THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ELECTRICAL ENGINEERING

ANALYSIS OF POWER ELECTRONIC CONVERTER TRANSFER FUNCTIONS USING A POINCARÉ MAP APPROACH

JOHNATHAN PAUL ROSS
Spring 2010

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Electrical Engineering
with Honors in Electrical Engineering

Reviewed and approved* by the following:

Jeffrey S. Mayer
Associate Professor
Thesis Supervisor

John D. Mitchell
Professor
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

Modern power converter analysis and design uses transfer functions based upon weighted state space representations of the circuit. The steady state value of the duty cycle must be known in order to determine the weighted steady state representation. Under complex control modes the duty cycle is dependent upon on a state variable and a reference signal, leading to particular solutions for each control mode. In this thesis a general approach to solving for the power converter transfer function is found using a Poincaré Map of the periodic, piecewise-linear-time-invariant state space model of the converter system. The Jacobians of the Poincaré Map are used to determine a discrete-time transfer function of the converter system including the control mode. The power converter transfer function is then found after applying the Bilinear Transform and deconstructing the closed loop transfer function model. The results of this method show that it accurately describes the response of the power converter to perturbations in the input for frequencies up to one order of magnitude below the switching frequency.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Stability analysis of power converters has been the focus of much research for over forty years owing to the practical importance of stability and the mathematical challenged posed by the inherent switching nonlinearity of power converter networks. The predominant method of stability analysis involves linearization of the converter model through State Space Averaging [1,2]. In this technique, the state and output equations associated with the successive network topologies are averaged using the duty cycle for each topology as a weighting factor. The resulting continuous-time state space model can be used to express s-domain transfer functions that relate perturbations in converter outputs, typically voltage, to perturbations in converter inputs, typically duty cycle or voltage. Each of these transfer functions is then incorporated into an *s*-domain model of the closed-loop system to study stability and performance. Generally, the converter transfer functions derived through State Space Averaging are accurate for frequencies up to one order of magnitude below the switching frequency. This places an upper limit on the bandwidth that can be achieved for closed-loop system.

An alternative method for modeling power converter dynamics is described in this thesis. In this method, the state space models for the various network topologies are not averaged but instead solved successively to establish a Poincaré map or nonlinear, discrete-time state-space model for the initial condition vector of the periodic, piecewise-linear-time-invariant (PPLTI) state-space model of the converter system. The Jacobian of the Poincare map – along with a matrix of partial derivatives with respect to the input variables – represents a linear discrete-time model for perturbations to the periodic orbits of the PPLTI system model. Thus, this model can be used to assess the periodic stability of the converter system, or it can be transformed to yield a continuous-time (s-domain) transfer function for the overall system.

Closed form expressions for the Jacobian and the matrix of partial derivatives with respect to input variables were recently derived [3]. As part of this thesis project, MATLAB code to evaluate these matrices has been implemented. Additional MATLAB code has been implemented to transform the discrete-time model to a continuous-time model and to extract the transfer function of the converter from the transfer function for the overall system.

The thesis is organized as follows. Chapter 2 provides the information pertaining to the State Space Averaging approach and the Poincaré Map approach to solving for the PPLTI system model. Chapter 3 comprises the computation of a small signal transfer function from the PPLTI state space model from the Poincaré Map. Chapter 4 details an example of a voltage mode controlled buck converter and compares the bode plots of the transfer functions. Chapter 5 explores the conclusions found in this thesis and states future opportunities for research.….

.

Chapter 2

Background

In this chapter, essential information regarding State Space Averaging and the Poincaré Map approach is provided. The first section describes the State Space Averaging approach. The second section describes the Poincaré map approach that was developed by Stanislav Kriventsov as part of his Ph.D. from The Pennsylvania State University [3]. A portion of his work is reproduced here in order to clarify notation and correct minor errors. His method is useful because it provides an explicit expression for the Jacobian of a periodic piecewise linear time invariant system such as those that arise in power electronics. The third section provides a transformation method that transforms a discrete-time transfer function to a continuous-time transfer function.

State Space Averaging

Power electronic converters are nonlinear devices owing to the operation of the switches that reconfigure the network into various topologies throughout converter operation. This nonlinearity makes modeling power converters a challenge, and various methods have been developed to derive a linear model of them. The State Space Averaging method provides was one of the first approaches to obtaining a linear model, and it remains one of the most general. State Space Averaging will serve as a benchmark for the Poincare map approach derived in the thesis. Consequently, it is useful to illustrate this method for the special but important class of power converters having two topologies with switching between the topologies determined through duty cycle control based on feedback of the converter output voltage. In this case, the goal is to establish a small signal transfer function that relates perturbations in the output voltage, v_0, to perturbations in the duty cycle, d. The first step is to obtain the state space representations of each circuit topology:

$$\dot{\mathbf{x}} = \mathbf{A}_1\mathbf{x} + \mathbf{B}_1 v_d \qquad 0 \le d\,t \le d\,T_s \qquad (1)$$
$$v_0 = \mathbf{C}_1\mathbf{x}$$

$$\dot{\mathbf{x}} = \mathbf{A}_2\mathbf{x} + \mathbf{B}_2 v_d \qquad d\,T_s \le d\,t \le T_s \qquad (2)$$
$$v_0 = \mathbf{C}_2\mathbf{x}$$

The next step is to average the state space representations using the duty cycle, d, and its compliant $(1-d)$ as weighting factors.

$$\dot{\mathbf{x}} = [\mathbf{A}_1 d + \mathbf{A}_2(1-d)]\mathbf{x} + [\mathbf{B}_1 d + \mathbf{B}_2(1-d)]v_d \qquad (3)$$
$$v_0 = [\mathbf{C}_1 d + \mathbf{C}_2(1-d)]\mathbf{x}$$

The next step is to introduce small perturbations in $\mathbf{x}$, v_0, and d:

$$\mathbf{x} = \mathbf{X} + \tilde{\mathbf{x}} \qquad (4)$$
$$v_0 = V_0 + \tilde{v}_0$$
$$d = D + \tilde{d}$$

Where $\mathbf{X}$, V_0, and D are the dc components of the state, output, and duty cycle respectively, and $\tilde{\mathbf{x}}$, $\tilde{v}_0$, and $\tilde{d}$ are small-signal perturbations of the state, output, and duty cycle respectively. For simplicity the perturbations in the input voltage, $\tilde{v}_d$, is assumed to be zero. After some algebra and removing of the dc components a new state space representation of the small signal perturbations can be obtained:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + [(\mathbf{A}_1 - \mathbf{A}_2)\mathbf{X} + (\mathbf{B}_1 - \mathbf{B}_2)V_d]\tilde{d} \qquad (5)$$
$$\tilde{v}_0 = \mathbf{C}\tilde{\mathbf{x}} + [(\mathbf{C}_1 - \mathbf{C}_2)\mathbf{X}]\tilde{d}$$

where

$$\mathbf{A} = [\mathbf{A}_1 d + \mathbf{A}_2(1-d)] \qquad (6)$$
$$\mathbf{C} = [\mathbf{C}_1 d + \mathbf{C}_2(1-d)]$$

The last step is to determine the small signal transfer function by applying the Laplace transform and solving for $\tilde{v}_0(s)/\tilde{d}(s)$:

$$\frac{\tilde{v}_0(s)}{\tilde{d}(s)} = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}[(\mathbf{A}_1-\mathbf{A}_2)\mathbf{X}+(\mathbf{B}_1-\mathbf{B}_2)V_d]+[(\mathbf{C}_1-\mathbf{C}_2)\mathbf{X}] \quad (7)$$

Where $\mathbf{I}$ is an identity matrix of the same size as $\mathbf{A}$.

Generally, the small signal transfer function is accurate for frequencies up to an order of magnitude below the switching frequency. [18]

Poincare Maps

Let us consider the following general piecewise-smooth system with constant inputs that we will use as a mathematical model for switched power converters:

$$\begin{aligned} &\frac{d}{dt}\mathbf{x} = \mathbf{F}(\mathbf{x},\mathbf{u},t) \\ &\mathbf{F}(\mathbf{x},\mathbf{u},t) = \mathbf{F}_j(\mathbf{x},\mathbf{u}) \text{ when } d_{j-1}^k T_s \le t - kT_s \le d_j^k T_s \quad j=1,2,...,J \quad 0 \le d_j^k \le 1 \end{aligned} \quad (8)$$

Here $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{F}$ are column vectors, and each $\mathbf{F}_j(\mathbf{x},\mathbf{u})$ is a smooth (continuously differentiable) function which will describe the behavior of the system during the j^{th} circuit topology of each switching cycle. The input vector $\mathbf{u}$ is assumed to be constant in all expressions in this thesis unless stated otherwise. The integer variable k is the switching cycle number, T_s is the switching period, and J is the number of circuit topologies that the system goes through during each cycle. The value of each d_j^k defines the instant measured from the beginning of the k^{th} switching cycle when the system is switched from the j^{th} to the $(j+1)^{st}$ topology of the cycle. The system is assumed to go through the same sequence of topologies during each cycle; that is, the functions $\mathbf{F}_j(\mathbf{x},\mathbf{u})$ are independent of the cycle number k. By definition we set $d_0^k = 0$ and $d_J^k = 1$ for all values of k, which means that the k^{th} switching cycle starts at $t = kT_s$ and ends at $t = (k+1)T_s$. All other instants of transition between topologies may be determined dynamically during each cycle due to the action of the controller and/or

presence of discontinuous conduction (or blocking) modes, which makes the system nonlinear and more difficult to analyze.

