Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

Design and Analysis of Boost Converter Sliding Mode Control Using Matlab Real-Time Environment

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Control Engineering

By

Wondimagegn Debebe

Advisor

Dr. Mengesha Mamo (PHD)

Nov, 2025

Addis Ababa, Ethiopia

Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

This document certifies that Wondimagegn Debebe's thesis, "**Design and Analysis of Boost Converter Sliding Mode Control Using Matlab Real-Time Environment**," which was submitted as part of the requirements for the Master of Sciences degree in Control Engineering, complies with university regulations and maintains accepted standards for originality and quality.

Approved by the Board of Examiners

Name	Signature	Date
Dr. Sosina Menegistu		
(School Head)		
Dr. Mengesha Mamo		
(Advisor)		
Dr. Nebiyu Tegene		
(External Examiner)		
Dr. Mesfin Tilahun		
(Internal Examiner))		

Declaration

I hereby affirm that the research study titled "**Design and Analysis of boost converter Sliding Mode Control Using Matlab Real-Time Environment**" is entirely the result of my own work and intelligence. This study has not been submitted for credit toward a degree at this university or any other. Every source and reference used in this thesis has been properly cited.

Wondimagegn Debebe		Nov , 2025
Name of Student	**Signature**	**Submission Date**

As a university advisor, I have authorized the submission of this thesis for review.

Name of Advisor	**Signature**
Dr. Mengesha Mamo	

Acknowledgment

I genuinely thank God for his constant guidance and favor, which have lit my path and given me the strength to go beyond challenges while working on my thesis. I would like to sincerely thank my thesis advisor, Dr. Mengesha Mamo, for his essential counsel, unwavering support, and professional insights that have been crucial in guiding this research. Additionally, I would like to thank the dedicated control engineering lecturers at the College of Technology and Built Environment (CTBE) for their excellent advice, training, and contributions to my academic development. My success has been largely attributed to my family's everlasting love, support, and belief in my skills. Your steadfast support inspires me to strive for greatness. I am appreciative of my classmates' company, experiences, and stimulating discussions, all of which have improved and made my academic path very exceptional. I am thankful to everyone who has supported me in my academic pursuits. Your combined influence has had a significant impact on my life, and I am appreciative of your guidance and encouragement.

Wondimagegn Debebe

Abstract

Power converters are electronic devices designed to transform and regulate electrical energy for wide range of uses, from small-scale devices like tablets that run on milli-Watts to large-scale power systems to control Mega-watts. There are three main types of DC-DC converters: boost (output voltage higher than input voltage), buck (output voltage lower than input voltage), and buck-boost (either step-up or step-down) converters.

In this thesis, two control techniques-proportional-integral-derivative (PID) and sliding mode control (SMC) are used to model and regulate a DC-DC boost converter. We developed a mathematical model of the boost converter using fundamental circuit concepts, resulting in a bilinear system representation. Classical linear control techniques, such as proportional-integral-derivative (PID) control, have been extensively employed for boost converter control. However, linear controllers may exhibit limitations when dealing with the nonlinearities, parameter variations, and uncertainties inherent in boost converter systems. To solve these problems, in this study, a sliding mode control (SMC) was designed based on its nonlinear dynamic modeling. This technique has the potential to address the nonlinear nature of the system, provide an improved transient response, and maintain stability across a wide range of operating conditions. The stability analysis of the boost converter was performed using Lyapunov's stability criterion.

The performance of the SMC and PID controllers under various disturbances is thoroughly compared in this work. Across a range of disturbances, the SMC outperformed the PID. measured by figuring out the disturbance Integral Time Absolute Error (ITAE) improvement as a percentage. In particular, it attains an improvement of 97.87% under nominal conditions and 94.86% under modest controller variation. Furthermore, as compared to PID, the SMC showed better robustness in the face of disruptions. By creating Matlab Real-Time Environment (RTE) simulation results using MATLAB software, the efficacy of the suggested control method was confirmed.

Keywords: Boost Converter, DC-DC, Matlab, PID, SMC, RTE, Lyapunov's stability

Contents

List of Figures

List of Tables

List of Acronyms

DC-AC	Direct-Current to Alternative-Current
EMC	Electromagnetic compatibility
HIL	hardware-In-loop
IGBTs	Insulated-Gate-Bipolar Transistor
LC	Inductor-Capacitor
LPF	Low-Pass-Filter
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
PSO	Particle Swarm Optimization
RTE	Real-Time-Environment
SMC	Sliding -Mode-Controller
VSC	Variable Structure Control

Chapter 1

Introduction

1.1 Background of the Thesis

Power converters are electronic circuits that convert, control, and regulate electrical power. Their power ratings range from milliwatts in tablet computers to several megawatts in electric power systems. Since the first power electronic devices were invented in the early 1900s, numerous topologies for buck, boost, and buck–boost converters have been devised, developed, and thoroughly researched, particularly with the rapid growth of semiconductor technology [1]. Power converter reliability has become a key issue in industrial applications in recent years. To ensure a long operational lifetime, both the electrical components and the control circuits that accompany them must have high robustness and fault tolerance [2, 3]. One of the main considerations during system design is the overall efficiency of power electronic circuits. Increased efficiency not only guarantees direct economic gains by lowering power consumption and associated costs, but even small improvements have a big effect on the profitability scale for electronic systems. Additionally, power producing plants and environmental pollutants are impacted by power use [4]. One of the most crucial performance factors when using power converters is rejection of disturbances. Certain events, such as brief short-circuit problems in transmission lines or variations in friction related to mechanical motion in generators and motors, frequently result in load shifts in power transmission systems [5]. Numerous control techniques must be investigated and put into practice because there are so many different operating circumstances and output levels. In order to demonstrate system stability, the current research aims to model a boost converter, develop different control systems, and evaluate their efficacy.

1.2 Statement of problem

Design and Analysis of boost converter Sliding Mode Control Using Matlab Real-Time Environment is the focus of this study. A boost converter is a power electronic device commonly used in various applications, such as DC-DC power conversion and renewable energy systems. The specific problem to be addressed in this study includes the following key aspects: Design of Linear Controller; such as Proportional-Integral-Derivative (PID) controllers, for the boost converter. The challenge is to determine the controller parameters that can provide good transient response, stability, and robustness across a wide range of operating conditions. Analysis of Linear Control Performance: Once the linear controllers are designed, it is important to evaluate their performance in terms of key metrics such as settling time, overshoot, steady-state error, and disturbance rejection. The analysis should consider variations in load conditions, input voltage fluctuations, and controller uncertainties to assess the controller's robustness and stability.

This thesis work, a sliding mode control method is well known its robustness will be designed and applied to boost converter for stabilization and reference tracking problem. Its properties of robustness against uncertainties in the model and external disturbances make it the best controller.

1.3 Objective

1.3.1 General objective

Design and Analysis of boost converter Sliding Mode

Control Using Matlab Real-Time Environment.

1.3.2 Specific objective

- To analysis mathematical model for boost converter.
- To develop a sliding mode control strategy for boost converter.
- To evaluate the performance of output voltage tracks to a desired value in the presence of various disturbances.
- To simulate the mathematical models with the proposed controller using Matlab RTE simulation in Matlab software

1.4 Methodology

The study will be conducted using the following approach in order to fulfill its objectives:

- Literature Review: Conduct a comprehensive literature review to understand the existing research and developments in the field of boost converter control approaches. Identify the strengths and limitations of different control methods, as well as the performance metrics used for evaluation.
- Obtaining mathematical modeling for Boost Converter.
- Designing Sliding Mode Controller Boost converter.
- Analyze the results in MATLAB/ Simulink

1.5 Significance of the Thesis

A boost converter is a popular circuit used to raise a lower DC voltage to a higher one. It finds applications in various industries, including renewable energy systems, electric vehicles, and power supplies. For instance: Efficiency and Performance improvement, stability and robustness and nonlinear behavior control are crucial issues. Generally, the research in the Design and Analysis of boost converter Sliding Mode Control Using Matlab Real-Time Environment aims to enhance the performance, stability, and efficiency of this converter.

1.6 Thesis Scope

All of the models are implemented in simulation method only using Matlab or Simulink software. The effectiveness between SMC and PID controller is studied in terms of performance indices and voltage derivations from desired value under different disturbances.Analysis of the controller have been carried out for continuous conduction mode only.

1.7 Thesis Outline

chapter 2: Previous studies and literature pertaining to the discussion of the overview of Boost converter functioning are covered in the theoretical background. The Boost converter's functioning, construction, and basic principles are covered, as well as the control strategies that guarantee the output voltage and current are accurate. **chapter 3:** Explains how

to describe the Boost converter mathematically and creates a comprehensive model that is utilized in this thesis. **Chapter 4:** Describes the modeling of the boost converter and the design of PID and SMC controllers. The gains of the PID controller were adjusted through a trial-and-error process, while the gains of the sliding mode controller (SMC) were fine-tuned using the particle swarm optimization technique (PSO) . **Chapter 5:** Explains the outcomes of employing the boost converter model with the PID and SMC controllers that were created. The results for voltage regulation and performance metrics such as settling time, overshoot, and robustness are discussed for each controller. **Chapter 6:** The assessment of PID and SMC controller performance is included in the conclusion derived from the investigation carried out in this thesis. Recommendations and suggestions for future improvements in boost converter control are also provided.

Chapter 2

Literature Review

2.1 Theoretical Background

2.1.1 Converter Classification

Power converters are used to regulate the flow of power from the input to the output in terms of voltage or current in order to meet the necessary application, as described in Chapter 1. Depending on how energy is transmitted from the input to the output, power conversion methods may typically be categorized [6].

- DC to DC converter
- DC to AC inverter
- AC to DC rectifier
- AC to AC transformer

2.1.1.1 DC to DC Converter

DC-DC converters alter the DC power level from one level to another. Non-isolated DC-DC converters and isolated DC-DC converters are the two primary categories into which DC-DC converters can be broadly classified based on their isolation. According to [7], the primary distinction between DC-DC converter types is the presence of a DC link from input to output. Even though isolated DC-DC converters are more complex, noise filtering and effective functioning are made possible by the use of a transformer to connect input and output. The six most common types of non-isolated DC-DC converters are buck, boost, buck-boost,

Cuk, Zeta, and SEPIC. According to [8], there are four common isolated DC-DC converters: the flyback converter, the forward converter, the single-ended two-transistor forward converter, and the phase-shift full-bridge converter. DC-DC converters are frequently used in low-power circuits, including those used in laptop computers and cell phones that run mostly on batteries. Smaller circuits that help regulate the voltage from batteries or other power sources to the proper level are typically found inside these devices.

2.1.1.2 DC to AC Inverters

In AC motor drives and uninterruptible power supply, DC-AC inverters are used to produce a sinusoidal AC signal [9]. These devices are capable of controlling both frequency and amplitude. However, a system such as this is very difficult to operate effectively because of the system design. A cascaded control system is used to regulate the DC-DC converters and the DC-AC inverters. DC-AC conversion systems are usually made up of several tiers.

2.1.1.3 AC to DC Rectifier

The AC voltage that arrives with the power supply can be transformed into a form that can be utilized in the circuit by AC-DC rectifiers, also known as AC-DC converters. The goal in many power electronics applications would be to transform the AC input sine wave form into an appropriate DC voltage. Uncontrolled diodes, silicon-controlled rectifiers, and controlled IGBTs can be used to construct these rectifiers. Rectifiers can be used in DC servo drives, power supply, and other applications, depending on the circuit requirements.

2.1.1.4 AC to AC Transformer

AC-AC transformers serve to tune an incoming AC voltage up or down to what's required. A transformer's efficiency hinges mainly on core (iron) losses and copper losses. Using a transformer can increase the general power conversion efficiency of a DC-DC boost converter connected to an AC grid [10]. Other more power-electronics-based AC-AC converter approaches include the cycloconverter and matrix converter approaches [11]. Cycloconverters convert AC power directly from one frequency to a lower one without first converting to DC, while matrix converters are smart-switch-lattice-based approaches that essentially steer AC power between forms in some kind of bidirectional current control manner.

2.1.2 Power converters in switched mode

Electronic converters that are most widely used are switched mode power converters (SMPC). By altering the input voltage, they provide DC loads with a consistent output voltage. Electrical switches are used in these converters. The switches flip on and off quickly. The device's efficiency is maintained by this quick switching, which reduces power loss. When it comes to power size and efficiency, SMPCs outperform linear devices. To reduce the output voltage ripple, you'll need a filter. Certain electromagnetic emissions(EME) are caused by the LC components in the filter. Despite this, SMPCs continue to be the most popular converters [12].

2.1.3 Boost Converter

A boost converter is also referred to as a step-up converter since it raises a lower DC voltage. The boost converter's basic topology-based working concept and some of its applications are presented in this section.

2.1.4 Basic Concept

A boost converter's basic circuit is depicted in Figure 2.1. A switching transistor, inductor L, capacitor C, and diode D are all part of the boost converter circuit. The switch device in this case is a MOSFET . It's important to remember that, depending on variables like current, voltage, switching speed, and cost, bipolar power transistors and IGBTs can also be utilized for power switching [13]. The fundamental principle of a boost converter is based on the

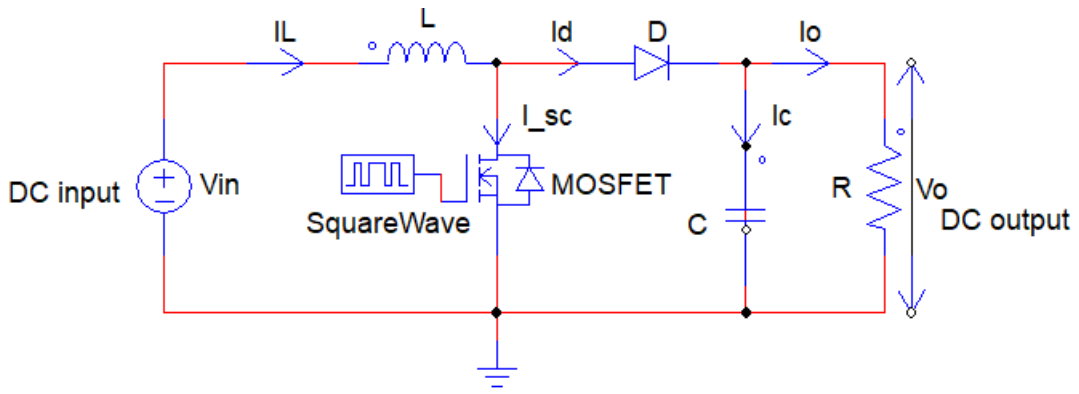

Figure 2.1: Basic Boost Converter

inductor's inherent resistance to abrupt changes in current by creating a magnetic field [14]. The inductor starts to store energy as a magnetic field when the switch is activated. The

inductor releases its stored magnetic energy and transfers it to the load whenever the switch is switched off. Figure 2.2 illustrates the circuit path when the switch is subjected to a

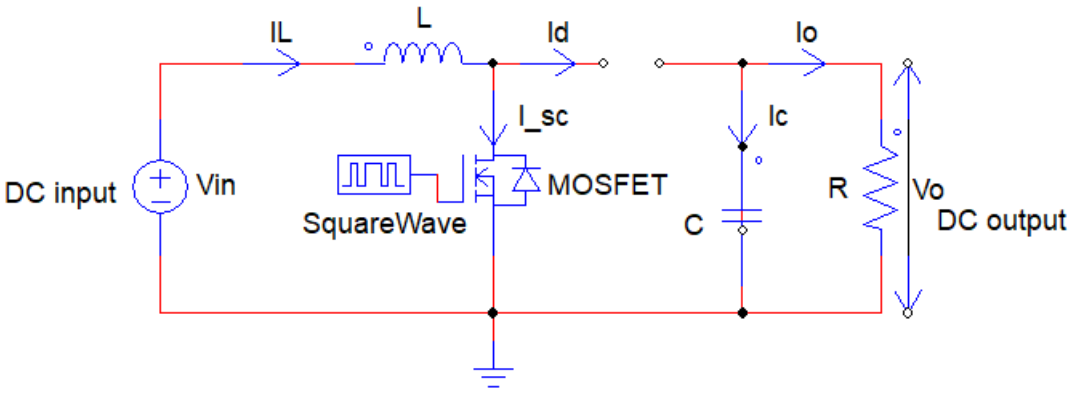

Figure 2.2: Switch ON

high voltage square wave cycle, which causes the MOSFET to turn on. The inductor stores energy by creating a magnetic field and is charged by input DC power during the switch's "ON" time. As a result, current moves from the positive to the negative terminal through inductor L. In the meantime, the load receives energy from capacitor discharge during this switch ON period. The circuit behavior during the low level of switching square wave, which

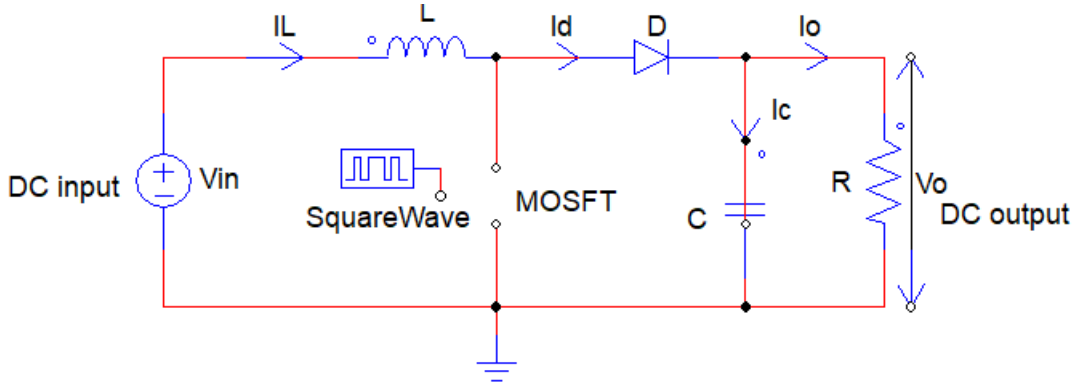

Figure 2.3: Switch OFF

causes the MOSFET to turn off, is shown in Figure 2.3. In order to maintain current flow from left to right, the magnetic field formed at switch ON is destroyed. This results in a greater voltage and charging capacitor C since the load receives energy from two sources: the inductor and input DC power. The switching function is displayed in figure 2.1 with a steady state duty cycle D. As seen in figure 2.1, the inductor voltage V_L pulses between two values: V_{in} and $-(V_o - V_{in})$. The volt-second area effects during the two subintervals balance out because they are identical in magnitude but opposite in direction because the average inductor voltage is zero 2.5.

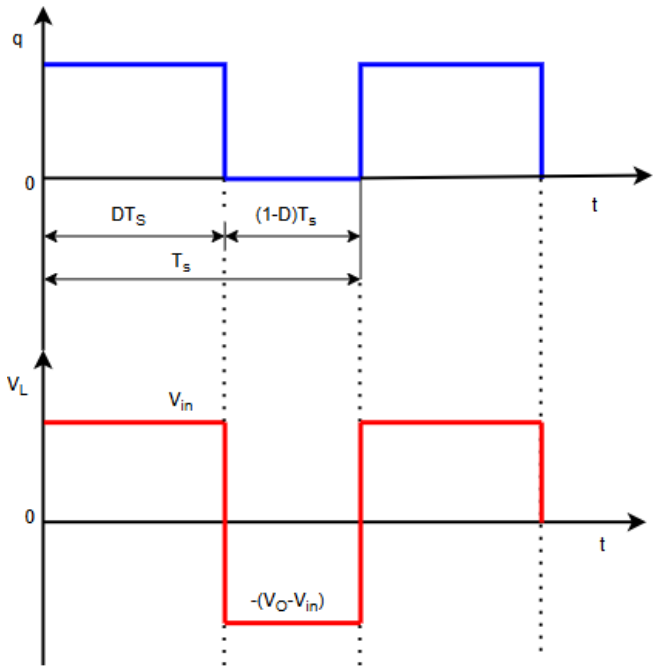

Figure 2.4: Boost Converter Waveforms

The inductor voltage waveform, whose average is zero in dc steady state [15], can be used to determine the input output voltage ratio. This provides:

$$(\mathrm{DTs}) = (\mathrm{V}_o - V_{in})(1 - D)Ts$$

Therefore,

$$\frac{V_o}{V_{in}} = \frac{1}{1 - D} \tag{2.1}$$

The following equation for the output voltage ripples

$$\Delta V_o \ll V_o$$

$$\Delta V_o = \frac{\Delta Q}{C} \tag{2.3}$$

$$\Delta V_o = \frac{I_o \cdot D \cdot T_s}{C} \tag{2.4}$$

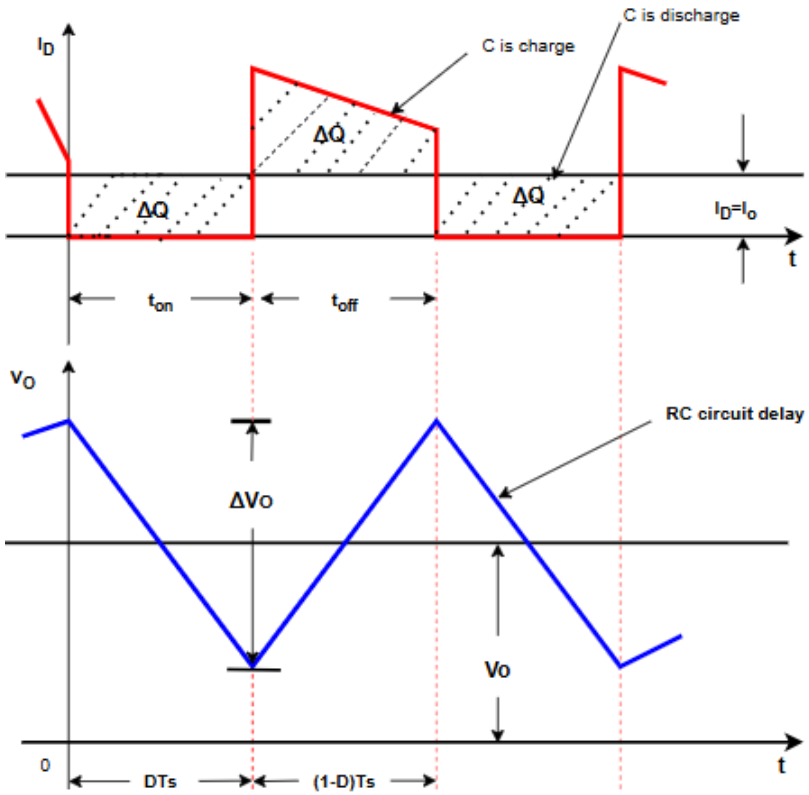

Figure 2.5: The output voltage ripple wave forms [16]

Where, $I_o = \frac{V_o}{R}$ and $T_s = \frac{1}{f_s}$.
Substituting I_o and T_s into equation (2.4):

$$\Delta V_o = \frac{V_o \cdot D}{f_s \cdot R \cdot C} \tag{2.5}$$

We know that $R \cdot C$ is the time constant τ, therefore:

$$\frac{\Delta V_o}{V_o} = \frac{D}{f_s \cdot \tau} \tag{2.6}$$

From equation (2.6), we can see that the output voltage ripple does not depend on the inductor value.

