2022

STABILITY APPROXIMATION METHOD FOR PULSE LOADS IN POWER ELECTRONIC SYSTEMS

Eduardo A. Marmol D
Michigan Technological University, emarmol@mtu.edu

STABILITY APPROXIMATION METHOD FOR PULSE LOADS IN POWER ELECTRONIC SYSTEMS

By

Eduardo A. Marmol D

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical and Computer Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2022

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

Department of Electrical and Computer Engineering

Thesis Advisor: *Dr. Wayne W. Weaver*

Committee Member: *Dr. Gordon Parker*

Committee Member: *Dr. Flavio Costa*

Department Chair: *Dr. Jin W. Choi*

Contents

List of Figures

List of Tables

Acknowledgments

During my time at Michigan Tech, I had the opportunity to meet a lot of people, and thanks to their support, advice, and friendship I was able to achieve my goals. I would like to extend my sincere gratitude to my advisor Dr. Wayne Weaver for his patience, guidance, and support, throughout the course of this research. Also, I would like to thank my committee members and Dr. Gowtham. Finally, I would like to thank my family and friends for all their help.

List of Abbreviations

P_L	Linear power
P_{nl}	Nonlinear power
w_n	Natural frequency
w_{no}	Natural frequency for $P = 0$
w_{nl}	Natural frequency for $P = P_{nl}$
w_{nP}	Natural frequency of the power understudy
T_n	Natural Period
T_{nl}	Natural Period for $P = P_{nl}$
T_{nP}	Natural Period of the power understudy
i_{Lo}	Inductor's current equilibrium point
$i_{Lo_{P=P_{nl}}}$	Inductor's current for P
$i_{Lo_{P=0}}$	Inductor's current for $P = 0$
V_{co}	Capacitor's voltage equilibrium point

Abstract

Pulse loads on power electronic distribution systems are becoming very popular nowadays as the components of ships and airplanes are moving to more electric power. However, the pulse loads have a destabilizing effect on the power distribution system. Usually, the method used to study the stability of this type of system are small-signal analyses and are based on a system where the load is modeled as a constant. Since DC-DC systems with pulsed loads are very nonlinear, a small-signal analysis does not provide helpful information related to the stability of the system. The work presented in this thesis focuses on the large-signal stability analysis of the system based on the average-mode model of dc-dc converters. Where the linear parameters of the system found using small-signal analysis are used to define a series of cases, equations, and relationships. This new method will give a more accurate approximation of the stability of the full nonlinear systems for a pulse load than small-signal analysis and much faster than the Hamiltonian Surface Shaping and Power Flow Control (HSSPFC) method. The objective of this investigation is to get an accurate approximation that does not require as much computational time.

Chapter 1

Introduction

This research work was developed with the specific objective of decreasing the computational time required to study the stability behavior of electric warships for pulse power loads since the Hamiltonian Surface Shaping method, which is a large-signal stability analysis that accurately describes the stability behavior of the system requires a lot of computational time. The electric warship is a new technology that was developed to support the electrical power demands of advanced weapons and combat systems of warships with the purpose of reducing fuel consumption and ownership costs [1]. Designing a power and energy system that meets the dynamic load demand of the electric warship has been a challenge due to the lack of efficiency of the current designing tools available since they consider a constant power demand, and this

does not accurately describe the ship loads [2]. Dynamic loads such as electromagnetic guns, electromagnetic launch systems, and free electron lasers usually operate as pulse loads signals with a power magnitude, duty cycle, and period. The integration of these loads on the system is challenging [3]. A lot of research has been done on the effects of constant power loads in DC systems and they all concluded that constant loads have a destabilizing effect on the system [4][5][6]. Also, it has been proven that pulse loads cause changes in the voltage of the system and depending on the values of the magnitude, duty cycle, and period of the pulse can lead to instability [7][8]. A typical approach to study the stability of these systems for a constant power load is by a small-signal analysis. However, the changes in the pulse power loads are so drastic that the small-signal analysis is not applicable because it gives inappropriate and inaccurate results. Since a pulse power load is a time-variant system, a good approach to study the stability of the system would be a linear time-variant method such as Floquet theory [9][10]. However, even though Floquet theory describes the time-variant nature of the power pulse load, this method still fails to capture the large-signal behavior of the system.

Other research has focused on designing controllers to mitigate the instability effects of pulse loads on the system. In [11], the energy storage of the electromagnetic launcher (EML) to feed both the free-electron laser (EFL) and the main power bus of the system is used, and instability effects of the pulse loads are mitigated. In [2] they mitigate the instability effects by implementing a simple guidance control scheme to

adjust power flows when a pulse load is active. However, these controllers might not be necessary for certain pulse power loads, for this reason, stability analysis plays an important role in the design of electric ships. For the reason mentioned above a new method to analyze pulse power loads on a DC electric distribution system, such as electric ships, was introduced in [12]. Where they represented the power system of the ship and pulse power load as a Hamiltonian surface which is a type of Lyapunov function which is able to capture the large-signal effects and predict the stability boundaries of the system for a pulse power load. Also, they introduced a new concept to define a system as metastable, because during the on period of the pulse power load the bus voltage grows exponentially but the off period of the pulse load damps the instability and keeps the voltage of the system bounded. They defined this bounded instability as meta-stable. However, even though this method is very accurate, it requires a lot of computational time to generate the data that describes the stability of the system for power loads with different values of power, duty cycle, and period. Usually, to determine the stability behavior of the system for a pulse power load with a specific magnitude of power can take around eight hours. Although the Hamiltonian surface method is able to accurately predict the stability of the system for pulse power load, this method requires a lot of computational time to generate the data that describes the stability behavior of the system for power loads with different values of power, duty cycle, and period. For this reason, an approximation method based on the linear parameters of the system is used to predict the stability of the

DC-DC system with a pulse power load. This method predicts the stability of the system for power loads and requires less computational time than the Hamiltonian surface method.

Chapter 2

System Dynamics and Theory

2.1 Stability

In control system design and study, stability plays an important role, since it determines the safe operation of a system. The stability of a system is determined by the ability of the system to maintain or restore its equilibrium whenever the system experiences an abrupt change caused by internal or external forces. In power systems, a system is defined as stable when its characteristics are able to maintain the system in a state of equilibrium under standard operating conditions, and when the system can reach equilibrium after being exposed to a disturbance [13].

A basic and typical approach to study the stability is through a small-signal Eigenvalues analysis, and it is only valid for a small region around the operating points of the system. However, in this work, the system that is being studied has a pulse power load which makes it a time-variant system and for this reason, the small signal approach is not accurate.

2.2 Floquet theory

The system under consideration is a periodic time-variant. Floquet theory is one of the few tools available to test the stability of a time-variant system. Floquet theory allows to study the stability of a periodic time-varying pulse power load. This method considers a set of differential equations that are linear, homogeneous, and time-periodic [14]. In Floquet theory a time-varying linear system represented as

$$\frac{dx}{dt} = A(t)x \tag{2.1}$$

where A(t) is an nxn matrix with minimal period T and x is an n-dimensional vector and the general solution for (2.1) is

$$x = \sum_{i=1}^{n} c_i e^{\mu_i t} p_i(t) \tag{2.2}$$

where c_i are constants related to initial conditions, $p_i(t)$ are vector-valued functions with period T, and μ_i are the Floquet exponents, which are equal to the eigenvalues. The Floquet multipliers are associated with the Floquet exponents by

$$\rho_i \;=\; e^{\mu_i T}. \tag{2.3}$$

The long-term behavior of the system is defined by the Floquet exponents. The zero equilibrium is stable if all the Floquet exponents have negative real parts, or equivalently the Floquet multipliers have real parts between -1 and 1. From (2.3) the Floquet exponents are defined as

$$\mu_i \;=\; \frac{\log \rho_i}{T}. \tag{2.4}$$

To demonstrate Floquet stability, a study is done with an illustrative example where the system under study is defined by the matrix $A(t)$ as

$$A(t) \;=\; \begin{bmatrix} -1 & \sin(t) \\ \cos(t)+5 & -1 \end{bmatrix}$$

where the Floquet multipliers are found using (2.1) as

$$\frac{dx}{dt} \;=\; \begin{bmatrix} -1 & \sin(t) \\ \cos(t)+5 & -1 \end{bmatrix} X$$

where x is an nxn matrix and has initial conditions of the identity matrix (x(0) = I). The Floquet multipliers (ρ_i) are the Eigenvalues of $x(t)$ and the solution of x must be found numerically. To solve the differential equation a MATLAB code was used. Then, the numeric solution of the system must be evaluated at a time equal to the period of the periodic signal, in this case, $T = 2\pi$. The solution of the system for $t = T$ is

$$x(T) = \begin{bmatrix} 0.403743 & 0.088559 \\ 0.314482 & 0.068989 \end{bmatrix}$$

the eigenvalues of $x(T)$ (Floquet multipliers) are

$$\rho(T) = \begin{bmatrix} 0.472772 \\ 7.377\,12 \times 10^{-6} \end{bmatrix}.$$

Using (2.4) the Floquet exponents are

$$\mu(T) = \begin{bmatrix} -0.119246 \\ -1.88075 \end{bmatrix}$$

since the real part of the Floquet exponents have negative real parts, the Floquet stability study concludes that the system is stable.

2.3 Metastability

In the case of pulse loads on a DC electric power system, such as electric ships, the concept of stability of the pulse power system is redefined as metastable. This new terminology to describe the stability of this system was introduced in [12]. The system might be unstable for a value of power during the "on" period of the pulse load, and this will cause the bus voltage to increase exponentially. However, during the "off" period of the pulse load the instability is damped and this will keep the bus voltage bounded. Due to these reasons, the "on" cycle of bound instability plus the damping cycle of the off period will define the metastability of a DC electric power distribution system.

Since power distribution systems are able to work between a range of voltages, when the system reaches a stable limit cycle for pulse load with a specific period and duty cycle, the system will be defined as metastable. Once the system reaches a stable limit cycle the bus voltage will be bounded between certain values of voltage. The range of bounded voltages used to define when a system is considered metastable will depend on the constraints established by the designer of the system. For the purpose of this research, the system is define as metastable when the minimum voltage of the system was greater than $0.4v_{co}$, where v_{co} is the nominal bus operating voltage of the system, which has a magnitude of 400 V.