We will define $\mathbf{x}_j^k$ to be the state variable vector of the system at the j^{th} switching instant of the k^{th} cycle, that is,

$$\mathbf{x}_j^k = \mathbf{x}((k + d_j^k)T_s) \tag{9}$$

In this case the value of the state vector at the beginning of the k^{th} switching cycle will be $\mathbf{x}_0^k$. Using the sampled-data approach with periodic sampling at the beginning of each cycle, we must to find a relationship between the state vector values $\mathbf{x}_0^k$ and $\mathbf{x}_0^{k+1} = \mathbf{x}((k+1)T_s)$, assuming that the values of all switching instants d_j^k between them, which can be combined in a $J \times 1$ column vector $\mathbf{d}$, are known. Let us assume that we can calculate this relationship in the form:

$$\mathbf{x}_0^{k+1} = \mathbf{f}(\mathbf{x}_0^k, \mathbf{u}, \mathbf{d}(\mathbf{x}_0^k)) \tag{10}$$

Expression (10) provides the Poincaré map of the system. We are interested in periodic solutions of (8) with period T_s; that is, solutions that satisfy the equation $\mathbf{x}_0^{k+1} = \mathbf{x}_0^k$. The simplest case is open-loop, continuous conduction-mode operation, wherein all switching instants d_j^k are known and the periodic solutions can be found by simply locating the fixed points of the map (10). The majority of practical converters, however, are operated with feedback and/or may have topologies associated with discontinuous conduction or voltage blocking [14], which means that some or all of the d_j^k are determined dynamically during the cycle based on the present values of the states and the inputs. Let us assume that the conditions for each switching instant of the cycle can be expressed as:

$$\sigma_j(\mathbf{x}(t), \mathbf{u}, t) = 0 \quad j = 1, 2, ..., J-1 \tag{11}$$

In this case the switching instants d_j^k can be determined for known inputs and initial conditions by finding $\mathbf{x}(t)$ from (8) and then solving (11) for the value of the time variable t, repeating this process for each circuit topology. In order to find the periodic solution of (8), the equation $\mathbf{x}_0^{k+1} = \mathbf{x}_0^k$ must be solved together with (8), (10), and (11). This usually cannot be done analytically, so a numerical technique must be employed. The most common approach is by means of the Newton-Raphson algorithm, wherein one starts with an initial point $\mathbf{x}(0)$ selected in the vicinity of the unknown vector $\mathbf{X}_0$ defining the point $\mathbf{x}_0^k$ for the desired periodic solution, and calculates each following point based on the value of the previous one using the following expression for the iterates [11]:

$$\mathbf{x}_0^{k+1} = \mathbf{x}_0^k - \left(\mathbf{I} - \mathbf{H}_x(\mathbf{x}_0^k, \mathbf{u}, \mathbf{d}(\mathbf{x}_0^k))\right)^{-1}\left[\mathbf{x}_0^k - \mathbf{f}(\mathbf{x}_0^k, \mathbf{u}, \mathbf{d}(\mathbf{x}_0^k))\right] \tag{12}$$

Here $\mathbf{f}$ defines the Poincaré map of the system as in (10), $\mathbf{I}$ is the identity matrix, and $\mathbf{H}_x$ is the Jacobian matrix of the system. The Jacobian matrix describes, in the first order, the evolution over one switching cycle of any small perturbation $\tilde{\mathbf{x}}$ applied to the point $\mathbf{x}_0^k$ for which $\mathbf{H}_x$ is calculated, which can be written as

$$\mathbf{f}(\mathbf{x}_0^k + \tilde{\mathbf{x}}, \mathbf{u}, \mathbf{d}(\mathbf{x}_0^k) + \tilde{\mathbf{d}}) - \mathbf{f}(\mathbf{x}_0^k, \mathbf{u}, \mathbf{d}(\mathbf{x}_0^k)) = \mathbf{H}_x(\mathbf{x}_0^k) \cdot \tilde{\mathbf{x}} \tag{13}$$

Knowledge of both the Poincaré map $\mathbf{f}$ and the Jacobian matrix $\mathbf{H}_x$ is necessary for application of the periodic solutions search algorithm described by (12). Once the periodic solution vector $\mathbf{X}_0$ is found, $\mathbf{H}_x$ is also useful for analyzing the stability of the corresponding orbit. In particular, if the eigenvalues of $\mathbf{H}_x(\mathbf{X}_0)$ lie inside the unit circle in the complex plane, then any small perturbation will decrease from one sampling point to the next, which means that the periodic orbit is stable. Conversely, if one or more of the eigenvalues is outside of the unit circle, there will always exist some perturbation that

will increase with time, so the solution defined by $\mathbf{X}_0$ is unstable. Thus, by examining the eigenvalues of $\mathbf{H}_x$, one can assess the stability of the corresponding periodic solution [11,12].

Let us derive a general closed-form expression for the Jacobian matrix $\mathbf{H}_x$ calculated at an arbitrary point $\mathbf{X}_0$ for a system with known input vector $\mathbf{U}$, Poincaré map $\mathbf{f}$, and switching condition vector $\boldsymbol{\sigma}$. Introducing small perturbations as

$$\begin{aligned} &\mathbf{x}_0^k = \mathbf{X}_0 + \tilde{\mathbf{x}}_0^k \\ &d_j(\mathbf{x}_0^k) = d_j(\mathbf{X}_0) + \tilde{d}_j^k \\ &\mathbf{u} = \mathbf{U} + \tilde{\mathbf{u}} \end{aligned} \tag{14}$$

we can find the Jacobian matrix $\mathbf{H}_x(\mathbf{X}_0, \mathbf{U}, \mathbf{d}(\mathbf{X}_0))$ by obtaining the first order expansion:

$$\tilde{\mathbf{x}}_0^{k+1} = \mathbf{H}_x \tilde{\mathbf{x}}_0^k + \mathbf{H}_u \tilde{\mathbf{u}} \tag{15}$$

where $\tilde{\mathbf{x}}_0^{k+1} = \mathbf{f}(\mathbf{x}_0^k, \mathbf{u}, \mathbf{d}(\mathbf{x}_0^k)) - \mathbf{f}(\mathbf{X}_0, \mathbf{U}, \mathbf{d}(\mathbf{X}_0))$ is the value of the state vector perturbation after one switching cycle.

Writing the effect of the perturbations on the Poincaré map expression (10) and the switching conditions (11), we obtain the following system:

$$\begin{cases} \dfrac{\partial \mathbf{f}}{\partial \mathbf{x}_0}\tilde{\mathbf{x}}_0^k + \dfrac{\partial \mathbf{f}}{\partial \mathbf{u}}\tilde{\mathbf{u}} + \dfrac{\partial \mathbf{f}}{\partial d_1}\tilde{d}_1^k + \cdots + \dfrac{\partial \mathbf{f}}{\partial d_J}\tilde{d}_J^k = \tilde{\mathbf{x}}_0^{k+1} \\ \dfrac{\partial \sigma_1}{\partial \mathbf{x}_0}\tilde{\mathbf{x}}_0^k + \dfrac{\partial \sigma_1}{\partial \mathbf{u}}\tilde{\mathbf{u}} + \dfrac{\partial \sigma_1}{\partial d_1}\tilde{d}_1^k = 0 \\ \dfrac{\partial \sigma_2}{\partial \mathbf{x}_0}\tilde{\mathbf{x}}_0^k + \dfrac{\partial \sigma_2}{\partial \mathbf{u}}\tilde{\mathbf{u}} + \dfrac{\partial \sigma_2}{\partial d_1}\tilde{d}_1^k + \dfrac{\partial \sigma_2}{\partial d_2}\tilde{d}_2^k = 0 \\ \cdots \\ \dfrac{\partial \sigma_{J-1}}{\partial \mathbf{x}_0}\tilde{\mathbf{x}}_0^k + \dfrac{\partial \sigma_{J-1}}{\partial \mathbf{u}}\tilde{\mathbf{u}} + \dfrac{\partial \sigma_{J-1}}{\partial d_1}\tilde{d}_1^k + \dfrac{\partial \sigma_{J-1}}{\partial d_2}\tilde{d}_2^k + \cdots + \dfrac{\partial \sigma_{J-1}}{\partial d_{J-1}}\tilde{d}_{J-1}^k = 0 \end{cases} \tag{16}$$

Here the derivatives must be calculated at the non-perturbed values of the switching instants. In order to represent $\mathbf{x}_0^{k+1}$ in the form (15) and thus find $\mathbf{H}_x$, all the terms $\tilde{d}_j^k$, $j=1,2,...,J$ in the first equation of (16) must be found from the rest of the equations as functions of $\tilde{\mathbf{x}}_0^k$ and $\tilde{\mathbf{u}}$. Noticing that the second equation of (16) contains only $\tilde{d}_1^k$, we can express the latter from this equation. Then, taking the next equation of (16) and substituting $\tilde{d}_1^k$ into it, we can express $\tilde{d}_2^k$ and continue the process until the last equation. This yields:

$$\begin{cases} \tilde{d}_1^k = -\left(\dfrac{\partial\sigma_1}{\partial d_1}\right)^{-1}\dfrac{\partial\sigma_1}{\partial\mathbf{x}_0}\tilde{\mathbf{x}}_0^k - \left(\dfrac{\partial\sigma_1}{\partial d_1}\right)^{-1}\dfrac{\partial\sigma_1}{\partial\mathbf{u}}\tilde{\mathbf{u}} \\ \tilde{d}_2^k = \left[-\left(\dfrac{\partial\sigma_2}{\partial d_2}\right)^{-1}\dfrac{\partial\sigma_2}{\partial\mathbf{x}_0} + \left(\dfrac{\partial\sigma_2}{\partial d_2}\right)^{-1}\left(\dfrac{\partial\sigma_1}{\partial d_1}\right)^{-1}\dfrac{\partial\sigma_2}{\partial d_1}\dfrac{\partial\sigma_1}{\partial\mathbf{x}_0}\right]\tilde{\mathbf{x}}_0^k ... \\ \qquad + \left[-\left(\dfrac{\partial\sigma_2}{\partial d_2}\right)^{-1}\dfrac{\partial\sigma_2}{\partial\mathbf{u}} + \left(\dfrac{\partial\sigma_2}{\partial d_2}\right)^{-1}\left(\dfrac{\partial\sigma_1}{\partial d_1}\right)^{-1}\dfrac{\partial\sigma_2}{\partial d_1}\dfrac{\partial\sigma_1}{\partial\mathbf{u}}\right]\tilde{\mathbf{u}} \\ ... \end{cases} \tag{17}$$

Analyzing the structure of (17) for various $\tilde{d}_j^k$, the following expression that is valid for all these terms can be obtained:

$$\begin{aligned} \tilde{d}_j^k &= \mathbf{g}_{xj}\tilde{\mathbf{x}}_0^k + \mathbf{g}_{uj}\tilde{\mathbf{u}} \\ \mathbf{g}_{xj} &= \sum_{m=0}^{j-1}(-1)^{m+1}\sum_{\substack{i_0>i_1>\cdots>i_m \\ i_0=j \\ i_n\in 1,2,...,j}}\frac{\prod_{n=0}^{m-1}\dfrac{\partial\sigma_{di_n}}{\partial d_{i_{n+1}}}}{\prod_{n=0}^{m}\dfrac{\partial\sigma_{di_n}}{\partial d_{i_n}}}\frac{\partial\sigma_{i_m}}{\partial\mathbf{x}_0} \\ \mathbf{g}_{uj} &= \sum_{m=0}^{j-1}(-1)^{m+1}\sum_{\substack{i_0>i_1>\cdots>i_m \\ i_0=j \\ i_n\in 1,2,...,j}}\frac{\prod_{n=0}^{m-1}\dfrac{\partial\sigma_{di_n}}{\partial d_{i_{n+1}}}}{\prod_{n=0}^{m}\dfrac{\partial\sigma_{di_n}}{\partial d_{i_n}}}\frac{\partial\sigma_{i_m}}{\partial\mathbf{u}} \end{aligned} \tag{18}$$

Here the inner summations are taken over all the possible combinations of j different integer numbers $i_n \in \{1, 2, \ldots, j\}$ $n = 1, 2, \ldots, j$ such that, with $i_0 = j$, $i_0 > i_1 > \cdots > i_j > 0$. When $j = 0$, these summations include only a single term with $i_0 = J$. The validity of (18) can be verified by means of mathematical induction.