2.1.5 Real Time Environment

In a MATLAB simulation, a real-time environment refers to a model that operates at the same rate as it would in reality. Essentially, one second in the simulation equals one second in the real world. It is a thing of utmost importance when it comes to the development and optimization of systems that are highly dependent on accurate timing, such as embedded software, control systems, and other time-sensitive technologies. Fast simulation helps us

to understand how such systems will work in the real world without the need for costly or even risky physical prototypes. It finds its use most in areas where accurate timing and fast response are vital, for instance, power electronics, automotive systems, robotics, and communication networks [17].

2.1.6 Real, Time Environment Simulation's Significance

Accuracy and Real-World Relevance: Real-time simulations give engineers a clearer picture of how the system will actually work once it is implemented in the real world. Real, time simulation enables them to spot those problems that a slower, offline simulation might not be able to detect, for example, inefficiencies in algorithms or instability of the system. When engineers thus get from the design stage to the real deployment, this results in more reliable outcomes and their confidence gets a boost [18].

Time and Cost Saving:

With real, time simulation, technicians can rapidly execute various scenarios that would have taken them several hours of manual testing and thus, they could avoid the need for purchasing expensive hardware. Therefore it is a great move in the development process and it makes the repetition of physical trials less necessary, thus saving both time and money [19].

Safe Testing:

The capability of securely performing the testing of the most dangerous or failure scenarios is one of the foremost advantages of a real, time simulation. As an example, it is considerably more secure to simulate power system faults or errors in a robotics configuration than to actually try those tests in the real world where accidents can cause both the damage and the downtime.

2.1.7 Advantages of Real-Time-Environment Simulation with MATLAB/Simulink:

Is a platform that can be trusted to carry out real-time simulation which is a feature of great importance in the engineering world. With the use of this tool, models can be executed in real-time thus the need for a hardware connection is eliminated. In simple terms, system testing in this manner gains in speed, flexibility, and convenience, and this is the major reason why the method is highly esteemed among others.

- **No Need for Physical Hardware:** Simulink Real Time makes it possible for a

system to be simulated in real time without the need for physical hardware. Even though the actual components are absent, the simulation behaves in such a way that it is almost identical to the system in the real world. Consequently, it is a tool that is very useful in the early stages of the development process of control algorithms and any other system, thus the delay and complexity of setting up hardware are eliminated [17] .

- **Easily Integrates with Hardware:** Real-time simulation may be performed without the need for hardware, but in the case where you want to connect to physical devices, it is very straightforward. The transition from the simulation to the real-world environment, where hardware is connected, is therefore very smooth; the algorithm you are testing will indeed be working when it is running on the real hardware [20].

- **Flexible and Scalable:** Simulink Real-time has the capability of being flexible to run on different operating systems and hardware thereby making it possible for one to be working on a small embedded device or a large industrial system and the software will still be valid. The flexibility feature is a great advantage to engineers since they can maintain the same standard across different types of projects [17] .

2.1.8 Real-Time Monitoring and Visualization

Simulink includes convenient built-in tools such as Scopes and To Workspace blocks, which allow engineers to monitor the performance of the system in real time. Engineers can see the important signals, such as voltage, current, and duty cycles, during the execution of the simulation. The live feedback makes it possible to detect problems at an early stage and to adjust the system immediately [20]. When control systems are fast or complex, real-time simulation cannot be dispensed with. Besides, it ensures that the control algorithms continue to perform their functions correctly, hence their reliability is maintained and they become capable of quickly adjusting themselves to the new situation if the system changes [19].

2.1.9 Applictaions

Boost converters with a switching mode have been extensively used in various applications. After a hundred years of its introduction, it has been broadly utilized in both mature areas, like electric power systems, and newly developed ones, e.g., electric vehicles [21]. As an example, an electric vehicle motor operating at a high voltage of 500V cannot be powered

by a single battery. Although it is technically feasible to satisfy such a demand by connecting several batteries in series, the resulting device would be too large and inconvenient. A more effective method is to use a smaller number of batteries along with a well, designed boost converter that can raise the battery's DC voltage to the required level [22]. The DC input of the boost converter can be taken either from batteries or from rectified AC from the mains. In addition, the boost converter can be fueled by renewables and other sources like solar panels, fuel cells, dynamos, and DC generators.

2.1.10 PID Controller

PID control is basically what you hear most of the time in industrial applications. It is one of the main solutions used in all types of systems such as robots, factories, chemical plants, cars, and so forth. PID controllers are favored by people because they are easy, efficient, and can be used in a vast number of scenarios without any problem. Fundamentally, a PID controller is a device that regulates the flow of operations. It is always changing the control fed into the system, it is always looking for that target value and it is also rectifying the differences between the actual and the expected values.

2.1.10.1 Components of PID Controller

A PID controller works by tweaking the control input using three things.

- **The proportional (P):** This is the one that leaps into the scene immediately after an error is detected. It virtually questions, "What is the current deviation?" and the farther the error is, the harder the effort to correct it is. That effort is determined by the proportional gain, or K_p. If you increase K_p, the controller will respond more aggressive. However, if you increase it to an extremely high level, you will have the problems of overshooting and a system that becomes unstable.

- **Integral (I):** The integral term is the one that remembers all the past errors that have been accumulating over time. In simple terms, it is the part that makes sure that after all residual, steady errors are eliminated and if needed it goes beyond that to the target. The integral gain, K_i, is the one that defines how much the controller influence the reaction to the accumulated errors. However, in the case that the integral action is very aggressive, the system may begin to oscillate intensely or it may even lose stability.

- **Derivative (D):** The derivative part is concerned with the rate at which the error is changing. Imagine it is a short look ahead; it being able to figure out where the error is going it arrives early to make sure that there is no overshoot or that the system does not bounce. The derivative gain, K_d, is the one that determines to what extent the adjustment is made depending on how fast the error is changing. If you manage to tune it properly, the system will be less of a wobble, more stable.

A PID controller figures out its response like this:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t)$$

Here's what all that means: $u(t)$ is the controller's output, or the control input to your system. $e(t)$ stands for the error at time t. The constants K_p, K_i, and K_d are the proportional, integral, and derivative gains. Each one tunes how the controller reacts to the error right now, in the past, and by predicting where it's headed.

2.1.10.2 Importance of PID Controller

PID controllers offer a straightforward and dependable method to handle various kinds of dynamic behavior in a system. In fact, you can adjust them to achieve certain objectives, for instance ensuring that the system reacts fast, keeps the error at a low level, or remains stable when getting unpredictable. One of the reasons they are finding so much use is that they are easy to make, whether you are dealing with hardware or software. We don't have to understand every single detail of the system for them to work. This feature makes them very effective in situations where things are complicated or you don't have a well-defined model. However, tuning the parameters Kp, Ki, and Kd is not always a piece of cake. To get the optimum performance, often one has to use some trial and error, or optimization tools and analytical tricks might be employed. In someone's actual world, folks have a tendency to regulate the gains on a per-job basis, thereby discovering the most effective way as they proceed.

2.1.10.3 Restrictions and Challenges

PID controllers are used in a wide range of applications, but they are not flawless. In the case of systems with large time delays or those that behave unpredictably, you cannot always rely on a PID controller to accomplish the task. If you do not adjust the parameters very

carefully, things can go terribly wrong, and sometimes you may end up with unstable or slow performance. Advanced methods such as model, based control, adaptive control, or sliding mode control are then available to take over. We will explore these issues in the following chapters. You will discover how PID controllers can be combined with contemporary control strategies, and we will examine the feasible solutions to the typical problems of basic PID control.

2.1.11 Sliding Mode Controller (SMC)

Sliding Mode Control (SMC) is generally regarded by a lot of people as a good method to come up with dependable controllers, especially for complicated nonlinear systems operating in unpredictable or dynamic environments. One of its biggest strengths is that it can tolerate changes in system characteristics and internal disturbances; hence, accurate system modeling is not always necessary. SMC helps to simplify the design of feedback controls by breaking down the overall system behavior into more basic, lower-dimensional parts. The control method is based on discontinuous state functions, which correspond to the binary nature of standard power converters that operate in simple "on/off" modes. Despite the benefits, SMC continues to be a fascinating topic of research in both the academic and industrial realms. It has been practically implemented with great success in such areas as robotics, control of electric motors and generators, industrial process automation, and vehicle motion systems [23].

2.1.11.1 Sliding Mode Control System Fundamentals

The following is a description of the two-step process for SMC design:

(i) In order to accomplish the appropriate system dynamics during sliding mode, a sliding surface is predefined.

(ii) A controller is then used to move the closed-loop dynamics to the sliding surface and maintain it there.

2.1.11.2 System Model and Sliding Mode Surface Design

We examine the following linear time-invariant (LTI) system without sacrificing generality [23]:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.8}$$

Where $x \in R^n$ is the system state vector, $u \in R^m$ is the control input vector, $A \in R^{n\times n}$ and $B \in R^{n\times m}$ are constant system matrices.

Assumption 2.1: The controllability matrix ([B AB A^2B ... $A^{n-1}B$] has full rank if n > m , B is of full rank (m , and the pair (A, B)[2] is fully controllable.The Let s(t) be a sliding variable vector that passes through the state space origin in R^m : $s(t) = Cx(t)$ (2.2) Where $C \in R^{m\times n}$ is the sliding mode parameter vector and $\|CB\| \neq 0$. The system equation (2.2) is said to attain a sliding mode surface when the state variable vector reaches and remains on the intersection of the m switching plane variables. The method of equivalent control is a way to determine the system motion restricted to the sliding mode surface $s(t) = 0$. On the sliding mode surface, $s(t) = 0$ and $\dot{s}(t) = 0$, using expressions equation (2.8) and (2.2), we have:

$$\dot{s}(t) = C\dot{x}(t) = 0 \tag{2.10}$$

Substituting $\dot{x}(t)$ from equation (2.8) into (2.10):

$$C(Ax(t) + Bu_{eq}(t)) = 0 \tag{2.11}$$

Where $u_{eq}(t)$ is viewed as the equivalent control. From expression equation 2.11, the equivalent control can be expressed as:to represent the equivalent control as follows:

$$u_{eq}(t) = -(CB)^{-1}CAx(t) \tag{2.12}$$

Substituting equation (2.12) into (2.8) yields the following differential equation:

$$\dot{x}(t) = \left[I - B(CB)^{-1}C\right] Ax(t) \tag{2.13}$$

The analogous system, represented by the system equation (2.13), explains the dynamic motion of the system (2.8) on the sliding mode surface. The similar system's attributes can be summed up as follows:

Remark 2.1: The control input has no effect on how the similar system behaves over time. Therefore, it is possible to determine the matrix C without knowing the nature of control input beforehand. In general, the system response limited on the sliding mode surface (2.13) has a desirable behavior, such as asymptotic stability and prescribed transient response, because of the design of the sliding parameter C. According to the linear control theory, in order to guarantee the solution of the differential equation in (2.13) to be asymptotically

stable, the sliding mode parameter vector C should be selected so that all the eigenvalues of the differential equation (2.8) have negative real parts. What's more, though the sliding surface (2.2) is linear, it indeed could be any other forms with nonlinearity to ensure a finite-time convergence of system dynamics in sliding mode. This will be looked at in the following part of this chapter.

Remark 2.2: Reduction of system order: For the matrix B with complete rank m, since s(t) = 0 in the sliding mode, there are m components of the state vector that depend on the remaining (n - m) components: $x_2 = s_o(x_1)$, $x_2, s_o \in R^m, x_1 \in R^{n-m}$, and thus, the sliding mode equation's order may be decreased by m .

$$\dot{x_2} = f_1(x_1, t, s_o(x_1)), \quad f_1 \in R^{n-m} \tag{2.14}$$

To put it another way, the analogous system (2.13) is a $(n-m)$-th order system, meaning that the sliding mode surface simplifies the system dynamic.

2.1.11.3 Reaching Phase

The reaching phase and the sliding phase are the two primary stages of Sliding Mode Control (SMC) design.In order for the system dynamics to reach and remain on the sliding surface, the reaching phase is essential. The idea of a sliding mode of a second-order system is shown in Figure 2.6. The next important problem is how to design a controller to guarantee the

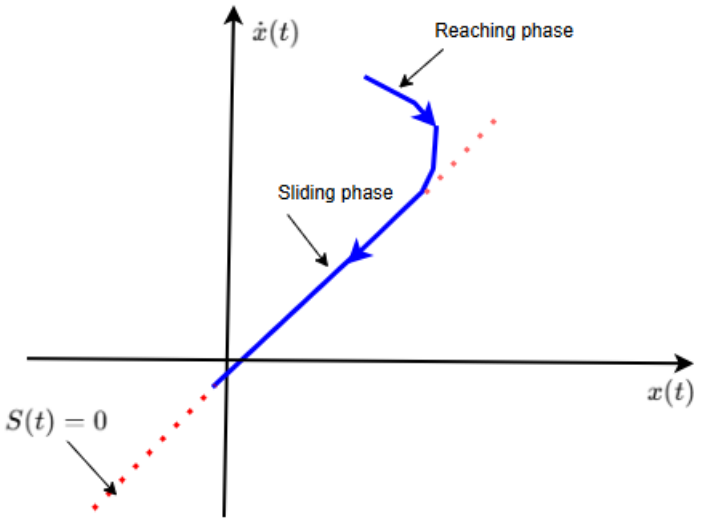

Figure 2.6: The two phases of ideal sliding mode

reachability of the sliding variable to the sliding mode surface. The goal of the sliding mode

controller is to guide the sliding variable s toward zero, ensuring that the system then follows the desired dynamics outlined in the equation (2.8) will be obtained [2, 24].

2.1.11.4 Reaching Condition

In ,he endena convergence problem is making sure the switching variables arrive at the sliding surfaceTherefore, Lyapunov's direct method has been widely used in SMC designs as a stability condition to ensure the convergence of the sliding mode variable onto the sliding surface during the reaching phase. All too often, the following Lyapunov's function candidate is used in the sliding mode controller design:

$$V(t) = \frac{1}{2}s(t)^T s(t) \tag{2.15}$$

In order to guarantee the asymptotic stability of the system equation (2.15) about the equilibrium point $x(t) = 0$, the following reaching condition must be satisfied:

$$\dot{V}(t) = \frac{1}{2}s(t)^T \dot{s}(t) < 0 \quad \text{for} \quad s(t) \neq 0 \tag{2.16}$$

Remark 2.3: Actually, a sufficient condition to guarantee is provided by the condition equation (2.16). It is important to note that the majority of sliding mode controllers are built using the reachability condition found in equation (2.16) in order to guarantee that the sliding mode controller can cause the sliding variable "s(t)" to asymptotically converge to zero.

2.1.11.5 Reaching Laws

To ensure the sliding mode's existence, SMC can be built using reaching laws. Lists some potential kinds of reaching laws [25] . Generally speaking, a reaching law can be expressed as follows:

$$\dot{s} = -\epsilon \ \text{sign}(s) - f(s), \quad \epsilon > 0 \tag{2.17}$$

Where $f(0) = 0$ and $sf(s) > 0$ when $s \neq 0$. In practice, three special reaching laws commonly used can be derived from equation (2.17) as follows:

i. Constant Rate Reaching Law

$$\dot{s} = -\epsilon \, \text{sign}(s), \quad \epsilon > 0 \tag{2.18}$$

The switching variable is constrained by this law to arrive to the switching manifold at a constant rate (s). This reaching law has the advantage of being straightforward. On the other hand, the reaching time will be excessively long if s is too little. However, excessive chattering will result from a too large s .

ii. Exponential Rate Reaching Law

$$\dot{s} = -\epsilon \text{sign}(s) - ks, \quad \epsilon > 0, \quad k > 0 \tag{2.19}$$

When s is big, the states are compelled to approach the switching manifold more quickly due to the addition of the proportional rate term (-k s).

iii. Power Rate Reaching Law

$$\dot{s} = -k|s|^{\alpha} \text{sign}(s), \quad 1 > \alpha > 0, \quad k > 0 \tag{2.20}$$

This reaching law increases the reaching speed when the states are far away from the switching manifold. However, it reduces the rate when the states approach the manifold. It is evident that the above three reaching laws can satisfy the reaching condition (2.15), and thus ensure the existence of the sliding mode. It is worth noting that a reaching law method simultaneously ensures the reaching condition, influences the dynamic quality of the system during the reaching phase, and provides the means for controlling the chattering level. Thus, a reaching law method can be applied to both linear and nonlinear SMC systems with system perturbations and external disturbances, to improve the performance of the reaching phase and reduce the amplitude of chattering [2, 24].

2.1.11.6 Equivalent Controller Design

In most of the VSC schemes, the control input usually consists of two components as followings:

$$u(t) = u_{\text{eq}}(t) + u_s(t) \tag{2.21}$$

Where the linear component $u_{\text{eq}}(t)$ is defined as in (2.12) and the nonlinear signal incorporates the discontinuous component given below:

$$u_s(t) = -\eta(CB)^{-1} \text{sign}(s(t)) \tag{2.22}$$

where a constant control gain is represented $\eta > 0$.

$$\dot{V}(t) = s^T(t)CAx(t) + s^T(t)CBu(t) = -\eta|s(t)| < 0 \tag{2.23}$$

is the result of Substituting (2.12) and (2.22) into (2.15) We can infer from (2.23) that there is a finite time guarantee for the sliding mode variable to reach the sliding mode surface.

Remark 2.4: The intended dynamics in equation (2.13) are the only factors that define the closed loop system dynamics once the sliding variable vector $s(t)$ is driven to zero. As a result, system uncertainties on the sliding mode surface have no effect on the closed loop system. Because of this, SMC systems are resilient to system uncertainties, which makes them an effective tool for controlling uncertain systems. This has greatly inspired later scholars in the field. However, it should be noted that throughout the reaching phase that is, prior to the sliding surface being reached—the system is still impacted by the disturbances.