Chapter 3

Small signal analysis and nonlinear model

3.1 System and load model

For the purpose of this study, an average-mode model of a boost converter was used to study the stability effect of a pulse load on a DC/DC grid as shown in Figure 3.1. In this research an average-mode model was considered where $\lambda = 1 - D$. Where D represents the duty cycle of the active switch [15]. The pulse load under study has the waveform shown in Figure 3.2 where P represents the magnitude of the pulse, T is the period and D is the duty cycle of the pulse power load. The system shown in

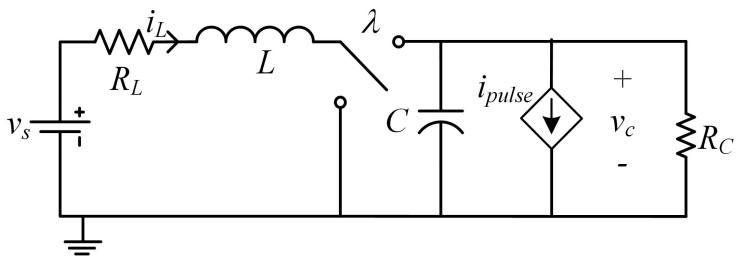

Figure 3.1: Average-mode model of a boost converter with pulse current load.

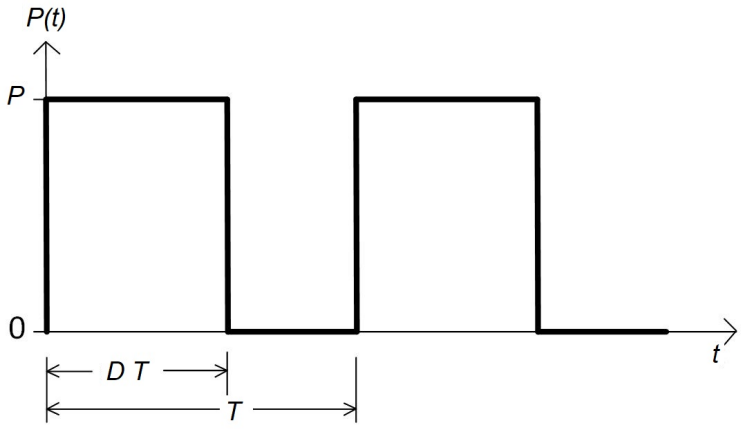

Figure 3.2: Pulse time-dependent power waveform.

Figure 3.1 can be described by the following mathematical equations:

$$L\frac{di_L}{dt} = V_s - \lambda v_c - R_L i_L \tag{3.1}$$

$$C\frac{dv_c}{dt} = \lambda i_L - \frac{v_c}{R_c} - i_{pulse} \tag{3.2}$$

where i_{pulse} represents the pulse power load, V_s is the voltage of the energy storage system, i_L is the current of the inductor, v_c is the voltage of the capacitor, λ is the

switching frequency of the boost converter and the load model is

$$i_{pulse} = \frac{P(t)}{v_c} \tag{3.3}$$

the effects of power loads on this type of system has been studied by several researchers and they concluded that these loads have de-stabilizing characteristics and have spent a lot of time trying to mitigate this effect [16][17][18].

3.2 Small-signal stability analysis

Though this method is not valid for the case that is being studied, it will provide important parameters such as the natural frequency and period of the system. These parameters are going to be used to define the equations and conditions of the approximation method developed in this investigation. The eigenvalues stability analysis is a simple approach to study the stability of a power system for a constant power load. From the system model (3.1)-(3.2), the linearized model of the system for a constant power load (P) is

$$\frac{dx}{dt} = \begin{bmatrix} \frac{-R_L}{L} & \frac{-\lambda}{L} \\ \frac{\lambda}{C} & \frac{1}{C}\left(\frac{P}{{v_{co}}^2} - \frac{1}{R_C}\right) \end{bmatrix} x + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u \tag{3.4}$$

where $x^T = \begin{bmatrix} i_L & v_c \end{bmatrix}$, $u = V_s$ and v_{co} is the linearized operating point of the system. From (3.4), the characteristic equation is

$$s^2 + \left(\frac{R_L}{L} + \frac{1}{CR_c} - \frac{P}{C{v_{co}}^2}\right)s + \left(\frac{R_L}{CLR_c} + \frac{\lambda^2}{CL} - \frac{PR_L}{CL{v_{co}}^2}\right) = 0 \tag{3.5}$$

which is stable when

$$\frac{R_L}{L} + \frac{1}{CR_C} - \frac{P}{C{v_{co}}^2} > 0 \tag{3.6}$$

$$\frac{R_L}{CLR_C} + \frac{\lambda^2}{CL} - \frac{PR_L}{CL{v_{co}}^2} > 0 \tag{3.7}$$

or

$$P < \frac{{V_{co}}^2}{R_C} + \frac{R_L C {v_{co}}^2}{L} \tag{3.8}$$

and

$$P < \frac{{V_{co}}^2}{R_C} + \frac{\lambda^2 {v_{co}}^2}{R_L} \tag{3.9}$$

where the maximum value of the power constraint is taken as (3.8) as defined in [12].

From (3.1) and (3.2), the linearized operating point of the inductor's current i_{Lo} and power source V_{so} are found solving the following system for a capacitor's voltage v_{co}

$$\frac{1}{L}(V_{so} - \lambda v_{co} - R_L i_{Lo}) = 0 \tag{3.10}$$

$$\frac{1}{C}\left(\lambda i_{Lo} - \frac{P}{v_{co}} - \frac{v_{co}}{R_C}\right) = 0 \tag{3.11}$$

from (3.10) and (3.11) the equation that defines i_{Lo} and V_{so} are

$$i_{Lo} = \frac{1}{\lambda}\left(\frac{P}{v_{co}} + \frac{v_{co}}{R_C}\right) \tag{3.12}$$

$$V_{so} = \lambda v_{co} + R_L i_{Lo}. \tag{3.13}$$

However, the eigenvalues stability analysis is only valid for a time-invariant system and is only accurate for a small region around the linearized operating point v_{co}. The characteristic equation of a second-order system has the following form

$$s^2 + 2\zeta\omega_n s + {\omega_n}^2 = 0. \tag{3.14}$$

From (3.14) the natural frequency of the linearized system can be determined. Comparing (3.5) and (3.14) the natural frequency of the system is defined as

$$\omega_n = \sqrt{\frac{R_L}{CLR_C} + \frac{\lambda^2}{CL} - \frac{PR_L}{CL{v_{co}}^2}} \tag{3.15}$$

from (3.15), the fundamental period of the system is calculated as

$$T_n = \frac{2\pi}{\omega_n} \tag{3.16}$$

and from (3.8) the maximum linear power that the system can support is defined as

$$P_L = \frac{{V_{co}}^2}{R_C} + \frac{R_L C {V_{co}}^2}{L}. \tag{3.17}$$

Using the parameters from Table 3.1 and (3.17), the maximum constant power that

Table 3.1
Example parameters

Parameter	**Value**
R_L	0.1 Ω
L	10 mH
C	1 mF
R_C	50 Ω
λ	0.5
v_{co}	400 V

the system can support is $P_L = 4800\ W$.

3.3 Floquet theory analysis

As discussed in Chapter 2.2, Floquet theory is one of the tools available to study the stability of time-varying systems. Where the stability of the system is determined by the Floquet exponents and multipliers. In this research, the Floquet stability of the system was considered to be stable base on the behavior of the Floquet exponents. The system is considered stable when the real part of the Floquet exponents are less

than or equal to zero.

This approach allows the analysis of the linearized system model of the pulse power load (3.1) and (3.2) with the load (3.3) where the linearized system is defined as

$$\frac{dx}{dt} = \begin{bmatrix} \frac{-R_L}{L} & \frac{-\lambda}{L} \\ \frac{\lambda}{C} & \frac{1}{C}\left(\frac{P(t)}{{v_{co}}^2} - \frac{1}{R_C}\right) \end{bmatrix} x \tag{3.18}$$

where $x^T = \begin{bmatrix} i_L & v_c \end{bmatrix}$ and v_{co} is the equilibrium point of the bus voltage. The stability of this system will be numerically solved with Floquet theory with different parameters and a periodic PWM pulse load shown in Figure 3.2.

Table 3.2
Example stability results using Floquet theory for a power of 6000 W and period of $T = 0.25\ s$

Duty cycle	**Re(Floquet exponents)**	**Floquet stability analysis**
0.9	1.9364	Unstable
0.8	0.0654	Unstable
0.6	-3.6456	Stable

The data in the Table 3.2 was generated following the steps described in Chapter 2.2, using the parameters from Table 3.1, a pulse load with a magnitude of 6000 W, a constant period of $T = 0.25\ s$ and different values of duty cycle of $D = [0.9, 0.8, 0.6]$. The system was defined as stable when the real part of the Floquet exponents is less or equal to zero.

Since the purpose of this research was to determine the maximum duty cycle for a specific value of power and period, a MATLAB code was developed following the steps described in Chapter 2.2, to test the stability of the system for different values of power, duty cycle, and period. The code was set up in a way that it will start the stability study of the system at a value of duty cycle equal to 1 (100% duty cycle) for a specific value of period and power, and if the stability analysis concludes that the system is unstable then it will decrease the value of the duty cycle by 0.001. It will continue testing the stability of the system until it found the maximum duty cycle for each value of period and power. The stability test is done at a value of time equal to the period of the pulse load. The code used to generate the Floquet theory data is shown in Appendix A and the data was gathered using the Michigan Tech HPC Superior.

3.4 Nonlinear model

The nonlinear model of the system defined by (3.1), (3.2), and (3.3), was simulated using MATLAB and Simulink [19][20]. In order to get the real data of the system and to be able to develop the approximation model. Different parameters were used to gather the data that is going to be compared with the approximation model. The nonlinear model was defined to be metastable when the minimum voltage of the system was greater than $0.4v_{co}$.

To gather the data a Simulink model was developed using the equations (3.1), (3.2), (3.3), and the parameters show in Table 3.1. Then, using MATLAB, a script was used to call the Simulink file and set the parameter values of the system for each simulation. The MATLAB code used to gather the data is shown in the Appendix B. Where the code was set up in a way that it will run thousands of simulations varying the values of the duty cycle and period of the pulse load of the system. Then from the data of a specific value of duty cycle and period of time if the output voltage of the load satisfied the voltage constraint defined above then the system is considered stable. The simulation time was set to 10 seconds because studying the stability of the full nonlinear system for just one or three times the pulse load period was not enough. There were cases where the system became unstable after 4 or 5 seconds. The objective of this code is to get the maximum value of the duty cycle for each period of time and for a specific value of power.