Substituting the expressions for $\tilde{d}_j^k$ into the first equation of (16), we get an expression of the form (15), where

$$\begin{aligned} \mathbf{H}_x &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \sum_{j=1}^{J-1} \frac{\partial \mathbf{f}}{\partial d_j} \mathbf{g}_{xj} \\ \mathbf{H}_u &= \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \sum_{j=1}^{J-1} \frac{\partial \mathbf{f}}{\partial d_j} \mathbf{g}_{uj} \end{aligned} \tag{19}$$

Thus, the first part of (19) gives us an expression for the Jacobian matrix $\mathbf{H}_x$. By substituting the expressions for $\mathbf{g}_{xm}$ and $\mathbf{g}_{um}$, and rearranging the terms in the summations, the following result can be obtained:

$$\begin{aligned} \mathbf{H}_x &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}_0} + \sum_{m=0}^{J-1} (-1)^{m+1} \sum_{\substack{i_0 > i_1 > \cdots > i_m \\ i_0 = J \\ i_n \in 1, 2, \ldots, J}} \frac{\prod_{n=0}^{m-1} \frac{\partial \sigma_{di_n}}{\partial d_{i_{n+1}}}}{\prod_{n=0}^{m} \frac{\partial \sigma_{di_n}}{\partial d_{i_n}}} \frac{\partial \mathbf{f}}{\partial d_{i_1}} \frac{\partial \sigma_{i_m}}{\partial \mathbf{x}_0} \\ \mathbf{H}_u &= \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \sum_{m=0}^{J-1} (-1)^{m+1} \sum_{\substack{i_0 > i_1 > \cdots > i_m \\ i_0 = J \\ i_n \in 1, 2, \ldots, J}} \frac{\prod_{n=0}^{m-1} \frac{\partial \sigma_{di_n}}{\partial d_{i_{n+1}}}}{\prod_{n=0}^{m} \frac{\partial \sigma_{di_n}}{\partial d_{i_n}}} \frac{\partial \mathbf{f}}{\partial d_{i_1}} \frac{\partial \sigma_{i_m}}{\partial \mathbf{u}} \end{aligned} \tag{20}$$

The first part of (20) represents the desired expression for the Jacobian matrix that can be used for finding periodic orbits and analyzing their stability. To use this expression, however, the partial derivatives within it must be expressed, and that requires knowledge of the particular form of the Poincaré map $\mathbf{f}$ and the switching surface $\boldsymbol{\sigma}$.

Piecewise-Linear Switched Systems

While (20) is valid for a very wide class of systems described by (8) and (11), even those with realistic (nonlinear) device models, in order to make further progress we will now make several assumptions. Let us assume that each topology of the circuit can be represented by a linear state-space model, which means that in (8) we have

$$\mathbf{F}_j(\mathbf{x},\mathbf{u}) = \mathbf{A}_j\mathbf{x} + \mathbf{B}_j\mathbf{u} \tag{21}$$

In this case it can be shown that the expression for the Poincaré map for known values of the switching instants d_j^k has the form: [3]

$$\mathbf{x}_0^{k+1} = \mathbf{x}_J^k = \mathbf{f}(\mathbf{x}_0^k, \mathbf{u}, d(\mathbf{x}_0^k)) = \mathbf{\Pi}_1^J \mathbf{x}_0^k + \sum_{j=1}^{J}\left(\mathbf{\Pi}_{j+1}^J \mathbf{\Psi}_j^k\right)\mathbf{u} \tag{22}$$

where

$$\begin{aligned}
&\mathbf{\Pi}_p^q = \prod_{j=p}^{q} \mathbf{\Phi}_j^k = \mathbf{\Phi}_q^k \mathbf{\Phi}_{q\text{-}1}^k \ldots \mathbf{\Phi}_p^k \\
&\mathbf{\Phi}_j^k = e^{\mathbf{A}_j \Delta_j^k} \\
&\Delta_j^k = d_j^k - d_{j-1}^k \\
&\mathbf{\Psi}\left(\mathbf{A},\mathbf{B},t\right) = \int_0^t e^{\mathbf{A}\tau}\mathbf{B}\,d\tau = \left(\mathbf{I}t + \frac{\mathbf{A}t^2}{2!} + \frac{\mathbf{A}^2t^3}{3!} + \frac{\mathbf{A}^3t^4}{4!} + \ldots\right)\mathbf{B} \\
&\mathbf{\Psi}_j^k = \mathbf{\Psi}\left(\mathbf{A}_j, \mathbf{B}_j, \Delta_j^k\right)
\end{aligned} \tag{23}$$

Here $\mathbf{I}$ is the identity matrix of the same size as the square matrix $\mathbf{A}$. In (23) and all the other expressions in this thesis that involve summation (Σ) or multiplication (Π), if the upper index of a sum has a smaller value than the lower index, that is, if the sum contains no terms, then the value of such empty sum is assumed to be zero. If a product contains no terms, its value is assumed to be unity (the identity matrix).

If the matrix $\mathbf{A}_j$ is invertible, then the expression for $\mathbf{\Psi}_j^k$ can be simplified to obtain:

$$\mathbf{\Psi}_j^k = \mathbf{A}_j^{-1}\left(e^{\mathbf{A}_j\Delta_j^k} - \mathbf{I}\right)\mathbf{B}_j \tag{24}$$

However, for many important cases, including, for example, the basic boost and buck-boost converter topologies, as well as discontinuous mode topologies, $\mathbf{A}_j$ is a singular matrix and the expression (24) is not valid, so the more complicated formula in (23) must be used. The problem of quickly computing the Taylor series for a matrix function is in general quite complex [15]. Fortunately, in practical power systems the switching frequency $f_s = 1/T_s$ is typically large enough that $\left\|\mathbf{A}_j T_s\right\| << 1$ (or at least $\left\|\mathbf{A}_j T_s\right\| \le 1$). In this case, the series in (12) converges quickly and can be computed by summation of terms until the desired tolerance is achieved. Another, more efficient method employs Padé approximation [15].

In the open-loop case (i.e., when the switching instants are known a priori), fixed points of (22) define possible periodic orbits of the system. If $\left|\mathbf{I} - \mathbf{\Pi}_1^J\right| \neq 0$, which is often the case, there is always exactly one such orbit for the given values of switching instants, which is

$$\mathbf{x}_0 = \left(\mathbf{I} - \mathbf{\Pi}_1^J\right)^{-1} \cdot \sum_{j=1}^{J}\left(\mathbf{\Pi}_{j+1}^J \mathbf{\Psi}_j^k\right) \tag{25}$$

In the closed-loop case (22) must be solved together with the switching condition equations of the type (11). Let us consider the following form of switching conditions:

$$\sigma_j\left(\mathbf{x}_0^k, \mathbf{u}, d_1^k, d_1^k, ..., d_j^k\right) = \mathbf{\sigma}_{xj}\mathbf{x}_j^k + \mathbf{\sigma}_{uj}\mathbf{u} + \sigma_{dj}d_j^k + \sigma_{cj} = 0 \tag{26}$$

Here $j = 1, 2, ..., J-1$; $\mathbf{\sigma}_{xj}$ and $\mathbf{\sigma}_{uj}$ are constant row vectors that define which state variables and inputs are to be used to determine the j^{th} switching instant, while σ_{dj} and σ_{cj} are scalar constants for each *j* that are employed to adapt (26) for different cases of switching. The condition of the form (26) is quite general, as it can be used to describe both switching due to PWM controllers and transitions into

discontinuous modes, as well as open-loop switching and other cases. Examples of use of (26) for practical systems are given in Appendix II.

From the Poincaré Map (22) and the switching conditions (26), we can now obtain the expressions for the derivatives in the Jacobian matrix expression (20). The derivation yields:

$$
\begin{aligned}
&\frac{\partial \mathbf{f}}{\partial \mathbf{x}_0} = \mathbf{\Pi}_1^J \\
&\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \sum_{j=1}^{J} \mathbf{\Pi}_{j+1}^J \mathbf{\Psi}(\mathbf{A}_j, \mathbf{B}_j, \Delta_j^k) \\
&\frac{\partial \mathbf{f}}{\partial d_j} = \mathbf{\Pi}_{j+1}^J \left(\dot{\mathbf{X}}(d_j^-) - \dot{\mathbf{X}}(d_j^+) \right) \\
&\frac{\partial \sigma_i}{\partial \mathbf{x}_0} = \sigma_{xi} \mathbf{\Pi}_1^i \\
&\frac{\partial \sigma_i}{\partial \mathbf{u}} = \sigma_{xi} \left[\sum_{j=1}^{i} \mathbf{\Pi}_{j+1}^i \mathbf{\Psi}(\mathbf{A}_j, \mathbf{B}_j, \Delta_j^k) \right] + \sigma_{ui} \\
&\frac{\partial \sigma_i}{\partial d_j} = \begin{cases} 0 & i < j \\ \sigma_{xi} \dot{\mathbf{X}}(d_j^-) + \sigma_{di} & i = j \\ \sigma_{xi} \mathbf{\Pi}_{j+1}^i \left(\dot{\mathbf{X}}(d_j^-) - \dot{\mathbf{X}}(d_j^+) \right) & i > j \end{cases}
\end{aligned}
\tag{27}
$$

where

$$
\begin{aligned}
\dot{\mathbf{X}}(d_j^-) &= \mathbf{A}_j \mathbf{X}_j + \mathbf{B}_j \mathbf{U} \\
\dot{\mathbf{X}}(d_j^+) &= \mathbf{A}_{j+1} \mathbf{X}_j + \mathbf{B}_{j+1} \mathbf{U}
\end{aligned}
\tag{28}
$$

are the left and right time derivatives of the state variable vector at d_j, which follows from (8) and (21).