2.1.11.7 Property of Robustness

One important aspect of the SMC system is robustness. The SMC controller design constantly takes system uncertainties and disturbances into account. The LTI system in equation (2.8) can be generalized as follows in the presence of uncertainties and disturbances:

$$\dot{x}(t) = (A + \Delta A)x(t) + (B + \Delta B)u(t) + d(t) \tag{2.24}$$

where (d(t) denotes the external disturbance and ΔA and ΔB is denote the system uncertainties. The equation (2.24) can be rewritten as:

$$Ax(t) + Bu(t) + f(t) = \dot{x}(t) \tag{2.25}$$

where, the disturbance $f(t) = \Delta Ax(t) + \Delta Bu(t) + d(t)$ is representing the lumped uncertainty or situation. The definition of equivalent control defined as equation (2.21), where $u_s(t)$ given by equation (2.22) and the equivalent control is given by the following equation:

$$u_{\text{eq}}(t) = -(CB)^{-1}C(Ax(t) + f(t)) \tag{2.26}$$

In (2.26) by replacing (2.25)the equivalent system equation in sliding mode is obtained:

$$\dot{x}(t) = \left[I - B(CB)^{-1}C\right] Ax(t) + \left[I - B(CB)^{-1}C\right] f(t) \tag{2.27}$$

(2.27) becomes (2.13), which is totally insensitive to system uncertainties and outside disturbances, if $f(t)$ satisfies the matching condition $f(t) = Bg(t)$. In other words, when it comes to matching system uncertainty and disturbances, the SMC system demonstrates a significant robustness. SMC is an effective tool for managing uncertain systems because of this invariance quality, which also serves as a compelling reason for the ongoing interest in the control field. Nevertheless, the corresponding control action (2.26) cannot be implemented in practice since it depends on the unknown external signal.

2.1.11.8 The phenomenon of chattering

In real world systems, the ideal sliding mode depicted in Figure (2.6) cannot be fully achieved since it would necessitate an infinite frequency switch in the control input. In real applications, limitations and imperfections in switching hardware introduce chattering in sliding mode control (SMC). As shown in Figure (2.7) [26, 27], this fast, undesired oscillation produces a discontinuous feedback response that leads to a unique dynamic behavior in the area of the sliding surface. In figure (2.7), the system trajectory in the region where $s(t) > 0$ moves towards the sliding surface $s(t) = 0$ and initially reaches it at point A. Under ideal sliding mode control, the trajectory would remain on the surface and begin sliding from point A onward. In practice, however, there is a delay between the instant when the sign of $s(t)$ changes and when the controller updates the switching action. Because of this delay, the trajectory overshoots the surface, reverses direction, and moves toward it again. Repeating this sequence produces a characteristic "zig-zag" pattern small oscillations around the desired sliding surface.

This chattering phenomenon reduces control precision and leads to excessive energy loss in

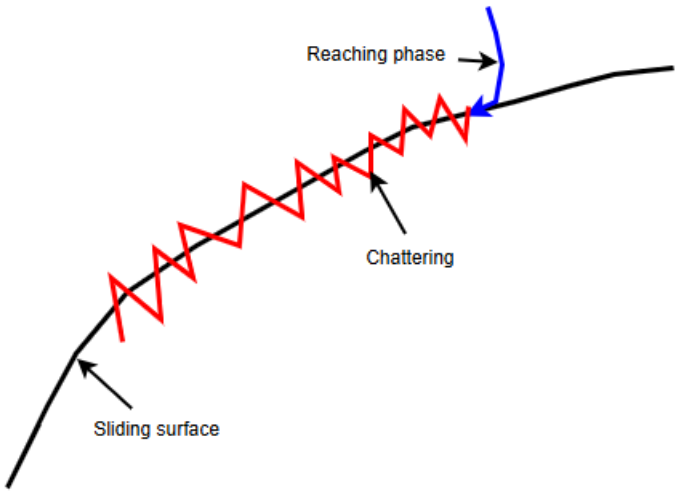

Figure 2.7: The chattering phenomenon

power electronics. It can also accelerate wear in mechanical components and may excite unmodeled high-frequency dynamics. As a result, chattering not only degrades overall system performance but can also, in severe cases, threaten system stability.

2.1.11.9 Boundary Layer Technique

A number of methods have been introduced in the literature to reduce or eliminate chattering in sliding mode control [28, 29]. One widely used approach is the boundary layer technique. As shown in equation (2.22), the discontinuous or switching part of the SMC controller is typically formulated using a sign function. As seen in Figure (2.7), the boundary layer method reduces chattering by substituting a smooth saturation function for this sign function. The expression for the modified control law is:

$$u_s(t) = -\eta(CB)^{-1}\,\mathrm{sat}(s(t)) \tag{2.28}$$

where the saturation function denoted by sat s is determined by:

$$\mathrm{sat}(s) = \begin{cases} \frac{s}{\rho} & \text{for } |s| \leq \rho \\ \mathrm{sign}(s) & \text{for } |s| > \rho \end{cases} \tag{2.29}$$

A positive constant $\rho > 0$ is typically selected in simulations or experiments to ensure that chattering is suppressed while maintaining acceptable control performance. This smoothing

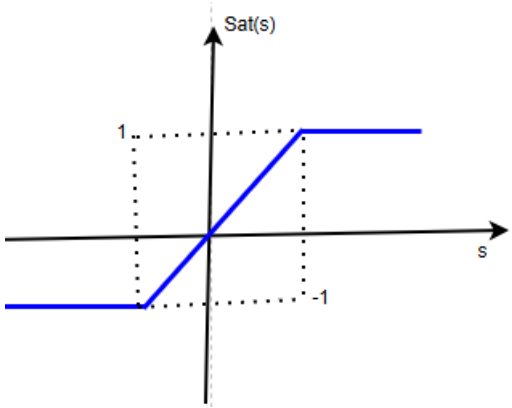

Figure 2.8: The Saturation function sat(s)

approach is widely used to reduce the rapid switching inherent in sliding mode control. However, although it effectively removes chattering, it also weakens the robustness that characterizes true sliding motion. As a result, the system may lose its asymptotic stability. For this reason, the boundary layer method cannot be considered a complete solution to the

chattering problem. Another strategy for addressing chattering involves using a continuous approximation—often referred to as the pseudo–sliding mode method where the sign function in equation (2.22) is replaced with a smooth alternative [26].

$$u_s(t) = -\eta(CB)^{-1}\frac{s(t)}{|s(t)| + \rho} \tag{2.30}$$

However, this method can result in excessively high control gains when the system states lie very close to the sliding surface.

2.1.11.10 Stability of Equilibrium Points

Let us examine a dynamical system that meets:

$$\dot{x} = f(x, t) \tag{2.31}$$

whereR$x \in R^n$ is the state variable vector, and n is the order of the system.A collection of functions of x(t) is denoted by f(x, t) $\in R^n$).$wLetx_e \in R^n$ be an equilibrium point of the system(2.31), i.e., $f(x_e, t) = 0$. without sacrificing generality, we present all definitions and theorems for the situation when the equilibrium point is at the origin ofR^n, i.e., $x_e = 0$.
Definition 2.1: The equilibrium point at the origin of (2.31) is:

- **Stable**, if, for any $\epsilon > 0$, there exists $sigma > 0$.

$$|x(t)| < \sigma \implies |x(t)| < \epsilon, \ \forall t \geq 0 \tag{2.32}$$

- **Unstable**, if the above condition is not satisfied.
- **Asymptotically Stable**, if it is stable and δ can be chosen such that:

$$|x(t)| < \delta \implies \lim_{t\to\infty} x(t) = 0 \tag{2.33}$$

addresses only local stability, describing the system's behavior in the vicinity of an equilibrium point. As a result, it has limited practical significance. To ensure global stability, Lyapunov's direct method is employed, which is introduced in the following section to overcome this limitation.

2.1.11.11 Direct Method of Lyapunov

Lyapunov's direct technique, often known as the second method of Lyapunov, makes it possible to evaluate a system's stability without actually solving the differential equation in (2.31). The concept originates from theoretical physics, where a stable conservative system's "energy" is a positive definite scalar that diminishes with time. In order to investigate the stability of nonlinear systems, we develop a generalized "energy-like" function known as a Lyapunov function.

Theorem 2.1: Let $D \subset R^n$ be a domain containing the system origin, and let $V(x) : D \to R$ be a continuously differentiable function such that:

$$V(0) = 0, \quad V(x) > 0, \ \forall x \neq 0 \tag{2.34}$$

And

$$\dot{V}(x) \leq 0 \tag{2.35}$$

$x = 0$ is hence stable. Furthermore, x = 0 is asymptotically stable if

$$\dot{V}(x) < 0, \forall x \neq 0 \tag{2.36}$$

.

Remark 2.5: A Lyapunov function is a continuously differentiable function V(x) that satisfies equations (2.34) and (2.35). It should be emphasized that the aforementioned Lyapunov criterion is not helpful since it does not provide a method for figuring out the Lyapunov function. Finding a suitable Lyapunov function that proves the stability of an equilibrium point is the responsibility of the control designer. In practice, this frequently seems like a difficult effort. Furthermore, the theorem's reverse is not always true because it simply offers sufficient circumstances.

2.1.12 Low-Pass Filter [30]

2.1.12.1 Purpose of the Low-Pass Filter

i. **Ripple Reduction**: The low-pass filter (LPF) reduces the voltage ripple and high-frequency noise that come from the boost converter's output.

ii. **Stability improvement** It stabilizes the system by removing any voltage fluctuations,

thus ensuring that the output remains stable. This is particularly necessary when you have sensitive loads connected.

2.1.12.2 Types of Filters

i. **RC Low-Pass Filter Components**: Just a resistor (R) and a capacitor (C).
Key points:

- It's straightforward to design and doesn't take up much space.
- The resistor can waste a bit of power as heat, but the setup stays simple.

Cutoff Frequency (f_c):

$$f_c = \frac{1}{2\pi RC} \tag{2.37}$$

Typical Values:

- R usually lands somewhere between 1 kΩ to 10 kΩ.
- Once we pick up R, figure out C based on the cutoff frequency we want.

ii. **LC Low-Pass Filter Components**: Inductor (L) and Capacitor (C).
Characteristics:

- More efficient, especially at higher currents.
- Better performance in filtering out ripple and noise.

Cutoff Frequency (f_c) [30]:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \tag{2.38}$$

Typical Values:

- Choose L in the range of 10 mH to 100 mH.
- Calculate C based on the desired cutoff frequency.

Thus, for most boost converter applications, particularly where efficiency and output ripple are important an LC low-pass filter is typically the preferred choice.

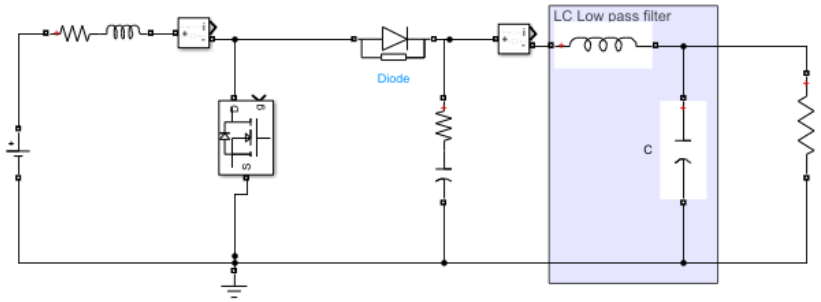

Figure 2.9: LC low pass filter

2.1.13 Particle Swarm Optimization

The collective behavior of social species, such fish schools and flocks of birds, is the source of inspiration for particle swarm optimization (PSO), a nature-inspired optimization method [31]. Kennedy and Eberhart created it in 1995. PSO has become well-known as a potent heuristic optimization method [31]. Its popularity is a result of its capacity to effectively explore intricate search areas, which makes it ideal for resolving a variety of optimization issues. Numerous fields, including as engineering, finance, biology, and machine learning, have used PSO [18]. It has been modified and expanded to tackle other optimization problems, including limited optimization, dynamic optimization, and multi-objective optimization [19].

Table 2.1: Mathematical algorithms of PSO

Step	**Description**
Initialization	Randomly initialize positions X and velocities V of particles.
Fitness Evaluation	Evaluate fitness values $f(X)$ for each particle's current position.
Personal Best Update	Update personal best positions P if $f(X)$ is better than previous best. Update global best position G as the best P among all particles.
Global Best Update	Update global best position G as the best P among all particles.
Velocity and Position Update	Update velocities V and positions X using formulas: $V = \omega V + \phi_1 R_1(P - X) + \phi_2 R_2(G - X)$ $X = X + V$
Termination	Repeat steps 2-5 until termination condition is met.

2.1.14 Gate Driver Design

2.1.14.1 Overview and Theory

Flipping a MOSFET switch doesn't work as if you just flipped a switch. There is more going on underneath. Each time you turn it on or off, the gate consumes some energy, and you also lose power as the drain source resistance (R_{DS}) changes from high to low. If you want to switch faster and produce less heat, then you have to get rid of that gate charge quickly.

Essentially, if you push more current into the gate, it will respond faster. One good way to understand this is to consider the MOSFET gate as a capacitor. This allows us to calculate the energy and power required for switching the device at a given voltage and frequency. The formula for energy stored in a capacitor is:

$$E_{\text{stored}} = \frac{1}{2}CV^2$$

And each time you switch, you lose that same amount of energy:

$$E_{\text{lost}} = \frac{1}{2}CV^2$$

So, for a complete charge and discharge, the total energy needed comes out to:

$$E_{\text{net}} = CV^2$$

Since $Q = CV$, you can also write it as:

$$E_{\text{net}} = QV$$

To find out how much power you need to switch the gate at a certain frequency, use this:

$$P = QfV$$

In simpler terms, we are interested mainly in the total gate capacitance (C_g), total gate charge (Q_g), and the voltage variation of the gate drive (ΔV_g). As a matter of example, if your driver results in a square wave going from 0 V to 15 V, then $\Delta V_g = 15\,\text{V}$. After that, knowing the power needed, we are able to calculate the lowest current the driver has to provide:

$$I_{g,\text{min}} = \frac{P}{\Delta V}$$

Generally, the MOSFET datasheet will provide you with both C_g and Q_g but remember that parameters such as C_{gg} and Q_g might change according to the voltage applied [32]. In order to get the closest value of Q_g, it is probably necessary to simulate or carry out an actual experiment in which you measure the gate current (I_g) while switching. The integration of

that current with respect to time would yield:

$$Q_g = \int I_g \, dt$$

Once you know Q_g either from measurements or the datasheet you can determine the peak gate current needed to achieve your desired switching time with this formula:

$$I_{g,\text{peak}} = \frac{Q_g}{\text{desired switch time (s)}}$$

2.1.14.2 Gate Driver Design

In this gate driver circuit, two NPN BJTs are used $'Q_2'$ acts as an inverter, while $'Q_1'$ serves to isolate the 12 V power supply from the PWM control signal.Here's how it works: when the PWM signal is **LOW**, $'Q_2'$ is turned **off**. This allows $'Q_1'$ to be biased **on**, which pulls the gate voltage (V_{gate}) **HIGH**.On the other hand, when the PWM signal goes **HIGH**, $'Q_2'$ turns **on** and pulls the base of $'Q_1'$ down to ground. This shuts $'Q_1'$ **off**, and a path opens up through diode D_1 to ground, allowing V_{gate} to discharge.This simple setup not only inverts the PWM signal but also helps cleanly switch the gate voltage while keeping the control circuitry isolated from the 12 V rail [30]. The behavior is illustrated in figure 2.10.

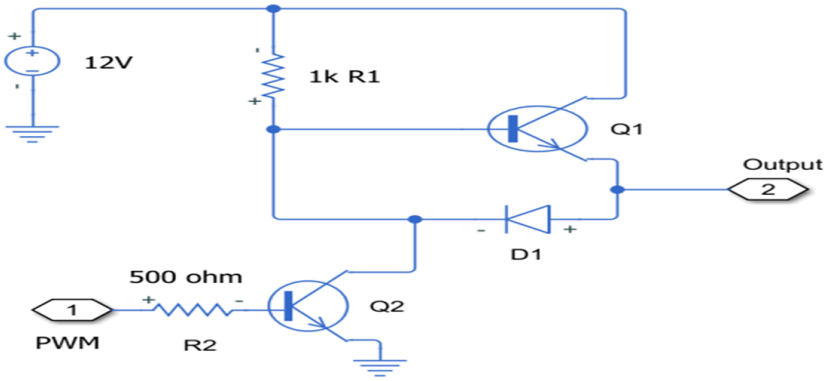

Figure 2.10: Gate driver Circuit

2.2 Review of Existing Literature

DC-DC converters are essential parts of switched mode power supplies. Their analysis, control, and stabilization are important. Although there are many control methods, practical and simple designs are often preferred in both industrial and high-performance systems. Each method has its own strengths and limitations based on operating conditions.

Tan et al. (2007) introduced a fast-response sliding-mode controller for boost converters. This controller quickly adjusts to changes in input voltage and load. It helps the system achieve the desired output while maintaining stability. However, this method experiences chattering because of the high-frequency switching needed for its quick response. The main challenge is to achieve quick regulation without excessive chattering. Too many oscillations can harm performance and reliability [33].

Gueldemir (2005) looks at how sliding mode control (SMC) works in boost converters. He points out its strength and effectiveness in keeping performance stable, even when there are disturbances or changes in system parameters. The sliding mode control method improves the system's stability across many operating conditions. This makes it particularly helpful in situations where load and input voltage changes are unpredictable. As a result, it maintains high efficiency and ensures a steady output despite external challenges. However, like Tan et al. (2007) found, one major drawback of Gueldemir's work is the possibility of chattering. The high-frequency switching that comes with the sliding mode control method can cause wear on system parts and lead to energy waste. To reduce these problems, careful design strategies are necessary, which adds complexity to the implementation process [23].

Ahmed et al. (2004) compare sliding mode control (SMC) and traditional PID control for buck converters. Their study highlights the greater robustness of SMC, especially when facing external disturbances and changing operating conditions. Unlike PID controllers, which can have trouble maintaining stable performance in these situations, sliding mode control offers a more reliable and efficient option. However, Ahmed et al. (2004) also point out the challenges of implementing SMC, particularly for buck converters. The need for careful design to avoid chattering, along with the difficulties of tuning the controller to work well at different operating points, makes sliding mode control harder to implement. This added complexity may make SMC less attractive for simpler applications or systems where easy implementation is a priority [34].

Foreyth and Mollov (1998) provide an overview of modeling and control strategies for DC-DC converters. They emphasize the need for good system modeling to achieve efficient and

stable performance. Their work highlights that understanding the system's dynamics and using precise models is essential for choosing the best control strategies. This ensures that the converter works well under different conditions. However, a significant limitation of their study is the lack of detailed exploration of modern control techniques, like sliding mode control (SMC) or adaptive controllers. While they stress the importance of accurate modeling, they do not completely address the challenges and potential issues of using contemporary control methods in real-life situations. This gap might limit the relevance of their findings for advanced or rapidly changing systems [13].

Rashid's "Power Electronics Handbook" (2001) is a guide to different control techniques. It gives helpful insights into the theoretical foundations and practical uses of methods like PID, SMC, and others in DC-DC converters and power electronics systems. The book helps readers understand how these techniques can improve system performance. However, it has a drawback. Rashid does not provide a detailed critique of the practical limitations of each control method, especially in real-time situations. While the book discusses the theoretical benefits of these control techniques, it does not fully address the real-world challenges, such as implementation issues, the need for careful adjustments, and the trade-offs in performance and complexity that can occur when applying these methods in actual systems. This gap may reduce the book's usefulness for practitioners who face the challenges of real-world implementations [14]. Racirah and Sen (1997) compare Proportional-Integral (PI), Sliding Mode Control (SMC), and Fuzzy Logic Controllers for power converters. Their study shows that SMC is strong, especially in systems with disturbances or changes in load and input. SMC can keep stability and performance in uncertain conditions, making it a better choice than PI controllers, which may have trouble in similar situations. However, the study also notes a major issue: chattering. This occurs due to high-frequency switching in SMC, which can harm system performance, wear out components, and possibly reduce the converter's lifespan. To address this, careful design strategies or filtering techniques are needed, which complicates the use of SMC [15].

Utkin's key works from 1978 and 1981 on sliding modes and control optimization set the stage for modern sliding mode control (SMC) applications. These studies highlight the benefits of SMC, especially its strength and flexibility in changing systems. Utkin's ideas have played an important role in developing SMC. However, a major drawback of these early works is their focus mainly on the theory of sliding mode control. They do not adequately address the practical challenges faced in implementing it in real-world situations. Issues like tuning the controller for specific systems, managing hardware limits, and reducing chattering

effects in real-time environments are not fully examined. This creates a gap between theory and practice, making it more difficult to apply SMC in the real world [21].

Tan et al. (2008) discuss the general design issues encountered when using sliding mode controllers (SMC) in DC-DC converters. Their work highlights how SMC can improve converter stability, especially in dynamic conditions. This makes SMC very effective for applications that need high precision and flexibility. By increasing robustness, SMC provides important performance benefits in tough environments. However, they also point out the ongoing problem of chattering, which can disrupt the controller's performance, particularly in systems that require quick adjustments. This creates a trade-off between the controller's robustness and its risk of becoming unstable due to too much switching. To address this, careful tuning and design changes are needed, which adds complexity to its use in real-world systems [33].

Spiazzi et al. (1997) discuss sliding control strategies for DC-DC converters and highlight the main benefits of sliding mode control (SMC) regarding robustness and performance. Their work shows that SMC can significantly improve both stability and performance, especially in converters that need to function under changing conditions or unpredictable disturbances. This makes SMC a useful method for systems that need to be flexible. However, they also point out a major drawback: chattering, which is a common problem with SMC. The high-frequency switching linked to SMC can cause inefficiencies, increased wear on components, and a shorter system lifespan. Reducing chattering takes careful design, which adds a lot of complexity to creating the controller [35].