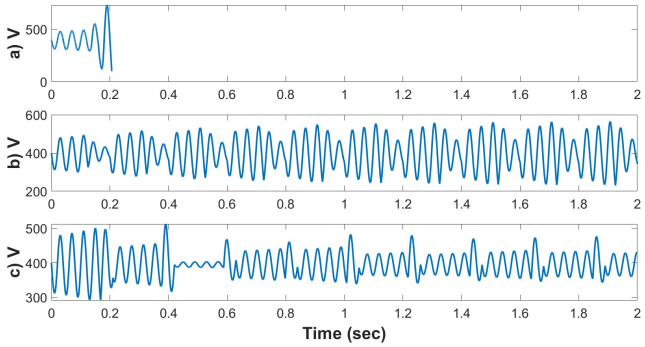

Figure 3.3: Simulation output voltage results for a $P = 5000$ W for demonstration of case a) $T = 0.16$ s, $D = 0.85$ (unstable), case b) $T = 0.2$ s, $D = 0.625$ (marginally meta-stable) and case c) $T = 0.21$ s, $D = 0.8$ (meta-stable).

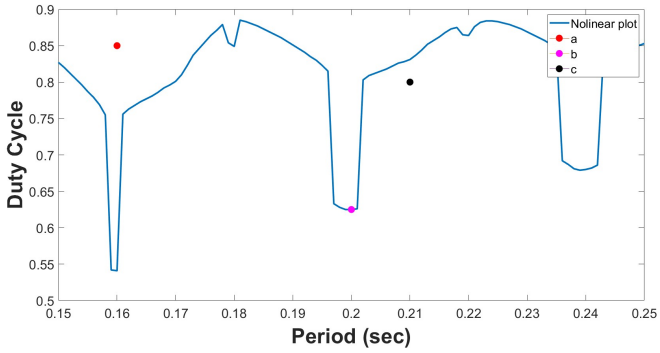

Figure 3.4: Zoomed in area of Figure 3.5 at $P = 5000$ W for demonstration of case a) $T = 0.16$ s, $D = 0.85$ (unstable), case b) $T = 0.2$ s, $D = 0.625$ (marginally meta-stable) and case c) $T = 0.21$ s, $D = 0.8$ (meta-stable).

As predicted by Figure 3.4, the boost voltage in Figure 3.3 (a) shows the bus voltage of the system where the voltage swing magnitude breaks the stability bounds. Figure 3.3 (b) shows the boost voltage limit just at the limit of metastability of the bus voltage, and Figure 3.3 (c) is a meta-stable operation. The abrupt change in response in Figure 3.3 caused by a minor change in the pulse load period. By regulating the period of the pulse, different stability results appear. If this particular example were a pulse power load weapon on an electric ship, such as a laser or an electromagnetic gun, then the designer of the electric ship could set the parameters to make sure that the system operates in a meta-stable condition such as point c) in Figure 3.4 and not point a).

In Figure 3.5, an example of the nonlinear stability plot of the system for a power of 5000 W is shown. The system is stable for the values of duty cycle which are below the blue line and unstable if they are above it. This method requires a large amount

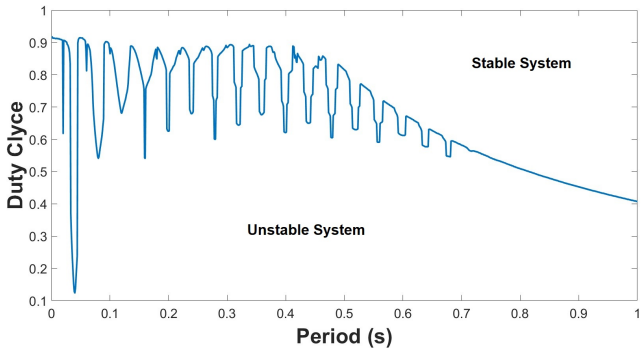

Figure 3.5: Nonlinear stability plot for a power of 5000 W

of computational time to generate the data. As explained above, this method checks the stability of the system by running thousands of simulations varying the duty cycle and period of the pulse power load. Since the stability study must be as accurate as possible, the variation of the duty cycle and period of the pulse power load must be very small. In this investigation the variation of duty cycle and period was defined as 0.001. Smaller the variations the larger will be the time required to determine the stability of the system. The nonlinear data was gathered using the Michigan tech HPC Superior since the time to generate the stability plot for a constant value of power could be around 8 hours. The data obtained by this method is used in the next chapter to generate the equation of the approximation method.

Chapter 4

Analytic Approximation Method

In this chapter, a new method to study the stability behavior of DC-DC systems with pulse loads is introduced. To determine whether the system is stable or not, three conditions were defined. Based on the natural period of the system (T_{nl}) and the maximum time (T_{max}) that the system can support a constant power load without violating the minimum voltage constraint. Where the magnitude of the power load is the power magnitude that we want to determine its stability for a pulse load. These conditions are: a) $T_{max} \geq 1$, b) $10T_{nP} > T_{max} > 1$ and c) $T_{max} \leq 10T_{nP}$. The stability of the system is established by the equations defined in each condition.

To define the equations that describe the stability behavior of the system, the plots generated in Chapter 3.4 for a constant magnitude of power were dived into sections,

based on the noticeable patterns of the plots. From Figure 3.5, it is observable that the plot has patterns, the valleys appear in multiples of the natural period (T_{nP}), after each valley the duty cycle increases until a value close to the maximum duty cycle and then the duty cycle decreases gradually.

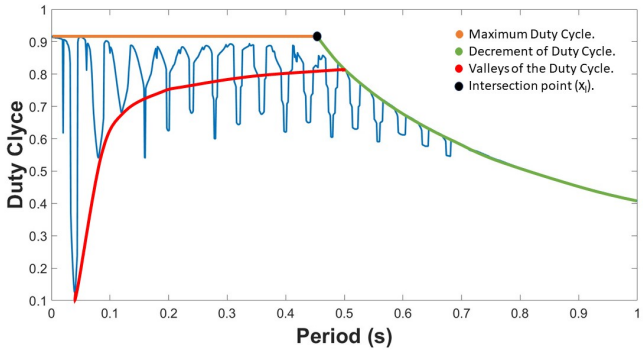

Figure 4.1: Plot approximation sections.

From Figure 4.1 the nonlinear plot for a constant power can be divided into three sections: a) In orange a constant line that defines the maximum duty cycle that makes the system stable, b) In green an equation to define the decreasing behavior of the duty cycle and c) In red an equation to define the valleys the first group of valleys. It is noticeable that the number of valleys of the system, which were defined as n_{max}, repeat in groups. These groups, were defined as max_{repts} and the number of valleys and how many times they repeat will depend on the conditions shown in Table 4.1. By shifting the equation of the read line down and to the right, the same equation predicted the value of the next group of valleys. To determine the coefficient to shift the equation, a relationship between the natural period and different parameters of

the system was defined. From the nonlinear data used to define these equations, a periodic behavior was noticed for values of the valleys. They were presented in multiples of the natural period of the system T_{nP}.

Where the equation used to determine the valleys is used just to determine the values of the valleys at multiples of the natural period of the system and this value is going to be a fixed value on the piece-wise function that defines the approximation. The final equation that defines the stability approximation of the system is a piece-wise function based on the intersection of the equation mentioned above.

First, based on the linearized parameters found using the small-signal analysis in Chapter 3.1, a series of conditions and equations were defined to analyze the metastability of a power electronic system. This method was defined based on the constraint that the system is going to be stable when the minimum voltage of the system is greater than $0.4v_{co}$. To implement this method, a series of parameters and equations must be determined using the equations defined on the small-signal analysis section.

In the system that is being studied the maximum linear power that the system can support P_L was found using a small-signal analysis, this value is not accurate and it is necessary to determine the real maximum constant power load that the system can support. This value is defined as the nonlinear power P_{nl} because it marks the edge of the stability of the system. To determine P_{nl} a system simulation was set up in MATLAB based on (3.1) and (3.2), using a constant power load. A trial-and-error

method was used to find this value, knowing that this value must be less than P_L.

Second, a small simulation was set up based on (3.1) and (3.2) using a constant power load to determine the maximum time that the system can support a constant power load (P), and this time was defined as T_{max}. Where P was the power that was being studied to find its stability behavior for a pulse load. In Figure 4.2 the maximum time that the system can support a constant power of 5000 W is shown, this data was obtained using the parameters of Table 3.1 and a constant power load. At a time of 0.4079 seconds, the voltage of the system is equal to the lower constrain that defines when the system is metastable, being the minimum voltage value constraint equal to 160 V. Therefore, the value of T_{max} is set as 0.4079 for these set of parameters and value of power.

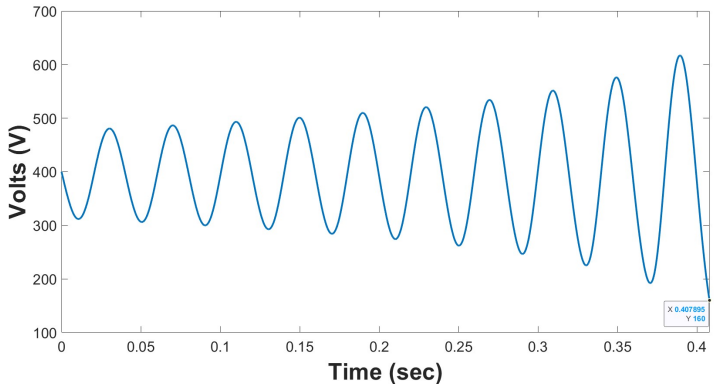

Figure 4.2: T_{max} for a power of 5000 W

Based on the value of T_{max}, three conditions were defined to approximate the behavior of the nonlinear model. These conditions are shown in Table 4.1 where T_{nL} is the

natural period of the system for the value of power that is being studied. These conditions are going to define the set of equations that are going to be used to approximate the stability plot of the system for a specific value of power. Figure 4.3 illustrates how the conditions of the method developed in this investigation defined in Table 4.1, will define the circumstances that determine the equations used to approximate the stability plot of a pulse power load of magnitude (P).