The resulting expressions can be easily programmed for automatic calculation of the Jacobian matrix for any system of the form (8), (21) with switching conditions of the type (23) and a given operating point $\mathbf{X}_0$. The Jacobian matrix can be used along with the Poincaré Map to search for periodic solutions or for stability assessment of a known periodic orbit.

The Bilinear Transform

The Poincare map method that is employed in this paper produces a discrete time model of the compensator, buck converter, and feedback loop. This leads to a discrete-time transfer function in z. The State Space Averaged small signal transfer function that serves as a benchmark is a continuous time transfer function in s. In order to perform a comparison of the two transfer functions the transfer function derived from the Poincare map must be converted into the continuous domain. The Bilinear Transform is one method which performs a transformation from the discrete-time domain, z, to the continuous-time domain, s.

A discrete-time transfer function has boundary limits at the unit circle. The challenge lies in mapping the area enclosed in the unit circle to the imaginary axis in the s domain. The following equation describes the transformation from the z domain to the s domain:

$$z = e^{T_s s} = \frac{e^{T_s s/2}}{e^{-T_s s/2}} \approx \frac{1+(T_s/2)s}{1-(T_s/2)s} \tag{29}$$

This utilizes a first order approximation of an exponential function and provides a reasonable mapping of the z domain on to the **s** domain. The first order approximation is accurate for frequencies an order of magnitude less than the switching frequency, $2\pi / T_s$.[19]

Chapter 3

Computation of a Power Converter Small Signal Transfer Function from a Poincare Map of the Closed-Loop Systems

In State Space Averaging the state and output equations of the different topologies are averaged before a state space representation of the perturbation model is obtained. One of the drawbacks of State Space Averaging is that the value of the steady state duty cycle is required to compute the transfer function. Often the duty cycle is determined by the mode of control, which can have a dependence on a state variable and a reference signal. [18] These various control methods lead to particular equations to solve for the steady state duty cycle.

The Poincaré Map approach to solving for the periodic solution of power converter systems provides a state space model of the whole power converter system that includes the plant, compensator, and control mode. The switching surface, $\sigma_j\left(\mathbf{x}_j^k, \mathbf{u}, d_j^k\right)$, factors in the control method and as the Jacobian matrices, $\mathbf{H}_x$ and $\mathbf{H}_u$, are calculated the effects of the control method are incorporated. The advantage of the Poincaré Map approach is that the switching surface has a closed form solution for each control mode, which is simpler to calculate than a particular solution to a complex control mode. Therefore, as the control method increases in complexity the Poincaré Map approach becomes the more worthwhile method of determining the steady state performance.

The challenge then is to extract the steady state transfer function representation of the power converter. The process described in this section calculates the power converter's transfer function using the equations and variables from the previous section.

The Jacobian matrices, $\mathbf{H}_x$ and $\mathbf{H}_u$ describe the state responses of the power converter system due to perturbations in the input and states. Therefore the equation

$$\begin{aligned}\tilde{\mathbf{x}}_0^{k+1} &= \mathbf{H}_x\tilde{\mathbf{x}}_0^{k} + \mathbf{H}_u\tilde{\mathbf{u}} \\ \tilde{v}_0 &= \mathbf{C}\tilde{\mathbf{x}}_0^{k} + \mathbf{D}\tilde{\mathbf{u}}\end{aligned} \tag{30}$$

can be used to determine the small signal transfer function of the overall system. The matrices $\mathbf{C}$ and $\mathbf{D}$ relate perturbations in the output directly to perturbations in the state and input respectively. For the sake of simplicity $\mathbf{D}$ is a zero matrix. The state space representation is a discrete model and the transfer function will be in terms of z.

$$\tilde{\mathbf{T}}_{PM\ Discrete}(z) = \mathbf{C}\left(z\mathbf{I} - \mathbf{H}_x\right)^{-1}\mathbf{H}_u \tag{31}$$

The resulting transfer function is a row vector that contains the transfer function response to the various inputs of the system. In order to determine the system response to one particular input the transfer function that correspond to the same index as the input value in the input column vector.

In order to compare the transfer functions obtained through State Space Averaging and the Poincaré Map directly, $\tilde{\mathbf{T}}_{PM\ Discrete}(z)$, must be transformed into the continuous domain, s. The Bilinear Transform (29) serves as a useful method to obtain $\tilde{\mathbf{T}}_{PM}(s)$,

$$\tilde{\mathbf{T}}_{PM}(s) = \tilde{\mathbf{T}}_{PM\ Discrete}\left(\frac{1+\frac{T_s s}{2}}{1-\frac{T_s s}{2}}\right) \tag{32}$$

The transfer function in (32) describes the response of the overall system response to perturbations in the reference voltage. The overall system consists of a controller, the power converter, and a feedback loop as shown in the figure below.

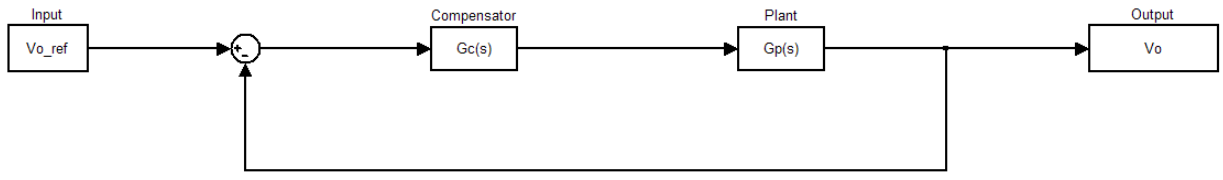

Figure 1: Power Converter System Model

The overall system transfer function can be determined based upon the transfer functions of the controller, $\mathbf{G}_C(s)$, and the plant, $\mathbf{G}_P(s)$.

$$\tilde{\mathbf{T}}_{PM}(s) = \left(\mathbf{I} + \mathbf{G}_P(s)\,\mathbf{G}_C(s)\right)^{-1} \mathbf{G}_P(s)\mathbf{G}_C(s) \tag{33}$$

Solving for $\mathbf{G}_P(s)$ will allow for calculation of the small signal transfer function of the power converter.

$$\mathbf{G}_{P\,PM}(s) = \frac{1}{\mathbf{G}_C(s)} \frac{\tilde{\mathbf{T}}_{PM}(s)}{\left(\mathbf{I} - \tilde{\mathbf{T}}_{NR}(s)\right)} \tag{34}$$

The small signal transfer function of this converter is now determined and can be compared to the State Space Averaged small signal transfer function.

Chapter 4

Implementation and Results

In this chapter State Space Averaging and the Poincaré Map approach are applied to derive the transfer function of a buck converter under voltage mode control. The system will utilize a single pole compensator and will only be analyzed for operation during continuous conduction mode. The first section of this chapter will describe the system to be analyzed and the various values needed for formulas of the next sections.

The second section will calculate the small signal transfer function through the use of the Poincaré Map method and the monodromy matrices. The MATLAB codes used to perform the computations will be cited and described for future ease of use and manipulation. The third section will comprise the derivation of the State Space Averaged small signal transfer function of the buck converter. The last section will house the results of the analysis and compare the Bode diagrams of the two different transfer functions.

An Example of a Voltage Mode Controlled Buck Converter

The circuit that we will utilize in the later analysis will be a voltage mode controlled buck converter. The components of the converter will be assumed to be ideal and DC voltage input to it will be constant. The compensator will have a single pole in the left hand plane. The voltage mode control will be implemented using a switching surface, $\sigma_j\left(\mathbf{x}_j^k, \mathbf{u}, d_j^k\right)$, which will be a function of the present states, the input, and the duty cycle. The switching surface allows for the calculation of the instant at which the topologies should be changed. So long as the value of the switching surface function is greater than zero, the present topology is maintained. When the value of the switching surface function is less than or equal to zero, the next topology is entered.

The states associated with the system are

$$\begin{aligned} x_1 &= Inductor\ Current \\ x_2 &= Capacitor\ Voltage \\ x_3 &= Output\ of\ the\ Controller \end{aligned} \tag{35}$$

The inputs of the system are

$$\begin{aligned} u_1 &= Input\ Voltage \\ u_2 &= Output\ Reference\ Voltage \end{aligned} \tag{36}$$

The values of the circuit components are

$$\begin{aligned} L &= 100\ \mu\text{H} \\ C_O &= 100\ \mu\text{F} \\ R &= 18\ \Omega \\ C_C &= 500\ \mu\text{F} \\ R_C &= 850\ \Omega \\ C_f &= 0.1\ \text{F} \end{aligned} \tag{37}$$

The state space matrices of the system are

$$\mathbf{A}=\begin{bmatrix} 0 & \frac{-1}{f_s L} & 0 \\ \frac{1}{f_s C_O} & \frac{-1}{f_s C_O R} & 0 \\ 0 & \frac{-1}{f_s C_C R_C} & \frac{-C_f}{f_s} \end{bmatrix}$$

$$\mathbf{B}_1=\begin{bmatrix} \frac{1}{f_s L} & 0 \\ 0 & 0 \\ 0 & \frac{1}{f_s C_C R_C} \end{bmatrix} \qquad (38)$$

$$\mathbf{B}_2=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{f_s C_C R_C} \end{bmatrix}$$

$$\mathbf{C}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where f_s is the switching frequency of the switches and has a value of 50 kHz. $\mathbf{B}_1$ is the $\mathbf{B}$ matrix associated with the first circuit topology where the switch from the input voltage to the inductor is closed and the diode is in reverse bias mode. $\mathbf{B}_2$ is the $\mathbf{B}$ matrix associated with the second circuit topology where the switch between the voltage input and the inductor is open and the diode is in conduction mode. The third topology is not considered because the small signal transfer function is not applicable for discontinuous conduction mode.