Cortes and Alvarez (2002) examine the use of robust sliding mode control (SMC) in boost converters. They highlight its role in improving stability and performance under difficult operating conditions. Their findings show that robust SMC can manage disturbances and changes in parameters effectively, maintaining high performance even in unpredictable environments. However, like other studies, they recognize that dealing with chattering remains a major challenge due to the need for complex design considerations. Achieving stability without losing responsiveness requires careful tuning and design. This makes implementing robust SMC more difficult, particularly for systems with limited computing resources or strict real-time requirements [36].

B. Johansson (2005) highlights the importance of dynamic modeling for DC-DC converters. He points out that accurate dynamic models are key for developing effective control strategies. Johansson shows how these models provide a basis for designing controllers that can adjust to different operating conditions. This ultimately improves converter efficiency and stability. However, while his study offers valuable insights into dynamic modeling, it does

not explore the challenges of applying advanced control techniques in real-time systems. The study mainly emphasizes the need for accurate models but does not discuss the practical difficulties or limitations that come up when using these models in dynamic environments or when combining them with advanced control strategies [1].

C. Albea (2010) provides a detailed analysis of different control methods for electronic power converters, such as PID, adaptive, and predictive control. These techniques are noted for their ability to improve the performance and reliability of power converters, especially in systems that face changing loads and input conditions. Adaptive and predictive controllers are particularly recognized for their ability to adjust to changes, which boosts overall system efficiency. However, Albea also points out important challenges in implementation, especially the sensitivity of these methods to changes in parameters and the difficulty of tuning. The challenge of tuning adaptive and predictive controllers, especially in uncertain or disturbed conditions, can limit their practical use. Additionally, the higher computational load needed for adaptive methods may restrict their application in real-time systems with tight resource limits [2].

Zhang et al. (2014) introduce a sensorless predictive current-controlled DC-DC converter. This design lessens dependence on sensors, which cuts out delays and inaccuracies from sensor feedback. The approach greatly improves efficiency, especially in applications that require high speed or precision, by speeding up the system's response time and lowering sensor-related errors. However, Zhang et al. also point out that while the sensorless method has benefits, it can have trouble with fast changes or nonlinearities in the system. These issues can impact system stability and performance, especially during rapid or highly nonlinear operating conditions. Even with the advantages of the sensorless technique, it is still prone to instability in certain dynamic situations [24].

B. K. Bose (2000) highlights the increasing need for energy efficiency in power electronics. He emphasizes the role of improved control systems in creating more sustainable power systems. Bose explains that better control strategies for power converters can lead to significant energy savings and lower environmental impact. This positions them as essential for future applications in power electronics. However, while the work stresses the importance of efficiency, it does not specifically explore the limitations or challenges of implementing certain control methods. The lack of detailed discussion on practical issues, like system adaptability and implementation complexity, may overlook important factors that could affect the real-world use of these strategiesKurucs et al. (2015) examine state-space control methods, specifically Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG)

approaches, for boost converters. These methods provide optimal control solutions. They improve system performance by minimizing a cost function that balances control effort with system state changes. This leads to high efficiency, especially in systems that need precise control. However, LQR and LQG methods have a major downside: they require heavy computation, making them less suitable for real-time applications that need quick adjustments. Additionally, these methods may not work well under different load conditions or in environments where system parameters can change unexpectedly, limiting their use in dynamic situations [4].

Kurucs et al. (2015) examine state-space control methods, specifically Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG) approaches, for boost converters. These methods provide optimal control solutions. They improve system performance by minimizing a cost function that balances control effort with system state changes. This leads to high efficiency, especially in systems that need precise control. However, LQR and LQG methods have a major downside: they require heavy computation, making them less suitable for real-time applications that need quick adjustments. Additionally, these methods may not work well under different load conditions or in environments where system parameters can change unexpectedly, limiting their use in dynamic situations [5].

Jiao et al. (2011) present voltage-lift-type switched-inductor cells to enhance the performance of DC-DC converters. This is especially true for applications that need high voltage conversion ratios. These cells increase the voltage gain and efficiency of the converter while lowering losses. However, adding voltage-lift-type cells makes control design more complex. This complexity can make the system harder to tune and implement. It can also create challenges in keeping the system stable, particularly under changing conditions. Moreover, the greater number of components and control needs can raise the risk of instability. This makes the method less appealing for systems that prioritize simplicity and ease of control [6].

Arulselvi et al. (2004) examine the design of PID controllers for boost converters. These converters are popular because they are simple and effective. The PID control method works well to keep the output voltage and current stable. It is also easy to implement and adjust. This simplicity is a key reason why it is widely used in power converter applications. However, a significant issue pointed out by Arulselvi et al. is the presence of a right half-plane (RHP) zero. This can negatively impact system stability and performance. RHP zeros make control design more complicated and require extra tuning to achieve the desired performance. Because of this, PID controllers might not be the best option for high-performance applications, where more complex control methods are necessary to address these dynamics [34].

Guo et al. (2001) describe a digital PID controller for buck converters. They highlight its benefits in digital settings where processors handle control tasks. Digital control enables more precise and flexible tuning, which improves the buck converter's overall performance and efficiency. However, the study points out some challenges in implementation, such as quantization effects and sampling issues. These can introduce errors and reduce performance when compared to analog systems. Additionally, digital PID controllers can be sensitive to changes in parameters and load disturbances. This sensitivity can affect their robustness and lead to decreased performance, especially in dynamic or uncertain situations [37].
Mingzhi and Jianping (2007) present a nonlinear PID control strategy for digital-controlled buck converters. They emphasize its ability to improve performance under changing conditions. The nonlinear PID method can respond better to system changes, which may improve both efficiency and stability, especially in converters facing unpredictable loads or input conditions. However, the added complexity of using nonlinear control laws creates significant challenges. This complexity can make the controller harder to tune and implement, which might reduce the benefits of improved performance. Additionally, the nonlinear aspects of the controller can raise computational demands, making it less suitable for systems with limited resources or real-time needs [12]. The literature shows a variety of controller methods for boost converters. Each method has its own advantages and limitations. PID controllers are appreciated for their simplicity; however, they can have stability problems, especially with RHP zeros. On the other hand, sliding mode control is known for its robustness and adaptability, but it often faces challenges like chattering and complex implementation.

2.2.1 Proposed SMC RTE for Boost Converter

In order to solve the problems inherent in this thesis, we developed a Sliding Mode Control (SMC) system for the Boost Converter in a Real, Time Environment (RTE). The system was designed to comprise several important techniques for performance enhancement:

- **Cascade Control:** This method improves the system's stability and responsiveness by managing multiple variables in a clear, step by step manner.
- **Equivalent Series R_L and R_C:** These resistances are used to better model and control the power conversion process, ensuring the system runs more efficiently.
- **Low-Pass Filter:** This filter helps reduce high-frequency switching noise, improving overall performance and stability.

The Real-Time Environment (RTE) plays a key role here. It lets the system change immediately in response to shifts and disturbances. This real-time feedback ensures better control and smoother operation, even when conditions change suddenly. The main goal of this controller is to keep the output voltage stable and constant while quickly responding to any fluctuations. Additionally, the controller keeps the benefits of Sliding Mode (SM) control. It offers better accuracy when handling switching delays and also addresses the common chattering issue found in traditional SM control systems.

Chapter 3

System Modelling

This chapter presents a mathematical model of the boost converter that will be utilized in controller design and performance analysis. Kirchhoff's voltage and current rules, among other common circuit theories, serve as the foundation for the circuit model's derivation. A popular technique for analyzing and designing controller and power electronics circuits based on linear RLC elements, independent power, and a network made up of switches is the state-space averaging method. State variables include the capacitor's voltage and the inductor's current. Circuit equations are written based on the power switching of ON and OFF states using fundamental circuit concepts. Finding an average model of the circuit that corresponds to the ON and OFF states is appropriate because the switching happens at extremely high frequencies. For the state space average model, this results in the bilinear model of the boost converter [38].

3.1 Averaging Method

In reality, a DC-DC boost converter circuit operates in two modes: switch open mode and switch close mode. The state equations of two modes can be expressed as follows in state space description: Section 3.1.

$$\dot{x} = A_1 x + B_1 u \qquad 0 <= t <= DT, \qquad Switch\ ON \tag{3.1}$$

$$\dot{x} = A_2 x + B_2 u \qquad DT <= t <= T, \qquad Switch\ OFF \tag{3.2}$$

Where D is the duty cycle in one T period, which is a positive number, $0 < D \leq 1$. Then the average state space model is:

$$\dot{x} = \bar{A}x + \bar{B}u \tag{3.3}$$

Where,

$$\bar{A} = A_1 D + A_2(1 - D)$$
$$\bar{B} = B_1 D + B_2(1 - D)$$

Below is a basic proof. We assume that in both ON and OFF operating modes, the capacitor voltage and inductor current vary linearly. Given that the converter switching happens at extremely high frequencies, this makes sense. Next, we have the following for the switch ON period:

$$\int_0^{DT} \dot{x}\, dt = \int_0^{DT} (A_1 x + B_1 xu)\, dt \tag{3.4}$$
$$x(DT) - x(0) \approx (A_1 x + B_1 xu)DT \tag{3.5}$$

Likewise, during the OFF phase, we have

$$\int_{DT}^{T} \dot{x}\, dt = \int_{DT}^{T} (A_2 x + B_2 xu)\, dt \tag{3.6}$$
$$x(T) - x(DT) \approx (A_2 x + B_2 xu)(T - DT) \tag{3.7}$$

By combining equations (3.5) and (3.7)

$$x(T) - x(0) = (A_1 x + B_1 u)DT + (A_2 x + B_2 u)(1 - D)T$$
$$\dot{x} = X(T) - X(0) = (A_1 D + A_2(1 - D))x + (B_1 D + B_2(1 - D))u \tag{3.8}$$

Which are gives the state space average model.

3.2 Average Model for Boost Converter State Space

To design controllers, a precise model is necessary. The system behavior covered in the previous part is approximated in this section using the averaging method. The precise

model of a boost converter with a number of parasitic parameters is shown in Figure 3.1. The equivalent series resistor (ESR) of the capacitor, the forward voltage of the diode, and the parasitic resistor of the inductor are denoted by R_L, V_D, and R_C. When the switch is

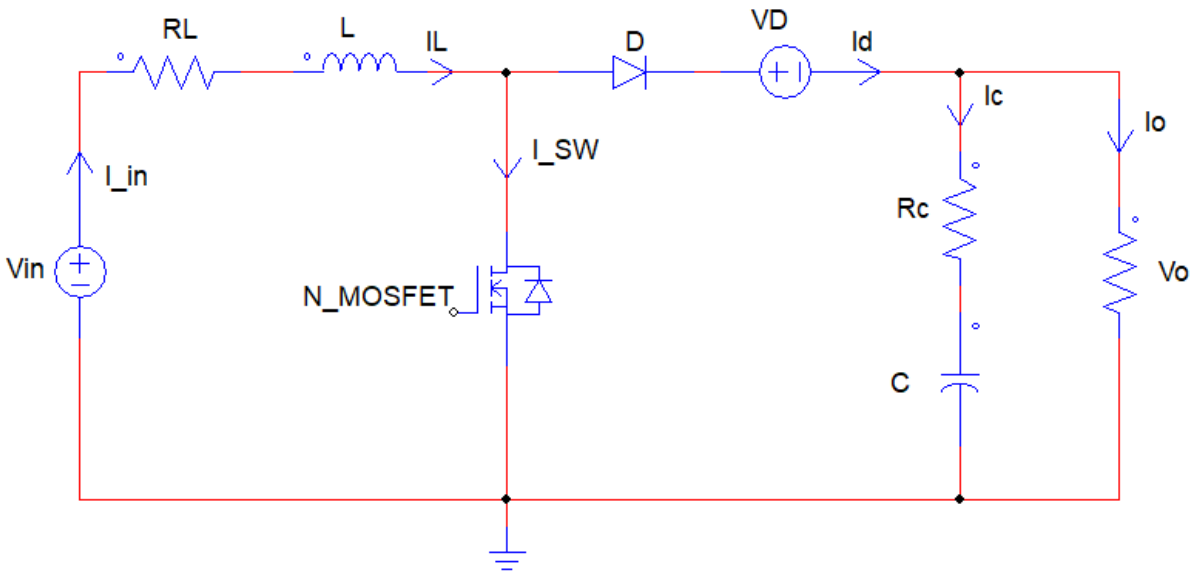

Figure 3.1: The Accurate Model of Boost Converter Circuit

closed, the battery circuit is closed through the switch, and at the same time the capacitor discharges through the load resistance [38].

i) During switch ''ON''

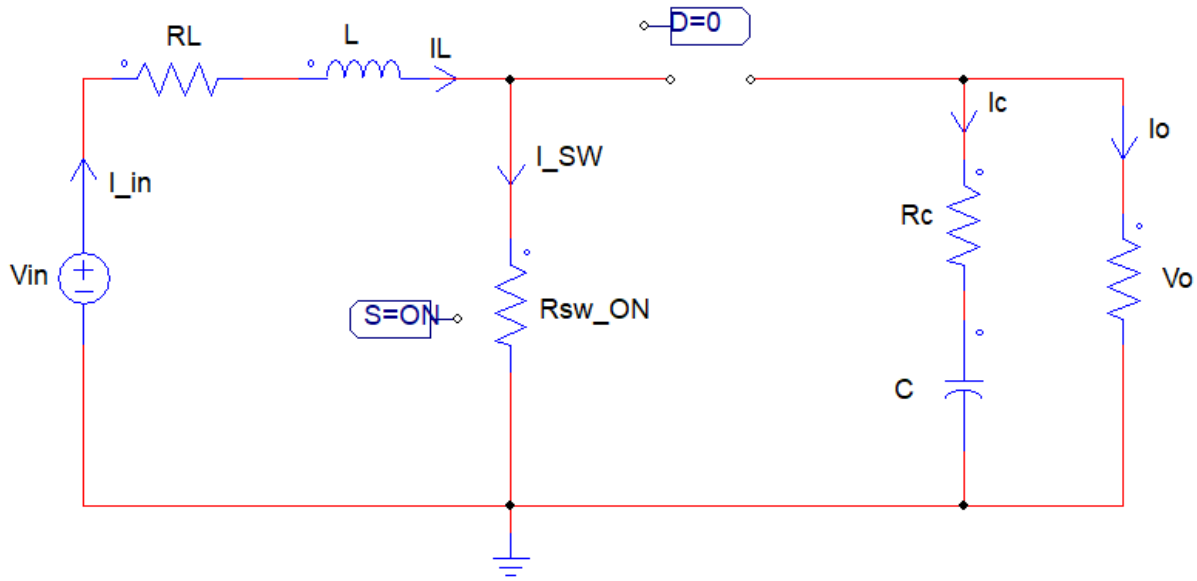

Figure 3.2: The equivalent circuit when switch is "ON"

The equivalent circuit for boost converter during interval of $0 < t \leq DT$. i.e. On period.

In this interval, the diode is OFF and the MOSFET switch is replaced by its ON-Time resistance (Ron). And the input current, inductor current and switch current are the same, i.e. I_{in} =I_L=I_{sw} are equal.

Using KVL and KCL, the fundamental equations for circuit as shown below:

$$\begin{aligned} \frac{dI_L(t)}{dt} &= V_{in} - I_L(t)(R_L + R_{ON}) \\ I_C &= -I_0 \\ I_C &= C\frac{dV_c(t)}{dt} = -\frac{V_o(t)}{R} \end{aligned} \tag{3.9}$$

To provide energy to the load, the capacitor is discharged:

$$V_o(t) = V_c(t) - V_o(t) \implies V_o(t) = \frac{R}{R + R_c}V_c(t) \tag{3.10}$$

Discharged capacitor current is $-(V_o(t))/R$. The state space equations are:

$$\begin{cases} L\frac{dI_L(t)}{dt} & = V_{in} - I_L(t)(R_{ON} + R_L) \\ C\frac{dV_C(t)}{dt} & = -\frac{V_C(t)}{R+R_C} \end{cases} \tag{3.11}$$

where V_c represents the capacitor's voltage and I_L represents the inductor's current, which is equal to I_{in}. Additionally, the output voltage V_o is expressed as follows:

$$V_o = \frac{R}{R + R_c}V_c \tag{3.12}$$

The capacitor's stored energy discharges through the diode when the switch is opened.

ii) During switch "OFF"

The equivalent circuit for boost converter during interval of $DT < t \leq T$. i.e. OFF period. In this interval, the MOSFET switch is OFF and Diode (D) is ON; Diode (D) is replaced by its equivalent model.i.e diode resistance (R_d) in series with forward drop voltage (V_D). The input current I_{in} the same as with inductor current, I_L and the switch current, I_{sw} is zero.

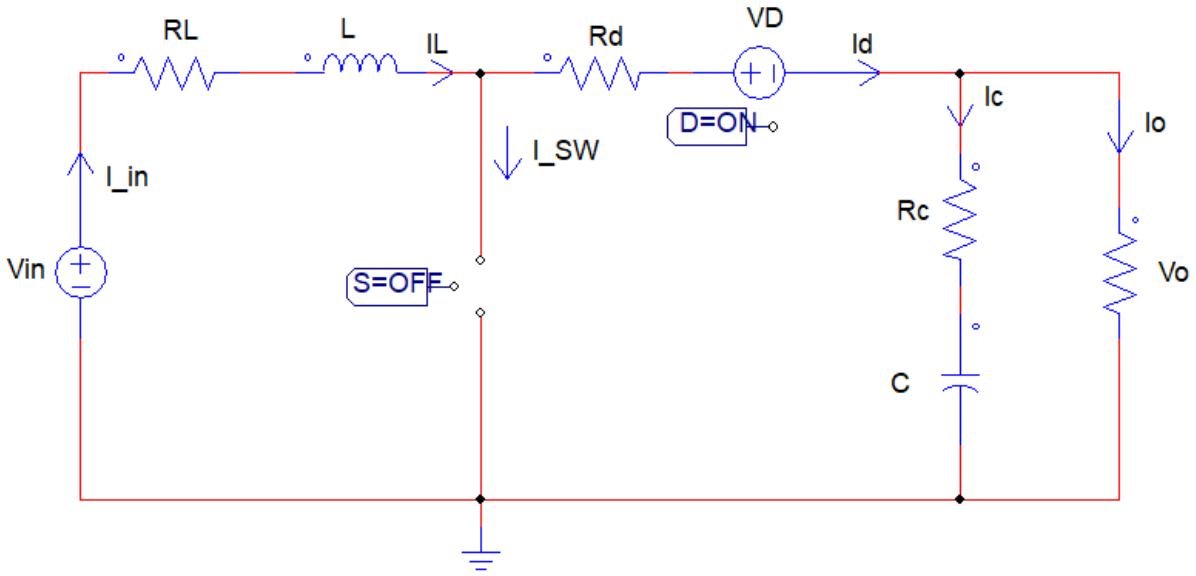

Figure 3.3: The equivalent circuit when switch is "OFF"

Using KVL and KCL, the basic fundamental equation for the circuit is shown as follows:

$$\begin{aligned} L\frac{dI_L(t)}{dt} &= V_{in} - V_D - \left((R_d + R_L + \frac{R_{RC}}{R+R_C})I_L(t) - \frac{R}{R+R_C}V_c(t)\right) \\ C\frac{dV_c(t)}{dt} &= I_L - \frac{V_o(t)}{R} \\ V_o(t) &= V_c(t) + I_L R_C \end{aligned} \tag{3.13}$$

$I_L(t) - (V_o(t))/R$ is capacitor's charging current.