Table 4.1
Approximation plot conditions

Condition	Constrain
1	$T_{max} \geq 1$
2	$10T_{nP} > T_{max} > 1$
3	$T_{max} \leq 10T_{nP}$

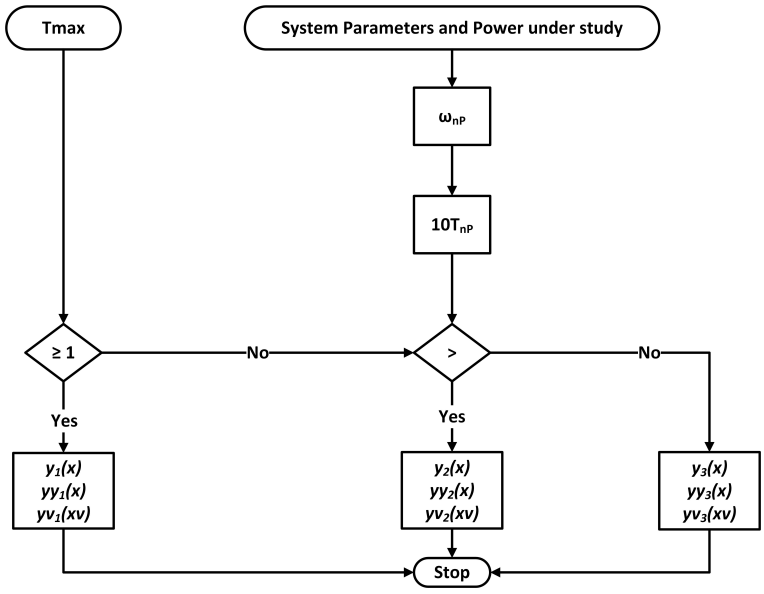

Figure 4.3: Flowchart of the conditions used to define the stability plot of the system for a pulse power load of magnitude (P).

In Figure 4.3, first the maximum time that the system can support the constant power that is being studied T_{max} is calculated, then this value is compared to 1 and if it is greater or equal than 1 then the first group of equations will define the stability plot of the system. If T_{max} is not greater or equal to 1, then using the parameters from the small-signal analysis the natural period of the system will be calculated and multiplied by 10 and this value will be compared with T_{max} and if it is greater than T_{max} then a second group of equations will define the stability plot of the system if not them a third group of equations will define the stability plot of the system.

Depending on the condition, different equations will define the behavior of the stability of the system based on the linear parameters from the linearized system. These equations were found using the MATLAB plot fit tool to determine the equation that describes best the behavior of the nonlinear plots from Chapter 3.4.

The equations that were more accurate in describing the behavior of the sections used to define each part of the nonlinear plot, as show in Figure 4.1, were the equation of a straight line of the form $y(x) = mx + b_1$ and a power equation of the form $y(x) = a_1 x^{b_2} + c_1$. Once the best fits found using the MATLAB plot fit tool were identified then the relationships that better describe the coefficients of the fit equations were determined defining relationships between the linear and nonlinear parameters of the system.

4.1 Condition 1: $T_{max} \geq 1$

In this section, the equations that described the behavior of the approximated stability plot of the first condition defined in Table 4.1 are presented. In this condition, when the T_{max} value for a specific value of power is greater than 1, the equation that defines the maximum duty cycle, which correspond to the orange line in Figure 4.1, is defined as

$$y_1(x) = \frac{P_{nl}}{P} \tag{4.1}$$

where P is the power that is being studied and P_{nl} is the maximum constant power load that the system can support. The equation that defines the decreasing part of the duty cycle, which correspond to the green line in Figure 4.1, is defined as

$$yy_1(x) = a + \frac{(\frac{P_{nl}}{P} - a)(x-1)}{a-1}. \tag{4.2}$$

where a is

$$a = 1 - \frac{T_{np}P_{nl}}{P} \tag{4.3}$$

The X-intersection point (x_I) between y and yy, which correspond to black dot in Figure 4.1, is used to determine the number of valleys of the approximation plot. The

equations that define the number of valleys (n_{max}) are

$$T_{max1} = \frac{Tmax}{T_n} \tag{4.4}$$

$$n_{max1} = \frac{x_I}{T_{nP}} - i_{L0_{P=P_{nl}}} + \frac{P_{nl}}{\omega_{nP}} - \frac{x_I}{\omega_{nP}{T_{nP}}^2} \tag{4.5}$$

$$n_{max} = \frac{2T_{max1}}{n_{max1}} + \frac{P_{nl}}{P} \tag{4.6}$$

where the value of n_{max} is a integer. These equations were derived by trying different combinations of the small-signal and nonlinear parameters from Chapter 3.2 and 3.4.

The valleys will repeat $n_{max} - 1$ times, and the equations that define their values, which correspond to the red line in Figure 4.1, are:

If $n_{max} \leq T_{nP}x$, the equation that defines the valleys is

$$yv_1(xv) = \frac{P_{nl}}{P} - \frac{T_{nP}}{(xv + \frac{T_{nP}}{8})^{\frac{P_{nl}}{P}}} \tag{4.7}$$

If $n_{max} < T_{nP}x \leq 2n_{max} - 1$, the equation that defines the valleys is

$$yv_1(xv) = \frac{P_{nl}}{P} - \frac{T_{nP}}{2} - \frac{T_{nP}}{(xv - \frac{n_{max}T_{nP}}{2})^{\frac{P_{nl}}{P}}} \tag{4.8}$$

If $m_i n_{max} - l_i < T_{nP} x \leq m_{i+1} n_{max} - l_{i+1}$, the equation that defines the valleys is

$$yv_1(xv) = \frac{P_{nl}}{P} - \frac{T_{nP}}{2} - \frac{T_{nP}}{(xv - \frac{n_{max} T_{nP}}{n})^{\frac{P_{nl}}{P}}} \tag{4.9}$$

where $m \in \{3, 4, 5, \ldots, m_{i+1}\}$, $l \in \{3, 6, 10, \ldots, (m_{i-1} + l_{i-1})\}$ and $l_0 = 3$, n $\in \{8/8, 13/8, 19/8, \ldots, (n_{i-1} + m_{i+2})\}$ and $n_0 = 1$, $xv = iT_{nP}$ and $i \in \mathbb{Z}^+$. The final equation that approximates the stability behavior of the system is

$$ys(x) = \begin{cases} y_1(x) & if \quad x < x_I \quad and \quad x \notin \left[N_i T_{nP} - \frac{TnP}{4}, \quad N_i T_{nP} - \frac{TnP}{4} \right] \\ yv_1(x) & if \quad x < x_I \quad and \quad x \in \left[N_i T_{nP} - \frac{TnP}{4}, \quad N_i T_{nP} - \frac{TnP}{4} \right] \\ yy_1(xv) & if \quad x \geq x_I \end{cases} \tag{4.10}$$

where $N_i \in \{1, 2, 3, \ldots, N_(i + 1)\}$ and it will increase when $xT_{nP} > N_i T_{nP}$.

4.2 Condition 2: $10T_{nl} > T_{max} > 1$

In this section, the equations that described the behavior of the approximated stability plot of the second condition defined in Table 4.1 are presented. In this condition, when the T_{max} value for a specific value of power is between ten times the natural period of the system and 1 second, the equation that defines the maximum duty cycle, which

correspond to the orange line in Figure 4.1, is defined as

$$y_2(x) = \frac{P_{nl}}{P} \tag{4.11}$$

where P is the power that is being studied and P_{nl} is the maximum nonlinear power that the system can support.

The equation that defines the decreasing part of the duty cycle, which correspond to the green line in Figure 4.1, is defined as

$$yy_2(x) = ax^b \tag{4.12}$$

where a and b are

$$a = T_{max} \tag{4.13}$$

$$b = -1 - \frac{T_{nP}(1 - P_{nl})}{P} \tag{4.14}$$

The X-intersection point (x_I) between y and yy, which correspond to black dot in Figure 4.1, is used to determine the number of valleys of the approximation plot. The equations that define the number of valleys (n_{max}) and how many times they repeat max_{repts} are

If $P < P_L$, the parameters that define the valleys are

$$T_{max1} = \frac{T_{max}}{T_n} \tag{4.15}$$

$$n_{max1} = \frac{x_I}{T_{nP}} - i_{L0_{P=P_{nl}}} + \frac{P_{nl}}{\omega_{nP}} - \frac{x_I}{\omega_{nP}{T_{nP}}^2} \tag{4.16}$$

$$n_{max1} = \frac{2T_{max1}}{n_{max1}} \tag{4.17}$$

$$max_{repts} = \frac{x_I}{\omega_{nP}{T_{nP}}^2} + 1 \tag{4.18}$$

If $P \geq P_L$, the parameters that define the valleys are

$$T_{max1} = \frac{T_{max}}{T_n} \tag{4.19}$$

$$max_{repts} = \frac{x_I}{\omega_{nP}{T_{nP}}^2} + 1 \tag{4.20}$$

$$n_{max} = \frac{T_{max1}}{max_{repts}} \tag{4.21}$$

where the values of max_{repts} and n_{max} are round values and determine the number of times that each set of valleys (n_{max}) are going to repeat. Each set of valleys will repeat max_{repts_i} times and it will have n_{max_i} valleys. These equations were derived by trying different combinations of the small-signal and nonlinear parameters from Chapter 3.2 and 3.4. The equations that define the values of the valleys, which correspond to the red line in Figure 4.1, are defined as

If $n_{max} \leq T_{nP}x$, the equation that defines the valleys is

$$yv_2(xv) = \frac{P_{nl}}{P} + T_{nP} - \frac{T_{nP}}{xv^{\frac{P_{nl}}{P}}} \tag{4.22}$$

If $n_{max} < T_{nP}x \leq n_{max}max_{repts}$, the equation that defines the valleys is

$$yv_2(xv) = \frac{P_{nl}}{P} + (1-n)T_{nP} - \frac{T_{nP}}{(xv - nT_{nP})^{\frac{P_{nl}}{P}}} \tag{4.23}$$

If $n_{max_i} < T_{nP} \leq n_{max_i}max_{repts_i}$, the equation that defines the valleys is

$$yv_2(xv) = \frac{P_{nl}}{P} + (1-n)T_{nP} - \frac{T_{nP}}{(xv - nT_{nP})^{\frac{P_{nl}}{P}}} \tag{4.24}$$

Where $n_{max_i} = n_{max_{i-1}} - 1$ and $n_{max_0} = n_{max}$, $max_{repts_i} = max_{repts_i} - 1$ and $max_{repts_o} = max_{repts}$ when $xv > n_{max_i}max_{repts_i}$, and $n \in \{2, 5/2, 7/2, \ldots, (n_i + 1/2)\}$, $xv = iT_{nP}$ and $i \in \mathbb{Z}^+$. The final equation that approximates the stability behavior of the system is

$$ys(x) = \begin{cases} y_2(x) & if \quad x < x_I \quad and \quad x \notin \left[N_iT_{nP} - \frac{TnP}{4}, \quad N_iT_{nP} - \frac{TnP}{4}\right] \\ yv_2(x) & if \quad x < x_I \quad and \quad x \in \left[N_iT_{nP} - \frac{TnP}{4}, \quad N_iT_{nP} - \frac{TnP}{4}\right] \\ yy_2(xv) & if \quad x \geq x_I \end{cases} \tag{4.25}$$

where $N_i \in \{1, 2, 3, \ldots, N_(i+1)\}$ and it will increase when $xT_{nP} > N_iT_{nP}$.