The last matrices to specify are for the switching surface, σ_j. The transition between the first and second topology occurs when the output of the controller is equal to the duty cycle.

$$\sigma_1 = x_3 - d_1^k \qquad (39)$$

In terms of the state and input vectors:

$$\begin{aligned}
&\sigma_1 = \sigma_{x1}\mathbf{x}_1^k + \sigma_{u1}\mathbf{u} + \sigma_{d1}d_1^k + \sigma_{c1} \\
&\sigma_{x1} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\
&\sigma_{u1} = \begin{bmatrix} 0 & 0 \end{bmatrix} \\
&\sigma_{d1} = -1 \\
&\sigma_{c1} = 0
\end{aligned} \tag{40}$$

The second switching surface to be calculated is for when the topologies transition from the second back to the first. This occurs when the duty cycle, d_2^k, is 1.

$$\sigma_{21} = d_2^k - 1 \tag{41}$$

In terms of the state vectors

$$\begin{aligned}
&\sigma_{21} = \boldsymbol{\sigma}_{x21}\mathbf{x}_2^k + \boldsymbol{\sigma}_{u21}\mathbf{u} + \sigma_{d21}d_2^k + \sigma_{c21} \\
&\sigma_{x21} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\
&\sigma_{u21} = \begin{bmatrix} 0 & 0 \end{bmatrix} \\
&\sigma_{d21} = 1 \\
&\sigma_{c21} = -1
\end{aligned} \tag{42}$$

The subscript here describes that this switching condition is for the transition from the second topology to the first. There are two other topologies that we will not consider for the transitions from the second topology to the third topology and for the transition from the third topology back to the first topology. These two cases are relevant for discontinuous conduction mode.

Calculation of the Poincaré Map Derived Buck Converter Transfer Function

The Jacobian matrix derived small signal transfer function for the buck converter is determined through the use of $\mathbf{H}_x$, $\mathbf{H}_u$, and the related small signal state space representation. Various MATLAB codes are used in the calculation of the Jacobians and they can be found in the appendices. The base code that houses all of the input of the previous section is in Appendix A. The state space matrices and the

switching surface vectors are saved here and are stored in cell arrays for later ease of access. The code is broken into three parts that compute the transient response, the steady state response, and the transfer functions respectively. The function that calculates the Jacobian matrices is in Appendix B. The function determines the periodic steady state response and Jacobian matrices using the initial conditions of the states.

Embedded in the code of Appendix B is a subroutine that performs the calculations of (20), (23), and (27). This subroutine can be found in Appendix C. The state transition matrices $\mathbf{\Phi}_j^k$ and matrices for the zero-state response $\mathbf{\Psi}_j^k$ are calculated and saved in cell arrays too be used in the partial derivatives. The next for loop calculates the partial derivatives that are common to both $\mathbf{H}_x$ and $\mathbf{H}_u$. The next two loops calculate $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ and $\frac{\partial \sigma_{i_j}}{\partial \mathbf{u}}$ respectively, while the last loop computes the summations and multiplications of the partial derivatives. Through this code the matrices $\mathbf{H}_x$ and $\mathbf{H}_u$ are as follows for the system described in the previous section

$$\mathbf{H}_x = \begin{bmatrix} 0.98 & -0.199 & 33.9904 \\ 0.1976 & 0.9691 & 0.8009 \\ -4.672\times10^{-6} & -4.686\times10^{-5} & 1 \end{bmatrix}$$
$$\mathbf{H}_u = \begin{bmatrix} 0.1751 & 1.411\times10^{-3} \\ 19.58\times10^{-3} & 33.24\times10^{-6} \\ -311.7\times10^{-9} & 47.06\times10^{-6} \end{bmatrix} \tag{43}$$

Using these matrices the overall system discrete time transfer function can be solved for. The function that inputs the Jacobians and other necessary variables is listed in Appendix D and returns the Poincaré Map derived buck converter small signal transfer function. In the ideal case the output voltage is the voltage across the buck converter filter capacitor. A specific $\mathbf{C}$ matrix will be used to output only the filter capacitor voltage, x_2, which is the second state of the state vector.

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{44}$$

Using (31) and the second transfer function in the matrix the discrete time system transfer function can be determined.

$$\tilde{\mathbf{T}}_{PM\ Discrete}(z) = \frac{3.325\times10^{-5}z^2 - 0.0002506z - 3.289\times10^{-5}}{z^3 - 2.949z^2 + 2.938z - 0.9889} \tag{45}$$

To obtain a continuous-time transfer function of the system, the Bilinear Transform (29) can be used as a first order approximation. In MATLAB the conversion is implemented using the function "d2c" and using the "Tustin" method.

$$\tilde{\mathbf{T}}_{PM}(s) = \frac{2.342\times10^{-5}s^3 - 2.351s^2 - 4.013\times10^{5}s + 4.022\times10^{10}}{s^3 + 557.2s^2 + 1.007\times10^{8}s + 4.028\times10^{10}} \tag{46}$$

Next using the compensator transfer function from Figure 1, the plant transfer function of the buck converter can be calculated using (34). The resulting transfer function has a high-order polynomial in both the numerator and the denominator. The order can be reduced by using the "minreal" command in MATLAB that will cancel zeros and poles that are within a given tolerance of each other. As a result of the pole-zero cancellation the buck converter small signal transfer function is:

$$\mathbf{G}_{P\,PM}(s) = \frac{9.955\times10^{-6}s^4 - .9993s^3 - 1.705\times10^{5}s^2 + 1.709\times10^{10}s + 1.709\times10^{}}{s^3 + 559.6s^2 + 1.011\times10^{8}s + 6.05\times10^{7}} \tag{47}$$

Calculation of the State Space Averaged Buck Converter Transfer Function

Use of State Space Averaging to determine a small signal transfer function of a nonlinear circuit is described in [18] and in the background section. The information required to begin the calculations are the state space representation of the two circuit topologies associated with continuous conduction mode. The state space representation is for only the buck converter, therefore there are only two states, the inductor current and the filter capacitor voltage. Other important values needed to computed the State

Space Averaged transfer function are the input voltage, V_d, and the duty cycle, d. The input voltage is the first value of the initial conditions vector, $\mathbf{U}$, and the duty cycle determined in the periodic steady state analysis is the second term of the vector, d_sequence.

$$\mathbf{A}_1 = \mathbf{A}_2 = \begin{bmatrix} 0 & \frac{-1}{f_s \mathbf{L}} \\ \frac{1}{f_s \mathbf{C}_O} & \frac{-1}{f_s \mathbf{C}_O \mathbf{R}} \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} \frac{1}{f_s \mathbf{L}} & 0 \\ 0 & 0 \end{bmatrix} \tag{48}$$

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Appendix E houses the function that calculates the State Space Averaging transfer function.

$$\mathbf{G}_{P\,SSA}(s) = \frac{1.7 \times 10^{10}}{s^2 + 555.6s + 10^8} \tag{49}$$

Comparison and Analysis

In the previous two sections the transfer functions of the example buck converter were determined using the Poincare Map and State Space Averaging. The results of a bode plot are in the figure below.

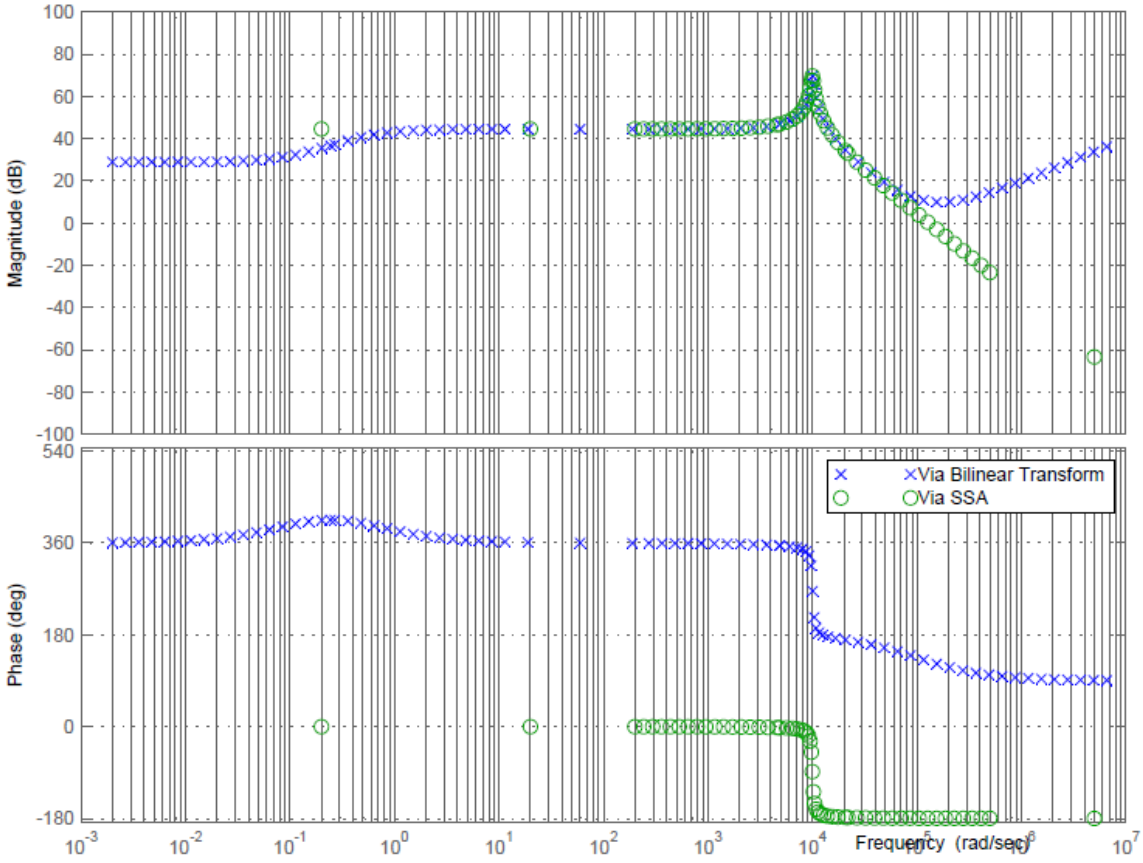

Figure 2: Bode Plots of the Buck Converter Transfer Functions

Upon inspection the transfer function of from the Poincaré Map method has disparity at lower frequencies. The magnitude is ~15dB different between the two transfer functions. The phase plot also shows a slight change from the otherwise steady 360°. The structure of the of the phase plot around 3.45 rad/s suggests that there are pole and zero near that value. This could be a result of the compensator not cancelling properly in (34). The poles and zeros of (47) are:

$$\begin{aligned} \mathbf{poles} &= (-0.02795 \pm \mathrm{j}1.004965)\times 10^4, -0.5985 \\ \mathbf{zeros} &= -1.308\times 10^5, 1.312\times 10^5, 10^5, -0.1 \end{aligned} \tag{50}$$

The lowest pole and zero are relatively close to each other when compared to the others. The zero at -0.1 is the compensator pole which became a zero when the compensator transfer function was inverted in (34). The pole at -0.5985 is a vestige of the Jacobians. The Jacobian matrices $\mathbf{H}_x$ and $\mathbf{H}_u$

represent the response of the entire system to disturbances in the discrete domain, this includes both the compensator and the buck converter. Therefore the compensator's effects should be present in the discrete system transfer function. When transformed into the continuous domain through the use of the Bilinear Transform the effects should still be present. When the plant transfer function is solved for then, in theory, the transformed effects of the compensator should be cancelled when multiplied by the inverse of the compensator transfer function in (34). The difference between the pole and zero may be attributed to the close to zero values in the Jacobians. In order to determine the relationship between the Poincaré Map derived transfer function and the State Space Averaged transfer function the pole and zero were removed and can be seen in the last part of the code in Appendix A.