The state equations are so as follows:

$$\begin{cases} L\frac{dI_L(t)}{dt} & = V_{in} - V_D - \left(\frac{R_d+R_l+\frac{R_{RC}}{R+R_C}}{R+R_C}\right) I_L(t) - \frac{R}{R+R_C}V_c(t) \\ C\frac{dV_c(t)}{dt} & = \frac{R_{RC}}{R+R_C}I_L(t) - \frac{V_c(t)}{R+R_C} \end{cases} \tag{3.14}$$

System voltage output $V_o(t)$ is represented by:

$$V_o(t) = \frac{R}{R+R_C}(V_c(t) + I_L R_C) \tag{3.15}$$

Let us assign state variables as follows: $x_1 = I_L, x_2 = V_C$ and $y = V_{\text{out}}$ also rewrite the equations (3.11),(3.12),(3.14) and (3.15) in to the state space form:

When switch closed

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{R_{ON}+R_L}{L} & 0\\ 0 & -\frac{1}{C(R+R_C)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}\frac{V_{in}}{L}\\ 0\end{bmatrix}$$
$$y = \begin{bmatrix}0 & \frac{R}{R+R_c}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{3.16}$$

When switch opened

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{RR_C+R_d(R+R_C)+(R+R_C)R_L}{L(R+R_C)} & -\frac{R}{L(R+R_C)}\\ \frac{R}{C(R+R_C)} & -\frac{1}{C(R+R_C)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}\frac{V_{in}-V_D}{L}\\ 0\end{bmatrix}$$
$$y = \begin{bmatrix}\frac{RR_C}{R+R_C} & \frac{R}{R+R_C}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{3.17}$$

Next, using the average method described in Section 3.1, the state space matrices of two distinct working modes are averaged to create the state space average model:
The average model is shown as follows:

$$\dot{x} = \bar{A}x + \bar{B}u$$
$$Y = \bar{C}x \tag{3.18}$$

Where,

$$\bar{A} = A_1D + A_2(1-D)$$
$$\bar{B} = B_1D + B_2(1-D)$$
$$\bar{C} = C_1D + C_2(1-D)$$

$$A_1 = \begin{bmatrix}\frac{-(R_{ON}+R_L)}{L} & 0\\ 0 & \frac{-1}{(C(R+R_C))}\end{bmatrix}, B_1 = \begin{bmatrix}\frac{V_{in}}{L}\\ 0\end{bmatrix}, C_1 = \begin{bmatrix}0 & \frac{R}{(R+R_C)}\end{bmatrix} \tag{3.19}$$

$$A_2 = \begin{bmatrix}\frac{-(RR_C+R_d(R+R_C)+(R+R_C)R_L)}{(L(R+R_C))} & \frac{-R}{(L(R+R_C))}\\ \frac{R}{(C(R+R_C))} & \frac{-1}{(C(R+R_C))}\end{bmatrix},$$
$$B_2 = \begin{bmatrix}\frac{(V_{in}-V_D)}{L}\\ 0\end{bmatrix}, C_2 = \begin{bmatrix}\frac{RR_C}{(R+R_C)} & \frac{R}{(R+R_C)}\end{bmatrix} \tag{3.20}$$

The state space model of boost converter becomes as follows:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}(1-D) - \frac{R_L}{L} & \frac{R_{ON}D}{L(R+R_C)} - \frac{R}{L(R+R_C)}(1-D) \\ \frac{R}{C(R+R_C)}(1-D) & -\frac{1}{C(R+R_C)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{V_D(1-D)}{L} + \frac{V_{in}}{L} \\ 0 \end{bmatrix} \tag{3.21}$$

$$y = \begin{bmatrix} \frac{RR_C}{R+R_C}(1-D) & \frac{R}{R+R_C} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.22}$$

The complete model symbolically expressed as $\dot{x} = Ax + Bu$ and $y = Cx$ which will be used in the next chapter to design a controller. First, the accuracy of the state space model is verified.

3.3 Boost Converter steady-state analysis

3.3.1 Assumptions and Steady-State Relations

For steady-state analysis, the following assumptions are considered:

- Voltage and current are assumed constant over one switching cycle.
- Voltage and currents are represented by their average steady-state values, i.e., $i_L(t) = I_L$, $v_{\text{in}} = V_{\text{in}}$, and $v_c(t) = V_C$.

The average value X of a variable $x(t)$ over a period T is given as:

$$X = \frac{1}{T}\int_0^T x(t)\,dt = Dx_{\text{ON}}(t) + D'x_{\text{OFF}}(t) \tag{3.23}$$

where $D' = 1 - D$. The variable $x(t)$ represents the voltage across or current through an element of the boost converter; the terms $Dx_{\text{ON}}(t)$ and $D'x_{\text{OFF}}(t)$ represent the variable during the switch on and off, respectively.

3.3.2 Volt-Second Balance and Charge Balance

According to Volt-Second balance theory, in steady state, the average inductor voltage must be zero.

$$V_L = D \cdot V_L(t)_{\text{ON}} + D' \cdot V_L(t)_{\text{OFF}} = 0 \tag{3.24}$$

Similarly, in steady-state, according to charge balance through the capacitor, the average capacitor current must be equal to zero.

$$I_C = D \cdot I_C(t)_{\text{ON}} + D' \cdot I_C(t)_{\text{OFF}} = 0 \tag{3.25}$$

The average output voltage is determined as follows:

$$V_o = D \cdot V_o(t)_{\text{ON}} + D' \cdot V_o(t)_{\text{OFF}} = 0 \tag{3.26}$$

3.3.3 Steady-State Output Voltage

From equations (3.24) and (3.25), we obtain steady-state relations:

$$I_L = \frac{V_o}{D'R} \tag{3.27}$$

$$V_o = V_C \tag{3.29}$$

Using equations (3.24) to (3.29), we obtain:

$$V_o = \frac{(V_{\text{in}} - D'V_D)D'R(R + R_C)}{(R_L + DR_{\text{ON}} + D'R_d)(R + R_C) + D'R(D'R + R_C)} = 11.491V \tag{3.30}$$

If all non-idealities are zero, the ideal output voltage of the boost converter is:

$$V_o = \frac{V_{\text{in}}}{D'} = \frac{V_{\text{in}}}{1 - D} = \frac{5V}{1 - 0.58} = 12V \tag{3.31}$$

3.3.4 Impact of Non-Idealities on Voltage Output

The duty cycle and input voltage alone determine the output voltage in a perfect boost converter. Equation (3.30) illustrates that, in addition to duty cycle D and input voltage V_{in}, the boost converter's output voltage V_o also depends on load resistance R and other parasitic factors, including non-idealities. Thus, if the practical boost converter's MOSFET switch is operated at a duty cycle that satisfies the relation in equation (3.31), it is evident that the practical boost converter's output voltage is always lower than the ideal boost converter.This is because the ideal boost converter relation neglects the non-idealities present in the practical boost converter. To achieve the desired output voltage from a practical boost converter, the duty cycle should be obtained by satisfying the input-output voltage relationship in equation (3.30).

3.3.5 Maximum Permissible Duty Cycle

From equation (3.30), we observe that V_o is a function of D. Therefore, by differentiating V_o with respect to D and equating it to zero, we get the information on the maximum permissible duty cycle ($D_{\max}$) as follows:

$$\begin{aligned} D'^2\left(V_o R^2 + V_D R(R+R_C)\right) - D'\left(V_o\left(R_{\text{ON}} - R_d\right)(R+R_C) - RR_C\right) \\ + V_{\text{in}} R(R+R_C) + V_o\left(R_L + R_{\text{ON}}\right)(R+R_C) = 0 \end{aligned} \tag{3.32}$$

Solving the quadratic equation (3.32) in terms of D':

$$\begin{aligned} D'^* = D'_{\text{ideal}} \Bigg(& \left[1 + \frac{V_O}{V_{\text{in}}}\left(\frac{R_{\text{ON}} - R_d - \frac{R}{R_C}}{R}\right)\right] \\ & + \sqrt{\left[1 + \frac{V_O}{V_{\text{in}}}\left(\frac{R_{\text{ON}} - R_d - \frac{R}{R_C}}{R}\right)\right]^2 - 4\frac{V_O^2}{V_{\text{in}}^2}\left[\frac{R}{R+R_C} + \frac{V_D}{V_O}\right]\left(\frac{R_L + R_{\text{ON}}}{R}\right)} \Bigg) \\ & \Big/ 2\left(\frac{R}{R+R_C} + \frac{V_D}{V_O}\right) \end{aligned} \tag{3.33}$$

The maximum permissible duty cycle can be further simplified as:

$$D_{\max} = \frac{\left[V_{\text{in}} R^2 + V_D r_1\right] - \sqrt{r_2\left[\left(V_{\text{in}} R\right)^2 + V_{\text{in}} V_D r_3 + V_D^2 r_2\right]}}{V_{\text{in}} R^2 + V_D r_3} \tag{3.34}$$

Where:

$$r_1 = (R_L + R_d)(R_C + R) + RR_C, \quad r_2 = (R_L + R_{\text{ON}})(R_C + R), \quad r_3 = (R_d - R_{\text{ON}})(R_C + R)$$

By substituting the equation (3.34) into equation (3.30), we can obtain the maximum achievable voltage ($V_{o\text{-}max}$) of the boost converter.

3.4 Power Loss Calculation

In the boost converter, there are different losses occur in the following components:

- MOSFET
- Power diode

- Inductor
- Capacitor

3.4.1 MOSFET Losses

There are two types of losses in the MOSFET:

- Conduction losses
- Switching losses

3.4.1.1 Conduction Losses

When any device conducts the current, heat is generated, and power loss occurs across it. The conduction loss is expressed as:

$$P_{\text{cond}} = I_{\text{rms}}^2 R_{\text{ds(on)}} \tag{3.35}$$

When the switch is "ON," $I_{\text{Sw}} = I_{\text{in}}$ for $0 < t < DT$.

When the switch is "OFF," $I_{\text{Sw}} = 0$ for $D < t < T$.

Therefore:

$$I_{s(\text{rms})} = \sqrt{\frac{1}{T}\int_0^T I_s^2\, dt} \tag{3.36}$$

Solving for $I_{s(\text{rms})}$:

$$I_{s(\text{rms})} = \sqrt{\frac{1}{T}\left(\int_0^{DT} I_{\text{in}}^2\, dt + 0\right)} = I_{\text{in}}\sqrt{D} \tag{3.37}$$

Thus, the RMS value of current is:

$$I_{s(\text{rms})} = I_{\text{in}}\sqrt{D} \tag{3.38}$$

Substitute into the conduction loss equation (3.35):

$$P_{\text{cond}} = I_{\text{in}}^2 R_{\text{ds(on)}} D \tag{3.39}$$

Now, using the relation $\frac{I_{\text{in}}}{I_{\text{out}}} = \frac{1}{1-D}$, we have:

$$I_{\text{in}} = \frac{I_{\text{out}}}{1-D} \tag{3.40}$$

Substituting into the conduction loss equation (3.39):

$$P_{\text{cond}} = I_{\text{out}}^2 R_{\text{ds(on)}} \frac{D}{(1-D)^2 R^2} \tag{3.41}$$

3.4.1.2 Switching Losses

During the switch ON and OFF, some power is lost. The total switching power loss is given by:

$$P_{\text{sw}} = I_d^2 R_{\text{ds}} \tag{3.42}$$

Assuming the loss during switch OFF is negligible, the switching loss during ON is:

$$P_{\text{sw}} = I_d^2 R_{\text{ds}} = 0.00339\,\text{W} \tag{3.43}$$

Therefore, the total MOSFET losses are the sum of equations (3.41) and (3.43):

$$P_{\text{total, MOSFET}} = P_{\text{cond}} + P_{\text{sw}} = 0.00899\,\text{W} \tag{3.44}$$

3.4.2 Power Diode Losses

The power diode has two losses:

- Conduction losses
- Forward voltage losses

3.4.2.1 Conduction Losses

The conduction loss is given by:

$$P_{\text{cond}} = I_{\text{d(rms)}}^2 R_{d(f)} \tag{3.45}$$

When the switch is ON, the diode does not conduct: $I_d = 0$ for $0 < t < DT$.

When the switch is OFF, the diode conducts the current: $I_d = I_{\text{in}}$ for $DT < t < T$.

Thus:

$$I_{\text{rms(d)}} = \sqrt{\frac{1}{T}\int_0^T I_d^2\, dt} = I_{\text{in}}\sqrt{1-D} \tag{3.46}$$

Substituting the equation (3.46) into (3.45) conduction loss becomes:

$$P_{\text{cond}} = \frac{V_{\text{out}} R_f(d)}{(1-D)^2 R^2} = 0.001224\,\text{W} \tag{3.47}$$

3.4.2.2 Forward Voltage Losses

The forward voltage loss is given by:

$$P_{\text{d(forward)}} = V_f I_{\text{out}} \tag{3.48}$$

$$P_{f(d)} = \frac{V_f V_{\text{out}}}{R} = 0.369\,\text{W} \tag{3.49}$$

Therefore, the total power losses is the sum of equation (3.47) and (3.49):

$$P_{\text{total, diode}} = P_{\text{cond}} + P_{\text{d(forward0}} = 0.3702\,\text{W} \tag{3.50}$$

3.4.3 Inductor Losses

The inductor in the boost converter stores energy. It is connected to the input side and is charged during the switch ON and discharged during the switch OFF. During this operation, some energy is lost across it due to parasitic inductive resistance R_L. The copper loss in the inductor is given by:

$$P_{\text{cu}} = I_{\text{rms}}^2 R_L \tag{3.51}$$

When the switch is ON, the inductor stores the energy, and $I_L = I_{\text{in}}$ for $0 < t < DT$.
When the switch is OFF, the inductor discharges through the diode, and $I_L = I_{\text{in}}$ for $DT < t < T$.

Substituting the RMS current value from equation (3.46):

$$P_{\text{cu}} = \frac{V_{\text{o}}^2 R_L}{(1-D)^2 R^2} = 0.2411\,\text{W} \tag{3.52}$$

3.4.4 Capacitor Losses

The capacitor is connected across the load to maintain the voltage level constant. The power loss in the capacitor is given by:

$$P_{\text{cap}} = I_{\text{c(rms)}}^2 R_c \tag{3.53}$$

The RMS of capacitor current equation is:

$$I_{c(\text{rms})} = I_{\text{out}}\sqrt{\frac{D}{1-D}} \tag{3.54}$$

Therefore, by substituting equation (3.54) into (3.53) the capacitor power loss is:

$$P_{\text{cap}} = \frac{V_o^2 D R_c}{(1-D)R^2} = 0.01177\,\text{W} \tag{3.55}$$

3.4.5 Total Losses and Efficiency

The total losses in the boost converter are the sum of equation (3.44),(3.50),(3.52) and (3.55) are:

$$P_{\text{total losses}} = P_{\text{MOSFET}} + P_{\text{diode}} + P_{\text{inductor}} + P_{\text{capacitor}} \tag{3.56}$$

The efficiency of the boost converter is given by:

$$\eta\% = \frac{\text{Output Power}}{\text{Output Power} + \text{Total Losses}} \times 100\% = 94.5\% \tag{3.57}$$

Where, output power is:

$$P_{\text{out}} = \frac{V_{\text{out}}^2}{R} \tag{3.58}$$

Therefore, with the above mathematical analysis, we can find the losses in the boost converter and as we can seen in the equation (3.57) the efficiency of converter is good.

3.5 Components design and selection

3.5.1 Boost converter components selection

The above modeled boost converter's parameter selection is shown in this section. As nominal working conditions, the input voltage (V_{in})$and input current (I_L) are chosen to be 5 volts and 2.2 amperes$, and 500 mΩ. Based on this, we've selected the inductor's ESR, R_L, to be 0.05 Ω. Similarly, for better ripple reduction of the output voltage, we've chosen the ESR of the capacitor to be in the range of 10 mΩ to 100 mΩ. After considering this, we selected the capacitor's ESR, R_C, to be 0.01 Ω. The nominal output voltage is chosen as 12 volts. The following formulas are used to determine the inductor (L) and capacitor (C):

$$I_{\text{out}} = I_{\text{in}}(1-D) \tag{3.59}$$

$$I_{\text{out}} = 2.2\,\text{A} \times (1 - 0.58) = 0.924\,\text{A} \tag{3.60}$$

The load resistor can be calculated as:

$$R = \frac{V_{out}}{I_o} = \frac{12V}{0.924A} = 13\,\Omega \tag{3.61}$$

For the inductor selection, the formula is:

$$L > \frac{DV_{\text{in}}(1 - D)}{2f_{sw}I_{\text{out}}} \tag{3.62}$$

Substituting the values in the equation (3.62) become:

$$L > \frac{0.58 \times 5 \times (1 - 0.58)}{2 \times 50000 \times 0.924} = 13.18 \times 10^{-6}\,\text{H} \tag{3.63}$$

Thus, $L > 13.18\,\mu\text{H}$.

For the capacitor selection, the formula is:

$$C > \frac{I_{\text{out}}}{V_{\text{ripple}}f_{sw}} \tag{3.64}$$

Substituting their values inthe equation (3.64) become:

$$C > \frac{0.924}{0.05 \times 50000} = 3.69 \times 10^{-4}\,\text{F} = 369\,\mu\text{F} \tag{3.65}$$

where, D is the nominal duty cycle selected as 58% to boost converter, V_{ripple} is the maximum allowable voltage ripple, and I_{out} is the output current calculated by input power and load resistance.

3.5.2 LC low pass filter design

For the design of an LC of LPF, let's assume:

$$f_{\text{cut}} = \frac{f_{sw}}{10} = \frac{50\text{ kHz}}{10} = 5\text{ kHz} \tag{3.66}$$

Choose the inductor value as $L = 10$ mH. Using the formula for f_{cut}, we can rearrange to find C:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \quad \Rightarrow \quad C = \frac{1}{(2\pi f_{\text{cut}})^2 L} \tag{3.67}$$

Substituting the values:

$$C = \frac{1}{(2\pi \times 5000)^2 \times 10 \times 10^{-3}} = 10\,\mu F \tag{3.68}$$

Thus, the required capacitance is $C = 10\,\mu F$.

3.5.3 MOSFET and Diode Selection [10]

3.5.3.1 MOSFET Specifications

- Output Voltage: 30V
- Load Current: 5A
- RDS(on): 0.002Ω
- Switching Frequency (f): 50 kHz
- Type: N-channel MOSFET

MOSFET Selection Considerations:

- Channel Type: The MOSFET's channel type determines the polarity of the voltage applied to the gate. Since the PWM signal from the gate driver has a positive magnitude, an N-channel MOSFET is chosen.
- Max Drain to Source Voltage (V_{DSS}): This is the voltage rating at which the MOSFET can block when it is off. A good rule of thumb is to choose a part with a voltage rating twice that of the expected voltage applied to the drain. Since the output voltage is 12V, a MOSFET rated for 24V or higher is required.
- Drain to Source Resistance ($R_{DS(on)}$): This is important because it influences heat dissipation. A lower $R_{DS(on)}$ will reduce the heat and power loss. It is preferred to select a MOSFET with low $R_{DS(on)}$.
- Gate Threshold Voltage (V_{th}): This is the voltage required to turn the MOSFET on. It is important to match this with the gate driver voltage.
- Maximum DC Drain Current: This quantifies the maximum current the MOSFET can handle under proper cooling. Ensure the MOSFET can handle the required current by checking the safe operating area curves in the datasheet.

Conclusion: The MOSFET chosen for the design is an N-channel MOSFET with a maximum drain to source voltage of 30V, capable of handling 5A with a low $R_{DS(on)}$ of 0.002Ω, and operating at a switching frequency of 50 kHz. This ensures efficient switching and low power dissipation for the boost converter.

3.5.3.2 Diode Specifications

- Part Number: IN5820
- Maximum Voltage: 30V
- Maximum Current: 5A
- Type: Schottky Diode
- Forward Voltage Drop at Peak Current: 0.5V
- Reverse Recovery Time (Rd): 1mΩ

Table 3.1: Design values for the boost converter [31]

No.	Components	Value	Unit	Description
1	V_{in}	5	V	Input Voltage
2	V_{out}	12	V	Output Voltage
3	L	9	mH	Inductance
4	C	1	mF	Capacitance
5	R_{Load}	13	Ω	Load Resistance
6	R_C	0.01	Ω	Capacitor parasitic resistance
7	R_L	0.05	Ω	Inductor parasitic resistance
8	P_{in}	11	W	Input power
9	f	50	kHz	Switching frequency
10	D	0.58	-	Nominal duty cycle
11	V_{ripple}	0.05	V	Maximum allowable voltage ripple
12	R_{on}	0.001	Ω	MOSFET switch on resistance
13	R_d	0.001	Ω	Diode resistance
14	V_D	0.5	V	Diode forward voltage drop

3.6 Average Model System Verification

The open-loop response simulation results for the average model and the switch model are shown in this section. Equation (3.21) describes the average model, while equations (3.16) and (3.17) describe the switch model. Figure 3.4 compares these two models to verify that the average model can accurately depict the switch model. The simulation makes use of nominal duty cycle and input voltage. Both the switching model (saw-wave) and the average model (solid wave) approach the anticipated output voltage and inductor current values, as seen in Figure (3.4). It is evident that the average model provides a very good approximation of the switch model, suggesting that the average model functions as intended. For the remainder of the thesis, controller performance will be designed and assessed using this model.

