F_s}} \right) \right|_{\text{db}} \\
 &= \frac{20 \text{Log} \left(\frac{\sqrt{(0.01524 + 0.06744 \cos[\frac{f\pi}{25000}])^2 + 0.004549 \sin[\frac{f\pi}{25000}]^2}}{\sqrt{(0.3679 - 1.361 \cos[\frac{f\pi}{25000}] + \cos[\frac{f\pi}{12500}])^2 + (-1.361 \sin[\frac{f\pi}{25000}] + \sin[\frac{f\pi}{12500}])^2}} \right)}{\text{Log}[10]}
 \end{aligned} \tag{9.41}$$

$$\text{Phase} \left(H \left(e^{\frac{j2\pi f}{F_s}} \right) \right) = \frac{180 \text{Arg} \left(\frac{0.01524 + 0.06744 e^{\frac{jf\pi}{25000}}}{0.3679 - 1.361 e^{\frac{jf\pi}{25000}} + e^{\frac{jf\pi}{12500}}} \right)}{\pi} \tag{8.42}$$

Which can be plotted as a bode plot. Then frequency control techniques can be used to design the converter feedback control insuring overall stability. The frequency response for this particular design is shown in figure 8-10. Frequency response methods of feedback control are very popular but not the only possibility, for other techniques like pole placement can be used.

Another aspect of the dynamic characteristics that can be easily visualized within this setting is the effect of parameter variance on the frequency response, as exposed in figures 8-11 to 8-13.