After the pole and zero are cancelled the transfer function from the two different approaches can be analyzed properly. The bode plot of the new Poincaré Map derived transfer function and the State Space Averaged transfer function is in the figure below.

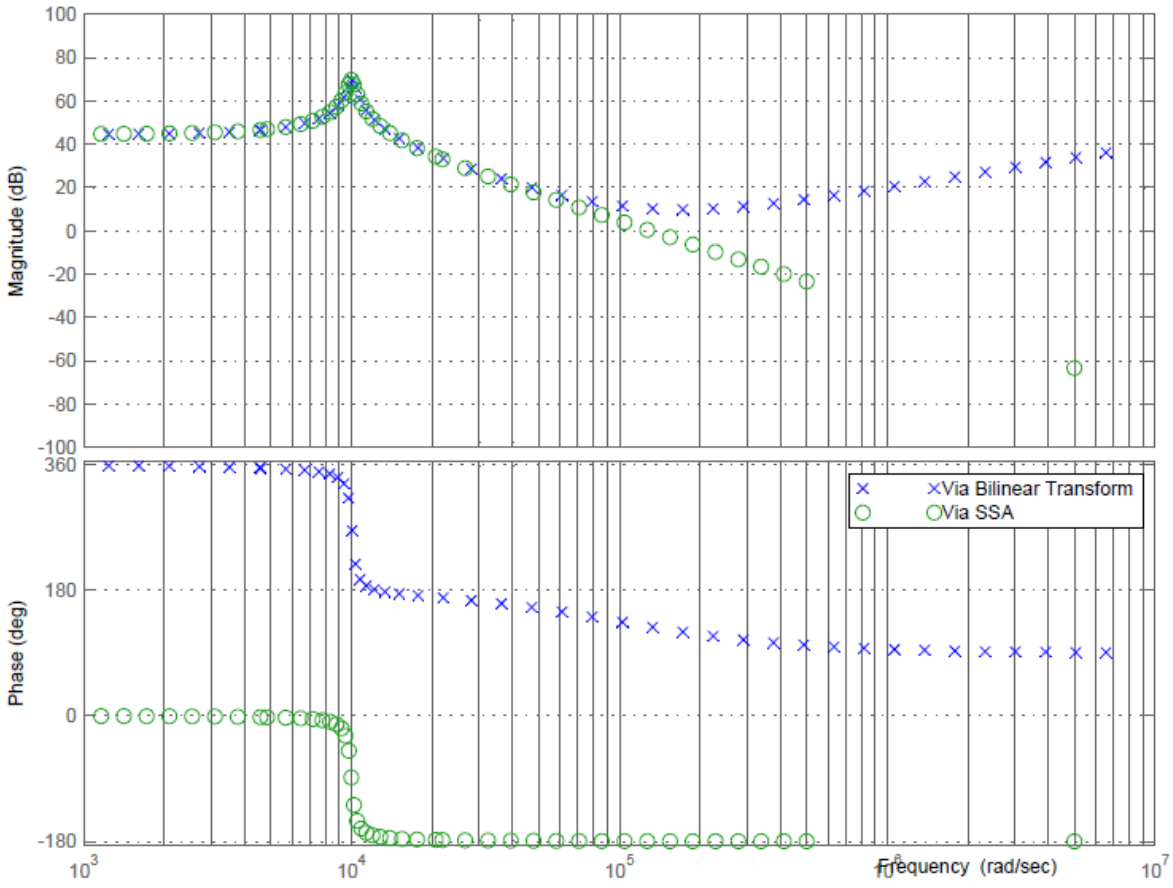

Figure 3: Bode Plots of the Buck Converter Transfer Functions

The switching frequency of the circuit is 314 krad/s and for frequencies an order of magnitude less than the switching frequency the transfer functions match very well. The magnitudes are very close and there is a corner frequency at approximately 10 krad/s. They both have a second order peak at the corner frequency that is caused by duplicate poles. The phase plots are off by 360^{o} which is caused by the two extra zeros in the Poincaré Map transfer function.

The bode plot of the Poincaré Map derived transfer function shows that at higher frequencies the system because of the higher order numerator. The poles and zeros of the Poincaré Map derived transfer function are:

$$\begin{aligned}\mathbf{poles} &= (-0.02795 \pm j1.004965)\times 10^4 \\ \mathbf{zeros} &= -1.308\times 10^5, 1.312\times 10^5, 10^5\end{aligned} \tag{51}$$

The poles of the State Space Averaged transfer function are:

$$\mathbf{poles} = (-0.02777 \pm j0.9996)\times 10^5 \tag{52}$$

The complex poles of both transfer functions are very close and are the reason for the similarities described before. The zeros of the Poincaré Map derived transfer function are from vestiges from the Bilinear Transform.

$$z = \frac{1+\frac{T_s s}{2}}{1-\frac{T_s s}{2}} = \frac{2f_s + s}{2f_s - s} = \frac{10^5 + s}{10^5 - s} \tag{53}$$

The zero at 10^5 comes directly from the Bilinear Transform while the other two are a result of the error attributed to using a first order approximation of $e^{T_s s}$. If more accuracy was required a higher order approximation could be used however the State Space Averaged method typically is only used for frequencies a magnitude below the switching frequency.

Chapter 5

Conclusions

In this thesis the transfer function of a dc-dc converter was determined thought the use of the monodromy matrices calculated in the Poincaré Map approach to solving for the periodic steady state operation of a nonlinear system. The Poincaré Map derived transfer function was shown to be similar to the State Space Averaged transfer function which is a popular method of determining the transfer function of a nonlinear system. Since the two transfer functions are similar it is another affirmation that the Floquét multiplier approach to solving the Poincaré Map of a nonlinear system is an accurate method of analysis of nonlinear devices such as power converters.

Opportunities for future research:

- The monodromy matrix approach and the software can be embedded in a graphical user interface that would allow for the evaluation of the circuit based upon a schematic diagram.
- More complex systems such as the Cuk converter, as well as systems with parallel converters or multiple converter topologies, should be evaluated using the monodromy matrix approach
- The current method of analysis does not work for singular Jacobians. Singular Jacobians can occur in circuits because of the second order nature of the model. There are many published methods and they should be researched and implemented to allow for the implementation of these programs for singular Jacobians.
- Circuits with other control architectures such as current mode control or with different controllers such as a PI compensator.

Bibliography

1. R. D. Middlebrook, "Input filter considerations in design and application of switching regulators," *Proc. IEEE Ind. Applicat. Ann. Meet.*, Oct. 1976.
2. R. D. Middlebrook and S. Cuk, “A general unified approach to modeling switching converter power stages”, *Proc. IEEE Power Electron. Spec. Conf.*, pp. 18-34, Jun. 1976.
3. S. Kriventsov, “Nonlinear techniques for analysis of dc-dc power converter systems”, Ph.D. Dissertation, Dept. Electr. Eng., The Pennsylvania State Univ., University Park, PA, May 2004.
4. P. T. Krein, J. Bentsman, R. M. Bass, and B. L. Lesieutre, “On the use of averaging for the analysis of power electronic systems”, *IEEE Trans. Power Electron.*, vol. 5, pp. 182-190**,** Apr. 1990.
5. S. R. Sanders, J. M. Noworolski, X. Z. Liu, and G. C. Verghese, “Generalized averaging method for power conversion circuits”, *IEEE Trans. Power Electron.*, vol. 6, pp. 251-259, Apr. 1991.
6. C. M. Wildrick, F. C. Lee, B. H. Cho, and B. Choi, "A method of defining the load impedance specifications for a stable distributed power system," *IEEE Trans. Power Electron.*, vol. 10, pp. 280-285, May 1995.
7. B. Lehman and R. M. Bass, “Switching frequency dependent averaged models for PWM dc-dc converters”, *IEEE Trans. Power Electron.*, vol. 11, pp. 89-98, Jan. 1996.
8. B. Lehman and R. M. Bass, “Extensions of averaging theory for power electronic systems”, *IEEE Trans. Power Electron.*, vol. 11, pp. 542-553, Jul. 1996.
9. S. K. Mazumder, A. H. Nayfeh, and D. Boroyevich, “Theoretical and experimental investigation of the fast- and slow-scale instabilities of a dc-dc converter“, *IEEE Trans. Power Electron.*, vol. 16, pp. 201-216, Mar. 2001.

10. C. C. Fang and E. H. Abed, “Sampled-data modeling and analysis of the power stage of PWM dc-dc converters“, *Int. J. Electron.*, vol. 88, pp. 347-369, Mar. 2001.

11. S. Banerjee and G. C. Verghese, “Nonlinear phenomena in power electronics“, New York, NY: IEEE Press, 2001.

12. C. K. Tse, “*Complex behavior of switching power converters*”, Boca Raton, FL: CRC Press, 2004.

13. G. C. Verghese, M. E. Elbuluk, and J. G. Kassakian, “A general approach to sample-data modeling for power electronic circuits,” *IEEE Trans. Power Electron.*, vol. PE-1, pp. 76-89, Apr. 1986.

14. N. Femia, G. Spagnuolo, and V. Tucci, “State-space models and order reduction for dc-dc switching converters in discontinuous modes”, *IEEE Trans. Power Electron.*, vol. 10, pp. 640-650, Nov. 1995.