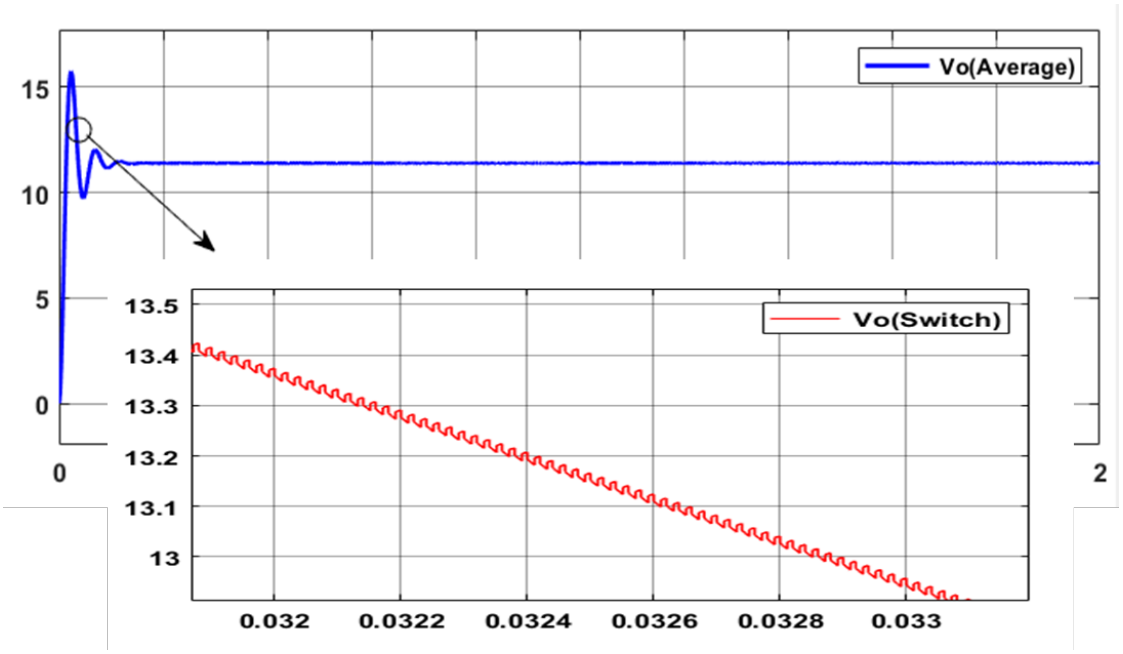

Figure 3.4: Average Model Verification for Boost Converter

Chapter 4

Controller Design

4.1 Introduction

This section presents the controller design utilizing the dynamic boost converter model for both the PID and the SMC. In this case, (PID) is utilized to compare SMC. Additionally, the trial and error method is used for PID, and particle swarm optimization (PSO) techniques are used to fine tune the controller gains for SMC. The boost converter's general block diagram is displayed in the accompanying figure 4.1.

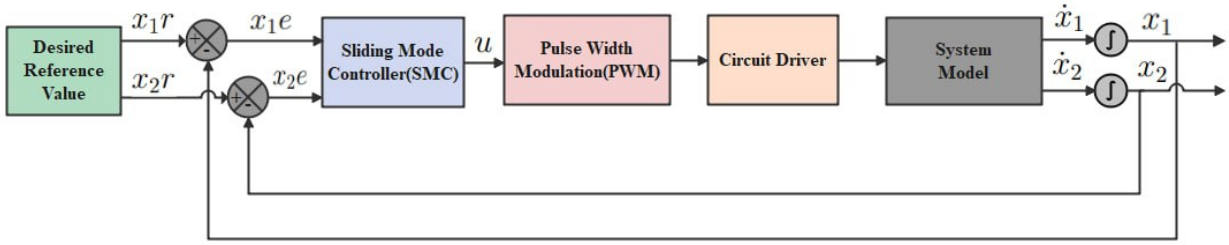

Figure 4.1: General block diagram of boost converter

4.1.1 PID Controller Design

In this section, we'll design a PID controller for the boost converter model. The block diagram below illustrates the overall workflow of the PID controller, as shown in the figure 4.2.

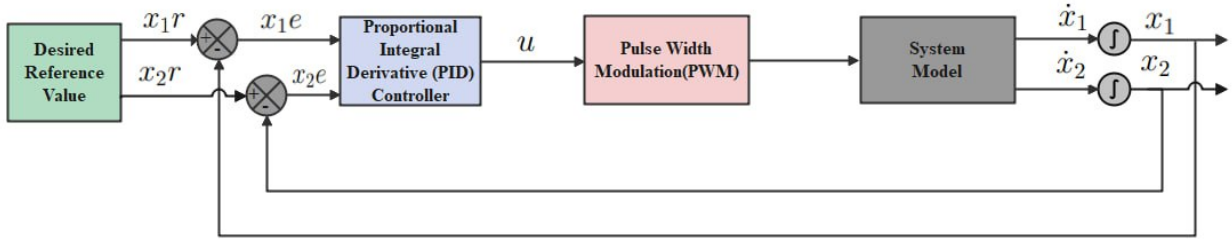

Figure 4.2: PID Controller Closed Loop Diagram

4.1.1.1 Linearization of Average Model

The average model shown below is nonlinear because the input variable, duty cycle D, is embedded inside the state space matrix. To implement a PID controller for the boost converter using linear controller design techniques, however, the average model must be linearized. Euler's approach is used for the simplest linearization method. It is assumed that the duty cycle (ΔD) , capacitor voltage (Δx_2), and inductor current (Δx_1) are appropriately small increments. Using these increments in the state-space model of the boost converter, we can proceed with the linearization:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}(1-D) - \frac{R_L}{L} & \frac{R_{ON}D}{L(R+R_C)} - \frac{R}{L(R+R_C)}(1-D) \\ \frac{R}{C(R+R_C)}(1-D) & -\frac{1}{C(R+R_C)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{V_D(1-D)}{L} + \frac{V_{in}}{L} \\ 0 \end{bmatrix} \tag{4.1}$$

$$y = \begin{bmatrix} \frac{RR_C}{R+R_C}(1-D) & \frac{R}{R+R_C} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4.2}$$

For x_1:

$$\begin{aligned} \dot{x_1} + \Delta\dot{x_1} = & \left(-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}(1-(D+\Delta D)) - \frac{R_L}{L} \right)(x_1 + \Delta x_1) \\ & + \left(\frac{R_{ON}(D+\Delta D)}{L(R+R_C)} - \frac{R}{L(R+R_C)}(1-(D+\Delta D)) \right)(x_2 + \Delta x_2) \\ & - \frac{V_D(1-(D+\Delta D))}{L} \end{aligned} \tag{4.3}$$

For x_2:

$$\dot{x_2} + \Delta\dot{x_2} = \frac{R}{C(R+R_C)}(1-(D+\Delta D))(x_1+\Delta x_1) - \frac{1}{C(R+R_C)}(x_2+\Delta x_2) \tag{4.4}$$

The following linearized equations are obtained by removing the second-order terms and canceling the equalities:

$$\begin{aligned}\Delta\dot{x_1} = &\left(-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}(1-D) - \frac{R_L}{L}\right)\Delta x_1 \\ &+ \left(\frac{R_{ON}D}{L(R+R_C)} - \frac{R}{L(R+R_C)}(1-D)\right)\Delta x_2 \\ &+ \left(\frac{RR_CI_{ss} + V_{ss}(R_{ON}+R)}{L(R+R_C)} + \frac{V_D}{L}\right)\Delta D\end{aligned} \tag{4.5}$$

$$\begin{aligned}\Delta\dot{x_2} = &\frac{R}{C(R+R_C)}(1-D)\Delta x_1 \\ &- \frac{1}{C(R+R_C)}\Delta x_2 + \frac{RI_{ss}}{C(R+R_C)}\Delta D\end{aligned} \tag{4.6}$$

The linear model of the boost converter is obtained by rewriting the aforementioned equations (4.5) and (4.6) in matrix form:

$$\begin{bmatrix}\Delta\dot{x_1}\\ \Delta\dot{x_2}\end{bmatrix} = \begin{bmatrix}-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}(1-D) - \frac{R_L}{L} & \frac{R_{ON}D}{L(R+R_C)} - \frac{R}{L(R+R_C)}(1-D)\\ \frac{R}{C(R+R_C)}(1-D) & -\frac{1}{C(R+R_C)}\end{bmatrix}\begin{bmatrix}\Delta x_1\\ \Delta x_2\end{bmatrix} + \begin{bmatrix}\frac{RR_CI_{ss}+V_{ss}(R_{ON}+R)}{L(R+R_C)} + \frac{V_D}{L}\\ -\frac{RI_{ss}}{C(R+R_C)}\end{bmatrix}\Delta D \tag{4.7}$$

where D is the steady-state duty cycle and V_{ss} and I_{ss} represent the steady state values of V_{out} and I_L. Additionally, the following is the linearization of the output equation:

$$\Delta y = \begin{bmatrix}\frac{RR_C}{R+R_C}(1-D) & \frac{R}{R+R_C}\end{bmatrix}\begin{bmatrix}\Delta x_1\\ \Delta x_2\end{bmatrix} + \frac{RR_CI_{ss}}{R+R_C}\Delta D \tag{4.8}$$

The continous time transfer function of linearized model is:

$$G(s) = \frac{-2195s + 5.571 \times 10^5}{s^2 + 83s + 2.004 \times 10^4} \tag{4.9}$$

Pole-Zero Mapping

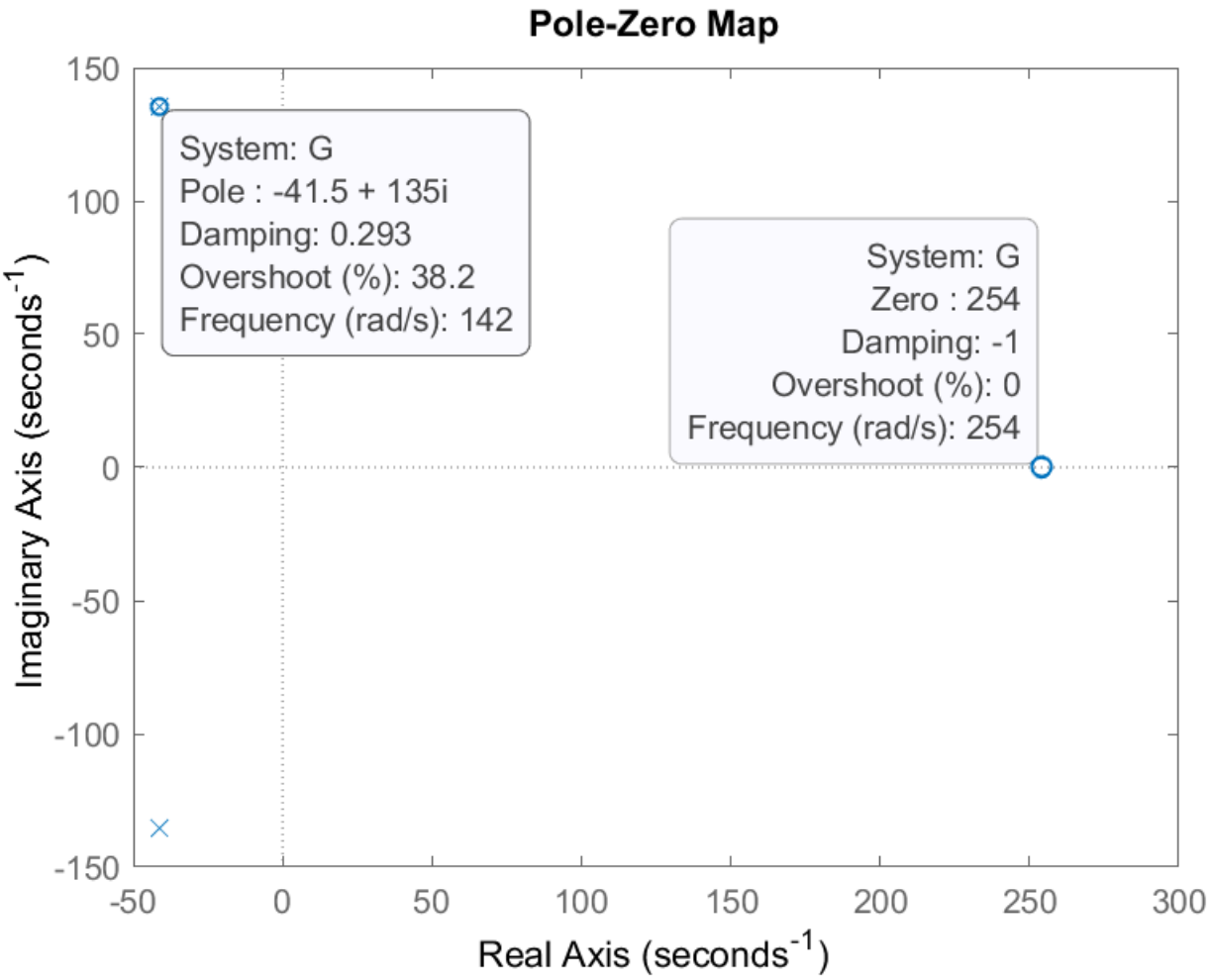

Figure 4.3: Linearized System Pole and Zero Map

Figure 4.3 shows that the system has a large zero at s = 254, which causes a large overshoot. Furthermore, since this is a non-minimum phase system, high gain control is not possible because it will cause instability in the closed loop system. The system's control objective is addressed in two ways: first, by regulating the duty cycle to track the intended output voltage, and second, by rejecting the load disturbance. Integral control is required to remove any steady state error because high gain control is not used.

The PID gains obtained based on the 'Trial and Error Method' are described as follows: The 'trial and error' method for finding the gains of a PID (Proportional Integral Derivative) controller involves manually adjusting the PID parameters: proportional gain (K_p), integral gain (K_i) and derivative gain (K_d) until the performance of the system meets the desired

criteria, such as stability, fast response, and minimal overshoot. The closed-loop system's root locus and Bode plot are then displayed in figures 4.4 and 4.5, respectively. The controller zero at the origin and the pole near the origin can make the system stable and enable error-free tracking of the reference since a gain and an integration element are added to the system. For the closed-loop system, the phase margin is about $-81.32°$ and the gain margin is about $-28.44\,\text{dB}$.Therefore ,the closed-loop system is unstable.Because of the phase margin and gain margin are negative.

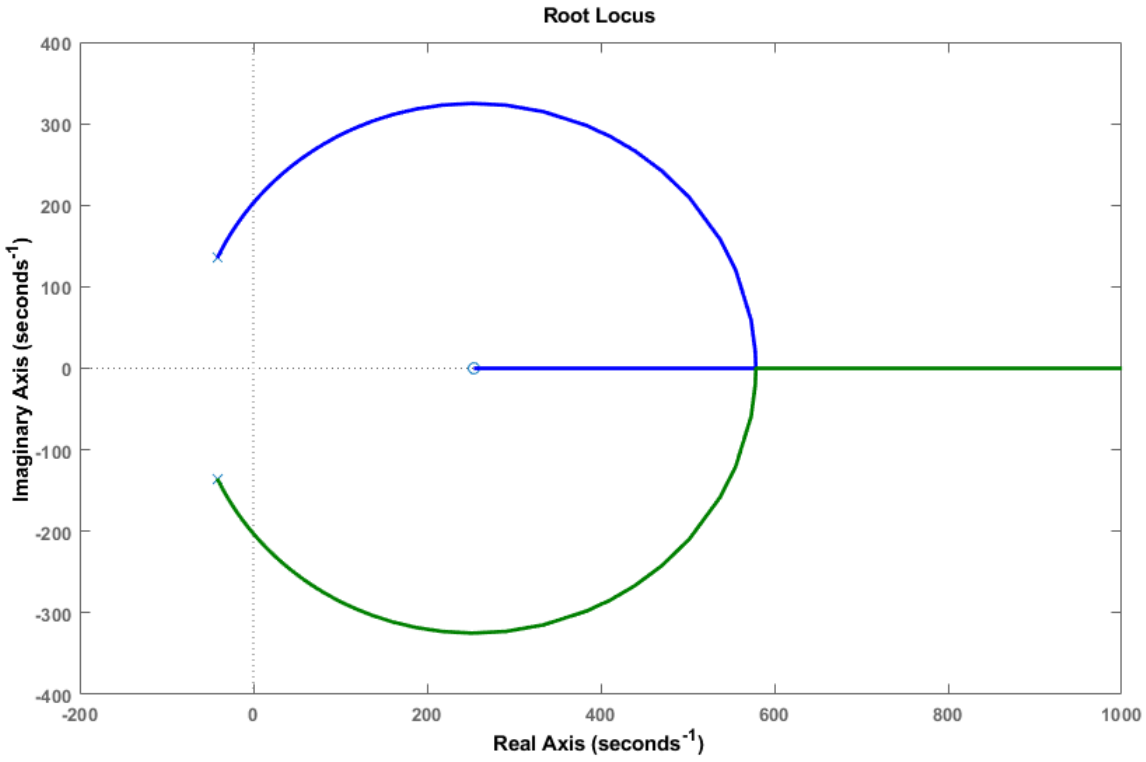

Figure 4.4: Root Locus Plot

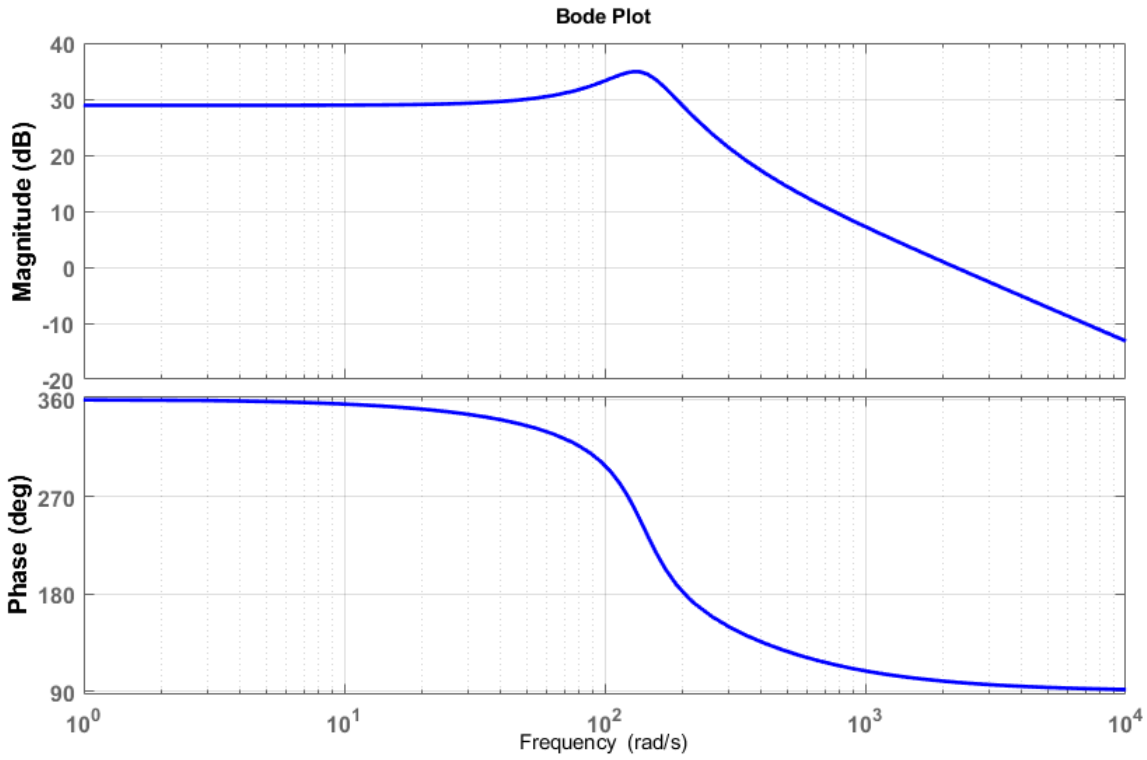

Figure 4.5: Bode Plot

4.1.2 Sliding Mode Controller design

In this section, the dynamic tracking error for both voltage and current control is eliminated using the SMC method. In order to improve stability and chattering problem of dynamic system, the Cascade control method is employed. Cascade control uses two or more feedback loops arranged in a layered setup. The inner loop takes care of the fast dynamics of the system, while the outer loop handles the slower, overall behavior. In a cascade SMC system, the outer loop is responsible for controlling the output voltage, while the inner loop manages current within the system as shown in the figure: 4.6. Therefore, SMC is designed based on the following dynamic equations of the boost converter's average model, which are presented in state-space form:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}(1-u) - \frac{R_L}{L} & \frac{R_{on}u}{L(R+R_C)} - \frac{R}{L(R+R_C)}(1-u) \\ \frac{R}{C(R+R_C)}(1-u) & -\frac{1}{C(R+R_C)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{V_D(1-u)}{L} + \frac{V_{in}}{L} \\ 0 \end{bmatrix} \tag{4.10}$$

The output voltage equation is:

$$y = \begin{bmatrix} \frac{RR_C}{R+R_C}(1-u) & \frac{R}{R+R_C} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4.11}$$

Where:

- x_1 is the inductor current
- x_2 is the capacitor voltage
- y is the output voltage
- u is the duty cycle

The duty cycle u in Sliding Mode Control (SMC) is usually calculated using the following rule: $u = \begin{cases} 1 & \text{if } s < 0 \\ 0 & \text{if } s > 0 \end{cases}$ where the sliding surface is represented by s. Controlling the output voltage y to a reference voltage $x_{2_{\text{ref}}}$ is the aim of boost converter control. The selection of the sliding surface is the first step in the design of the boost converter's sliding mode controller. As shown in [13] , it is clear that the surface $x_2 - x_{2_{\text{ref}}}$ can tend to zero only if the current increases continuously. Usually, a cascade control structure is applied to control the boost converter, which leads to solving the control problem using two control loops [14, 15, 21]. The outer voltage loop generates the reference current from the voltage error, and the inner current loop controls the inductor current via sliding mode figure 4.6.