4.3 Condition 3: $T_{max} \leq 10T_{nl}$

In this section, the equations that described the behavior of the approximated stability plot of the second condition defined in Table 4.1 are presented. In this condition, when the T_{max} value for a specific value of power is less or equal to ten times the natural period of the system, the equation that defines the maximum duty cycle, which correspond to the orange line in Figure 4.1, is defined as

$$y_3(x) = \frac{P_{nl}}{P} \tag{4.26}$$

where P is the power that is being studied and P_{nl} is the maximum nonlinear power that the system can support. The equation that defines the decreasing part of the duty cycle, which correspond to the green line in Figure 4.1, is defined as

$$yy_3(x) = ax^b \tag{4.27}$$

where

If $T_{max} \geq 6T_{nP}$ and $\frac{p_{nl}}{\omega_n} \notin \left[i_{Lo_{P=0}} \quad i_{Lo_{P=P_{nl}}} \right]$ and $\frac{P_{nl}}{\omega_n} > \omega_n T_{nP}$, the values of a and b are

$$a = T_{max} - \frac{T_{nP}P_{nl}}{P} \tag{4.28}$$

$$b = -1 + \frac{T_{nP}(1 - P_{nl})}{P} \tag{4.29}$$

If $T_{max} < 6T_{nP}$ and $\frac{p_{nl}}{\omega_n} \notin \begin{bmatrix} i_{Lo_{P=0}} & i_{Lo_{P=P_{nl}}} \end{bmatrix}$ and $\frac{P_{nl}}{\omega_n} > \omega_n T_{nP}$, the values of a and b are

$$a = T_{max} \tag{4.30}$$

$$b = -1 + \frac{T_{nP}(1 - P_{nl})}{P} \tag{4.31}$$

If $T_{max} < 6T_{nP}$ and $\frac{p_{nl}}{\omega_n} \in \begin{bmatrix} i_{Lo_{P=0}} & i_{Lo_{P=P_{nl}}} \end{bmatrix}$ and $\frac{P_{nl}}{\omega_n} > \omega_n T_{nP}$, the values of a and b are

$$a = T_{max} - \frac{T_{nP} P_{nl}}{P} \tag{4.32}$$

$$b = -1 - \frac{T_{nP}(1 - P_{nl})}{P} \tag{4.33}$$

If $T_{max} < 6T_{nP}$ and $\frac{p_{nl}}{\omega_n} \notin \begin{bmatrix} i_{Lo_{P=0}} & i_{Lo_{P=P_{nl}}} \end{bmatrix}$ and $\frac{P_{nl}}{\omega_n} < \omega_n T_{nP}$, the values of a and b are

If $a > 0$ and $T_{max} < 4T_{nP}$

$$a = \frac{T_{max}}{1 - T_{nP}} - T_{nP} \tag{4.34}$$

$$b = T_{nP} - \frac{\omega_{nP}}{\omega_{no}} \tag{4.35}$$

If $a \leq 0$ and $T_{max} < 4T_{nP}$

$$a = \frac{T_{max}}{1 - T_{nP}} \tag{4.36}$$

$$b = T_{nP} - \frac{\omega_{nP}}{\omega_{no}} \tag{4.37}$$

If $a \leq 0$ and $T_{max} \geq 4T_{nP}$

$$a = T_{max} \tag{4.38}$$

$$b = T_{nP} - \frac{\omega_{nP}}{\omega_{no}} \tag{4.39}$$

The X-intersection point (x_I) between y and yy, which correspond to black dot in Figure 4.1, is used to determine the number of valleys of the approximation plot. The equations that define the number of valleys (n_{max}) and how many times they repeat max_{repts} are

If $\frac{P_{nl}}{\omega_{nl}} < \omega_{nl} T_{nP}$

$$n_{max1} = \frac{x_I}{T_{nP}} - 1 \tag{4.40}$$

$$max_{repts} = \frac{x_I}{\omega_{nP} {T_{nP}}^2} + 1 \tag{4.41}$$

$$n_{max} = \frac{n_{max1}}{max_{repts}} \tag{4.42}$$

in this case, the values of n used in the equation to determine the value of the valleys are going to start at $n = 2$.

If $\frac{P_{nl}}{\omega_{nl}} \geq \omega_{nl} T_{nP}$

$$T_{max1} = \frac{T_{max}}{T_{nP}} \tag{4.43}$$

$$max_{repts} = \frac{x_I}{\omega_{nP} {T_{nP}}^2} \tag{4.44}$$

$$n_{max} = \frac{T_{max1}}{max_{repts}} - 2 \tag{4.45}$$

in this case, the values of n used in the equation to determine the value of the valleys are going to start at $n = 1$. Each set of valleys will repeat max_{repts_i} times and it will have n_{max_i} valleys. These equations were derived by trying different combinations of the small-signal and nonlinear parameters from Chapter 3.2 and 3.4. The equations that define the value of the valleys, which correspond to the red line in Figure 4.1, are

If $n_{max} \leq T_{nP}x$, the equation that defines the valleys is

$$yv_3(xv) = \frac{P_{nl}}{P} + T_{nP} - \frac{T_{nP}}{xv^{\frac{P_{nl}}{P}}} \tag{4.46}$$

If $n_{max} < T_{nP}x \leq 2n_{max} - 1$, the equation that defines the valleys is

$$yv_3(xv) = \frac{P_{nl}}{P} + T_{nP} - (3nT_{nP}) - \frac{T_{nP}}{xv - nT_{nP}{}^{\frac{P_{nl}}{P}}} \tag{4.47}$$

If $m_i n_{max} - l_i < T_{nP}x \leq m_{i+1}n_{max} - 1l_{i+1}$, the equation that defines the valleys is

$$yv_3(xv) = \frac{P_{nl}}{P} + T_{nP} - (3nT_{nP}) - \frac{T_{nP}}{xv - nT_{nP}{}^{\frac{P_{nl}}{P}}} \tag{4.48}$$

where $m \in \{3, 4, 5, \ldots, m_{i+1}\}$, $l \in \{3, 6, 10, \ldots, (m_{i-1} + l_{i-1})\}$ and $l_0 = 3$, $n =$

$n_{i-1} + 1/2$ and $n_0 = 1$ or 2, depending on the conditions defined above, $xv = iT_{nP}$ and $i \in \mathbb{Z}^+$. The final equation that approximates the stability behavior of the system is

$$ys(x) = \begin{cases} y_3(x) & if \quad x < x_I \quad and \quad x \notin \left[N_i T_{nP} - \frac{TnP}{4}, \quad N_i T_{nP} - \frac{TnP}{4} \right] \\ yv_3(x) & if \quad x < x_I \quad and \quad x \in \left[N_i T_{nP} - \frac{TnP}{4}, \quad N_i T_{nP} - \frac{TnP}{4} \right] \\ yy_3(xv) & if \quad x \geq x_I \end{cases} \tag{4.49}$$

where $N_i \in \{1, 2, 3, \ldots, N_{i+1}\}$ and it will increase when $xT_{nP} > N_i T_{nP}$. When the values of n_{max} or max_{repts} are negatives or equal to zero these mean that there are no valleys for the particular value of power that is being studied.

To generate the data of the approximation method, a MATLAB code was developed using the linear parameters of Chapter 3.2 and the conditions and equations described above. The code used to generate this data is shown in the Appendix C. This data requires less computational time than the nonlinear data from Chapter 3.4 and this was the objective of this investigation, the results will be discussed in the next Chapter.

Chapter 5

Comparison Nonlinear, Floquet theory and Approximation Methods

In this section, the results of the nonlinear system are compared with the results of the approximation method defined in this thesis and with the Floquet theory stability study. To generate the results used in this work multiple simulations were done using MATLAB, Simulink, and Michigan Tech's High-Performance Computer (Superior) due to the extensive time that is required to generate each stability plot for the nonlinear system and for the Floquet theory method. The parameters in Table 3.1 were the parameters used to generate the nonlinear, Floquet, and approximation data

used to validate the results of this investigation.

From Figure 5.1 and Figure 5.2, it is noticeable that the approximated map and the nonlinear map have similar behaviors. Both have valleys and the number of valleys decreases when the power increases. Also, in both the nonlinear model and the approximation model when the power increases the maximum duty cycle for higher periods of time decreases.

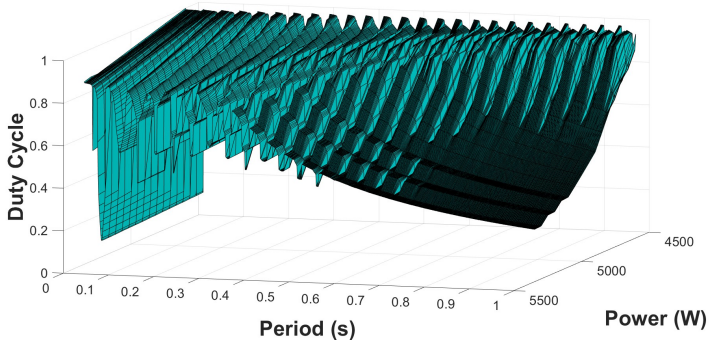

Figure 5.1: Nonlinear stability surface.