15. C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later“, *SIAM Rev.*, vol. 45, pp. 3-49, 2003.

16. S. R. Sanders and G. C. Verghese, “Lyapunov-based control for switched power converters”, *IEEE Trans. Power Electron.*, vol. 7, pp. 17-24, Jan. 1992.

17. M. di Bernardo, F. Garofalo, L. Glielmo, and F. Vasca, “Quasi-periodic behaviors in dc-dc converters”, *Proc. IEEE Power Electron. Spec. Conf.*, pp. 1376-1381, Jun. 1996.

18. N. Mohan, Power Electronics Converters, Applications, and Design, 3rd ed., Hoboken, NJ. John Wiley & Sons, 2003.

19. C. L. Phillips, Feedback Control Systems, 4th ed. , Upper Saddle River, NJ: Prentice Hall, 2000.,

Appendix A

buck_converter_voltage_mode_control_single_pole.m

```
%: buck_converter_voltage_mode_control_single_pole.m

%=========================================================================
% Demonstrate periodic steady state stability of a buck converter with
% a single-pole voltage mode controller as in [1].
%
% [1] Kriventsov, S., "Nonlinear techniques for analysis of DC-DC power
% converter systems", Pennsylvania State University, Ph.D.
% dissertation, 2004.

%=========================================================================


%=========================================================================
% 11-Jun-09 JSM Created.

%=========================================================================
clf;
clc;
clear;
pretty_plotting_18
%% Set values of essential parameters.
fs = 50e3;
L = 100e-6;
Co = 100e-6;
R = 18;
Cc = 500e-6;
Rc = 850;
Cf = 0.1;

f0 = 1/sqrt(L*Co)/(2*pi)

%% Define matrices.
A1 = [0 -1/(fs*L) 0;
1/(fs*Co) -1/(fs*Co*R) 0;
0 -1/(fs*Cc*Rc) -Cf/fs];

A3 = A1(2:3,2:3);

B1 = [1/(fs*L) 0; 0 0; 0 1/(fs*Cc*Rc)];
B2 = [0 0; 0 0; 0 1/(fs*Cc*Rc)];
B3 = B2(2:3,:);

C1 = eye(3);
```

```
C3 = [zeros(1,2); eye(2)];

D1 = zeros(3, 2);

A_all = cell(1, 3);
A_all{1} = A1;
A_all{2} = A1;
A_all{3} = A3;

B_all = cell(1, 3);
B_all{1} = B1;
B_all{2} = B2;
B_all{3} = B3;

C_all = cell(1, 3);
C_all{1} = C1;
C_all{2} = C1;
C_all{3} = C3;

D_all = cell(1, 3);
D_all{1} = D1;
D_all{2} = D1;
D_all{3} = D1;

si_x_all = cell(1, 3);
si_x_all{1} = [0 0 1];
si_x_all{2} = [0 0 0; 1 0 0];
si_x_all{3} = [0 0];

si_u_all = cell(1, 3);
si_u_all{1} = [0 0];
si_u_all{2} = [0 0; 0 0];
si_u_all{3} = [0 0];

si_d_all = cell(1, 3);
si_d_all{1} = -1;
si_d_all{2} = [-1; 0];
si_d_all{3} = -1;

si_c_all = cell(1, 3);
si_c_all{1} = 0;
si_c_all{2} = [1; 0];
si_c_all{3} = 1;

m_all = cell(1, 3);
m_all{1} = 2;
m_all{2} = [1; 3];
m_all{3} = 1;

R_all = cell(3,3);
R_all{1, 2} = eye(3);
R_all{2, 1} = eye(3);
R_all{2, 3} = [zeros(2,1) eye(2)];
R_all{3, 1} = [zeros(1,2); eye(2)];
R_all{1, 3} = [zeros(2,1) eye(2)];
```

```
97
98 %% Set initial conditions.
99 U = [170 150]';
100
101 X0 = [7 148 0.8]';
102
103 %% Simulate the transient response (which will not be the PSS response).
104 [m_sequence, d_sequence, x_sequence] ...
105 = ppltici_transient_switching_sequence_via_transition_matrices( ...
106 1, X0, U, ...
107 A_all, B_all, ...
108 si_x_all, si_u_all, si_d_all, si_c_all, m_all, R_all)
109
110 y_transitions ...
111 = pltici_output_transitions(m_sequence, x_sequence, U, C_all, D_all);
112
113 [d_waveform, y_waveform] ...
114 = pltici_output_waveforms_via_transition_matrices( ...
115 m_sequence, d_sequence, x_sequence, U, ...
116 A_all, B_all, C_all, D_all);
117
118 %% Illustrate the transient response.
119 figure(1)
120 clf
121
122 subplot(2,1,1)
123 plot(d_sequence, y_transitions(1, :), '--o', d_waveform, y_waveform(1,:))
124 ylabel('\iti_L\rm [A]', YLabelStyle{:})
125 grid
126
127 subplot(2,1,2)
128 plot(d_sequence, y_transitions(2, :), '--o', d_waveform, y_waveform(2,:))
129 ylabel('\itv_o\rm [V]', YLabelStyle{:})
130 grid
131
132 %% Determine the periodic steady state response.
133 pplti_function = @(x0) ...
134 ppltici_transient_switching_sequence_via_transition_matrices( ...
135 1, x0, U, ...
136 A_all, B_all, ...
137 si_x_all, si_u_all, si_d_all, si_c_all, m_all, R_all);
138
139 [m_sequence, d_sequence, x_sequence, Hx, Hu] ...
140 = pplti_steady_state_switching_sequence_via_newtons_method( ...
141 pplti_function, X0, A_all, B_all, R_all);
142 Hx
143 Hu
144 abs(eig(Hx))
145
146 y_transitions ...
147 = pltici_output_transitions(m_sequence, x_sequence, U, C_all, D_all);
148
149 [d_waveform, y_waveform] ...
150 = pltici_output_waveforms_via_transition_matrices( ...
151 m_sequence, d_sequence, x_sequence, U, A_all, B_all, C_all, D_all);
```

```
152
153 %% Illustrate the PSS response.
154 figure(2)
155 clf
156
157 subplot(2,1,1)
158 plot(d_sequence, y_transitions(1, :), '--o', d_waveform, y_waveform(1,:))
159 ylabel('\iti_L\rm [A]', YLabelStyle{:})
160 grid
161
162 subplot(2,1,2)
163 plot(d_sequence, y_transitions(2, :), '--o', d_waveform, y_waveform(2,:))
164 ylabel('\itv_o\rm [V]', YLabelStyle{:})
165 grid
166
167 %% Calculate the monodromy derived steady state transfer function
168
169 [Gp_NR, Gc]...
170 = Calculate_Monodromy_Derived_SSTF(Hx, Hu, fs, Rc, Cc, Cf);
171
172 %% Calculating the State Space Averaged Tranfer Function
173
174 A1 = [0 -1/(L);
175 1/(Co) -1/(Co*R)];
176 A2 = A1;
177
178 B1 = [1/(L); 0];
179 B2 = [0; 0];
180
181 C1 = [0 1];
182 C2 = C1;
183
184 D1 = zeros(2, 1);
185
186 [Gp_SSA] = SSAtf(A1,A2,B1,B2,C1,C2,D1,D1,d_sequence,y_waveform,U)
187
188 %% Output Results
189 format compact
190 format long
191 disp('Plant Transfer Function using Tustin Method')
192 Gp_NR
193 disp('Poles of TF')
194 Poles_Gp_NR = roots(Gp_NR.den{1})
195 disp('Zeros of TF')
196 Zeros_Gp_NR = roots(Gp_NR.num{1})
197
198 disp('Plant Transfer Function from State Space Averaging')
199 Gp_SSA
200 disp('Poles of TF')
201 roots(Gp_SSA.den{1})
202
203 figure(3)
204 hold on
205 bode(Gp_NR,'bx')
206 bode(Gp_SSA,'go')
```

```
grid on
legend('Via Bilinear Transform','Via SSA')

%% Cancelling out the Poles and Zeros Manually
Gp_NR = Gp_NR*tf([1 -Poles_Gp_NR(3)],[1 -Zeros_Gp_NR(4)]);
Gp_NR = minreal(Gp_NR,10E-9);

%% Output Results
format compact
format long
disp('Plant Transfer Function using Tustin Method')
Gp_NR
disp('Poles of TF')
Poles_Gp_NR = roots(Gp_NR.den{1})
disp('Zeros of TF')
Zeros_Gp_NR = roots(Gp_NR.num{1})

disp('Plant Transfer Function from State Space Averaging')
Gp_SSA
disp('Poles of TF')
roots(Gp_SSA.den{1})

figure(4)
hold on
bode(Gp_NR,'bx')
bode(Gp_SSA,'go')
grid on
legend('Via Bilinear Transform','Via SSA')
```

Appendix B

[m_sequence, d_sequence, x_sequence, Hx, Hu] = pplti_steady_state_switching_sequence_via_newtons_method(pplti_function, X0, A_all, B_all, R_all)

```
% PPLTI_STEADY_STATE_SWITCHING_SEQUENCE_VIA_NEWTONS_METHOD Determine
% the periodic steady state response of a periodic piecewise LTI system
% using Newtwon's method applied to the Poincare map.
%
% [m_sequence, d_sequence, x_sequence, Hx, Hu] ...
% = pplti_steady_state_switching_sequence_via_newtons_method( ...
% pplti_function, x0, A_all, R_all, varargin)
% determines the periodic steady state response and corresponding
% monodromy matrix Hx of a periodic piecewise LTI system given the
% initial value of the state vector.
%
% The outputs:
% m_sequence : Sequence of model indices.
% d_sequence : Sequence of duty cycles corresponding to transitions
% between LTI models;
% d_sequence(1) = 0 and d_sequence(end) = 1.
% x_sequence : Sequence of state vectors immediately before and
% after each transition between LTI models;
% except x_sequence(:,1) is the initial condition and
% x_sequence(:,end) is the final condition.
% Hx : Monodromy matrix.
% Hu :
%
% The inputs:
% pplti_function : Function to compute transient switching sequence.
% x0 : Estimated initial value of the state vector.
% A_all : A matrices for all models.
% B_all : B matrices for all models
% R_all : R matrices to map a state vector from one model
% to another.
%
% See also: pplti_monodromy_matrix.
%
% !!!THIS CODE DOES NOT HANDLE SINGULAR JACOBIANS!!!

function [m_sequence, d_sequence, x_sequence, Hx, Hu] ...
= pplti_steady_state_switching_sequence_via_newtons_method( ...
pplti_function, x0, A_all, B_all, R_all, varargin)

tolerance = 1e-6; if nargin > 5; tolerance = varargin{1}; end
max_iterations = 20; if nargin > 6; max_iterations = varargin{2}; end
al = 1; if nargin > 7; al = varargin{3}; end
p = 2; if nargin > 8; p = varargin{4}; end

for iteration = 1:max_iterations
```

```
% Determine the transient response over one period.
[m_sequence, d_sequence, x_sequence, ...
si_x_sequence, si_u_sequence, si_d_sequence, xdot_sequence] ...
= pplti_function(x0);

j = size(m_sequence, 2) - 1;

x0 = R_all{m_sequence(1), m_sequence(j)}*x0;

x0_next = x_sequence{2*j};

f = x0_next - x0;

[Hx, Hu] = pplti_monodromy_matrix( ...
m_sequence, d_sequence, ...
si_x_sequence, si_u_sequence, si_d_sequence, xdot_sequence, ...
A_all, B_all, R_all);

J = Hx - eye(size(Hx));

dlx0 = J\f;

x0 = R_all{m_sequence(j), m_sequence(1)}*(x0 - al*dlx0);

err = norm(dlx0, p);

%disp(['iteration ' num2str(iteration) ': error = ' num2str(err)])

if err < tolerance
break;
end
end
```