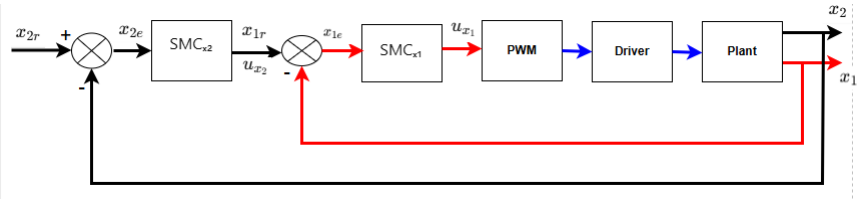

Figure 4.6: Cascade Sliding Mode Control Structure

The control of the output voltage of the DC-DC converter meets the criteria of stability and existence of sliding mode. However, because sliding mode control (SMC) is a highly nonlinear method, it has been demonstrated in [22] that it is challenging to determine the gains of the voltage loop. Additionally, the voltage loop will be more vulnerable to high-frequency phenomena and reference current uncertainties because SMC is only used for

current regulation. We suggest investigating a control mode based on a sliding surface that involves output voltage in order to enhance the controller's performance.

Steps in the Design of Sliding Mode Control for System Dynamics

The inner current control loop (SMC) and the outer voltage control loop (SMC) are the two loops used in cascade control.

The SMC's Inner Current Control Loop

The inner current control loop's goal is to control the inductor current x_1 to a target reference value $x_{1_{\text{ref}}}$.

- Let the current tracking error be defined as:

$$e(t)_{x_1} = x_{1_{\text{ref}}} - x_1 \tag{4.12}$$

- Select PI sliding surface of the inductor current is defined as:

$$s(t)_{x_1} = e(t)_{x_1} + \lambda_{x_1} \int e(t)_{x_1}\, dt \tag{4.13}$$

where λ_{x_1} is the slope of the sliding surface or the coefficient of the sliding surface. These coefficients must be chosen to ensure that the sliding mode exists at least around the desired equilibrium point, and the dynamics of the system will reach the surface and lead toward the equilibrium point.

Existence Condition

The existence condition of sliding mode implies that both $s(t)$ and $\dot{s(t)}$ will tend to zero as $t \to \infty$, meaning that the dynamics of the system will stay on the sliding surface. The existence condition for the sliding mode is:

$$s(t)\dot{s(t)} < 0 \quad (\text{when } s(t) \to 0),$$

where the fulfillment of this inequality ensures the existence of sliding mode around the commutation surface. Let us write the model of the boost converter equation (3.21) in state-space form, where the equilibrium point is the origin.We obtain:

$$\begin{aligned} \dot{x}_1 &= -\frac{R_d}{L}x_1 - \frac{RR_C}{L(R+R_C)}(1-u)x_1 - \frac{R_L}{L}x_1 + \frac{R_{on}u}{L(R+R_C)}x_2 \\ &\quad - \frac{R}{L(R+R_C)}(1-u)x_2 + \frac{V_{in}}{L} - \frac{V_D(1-u)}{L} \\ \dot{x}_2 &= \frac{R}{C(R+R_C)}(1-u)x_1 - \frac{1}{C(R+R_C)}x_2 \end{aligned} \tag{4.14}$$

By calculating $S\dot{S}$ to be negative for $u = 0$ and $u = 1$, we can deduce that the sliding region is limited by the following inequalities: For $u = 0$:

$$\dot{s}_{x_1} = \dot{e}_{x_1} + \lambda_{x_1} e_{x_1} < 0 \tag{4.15}$$

Substituting e_{x_1} from Equation (4.12) into Equation (4.16):

$$\dot{s}_{x_1} = -\dot{x}_1 + \lambda_{x_1}(x_{1_{\text{ref}}} - x_1) < 0 \tag{4.16}$$

Substituting Equation(4.14) into Equation (4.16):

$$\dot{s}_{x_1} = -\left[-\left(\frac{R_d}{L} + \frac{R_L}{L} + \frac{RR_C}{L(R+R_C)}\right)x_1 - \frac{R}{L(R+R_C)}x_2 + \frac{V_{in} - V_D}{L}\right] + \lambda_{x_1}(x_{1_{\text{ref}}} - x_1) < 0 \tag{4.17}$$

For $u = 1$:

$$\dot{s}_{x_1} = -\dot{x}_1 + \lambda_{x_1}(x_{1_{\text{ref}}} - x_1) > 0 \tag{4.18}$$

Substituting Equation (4.10) into Equation (4.18):

$$\dot{s}_{x_1} = -\left[-\frac{R_d + R_L}{L}x_1 + \frac{R_{on}}{L(R+R_C)}x_2 + \frac{V_{in}}{L}\right] + \lambda_{x_1}(x_{1_{\text{ref}}} - x_1) > 0 \tag{4.19}$$

From these two inequalities, and to ensure that the sliding mode exists at least around the equilibrium point $(x_1 = 0, x_2 = 0)$, the following condition must be satisfied:

$$\lambda_{x_1} = \frac{V_{in} - V_D}{L x_{1_{\text{ref}}}} \tag{4.20}$$

4.1.2.1 Stability condition

The stability of the system is guaranteed if the dynamics of the system in the sliding regime are directed toward the desired equilibrium point. Our aim is to determine the dynamics of the state variables x_1 and x_2 when the sliding regime is reached. The equivalent average control that must be applied to the system in order for the system state to slide along the commutation surface (4.15), the state-space model (4.14), and from $\dot{s} = 0$ is given by:

Step 1: Differentiating the sliding mode surface form given in equation (4.13)

$$\dot{s_{x_1}} = -\dot{x_1} + \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right) = 0 \tag{4.21}$$

Step 2: Substituting equation (4.14) into (4.21)

Let $u_{\text{eq}} = 1 - u$

$$\begin{aligned} \dot{s_{x_1}} = &\left(-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}u_{\text{eq}} - \frac{R_L}{L}\right)x_1 + \\ &\left(\frac{R_{\text{on}}}{L(R+R_C)} - \frac{R_{\text{on}}u_{\text{eq}}}{L(R+R_C)} - \frac{R}{L(R+R_C)}u_{\text{eq}}\right)x_2 + \\ &\frac{V_{\text{in}}}{L} - \frac{V_D u_{\text{eq}}}{L} - \lambda_c\left(I_{\text{ref}} - x_1\right) = 0 \end{aligned} \tag{4.22}$$

Step 3: Collect the equivalent u_{eq} terms

$$\begin{aligned} \left(-\frac{RR_C}{L(R+R_C)}u_{\text{eq}}\right)x_1 - \left(\frac{R_{\text{on}}u_{\text{eq}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}u_{\text{eq}}\right)x_2 \\ - \frac{V_D u_{\text{eq}}}{L} = \left(\frac{R_d}{L} + \frac{R_L}{L}\right)x_1 \\ - \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_c\left(x_{1_{\text{ref}}} - x_1\right) = 0 \end{aligned} \tag{4.23}$$

$$\begin{aligned} u_{\text{eq}}\left(\frac{RR_C}{L(R+R_C)}x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}\right)x_2 + \frac{V_D}{L}\right) = -\left(\frac{R_d}{L} + \frac{R_L}{L}\right)x_1 \\ + \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right) \end{aligned} \tag{4.24}$$

Step 4: The final equivalent control law becomes:

$$u_{x_1} = 1 - \frac{-\left(\frac{R_d}{L} + \frac{R_L}{L}\right)x_1 + \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right)}{\frac{RR_C}{L(R+R_C)}x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}\right)x_2 + \frac{V_D}{L}} \tag{4.25}$$

SMC's External Output Voltage Control Loop

Here, controlling the output voltage y to the intended reference value $x_{2_{ref}}$ is the aim of the outer output voltage control loop.

- Define the voltage tracking error:

$$e_{x_2} = x_{2_{\text{ref}}} - x_2 \tag{4.26}$$

- PI sliding surface design for Voltage:

$$s(t)_{x_2} = e(t)_{x_2} + \lambda_{x_2} \int e(t)_{x_2}\, dt \tag{4.27}$$

where λ_{x_2} is the slope of the sliding surface or the sliding surface coefficient of the control x_2. Differentiating equation (4.27) to get the equivalent control law u_{eq} for the output voltage control loop, we have:

$$\dot{s}_{x_2} = -\dot{e}_{x_2} + \lambda_{x_2} e_{x_2} \tag{4.28}$$

$$\dot{s}_{x_2} = \lambda_{x_2}(x_{2_{\text{ref}}} - x_2) - \dot{x}_2 = 0 \tag{4.29}$$

Substituting the dynamic equation from (4.14) into (4.29), and solving for u_{eq} for the outer control loop, we get:

$$\lambda_{x_2}(x_{2_{\text{ref}}} - x_2) - \frac{R}{C(R+R_C)}(1-u)(x_1) + \frac{1}{C(R+R_C)}x_2 = 0 \tag{4.30}$$

Solving for u_{eq}:

$$\frac{R}{C(R+R_C)}(u_{\text{eq}})x_1 = \lambda_{x_2}(x_{2_{\text{ref}}} - x_2) + \frac{1}{C(R+R_C)}x_2 \tag{4.31}$$

Thus, the equivalent control law is:

$$u_{x_2} = 1 - \frac{\lambda_{x_2}(x_{2_{\text{ref}}} - x_2) + \frac{1}{C(R+R_C)}x_2}{\frac{R}{C(R+R_C)}x_1} \tag{4.32}$$

4.1.2.2 Lyapunov Stability Analysis of SMC

For Lyapunov stability, we need to find a Lyapunov function $V(x)$, such that:

- $V(x) > 0 \quad \text{for all}\, x \neq 0$
- $\dot{V}(x) \leq 0$, ensuring that the system's energy decreases over time, which means the system is stable.

Define the Lyapunov function for each subsystem in the cascade. Let the system have two subsystems (one for the outer loop and one for the inner loop). Typically, for each subsystem

x_1 and x_2, Let us define a Lyapunov function as follows:

$$V(x) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 \tag{4.33}$$

The derivative equation (4.33) of the Lyapunov function $V(x)$ is:

$$\dot{V}(x) = x_1\dot{x}_1 + x_2\dot{x}_2 \tag{4.34}$$

Lyapunov Stability Analysis for inner loop

Let $V(x_1) = \frac{x_1^2}{2}$ be a Lyapunov candidate function of inner loop , we have: By substituting the control equivalent equation (4.25) in the state-space model (4.14), we deduce the dynamics of x_1 in the sliding regime:

$$\begin{aligned}
\dot{x_1} = (&-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)}\left(\frac{-\left(\frac{R_d}{L}+\frac{R_L}{L}\right)x_1 + \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right)}{\frac{RR_C}{L(R+R_C)}x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}\right)x_2 + \frac{V_D}{L}}\right) \\
&- \frac{R_L}{L})x_1 \\
&+ (\frac{R_{\text{on}}}{L(R+R_C)}\left(1 - \frac{-\left(\frac{R_d}{L}+\frac{R_L}{L}\right)x_1 + \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right)}{\frac{RR_C}{L(R+R_C)}x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}\right)x_2 + \frac{V_D}{L}}\right) \\
&- \frac{R}{L(R+R_C)}\left(\frac{-\left(\frac{R_d}{L}+\frac{R_L}{L}\right)x_1 + \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right)}{\frac{RR_C}{L(R+R_C)}x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}\right)x_2 + \frac{V_D}{L}}\right))x_2 \\
&+ \frac{V_{\text{in}}}{L} \\
&- \frac{V_D\left(\frac{-\left(\frac{R_d}{L}+\frac{R_L}{L}\right)x_1 + \frac{R_{\text{on}}}{L(R+R_C)}x_2 - \lambda_{x_1}\left(x_{1_{\text{ref}}} - x_1\right)}{\frac{RR_C}{L(R+R_C)}x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)}\right)x_2 + \frac{V_D}{L}}\right)}{L}
\end{aligned} \tag{4.35}$$

Let us simplify the complex part of equation (4.35):

$$\begin{aligned} \text{a'} = &\left(-\left(\frac{R_d}{L} + \frac{R_L}{L} \right) x_1 + \frac{R_{\text{on}}}{L(R+R_C)} x_2 - \lambda_{x_1}(x_{1_{\text{ref}}} - x_1) \right) \\ &\Big/ \left(\frac{RR_C}{L(R+R_C)} x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} + \frac{R}{L(R+R_C)} \right) x_2 + \frac{V_D}{L} \right) \end{aligned} \tag{4.36}$$

Simplified form of equation will be:

$$\begin{aligned} \dot{x_1} = &\left(-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)} a' \right) x_1 + \left(\frac{R_{\text{on}}}{L(R+R_C)} (1-a') - \frac{R}{L(R+R_C)} a' \right) x_2 \\ &+ \frac{V_{\text{in}}}{L} - \frac{V_D a'}{L} \end{aligned} \tag{4.37}$$

Where, 'a' is defined as the equation (4.36)

$$\dot{V}(x_1) = x_1 \dot{x_1} \tag{4.38}$$

$$\begin{aligned} x_1 \dot{x_1} = &\left(-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)} a' - \frac{R_L}{L} \right) x_1^2 \\ &+ \left(\frac{R_{\text{on}}}{L(R+R_C)} (1-a') - \frac{R}{L(R+R_C)} a' \right) x_1 x_2 \\ &+ \frac{V_{\text{in}}}{L} x_1 - \frac{V_D a'}{L} x_1 \end{aligned} \tag{4.39}$$

The term $\left(-\frac{R_d}{L} - \frac{RR_C}{L(R+R_C)} a' - \frac{R_L}{L} \right) x_1^2$, show that negative definite because of $-x_1^2 \leq 0$ for all x_1. Therefore,the given differential equation has a Lyapunov stability solution.

Discontinuous control law:

The key feature of Sliding Mode Control (SMC) is the control law, which guides the system to a specific point called the "sliding surface" and keeps it there. In the case of a boost converter, this discontinuous control law takes the form:

$$u_{\text{sw}}(t) = -k_1 \cdot \text{sign}(s(x_1)) \tag{4.40}$$

where, k_1 is a positive constant, and the sign function creates a control signal that switches between two states: on or off. This on/off switching ensures that once the system reaches the sliding surface, it remains on it, maintaining stability and control. Therefore, the total control law will be:

$$u_c(t) = u_{x_1}(t) + u_{\text{sw}}(t) \tag{4.41}$$

Lyapunov Stability Analysis for outer loop

By replacing the equivalent control equation (4.32) in the state-space model (4.14) , we deduce the dynamic of x_2 at the sliding regime:

$$\dot{x}_2 = \frac{R}{C(R+R_C)} \left(\frac{\lambda_{x_2}(x_{2_{\text{ref}}} - x_2) + \frac{1}{C(R+R_C)} x_2}{\frac{R}{C(R+R_C)} x_1} \right) x_1 - \frac{1}{C(R+R_C)} x_2 \tag{4.42}$$

Let $V(x_2) = \frac{x_2^2}{2}$ be a Lyapunov candidate function, then:

$$\dot{V}(x_2) = x_2 \dot{x}_2 \tag{4.43}$$

Substituting $\dot{x}_2$ from (4.42) into equation (4.43) :

$$\dot{V}(x_2) = x_2 \left(\frac{R}{C(R+R_C)} \left(\frac{\lambda_{x_2}(x_{2_{\text{ref}}} - x_2) + \frac{1}{C(R+R_C)} x_2}{\frac{R}{C(R+R_C)} x_1} \right) x_1 - \frac{1}{C(R+R_C)} x_2 \right) \tag{4.44}$$

Since $x_2 \left(-\frac{1}{C(R+R_C)} x_2 \right) \leq 0$ for all x_2, the expression is negative definite. Therefore, the overall expression for $\dot{V}(x_2)$ is negative definite, and the given differential equation has a Lyapunov stability solution.

Discontinuous control law:

The discontinuous switching controller is the same as in the equation control for the outer loop will be:

$$u_{\text{sw}}(t) = -k_2 \cdot \text{sign}(s(x_2)) \tag{4.45}$$

where k_2 is a positive constant,and thesign function creates a control signal that switches between two states: on or off. This on/off switching ensures that once the system reaches the sliding surface, it remains on it, maintaining stability and control. Therefore, the total control law will be:

$$u_v(t) = u_{x_2}(t) + u_{\text{sw}}(t) \tag{4.46}$$

Therefore, using equation(4.41) and (4.46) controller law ,we can do cascade control of the system.

4.1.2.3 Real-Time Environment Simulation Configuration

Time Allocation for Each Simulation Step

At a sampling frequency of 50 kHz, the time available per simulation step is [39]:

$$T_s = \frac{1}{f_s} = \frac{1}{50{,}000} = 20\ \mu\text{s}$$

Given a CPU clock speed of 2.42 GHz, the number of CPU cycles available per step is:

$$\text{CPU cycles per step} = 2.42 \times 10^9 \times 20 \times 10^{-6} = 48{,}400 \text{ cycles}$$

This computational budget must support:

- Numerical integration of the power electronics equations (stiff dynamics).
- Execution of the Sliding Mode Control (SMC) algorithm, including sign/saturation functions and switching logic
- Data logging or I/O handling
- Operating system task scheduling and overhead

On lower end CPUs complex models such as a nonlinear, switching, mode boost converter with high resolution PWM typically need more than 50, 100 μs per simulation step. There are two different types of modes for desktop real time environment simulation:

i. **Connected I/O mode**

- Connected I/O mode also known as normal mode with real time I/O
- The model runs on desktop CPU within simulink environment
- Real-time I/O is performed via SLDRT blocks (digital/analog I/O,serial, etc.)
- It is rapid prototyping without c code generation
- No need to supercomputer

ii. **Real-time (external) mode**

- This also known as standalone real-time mode
- The simulink model is compiled into c code, then built into an executable using SLDRT kernel
- Supports hard real time operation with accurate timing
- Require code generation or simulink coder
- Need supercomputer

As a result, this dissertation will employ connected I/O mode since there is no requirement for a supercomputer and C code. Setting up the real-time simulation environment for the model involves the following steps [20].

i. **Access Hardware Settings**

- Go to the `Desktop Real Time` tab and click on `Hardware Settings`.
- In the `Configuration Parameters` dialog box, select the `Solver` tab.

ii. **Set the Simulation Time:**

- In the `Start time` field, enter 0.0 to start the simulation at the beginning.
- In the `Stop time` field, specify how long we want to run the system model, such as 2 seconds.

iii. **Select the Solver Type:**

- From the `Type` list, choose `Fixed step`. This is because Simulink Coder does not support variable step solvers.

iv. **Choose a Solver**

- In the `Solver` list, pick an appropriate solver. For example, we can select the `ode5 (Dormand Prince)` solver, which is general-purpose.

v. **Set the Fixed Step Size**

- The given switching frequency f_{sw} is 50 kHz.

- Calculate Sampling time

$$\text{Sampling time} = \frac{1}{f_{\text{sw}}} = \frac{1}{50,000} = 0.00002\,\text{seconds (or 20 microseconds)}.$$

- The step size determines how frequently the simulation updates its calculations. To ensure the simulation is accurate, the step size is typically kept smaller than the sample time, especially in discrete systems.

- For a switching frequency of 50 kHz, a common approach is to set the step size to about 1/10th of the sample time. This ensures a good balance between keeping the simulation efficient and capturing the necessary details for accuracy.

- Under `Additional options`, find the fixed step size field and enter our desired sample time, such as 0.000002 seconds (20 µs), which corresponds to a sample rate of 50,000 samples per second or 50 kHz.

By following the above manuals, we will ensure that the real time environment simulation runs smoothly with the correct time-stepping and solver setting solver settings.

Chapter 5

Simulation Results

5.1 PID Simulation

This section presents simulation results using the boost converter model linearized around a set of operating points and controllers created in the previous section. MATLAB/Simulink is used to simulate the boost converter with closed-loop PID controller action. Figures display the output voltage, output current, and inductor current responses. (5.1), (5.2), and (5.3). The system with the PID controller rapidly reaches the intended steady-state value of 12 volts, as seen in Figure (5.1). The PID controller's effectiveness is confirmed by the settling time of 0.251 seconds and the overshoot of 0.22 volts. The closed-loop response under input voltage variation 2.5V and 7.5V disturbance at t=0.6 seconds was displayed in Figures (5.4), (5.1.2), and (5.5). It is evident that once the disturbance is applied, the output voltage, output current, and inductor current all rapidly return to steady-state values.The closed-loop response under load 6.5Ω and 19.5Ω disturbance at t=0.6 seconds is depicted in Figures (5.6), (5.7), and (5.8). It is evident that once the disturbance is applied, the output voltage, output current, and inductor current all rapidly return to steady-state values. The closed-loop response under controller variation 0.29 and 0.87 disturbance at t=0.6 seconds was displayed in Figures (5.14), (5.15), and (5.16). It is evident that once the disturbance is applied, the output voltage, output current, and inductor current all rapidly return to steady-state values.