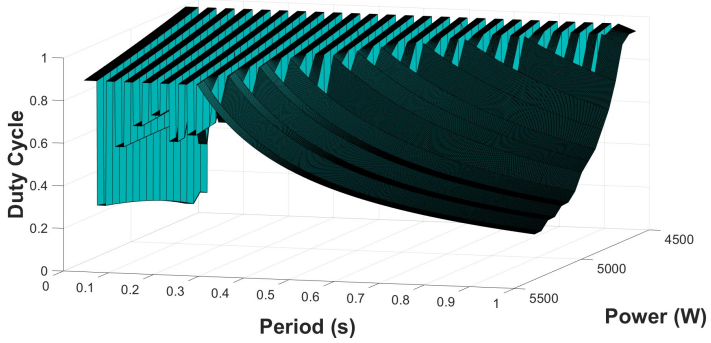

Figure 5.2: Approximated stability surface.

From Figure 5.3, the stability behavior of the system for a constant period. Where it is noticeable that Floquet theory and the approximation method have similar behavior for specific values of power. However, both cases are overestimating the stability of the system even though there are cases where the approximation method gives less overestimated results and vice versa.

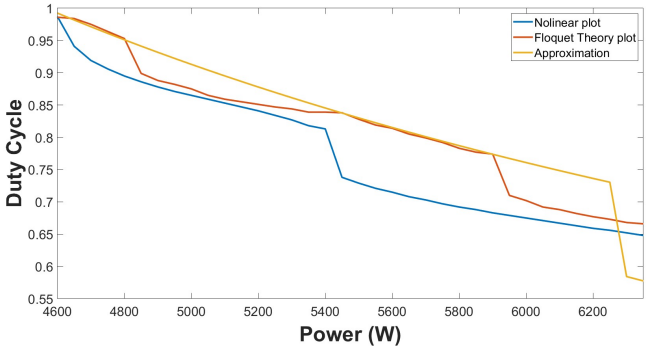

Figure 5.3: Stability behavior of the system for a constant period of time equal to 0.1 sec.

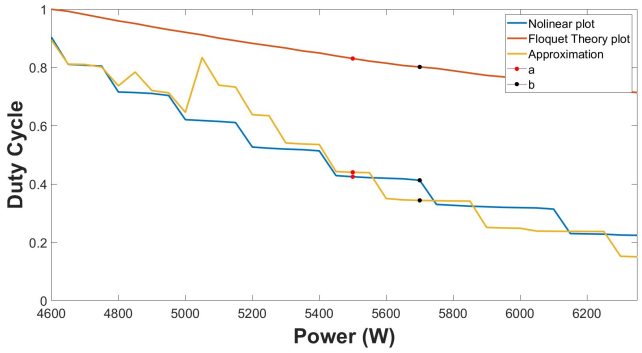

Figure 5.4: Stability behavior of the system for a constant period of time equal to 0.4 sec.

From Figure 5.4, it is observable that when the period increases the Floquet method

is not accurate anymore. However, the approximation method proposed in this investigation is able to describe the stability behavior of the system. Though there are cases where the approximation method overestimates the stability of the system, this case gives more accurate results than the Floquet method.

Table 5.1
Approximation plot conditions

Method	**Power (W)**	**Duty cycle**	**Period (sec)**	**case**
Nonlinear	5500	0.425	0.4	**a**
Floquet	5500	0.831	0.4	**a**
Approximation	5500	0.440	0.4	**a**
Nonlinear	5700	0.413	0.4	**b**
Floquet	5700	0.802	0.4	**b**
Approximation	5700	0.344	0.4	**b**

The values of cases a and b from Figure 5.4 were used to validate how accurate Floquet and the approximation method are. The values shown in Table 5.1 were used to generate the results of Figure 5.5 and Figure 5.6. From these figures it is noticeable that the Floquet method only yields overestimating stability behaviors of the system while the approximation method is able to give a better prediction of the stability of the system in certain cases. In Figure 5.6 the duty cycle obtain from the approximation method was more accurate than the value obtained using Floquet.

Figure 5.5 and Figure 5.6 validated that the approximation method gives better stability predictions than the Floquet theory. Though the Floquet method is a method

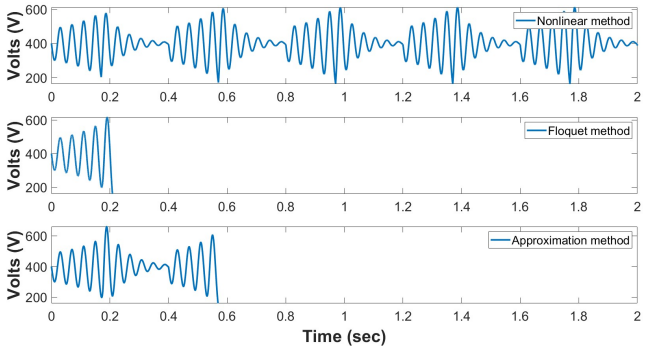

Figure 5.5: Stability behavior of the system for a power of 5500 W, a period of 0.4 sec, and the different values of duty cycle (case *a*).

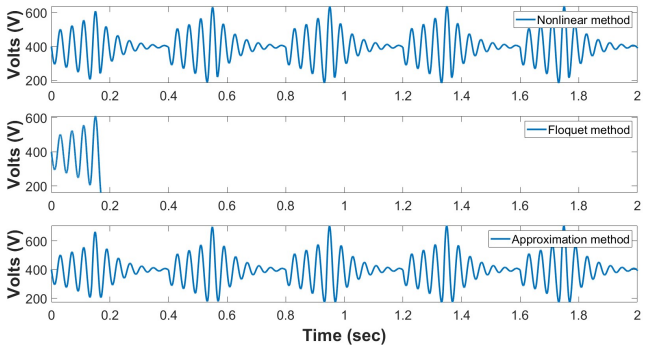

Figure 5.6: Stability behavior of the system for a power of 5700 W, a period of 0.4 sec, and the different values of duty cycle (case *b*.)

used to determine the stability of time-varying systems, it is not able to give an accurate approximation for the stability of the system for pulse loads.

Table 5.2
Error and simulation time for a power of 4600 W

Method	RMSE	Simulation time (sec)
Nonlinear	0	4.7959×10^4
Approximation	0.1415	44.1218

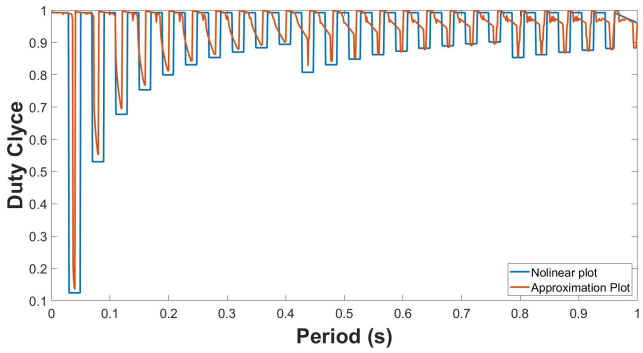

Figure 5.7: Stability behavior of the system for a power of 4600 W.

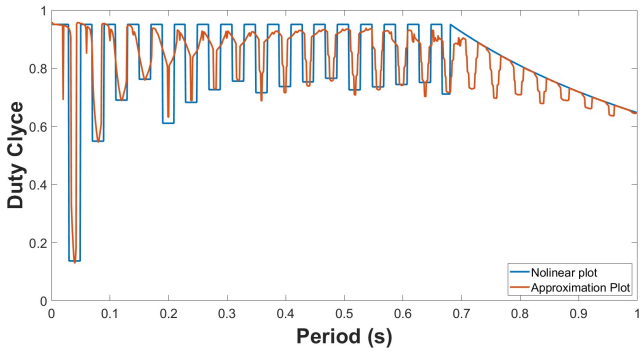

Figure 5.8: Stability behavior of the system for a power of 4800 W.

Table 5.3
Error and simulation time for a power of 4800 W

Method	RMSE	Simulation time (sec)
Nonlinear	0	1.4134×10^5
Approximation	0.1187	43.0165

Table 5.4
Error and simulation time for a power of 6000 W

Method	RMSE	Simulation time (sec)
Nonlinear	0	3.3749×10^5
Approximation	0.0967	48.1451

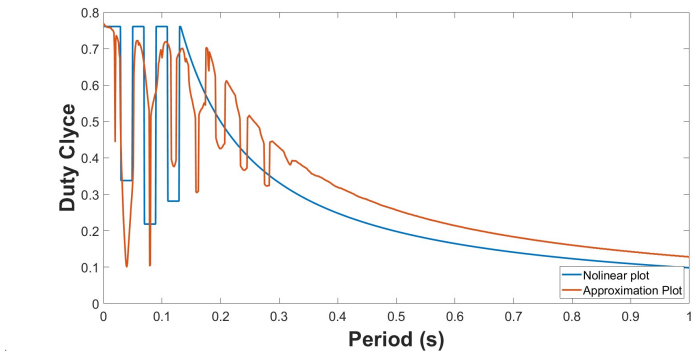

Figure 5.9: Stability behavior of the system for a power of 6000 W.

From Figures 5.1 and 5.2 by observing the behavior of the system for pulses power load with constant power magnitude as observe in Figures 5.7, 5.8, and 5.9 the comparison between the nonlinear and the approximation method is shown. In Tables 5.2,5.3 and 5.4, the values of the root mean square error (RMSE) is shown, this value represents the error between the nonlinear data and approximation data.

This error method measures the accuracy of each method, the smaller the value of the RMSE the better the approximation method is. The approximation method described in this investigation was able to track in some way the behavior of the nonlinear data. Also, the value of the RMSE error for the approximation method decreases when the value of the power increases. The time required to generate the stability behavior of the system (simulation time) is indicated in Tables 5.2,5.3 and 5.4. It is noticeable that the nonlinear method required a lot of time, at least several hours to generate the data for the stability plot for just one value of power, which means that to generate a stability map a lot of time will be required. However, the approximation method

is able to generate the stability data accurately and in less than a minute. This will allow the stability study of the DC-DC systems with pulse loads much faster than the Hamiltonian Surface Shaping Method described in [12] and more accurately than a small signal analysis.

Chapter 6

Conclusion and Future Work

This investigation has presented a new approach to study the stability of DC-DC system with a pulse power load, which had the goal to decrease the time that would take to generate the data needed for the stability analysis. This new method has proven to be much faster than the Hamiltonian shaping method and more accurate than the Floquet theory stability analysis. The results show that the method presented in this work was able to generate an accurate approximation of the full nonlinear method in a small time. Which is convenient at the moment of designing a DC-DC system. By using this method, the designer will be able to have an accurate idea about the stability behavior of the system in just a few seconds and by doing this the controller of the system does not need to be over-dimensioned, or maybe there is no need of a controller because the system could be stable for the particular values of pulse load

under study.