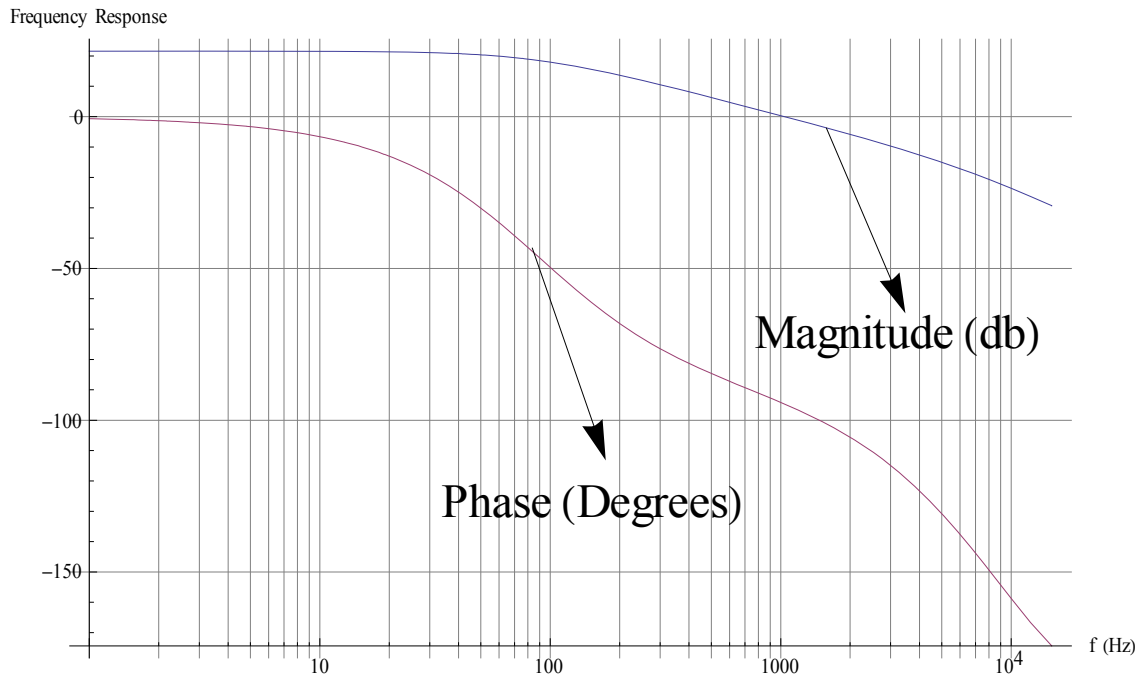


Figure 8-10 : Bode Plot of frequency response

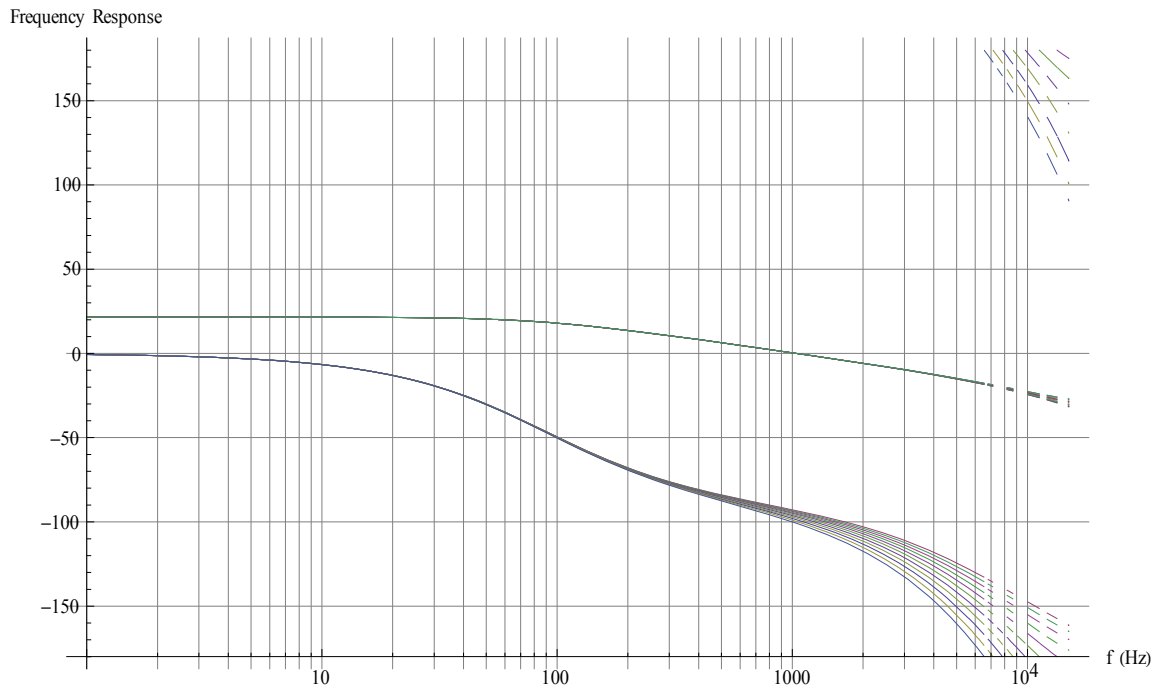


Figure 8-11 : Frequency response versus variance in operating point (Duty Cycle)