Appendix C

[Hx, Hu] = pplti_monodromy_matrix(m_sequence, d_sequence, si_x_sequence, si_u_sequence, si_d_sequence, xdot_sequence, A_all, B_all, R_all)

```
% PPLTI_MONODROMY_MATRIX
% 3-11-2010 JPR added code to calculate Hu
function [Hx, Hu] ...
= pplti_monodromy_matrix( ...
m_sequence, d_sequence, ...
si_x_sequence, si_u_sequence, si_d_sequence, xdot_sequence, ...
A_all, B_all, R_all)

J = size(m_sequence, 2) - 1;

% Generate state transition matrices.
PHI = cell(1, J);
PSI = cell(1, J);
for j = 1:J
Aj = A_all{m_sequence(j)};

Bj = B_all{m_sequence(j)};

dldj = d_sequence(2*j) - d_sequence(2*j - 1);

PHI{j} = expm(Aj*dldj);

PSI{j} = inv(Aj)*(PHI{j}-eye(size(PHI{j})))*Bj;
end

N = size(Aj, 1);
M = size(Bj,2);
% Generate partial derivative matrices.
partial_f_over_partial_d = zeros(N, J);
partial_si_over_partial_x0 = zeros(J - 1, N);
partial_si_over_partial_d = zeros(J - 1);
partial_si_over_partial_u = zeros(J,size(si_u_sequence{1},2));
partial_f_over_partial_u = zeros(size(PSI{1}));
mprevious = m_sequence(J);

PI_1_to_j = eye(size(A_all{mprevious}));

for j = 1:(J - 1)
PI_1_to_j = PHI{j}*R_all{mprevious, m_sequence(j)}*PI_1_to_j;
mprevious = m_sequence(j);

mj = m_sequence(j);
mj_plus_1 = m_sequence(j + 1);

R = R_all{mj, mj_plus_1};

xjdot = xdot_sequence{2*j};
```

```
48
49 dl_xjdot = R*xjdot - xdot_sequence{2*j + 1};
50
51 partial_si_over_partial_x0(j, :) = si_x_sequence{j}*PI_1_to_j;
52
53 partial_si_over_partial_d(j, j) ...
54 = si_x_sequence{j}*xjdot + si_d_sequence(j);
55
56 PI_j_plus_1_to_i = PHI{j + 1};
57
58 for i = (j + 1):(J - 1)
59 partial_si_over_partial_d(i, j) ...
60 = si_x_sequence{i}*PI_j_plus_1_to_i*dl_xjdot;
61
62 PI_j_plus_1_to_i ...
63 = PHI{i + 1}*R_all{m_sequence(i), m_sequence(i + 1)} ...
64 *PI_j_plus_1_to_i;
65
66 end
67
68 partial_f_over_partial_d(:, j) = PI_j_plus_1_to_i*dl_xjdot;
69
70 end
71
72 %% Partial f over partial u
73
74 for j = 1:J
75
76 PI_j_plus_1_to_J = eye(size(PHI{J-1}));
77
78 for k = j+1:J
79 PI_j_plus_1_to_J ...
80 = PHI{k}*R_all{m_sequence(k-1), m_sequence(k)} ...
81 *PI_j_plus_1_to_J;
82 end
83
84 partial_f_over_partial_u = ...
85 R_all{m_sequence(J),m_sequence(1)}*PI_j_plus_1_to_J*...
86 R_all{m_sequence(j),m_sequence(j+1)}*PSI{j} + ...
87 partial_f_over_partial_u;
88
89 end
90
91 %% Partial si over partial u
92
93 for i = 1:J
94 partial_si_over_partial_u(i,:) = si_u_sequence{i};
95 sum_PI_j_plus_1_to_i_times_PSI_j
=zeros(size(PHI{i},1),size(si_u_sequence{i},2));
96 for j = 1:i
97
98 PI_j_plus_1_to_i = eye(size(PHI{j}));
99 for k = j+1:i
100 PI_j_plus_1_to_i ...
101 = PHI{k}*R_all{m_sequence(k), m_sequence(k + 1)} ...
```

```
*PI_j_plus_1_to_i;
end

sum_PI_j_plus_1_to_i_times_PSI_j = ...
PI_j_plus_1_to_i*R_all{m_sequence(j), m_sequence(j+1)}*PSI{j}...
+ sum_PI_j_plus_1_to_i_times_PSI_j;

end

partial_si_over_partial_u(i,:) = ...
si_x_sequence{i}*sum_PI_j_plus_1_to_i_times_PSI_j...
+ partial_si_over_partial_u(i,:);

end

% Compute Hx and Hu matrix.
H_x = PHI{J}*R_all{m_sequence(J - 1), m_sequence(J)}*PI_1_to_j;

H_u = partial_f_over_partial_u;

correction_Hx = zeros(size(H_x));

correction_Hu = zeros(size(H_u));

i_set = (J - 1):-1:1;

for j = 1:(J - 1)
i_subsets_of_size_j = combnk(i_set, j);

for n = 1:size(i_subsets_of_size_j, 1)
i = i_subsets_of_size_j(n, :);

product_ratio = (-1)^j/partial_si_over_partial_d(i(j), i(j));

for k = 1:(j - 1)
product_ratio ...
= product_ratio ...
*partial_si_over_partial_d(i(k), i(k + 1)) ...
/partial_si_over_partial_d(i(k), i(k));
end

correction_Hx ...
= correction_Hx ...
+ product_ratio ...
*partial_f_over_partial_d(:, i(1)) ...
*partial_si_over_partial_x0(i(j), :);

correction_Hu ...
= correction_Hu ...
+ product_ratio ...
*partial_f_over_partial_d(:, i(1)) ...
*partial_si_over_partial_u(i(j), :);
end
end

```

```
157 Hx = H_x + correction_Hx;
158 Hu = H_u + correction_Hu;
159
```

Appendix D

Calculate_Monodromy_Derived_SSTF.m

```
% Calculate_Monodromy_Derived_SSTF
% This function will calculate the small signal transfer function of a buck
% converter using the jacobian matrices Hu and Hx as in [1]

% [1] Kriventsov, S., "Nonlinear techniques for analysis of DC-DC power
% converter systems", Pennsylvania State University, Ph.D.
% dissertation, 2004.


function [Gp_NR, Gc]...
= Calculate_Monodromy_Derived_SSTF(Hx, Hu, fs, Rc, Cc, Cf)
syms z s
Ts = 1/fs;
% C matrix from the continuous State Space

C = [0 1 0];

% Calculate the Discrete Time Transfer function based upon Hx = A, Hu = B

T_NR_discrete = tf(ss(Hx,Hu,C,0,1/fs))

% Perform the Bilinear or Tustin Transformation to continous time

T_NR = d2c(T_NR_discrete(2),'tustin')

T_NR = minreal(T_NR)
% Generate the Compensator Transfer Function

Gc = tf([1/(Rc*Cc)],[1 Cf])
% Calculate Gp_d_H

Gp_NR = 1/Gc*T_NR/(1-T_NR)

Gp_NR = minreal(Gp_NR,10E-9)
```

Appendix E

SSAtf.m

```
1 function [Gp_SSA] = SSAtf(A1,A2,B1,B2,C1,C2,D1,D2,d_sequence,y_waveform,U)
2 d = d_sequence(2);
3
4 X = [mean(y_waveform(1,:));
5 mean(y_waveform(2,:))];
6
7 A = A1*d + A2*(1-d);
8 B = B1*d + B2*(1-d);
9 C = C1*d + C2*(1-d);
10 D = D1*d + D2*(1-d);
11
12 Vd = U(1);
13 syms s
14 Gp_SSA = C*inv(s*eye(2) - A)*((A1 - A2)*X + (B1 - B2)*Vd) + (C1 - C2)*X
15 Gp_SSA = sym2tf(Gp_SSA);
```

VITA

Johnathan Paul Ross

EDUCATION

The Pennsylvania State University
The Schreyer Honors College
B.S. in Electrical Engineering
Graduate in May 2010
Advisor: Dr. J. Mitchell

EXPERIENCE

Lutron Electronics, Coopersburg, Pennsylvania
Project Electrical Engineer (start in 2010)

GE Energy Infrastructure, Schnectady, New York
Electrical Engineering Intern in Generator Rotor and Collector Engineering (June 2009 – Aug. 2009)

GE Transportation, Erie, Pennsylvania
Electrical Engineering Intern in Train Control (June 2008 – Aug. 2008)

Aberdeen Test Center, Aberdeen, Maryland
Engineering Intern in Chemical Sampling and Analysis (June 2007 – Aug. 2007)

PROJECTS

Senior Design Project
Designed a large scale electric motor for use in the drive train in an electric bus

Renewable Energy for Central America
Travelled to Roatán, Honduras to build a photovoltaic system to provide electricity for a school

Professional Organizations

Tau Beta Pi
Eta Kappa Nu
NECA Student Chapter

Awards

Matthew P. Blickley Memorial Fund
Dean's list for seven semesters (For having a 3.50 GPA or higher)
Lockheed Martin Corporation Scholarship