The future scope of this work is vast. This method could be implemented as a starting point for the designing process of a DC-DC system with pulse loads such as electric ships. Also, a machine learning algorithm can be developed using as a starting point the relationships defined in this work and this could increase the accuracy of the stability analysis.

References

[1] N. Doerry and K. Moniri, "Specifications and standards for the electric warship," *IEEE Electric Ship Technologies Symposium (ESTS)*, pp. 21–28, 2013.

[2] J. D. Stevens, D. F. Opila, E. S. Oh, and E. L. Zivi, "All-electric warship load demand model for power and energy system analysis using exogenously initiated threats," *2017 IEEE Electric Ship Technologies Symposium (ESTS)*, pp. 486–492, 2017.

[3] J. Neely, L. Rashkin, M. Cook, D. Wilson, and S. Glover, "Evaluation of power flow control for an all-electric warship power system with pulsed load applications," *IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 3537–3544, 2016.

[4] A. Riccobono and E. Santi, "Comprehensive Review of Stability Criteria for DC Power Distribution Systems," *IEEE Transactions on Industry Applications*, vol. 50, no. 5, pp. 3525–3535, 2014.

[5] W. Inam, J. A. Belk, K. Turitsyn, and D. J. Perreault, "Stability, control, and power flow in ad hoc DC microgrids," *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*, pp. 1–8, 2016.

[6] M. N. Marwali, J. W. Jung, and A. Keyhani, "Stability Analysis of Load Sharing Control for Distributed Generation Systems," *IEEE Transactions on Energy Conversion*, vol. 22, no. 3, pp. 737–745, 2017.

[7] S. Kulkarni and S. Santoso, "Impact of pulse loads on electric ship power system: With and without flywheel energy storage systems," *2009 IEEE Electric Ship Technologies Symposium*, pp. 568–573, 2009.

[8] V. Salehi, B. Mirafzal, and O. Mohammed, "Annual Conference on IEEE Industrial Electronics Society," *2009 IEEE Electric Ship Technologies Symposium*, pp. 3353–3358, 2010.

[9] Klausmeier, "C.A. Floquet theory: a useful tool for understanding nonequilibrium dynamics. Theor Ecol 1," *IEEE Electric Ship Technologies Symposium*, p. 153–161, 2018.

[10] J. A. Álvarez Martín, J. R. Melgoza, and J. J. R. Pasaye, "Exact steady state analysis in power converters using Floquet decomposition," *2011 North American Power Symposium*, pp. 1–7, 2011.

[11] L. N. Domaschk, R. E. H. A. Ouroua, O. E. Bowlin, and W. B. Colson, "Coordination of Large Pulsed Loads on Future Electric Ships," *IEEE Transactions on Magnetics*, vol. 43, no. 1, pp. 450–455, 2007.

[12] W. W. Weaver, R. D. Robinett, D. G. Wilson, and R. C. Matthews, "Metastability of Pulse Power Loads Using the Hamiltonian Surface Shaping Method," *IEEE Transactions on Energy Conversion*, vol. 32, no. 2, pp. 820–828, 2017.

[13] P. S. Kundur and O. P. Malik, *Power System Stability and Control*. McGraw Hill, 2 ed., 2022.

[14] K. C.A, "Floquet theory: a useful tool for understanding nonequilibrium dynamics," *Theor Ecol*, p. 153–161, 2008.

[15] 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 Transactions on Power Electronics*, vol. 5, no. 2, pp. 182–190, 1990.

[16] R. C. Matthews, L. J. Rashkin, S. F. Glover, and N. H. Doerry, "Stabilization of Generator Frequency Under Pulsed Load Condition Using Regenerative Propeller Braking," *IEEE Electric Ship Technologies Symposium (ESTS)*, pp. 1–6, 2021.

[17] M. Hosseinzadehtaher, A. Khan, M. Easley, M. B. Shadmand, and P. Fajri, "Self-Healing Predictive Control of Battery System in Naval Power System with Pulsed Power Loads," *IEEE Transactions on Energy Conversion*, vol. 36, no. 2, pp. 1056–1069, 2021.

[18] W. W. Weaver, M. M. Bijaieh, R. D. Robinett, and D. G. Wilson, "Energy Storage Baseline Requirements for Pulsed Power Loads," *IEEE Electric Ship Technologies Symposium (ESTS)*, pp. 52–59, 2019.

[19] "MATLAB," *[Online] Available: https://www.mathworks.com/.*

[20] "Simulink," *[Online] Available: https://www.mathworks.com/.*

[21] P. S. Kundur and O. P. Malik, *Power System Stability and Control*. Philadelphia, PA: McGraw Hill, 2 ed., 2022.

Appendix A

Floquet data code

```
clearvars

% Paremters

RL = 0.1;

L = 10*10^-3;

c = 1*10^-3;

Rc = 50;

lambda=0.5;

l=lambda;

u =200;

syms x1o x2o P
```

```
equ=[(u-RL*x1o -l*x2o)/L== 0,(l*x1o-x2o/Rc-P/x2o)/c== ↩
   0];
[x1o,x2o]=solve(equ,[x1o,x2o])
x20 = x2o(2,1);


syms x1 x2  P
f1 = (u - RL*x1 - lambda*x2)/L;
f2 = (lambda*x1 - x2/Rc - P/x2)/c;
a = [diff(f1,x1) diff(f1,x2); diff(f2,x1) diff(f2,x2)];
a = subs(a,x2,x20)


% Simualtion Code


conds = [1 0 0 1];
k = 1;
dc = 0 : 0.001 : 1;
T = 0.001 : 0.001 : 1;
POWER = p;
%{
for jj = 1  : (length(T))
    floquet_data(jj,:) = [0 0 0];
```

```
end
%}
tic
for ii = 1 : length (T)
    tspan = [0 T(ii)];
    for i = 1 : length(dc)
        f= (2*pi)/T(ii);
        dc1 =100-100*dc(i);
        dc2 = dc1/100;
        conds = [1 0 0 1];
        solu = ode15s(@(t,z) odefcn(t,z,a,f,dc1,POWER),↩
           tspan,conds);
        y = deval(solu,T(ii));
        zz = [y(1,1) y(2,1); y(3,1) y(4,1)];
        f_multipliers = eig(zz);
        f_exponents = log(f_multipliers)/T;
        f_max_exponent = max(real(f_exponents));
        if f_max_exponent > 0
        else
           floquet_data_p(k,:) = [POWER dc2 T(ii)]
           k = k + 1;
```

```
            break
        end
    end
end
time_p = toc;
save("floq_data_p.mat");
exit


function dzdt = odefcn(t,z,a,f,dc1,POWER)
    P = 0.5*(POWER+POWER*square(f*t,dc1));
    a = subs(a,P);
    dzdt = zeros(4,1);
    dzdt(1) = a(1,1)*z(1) + a(1,2)*z(3);
    dzdt(2) = a(1,1)*z(2) + a(1,2)*z(4);
    dzdt(3) = a(2,1)*z(1) + a(2,2)*z(3);
    dzdt(4) = a(2,1)*z(2) + a(2,2)*z(4);
end
```

Appendix B

Nonlinear data code

```
clearvars

%close('all')

RL=0.1;

L=10*10^-3;

c=1*10^-3;

Rc=50;

lambda=0.5;

l=lambda;

ST=1E-5;

% Initial conditions
```

```
Stime = 10;
P=0;
syms x1o u0
x20=400;
equ=[(u0-RL*x1o -l*x20)/L== 0,(l*x1o-x20/Rc-P/x20)/c== ↩
   0];
[x10,u0]=solve(equ,[x1o,u0]);
x10=double(x10);
u0=double(u0);
minV = x20 - x20*0.6;
% Period and Dc variable for a constat Power
Power = 4600;
name = sprintf('NL_%d.mat',Power);
T = 0.001 : 0.001 : 1;
dc = 0 : 0.001 : 1;

for ll = 1 : length(Power)*length(T)
    data(ll,:) = [0 0 0];
end

% New code
```

```
tic
k = 1;
for j = 1 : length(T)
    for i = 1 : length(dc)
        Simulink.sdi.clear;
        iii = 1 + length(dc) - i;
        test = sim('constantPower.slx');
        vData = (test.logsout.getElement('Vc').Values.↩
           Data);
        vmax = max(vData);
        vmin = min(vData);
        if (vmin>=minV)
            data(k,:) = [Power dc(iii) T(j)];
            [Power dc(iii) T(j)]
            k = k + 1;
            break
        end
    end
end
timeNL = toc;
save(name)
```

```
exit
```