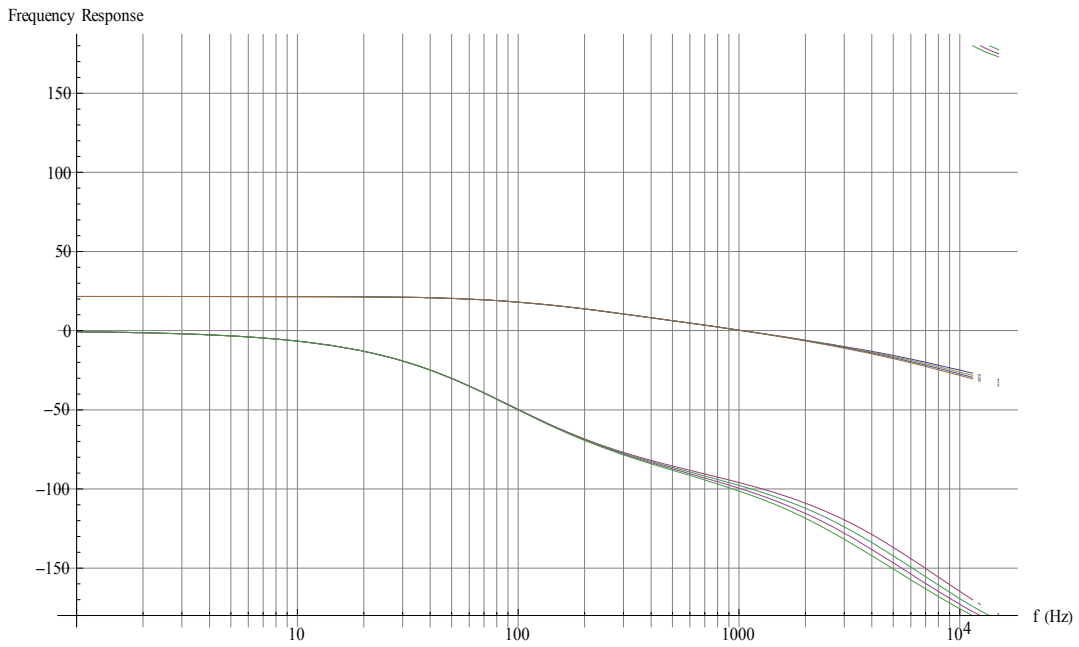


Figure 8-12 : Frequency response versus output filter capacitance

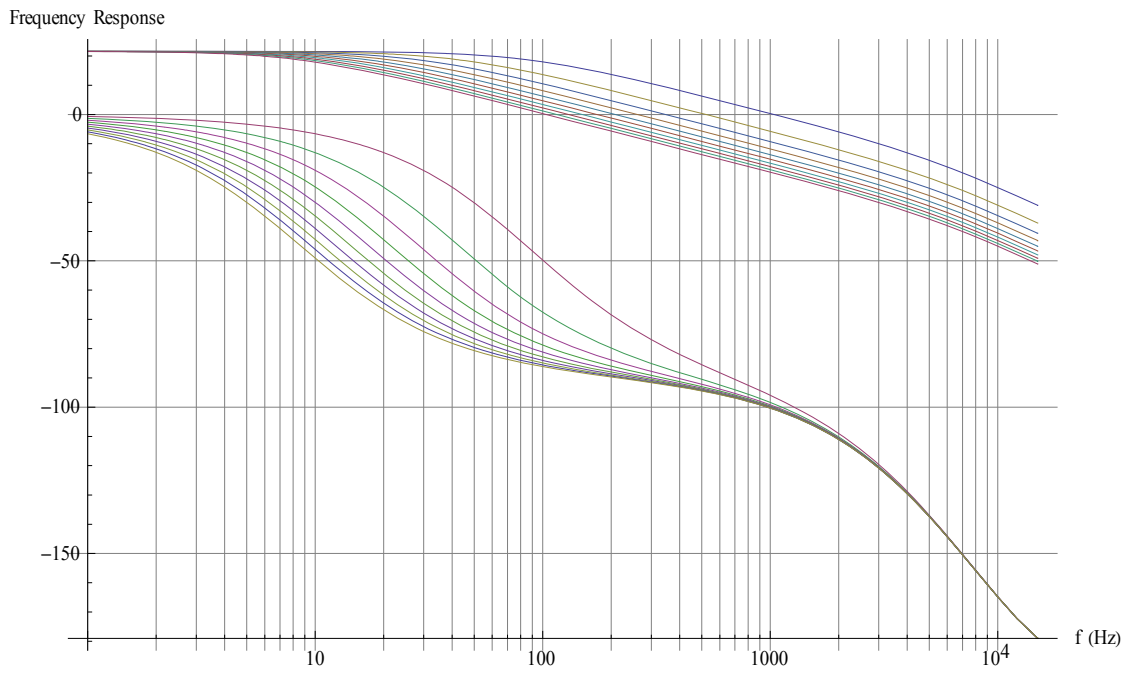


Figure 8-13 : Frequency response versus output filter Inductance

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

In this Dissertation, after a review of the main average modeling techniques for switching DC/DC converters a general small-signal discrete state-space model was developed. The main approximation in deriving this model was representing the duty cycle modulations that control the SMPS switches as weighted impulse functions. To make the model more accurate steady state waveform acquisition was utilized using Fourier analysis.

Then a large signal model was derived which turned out to be a linear function with respect to the converter inputs but nonlinear with respect to the duty cycle modulations as expected. Building upon that model an alternate small signal model was derived. Throughout the derivations certain symmetric properties in the form of the state-space equations were helpful in introducing definitions that simplified the notation to a large extent.