Appendix C

Approximation data code

```
% System Parameters


clearvars


RL = 0.1;

L = 10*10^-3;

c = 1*10^-3;

Rc = 50

lambda=0.5

l=lambda
```

```
ST=1E-5
x20=400                          % Voltage of the load
POWER = 5000

% Linearization of the system

syms x1 x2 u P s
f1 = (u - RL*x1 - lambda*x2)/L;
f2 = (lambda*x1 - x2/Rc - P/x2)/c;
a = [diff(f1,x1) diff(f1,x2); diff(f2,x1) diff(f2,x2)];
a = subs(a,x2,x20);
b = [ diff(f1,u);diff(f2,u)];
c1 = [0 1];
d = 0;
sys= c1*inv(s*eye(2)-a)*b+d;
polos = real(poles(sys));
value = polos(1,1)==0;

PL = double(round(solve(value,P))) % Maximun linearized ↩
   Power that the system can support
```

```
% Voltage limiits to define the stability


maxV = x20 + x20*0.6;

minV = x20 - x20*0.6;


%{

% Nonlienar Power value

T=1;

dc=1;

j = 0 : 1 : 10000;

P = 4800;

for j =1 : P

    P = Plinear - j

    Simulink.sdi.clear;

    test = sim("new_parameters.slx");

    vData = (test.logsout.getElement('Vc').Values.Data);

    vmax = max(vData);

    vmin = min(vData);

    if(vmax<=maxV && vmin>minV)

       Pnl = P

       break
```

```
    end
end
%}

% Initial conditions and linearized system values for Pnl
Pnl = 4565
P = Pnl;
syms x1o u0
equ=[(u0-RL*x1o -l*x20)/L== 0,(l*x1o-x20/Rc-P/x20)/c== 0];
[x10,u0]=solve(equ,[x1o,u0]);
x10=double(x10);
u0=double(u0);
ilPnl = x10;
VsP = u0;
Vco = x20;
wn = sqrt(RL/(c*L*Rc) - (P*RL)/(c*L*Vco^2) + lambda^2/(c*L));     % Linearized Natural Frequency (rad/sec)
zeta = (RL/L + 1/(c*Rc) - P/(c*Vco^2))/(2*wn);
                        % Linearized Damping Ratio
```

```
Q = 1/(2*zeta); ↩
                                                    % ↩
   Linearized Quality Factor of the System
fn = wn/(2*pi); ↩
                                                    % ↩
   Linearized Frequncy of the System (Hz)
tn = 1/fn;      % Linearized Fundamental Period (sec)
Relation = wn*tn;

% Initial conditions and linearized values for a Power ↩
   of P=0

P=0;
syms x1o u0
equ=[(u0-RL*x1o -l*x20)/L== 0,(l*x1o-x20/Rc-P/x20)/c== ↩
   0];
[x10,u0]=solve(equ,[x1o,u0]);
x10=double(x10);
u0=double(u0);
il0 = x10
Vs0 = u0;
```

```
Vco = x20;
wn0 = sqrt(RL/(c*L*Rc) - (P*RL)/(c*L*Vco^2) + lambda^2/(↩
   c*L));
zeta0 = (RL/L + 1/(c*Rc) - P/(c*Vco^2))/(2*wn0);
Q0 = 1/(2*zeta0);
fn0 = wn0/(2*pi);
tn0 = 1/fn0;
Relation0 = wn0*tn0;

% Simulation to determine Tmax

Simulink.sdi.clear;
Stime=10;
Power for Approx Plot
zz =1;
tic
k = 1;
xd = 0.001 : 0.001 : 1;
N = 1;
P=6000;                          % Value of POWER
syms x1p up
```

```
equ=[(up-RL*x1p -l*x20)/L== 0,(l*x1p-x20/Rc-P/x20)/c== ↩
   0];
[x1P,uP]=solve(equ,[x1p,up]);
ilP=double(x1P)
VsP=double(uP);
Vco = x20;
wnP = sqrt(RL/(c*L*Rc) - (P*RL)/(c*L*Vco^2) + lambda^2/(↩
   c*L))
zetaP = (RL/L + 1/(c*Rc) - P/(c*Vco^2))/(2*wnP);
QP = 1/(2*zetaP);
fnP = wnP/(2*pi);
tnP = 1/fnP
RelationP = wnP*tnP;
T=1;
dc=1;
test1 = sim("Test_for_Tmax1.slx");
Tmax = max(test1.tout)
Approx code
syms y x y1

% condition 1
```

```
if Tmax >= 1
    a = 1 - (tnP*Pnl)/P
    yy = ((Pnl/P - a)/(a - 1)).*(x - 1) + a
    y1 = Pnl/P;
    xI = double(solve(yy==y1));
    yI = double(((Pnl/P - a)/(a - 1)).*(xI - 1) + a);
    %nmax = round(ilPnl-Pnl/wnP) %borrar

    tmax1 = Tmax/tn
    nmax1 = round((xI/tnP) - (ilPnl-Pnl/wnP) - (xI/(↩
       wnP*tnP^2)))
    nmax = round(2*tmax1/nmax1 + Pnl/P)



    y1 = Pnl/P;

    % The valleys repets nmax - 1
    n = 1;
    nn = 3;
    nnn = 3;
    nmin = 2*nmax -1;
```

```
n1 = (nn+2)/8;
for z = 1 : round(1/tnP)
    if z <= nmax
        xv(z) = tnP*z;
        yv(z) = -tnP.*(xv(z) + tnP/8).^(-Pnl/P)+↩
           Pnl/P;
    end

    if (z > nmax) && (z <= 2*nmax -1)
        xv(z) = tnP*z;
        yv(z) = -tnP.*(xv(z)-tnP*nmax/2).^(-Pnl/P↩
           )+Pnl/P-tnP/2;
    end
    if (z > nmin) && (z <=nn*nmax -nnn)
        xv(z) = tnP*z;
        yv(z) = -tnP.*(xv(z)-tnP*nmax*n).^(-Pnl/P↩
           )+Pnl/P-tnP*n;
    end
    if z == (nn*nmax -nnn)
        nmin = nn*nmax -nnn;
        nnn = nnn + nn;
```

```
            nn = nn + 1;
            n = n + n1;
            n1 = (nn+2)/8;
        end
    end

    for xi = 1 : length(xd)
        if xd(xi) > N*tnP + tnP/4
            N = N + 1;
        end
        y(xi) = pie(xd(xi),tnP,N,yv,y1,xI,Pnl,P,a);
        data(k,:) = double([P y(xi) xd(xi)]);
        k = k + 1;
    end

end
% Condition 2
    if (Tmax > 10*tn) && (Tmax < 1)
        a = Tmax
        b = -1 - (tnP*(1-Pnl/P))
        y = a*x.^b;
```

```
y1 = Pnl/P;

xI = double(solve(y==y1));


if P < PL

    tmax1 = Tmax/tn

    nmax1 = round((xI/tnP) - (ilPnl-Pnl/wnP) - (↩
       xI/(wnP*tnP^2)))

    nmax = round(2*tmax1/nmax1)

    maxrepts = round(xI/(wnP*tnP^2)) + 1

else

    tmax1 = Tmax/tn

    maxrepts = round(xI/(wnP*tnP^2)) + 1

    nmax = round(tmax1/maxrepts)

end

n = 2;

nn = 1;

m = nmax;

nmax1 = nmax;

nmax2 = nmax + m;

% valleys code

for z = 1 : round(1/tnP*Pnl/P)
```

```
if z <= nmax
    xv(z) = tnP*z;
    yv(z) = -tnP*(xv(z).^(-Pnl/P))+(Pnl/P+↩
       tnP);
end


if (z > nmax1) && (z <= nmax2)
    xv(z) = tnP*z;
    yv(z) = -tnP*((xv(z)-n*tnP).^(-Pnl/P))+(↩
       Pnl/P+tnP-n*tnP);
    if z == nmax2
        n = n + 1;
        nn = nn + 1;
        if nn == maxrepts
            m = m - 1;
            if m == 0
                m = 1;
            end
            nn = 0;
            maxrepts = maxrepts - 1;
         end
```

```
                    nmax1 = nmax2;
                    nmax2 = nmax2 + m;
                end
            end
        end
        for xi = 1 : length(xd)
            if xd(xi) > N*tnP + tnP/4
                N = N + 1;
            end
            y(xi) = pie1(xd(xi),tnP,N,yv,y1,xI,a,b);
            data(k,:) = double([P y(xi) xd(xi)]);
            k = k + 1;
        end
    end
% Condition 3
    if (Tmax <= 10*tn)
        if Tmax >= 10*tnP*0.6
            a = Tmax - (tnP*Pnl/P)
        else
            a = Tmax
        end
```

```
b = -1 + tnP*Pnl/P
if (il0 <= Pnl/wn) && (Pnl/wn < ilP)
    a = Tmax - (tnP*Pnl/P)
    b = -1 - (tnP*(1-Pnl/P))
end
if (Pnl/wn < wn*tn)
    a = Tmax/(1-tnP) - tnP
    if a <=0
        a = Tmax/(1-tnP)
    end
    if Tmax >= 10*tn*0.4
        a = Tmax;
    end
    b = -(wnP/wn0 - tnP)
end
y = a*x.^b
y1 = Pnl/P;
xI = double(solve(y==y1));
tmax1 = Tmax/tn
if (Pnl/wn < wn*tn)
    nmax = round(xI/tnP) - 1
```

```
    maxrepts = round(xI/(wnP*tnP^2)) + 1
    nmax = nmax/maxrepts
    n = 2;
else
    maxrepts = round(xI/(wnP*tnP^2))
    nmax = round(tmax1/maxrepts) - 2
    n = 1;
end


nn = 1;
m = nmax;
nmax1 = nmax;
nmax2 = nmax + m;
% valleys code
for z = 1 : round(xI/tnP) + 1
    if z <= nmax
        xv(z) = tnP*z;
        yv(z) = -tnP*(xv(z).^(-Pnl/P))+(Pnl/P+↩
           tnP);
    end
```

```
if (z > nmax1) && (z <= nmax2)
    xv(z) = tnP*z;
    yv(z) = -tnP*((xv(z)-n*tnP).^(-Pnl/P))+(↩
       Pnl/P+tnP-3*n*tnP);
    if z == nmax2
        if nn == maxrepts
            m = m - 1;
            if m == 0
                m = 1;
            end
            nn = 0;
            maxrepts = maxrepts - 1;
            if maxrepts == 0
                maxrepts = 1;
            end
        end
        n = n + 1/2;
        nn = nn + 1;
        nmax1 = nmax2;
        nmax2 = nmax2 + m;
    end
```

```
            end
        end
        for xi = 1 : length(xd)
            if xd(xi) > N*tnP + tnP/4
                N = N + 1;
            end
            if nmax > 0
                y(xi) = pie1(xd(xi),tnP,N,yv,y1,xI,a,b)↩
                   ;
                data(k,:) = double([P y(xi) xd(xi)]);
                k = k + 1;
            else
                y(xi) = pie2(xd(xi),y1,xI,a,b);
                data(k,:) = double([P y(xi) xd(xi)]);
                k = k + 1;
            end
        end
    end
time = toc;
name = sprintf('Aprox_casetest_%d.mat',POWER);
save(name)
```

```
function y = pie(xd,tnP,N,yv,y1,xI,Pnl,P,a)
    if (xd > xI)
        y = ((Pnl/P - a)/(a - 1)).*(xd - 1) + a;
    elseif (xd > tnP*N - tnP/4) && (xd < N*tnP + tnP/4)
        y = yv(N);
    else
        y = y1;
    end
end
function y = pie1(x,tnP,N,yv,y1,xI,a,b)
    if (x > xI)
        y = a*x.^b;
    elseif (x > tnP*N - tnP/4) && (x < N*tnP + tnP/4)
        y = yv(N);
    else
        y = y1;
    end
end
function y = pie2(x,y1,xI,a,b)
    if (x > xI)
```

```
    y = a*x.^b;
else
    y = y1;
end
```