Both the Large and the small signal model were implemented for the case example of the Half-Bridge DC/DC with current doubler topology. All 3 main control schemes of the Half-Bridge (Symmetric, Asymmetric, DCS) were considered and their dynamic properties such as the frequency response and time response were compared. Finally a lab prototype was built with a compensator and its frequency response was compared with the frequency response obtained from the developed small signal models.

Chapter 6 of this dissertation reviewed an implementation of the derived models in MATLAB as a Design Platform GUI. After the description of the different components of this platform a design example was presented.

Chapter 7 embarked on detailing a Java based symbolic simulation tool that utilized a number of novel techniques to improve symbolic simulation time. While Chapter 8 dealt with a general state-space symbolic simulation approach tailored for DC-DC PWM converters.

9.2 Future Work

Future work will include more verification of the models derived here for both open loop and close loop. The main two expansions on the models will be to generalize them more so that they work for a converter with an arbitrary number of switches, the symmetry inherent in the model equations will enable the induction of this most general model. Then most importantly an extension to the case of DCM will be attempted.

Other ideas in Discrete modeling will be investigated, such as the use of input decimation so that it is possible to sample the input at a much higher sampling rate than the switching frequency. The reason this could be interesting is that continuous models in general account for both input and duty cycle as continuous, while discrete models account for both input and duty cycle as discrete, while in reality the input is continuous and the duty cycle is discrete. This idea if successfully implemented can mitigate any inaccuracies that abound when attempting to get an input to output transfer function from a discrete model.

All of this future work will be published in the open literature.

REFERENCES

1. *A general unified approach to modeling switching-converter power stages.* **Cuk, S. and Middlebrook, R. D.** 1976. IEEE PESC Proc. pp. 18–34.
2. *Simplified Analysis of PWM Converters using Model of PWM Switch Part I and Part II.* **Vorperian, Vatche.** 3, s.l. : IEEE Transactions on Aerospace and Electronic Systems, May 1990, Vol. 26.
3. *A New Continuous-Time Model for Current-Mode Control.* **Ridley, Raymond B.** 2, s.l. : IEEE Transactions on Power Electronics, April 1991, Vol. 6.
4. *Study of System instability in Solar Array Based Power Systems.* **Siri, Kasemsan.** 3, s.l. : IEEE Transactions on Aerospace and Electronic Systems, July 2000, Vol. 36.
5. *Sampled-Data Modeling of Switching regulators.* **Brown, Arthur R. and Middlebrook, R. D.** s.l. : IEEE Power Electronics Specialists Conference, 1981. pp. 349-369.
6. *Unified Analog and Digital Models for Half Bridge Converter with Current Doubler Rectifier.* **Yao, Liangbin, Abu-Qahouq, Jaber and Batarseh, Issa.** s.l. : IEEE Applied Power Electronics Conference and Exposition, APEC'2005, 2005.
7. **Packard, Dennis J.** *Discrete Modeling and Analysis of Switching Regulators.* s.l. : California institute of technology, 1976. PhD thesis.
8. **Chen, Wai-Kai.** *Active Network and Feedback Amplifier Theory.* s.l. : Hemisphere Pub, 1980.

9. *Zero-Voltage-Switching Half-Bridge DC–DC Converter With Modified PWM Control Method.* **Mao, Hong, et al.** 4, s.l. : IEEE Transactions on Power Electronics, July 2004, Vol. 19.
10. *Discrete model of DCS controlled half-bridge with current doubler for digital controller design.* **Shoubaki, Ehab, Abu-Qahouq, Jaber and Batarseh, Issa.** s.l. : Applied Power Electronics Conference and Exposition, APEC 2005, 2005.
11. **Eckel, Bruce.** *Thinking in Java.* s.l. : Prentice Hall, 2006.
12. *Symbolic Network Analysis Methods for Practical Analog Integrated Circuits: A Survey.* **Wambacq, Piet, Gielen, Georges G. E. and Sansen, Willy.** 10, s.l. : IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, October 1998, Vol. 45.
13. *An early introduction to circuit simulation techniques.* **Pillage, L.T.** 1, s.l. : Education, IEEE Transactions, Feb 1993, Vol. 36, pp. 16-19.