5.1.1 PID Controller Performance at nominal condition

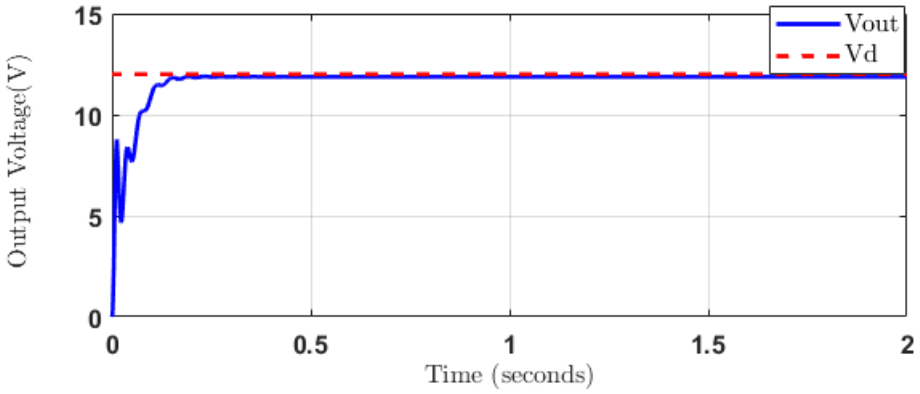

Figure 5.1: PID Controller Simulation of Output Voltage (V_o)

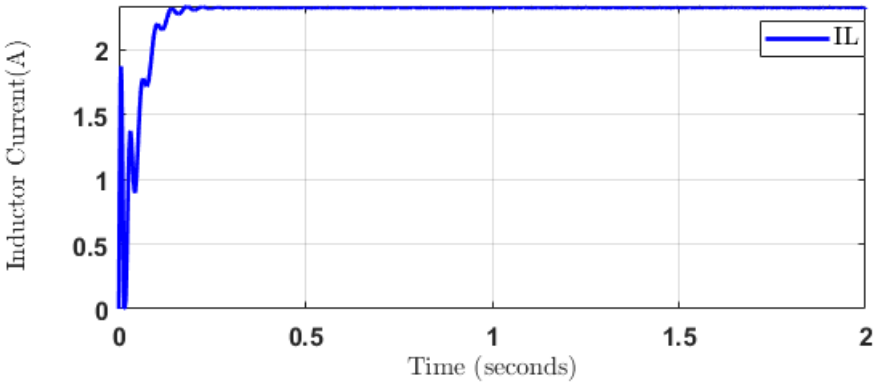

Figure 5.2: PID Controller Simulation of Inductor Current (I_L)

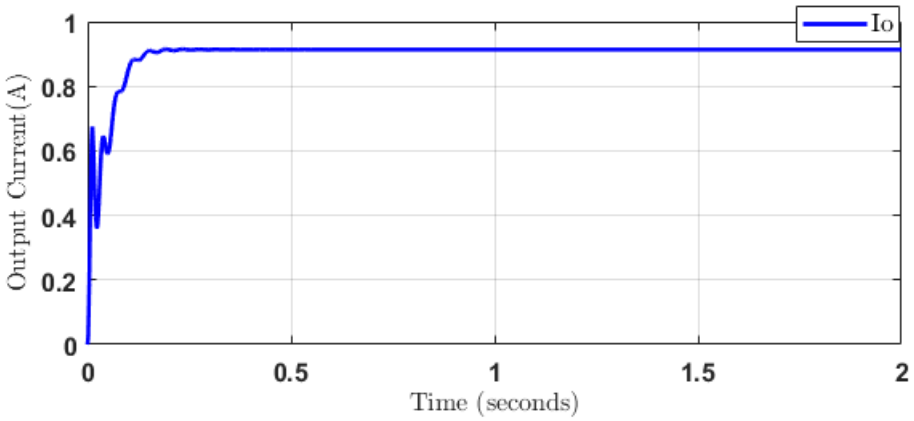

Figure 5.3: PID Controller Simulation of Output Current (I_O)

5.1.2 PID Controller Performance due to Input Voltage Variation at t=0.6sec

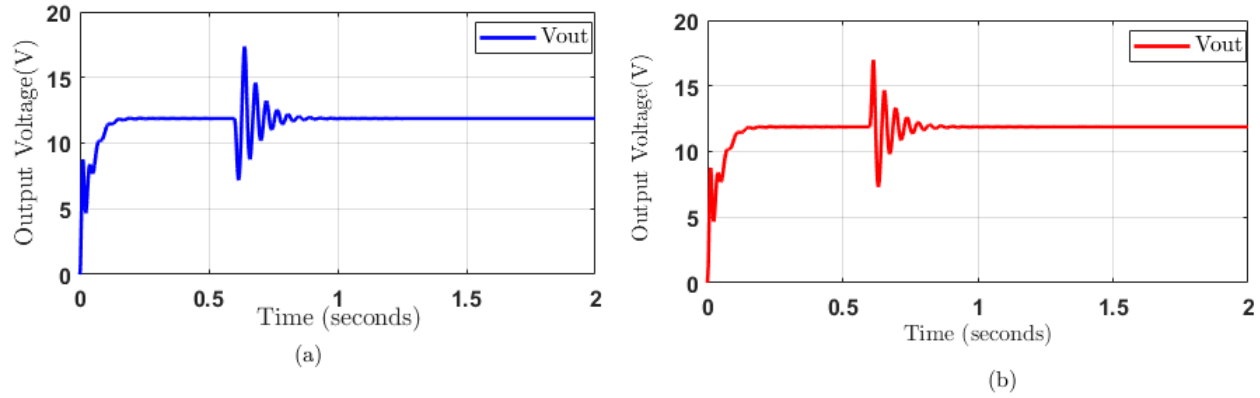

Figure 5.4: PID Controller Simulation of Output Voltage with Input voltage Variation,when (a) V_i = 2.5V and (b) V_i = 7.5V

[ht!] =0.7]PID/IL$_V$*ids.png*

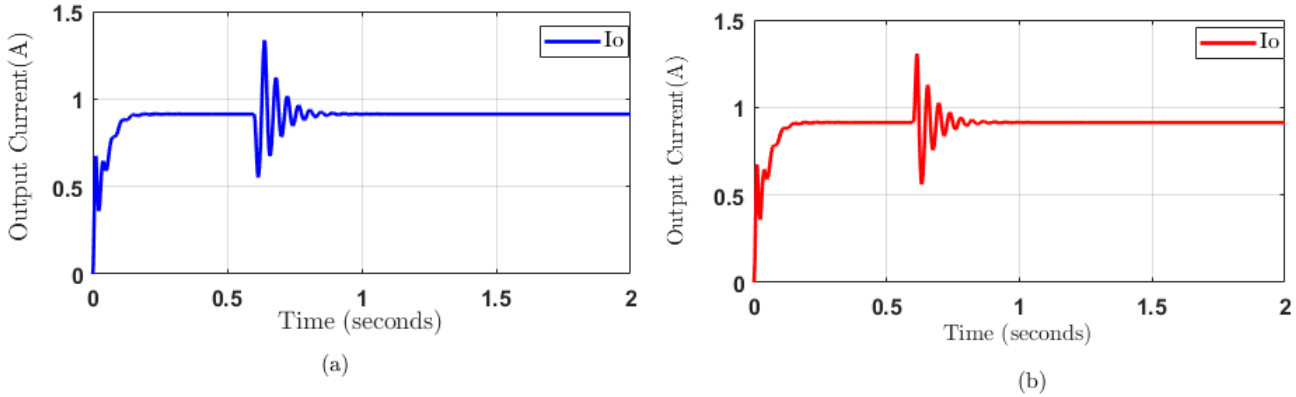

Figure 5.5: PID Controller Simulation of output Current with Input voltage Variation,when (a) V_i = 2.5V and b) V_i = 7.5V

5.1.3 PID Controller Performance due to load Variation at t=0.6sec

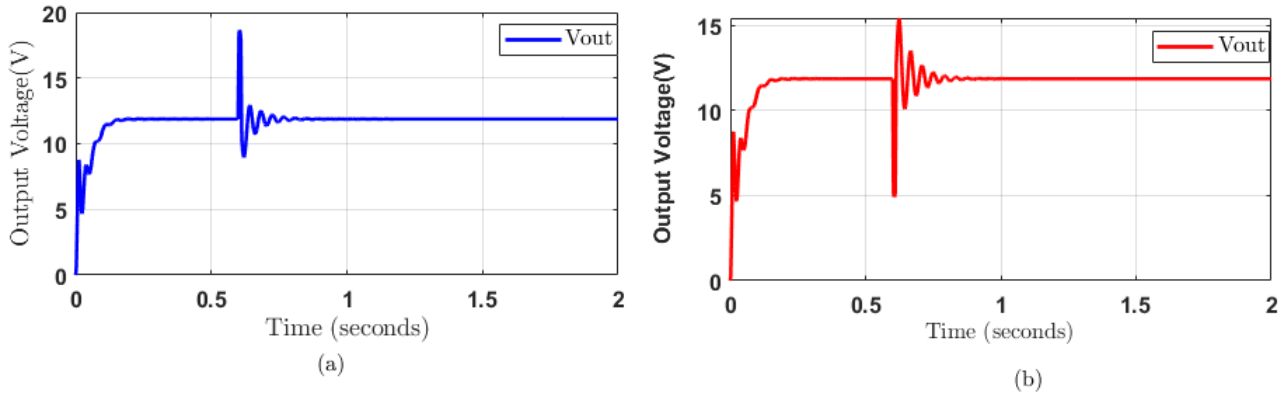

Figure 5.6: PID Controller Simulation of Output Voltage with Load Variation , when (a) $R = 6.5\,\Omega$ and (b) $R = 19.5\,\Omega$

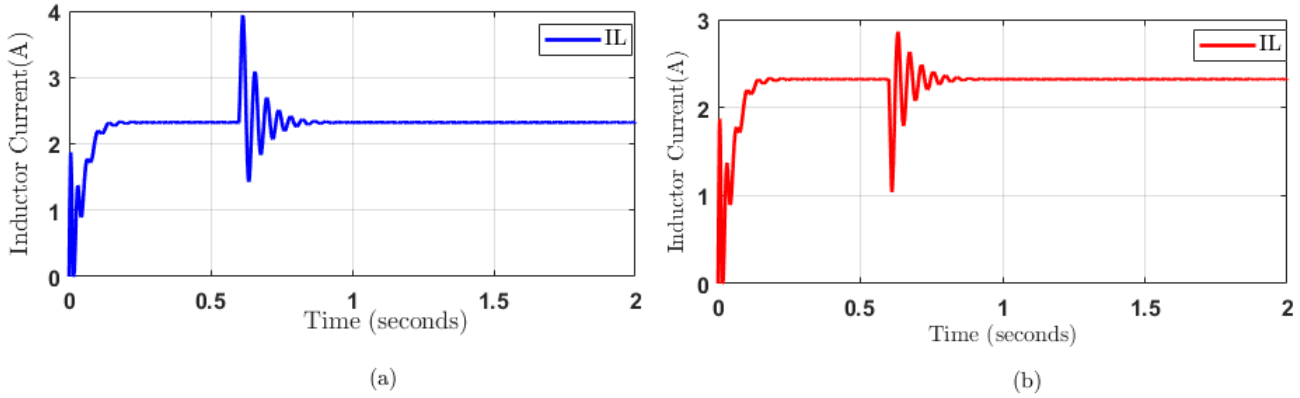

Figure 5.7: PID Controller Simulation of inductor current with Load Variation , when (a) $R = 6.5\,\Omega$ and (b) $R = 19.5\,\Omega$

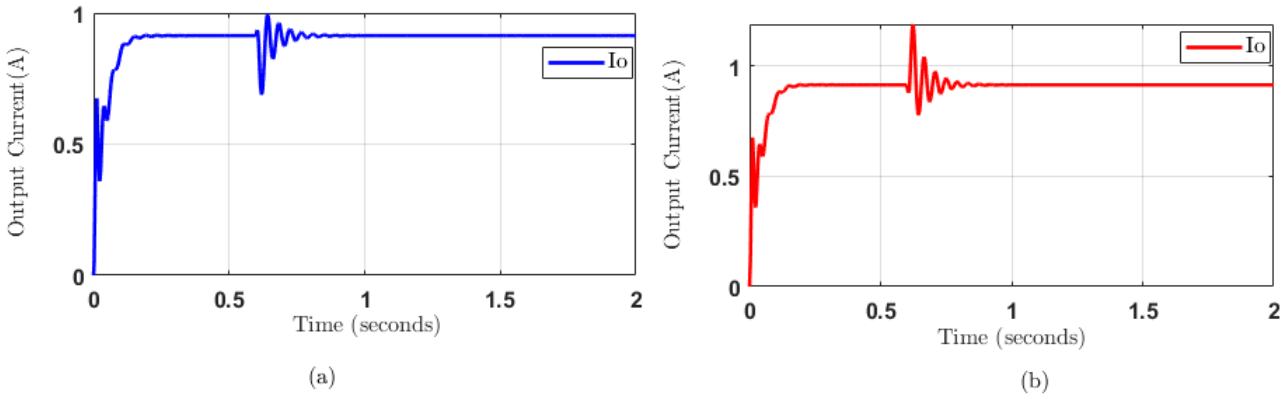

Figure 5.8: PID Controller Simulation of Output current with Load Variation , when (a) $R = 6.5\,\Omega$ and (b) $R = 19.5\,\Omega$

5.1.4 PID Controller Performance due to controller variation at t=0.6sec

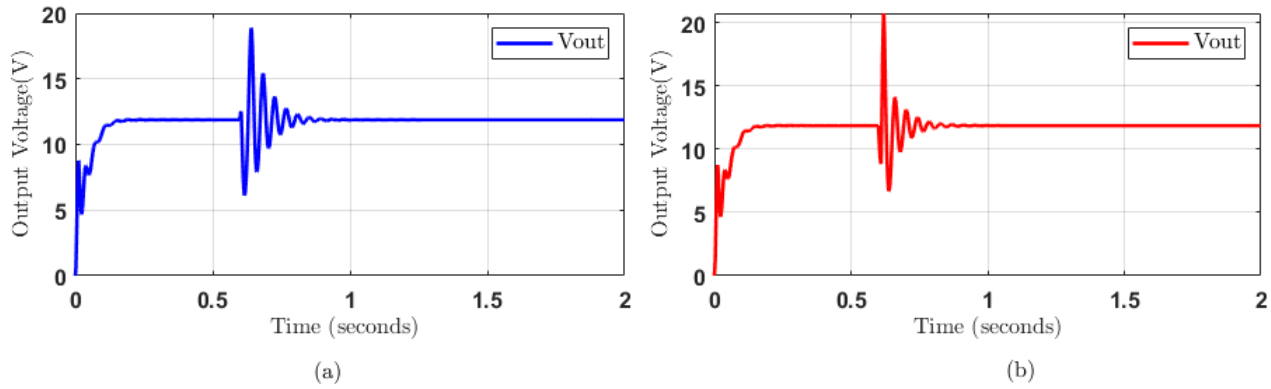

Figure 5.9: PID Controller Simulation of output voltage with controller Variation ,when (a) $u = 0.29$ and (b) $u = 0.87$

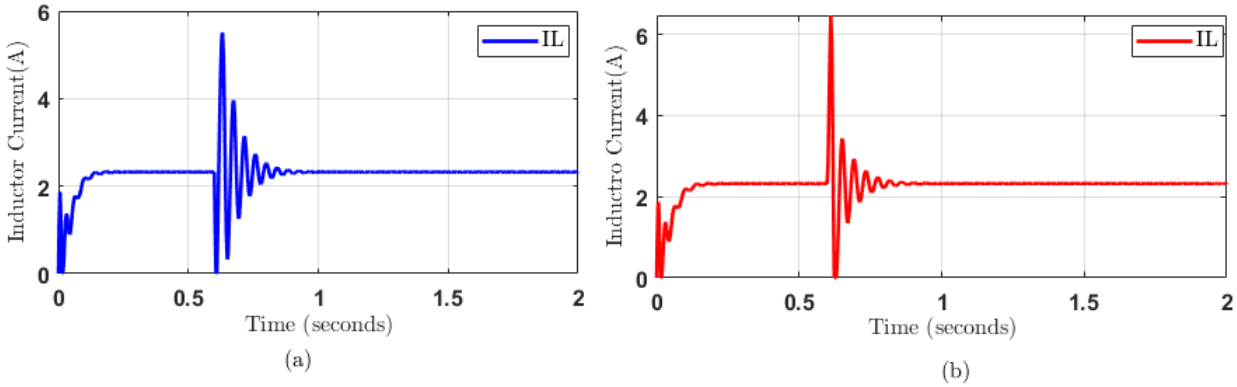

Figure 5.10: PID Controller Simulation of Inductor Current with controller Variation, when (a) $u = 0.29$) and (b) $u = 0.87$

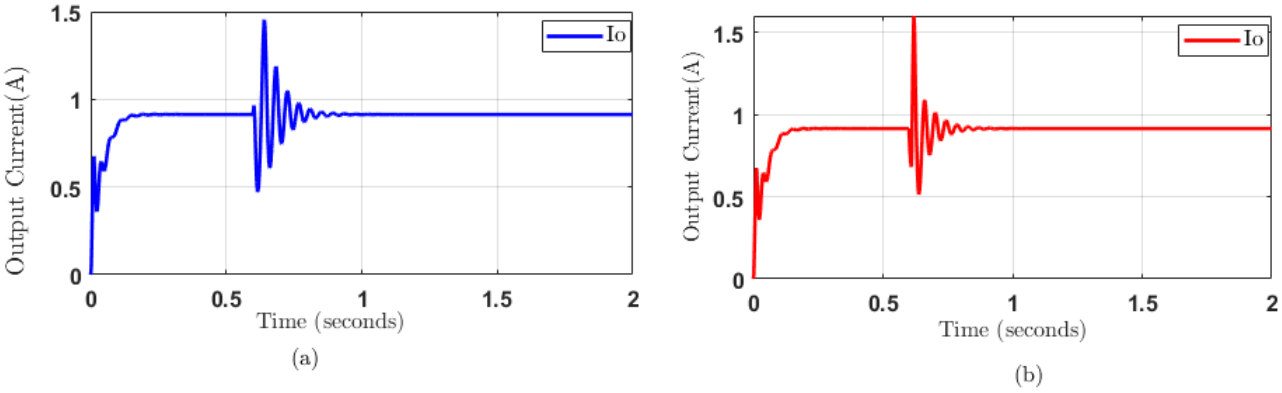

Figure 5.11: PID Controller Simulation of Output Current with controller Variation, when (a) $u = 0.29$ and (b) $u = 0.87$

5.2 SMC Simulation Results

The Sliding Mode Controller (SMC) desktop Real-time-Environment simulation results are shown in this section. The circuit parameters for the PID controller in use are the same as those previously mentioned. The boost converter with closed-loop Sliding Mode (SMC) controller action is simulated in MATLAB/Simulink. The SM controller chattering problem shown in the figure (5.12) without cascade control and LC filter using. The responses of the output voltage, the inductor current, the output current, and the controller (u_c) are respectively shown in Figures (5.13), (5.14), (5.15) and (5.16). Figure (5.13) shows that the system with the SM controller quickly settles to the desired steady-state value of 12 volts. The SM controller's effectiveness on the system is confirmed by the settling time of one second and the overshoot of 0.32 volts. Figures (5.17), (5.18) and (5.19) respectively show the closed-loop response under input voltage variation of 2.5V and 7.5v at 0.6 seconds applied for output voltage, inductor current and output current.Load variation of 6.5Ω and 19.5Ω at 0.6 seconds applied for the output voltage, the input current and the output current shown in the figures (5.20), (5.21) and (5.22).And controller variation of 0.29 and 0.87 at 0.6 seconds applied for the output voltage, the input current and the output current shown in the figures (5.23), (5.24) and (5.25).As a result, we can see that following the application of a disturbance or variation, the output voltage, inductor current, and output current rapidly return to steady state values.

Table 5.1: Optimized Gain Values of SMC Coefficients

No.	Coefficient Name	Value	Description
1	λ_{x_1}	1.5	Inductor current sliding surface coefficient
2	λ_{x_2}	1.5	Voltage sliding surface coefficient
3	K_{x_1}	0.1	Discontinuity control coefficient of inductor current
4	K_{x_2}	0.001	Discontinuity control coefficient of voltage
5	$K_{i_{x1}}$	0.4	Integral discontinuity control coefficient of inductor current
6	$K_{i_{x2}}$	0.5	Integral discontinuity control coefficient of voltage

5.2.1 SMC simulation performance at nominal condition t=0.6sec

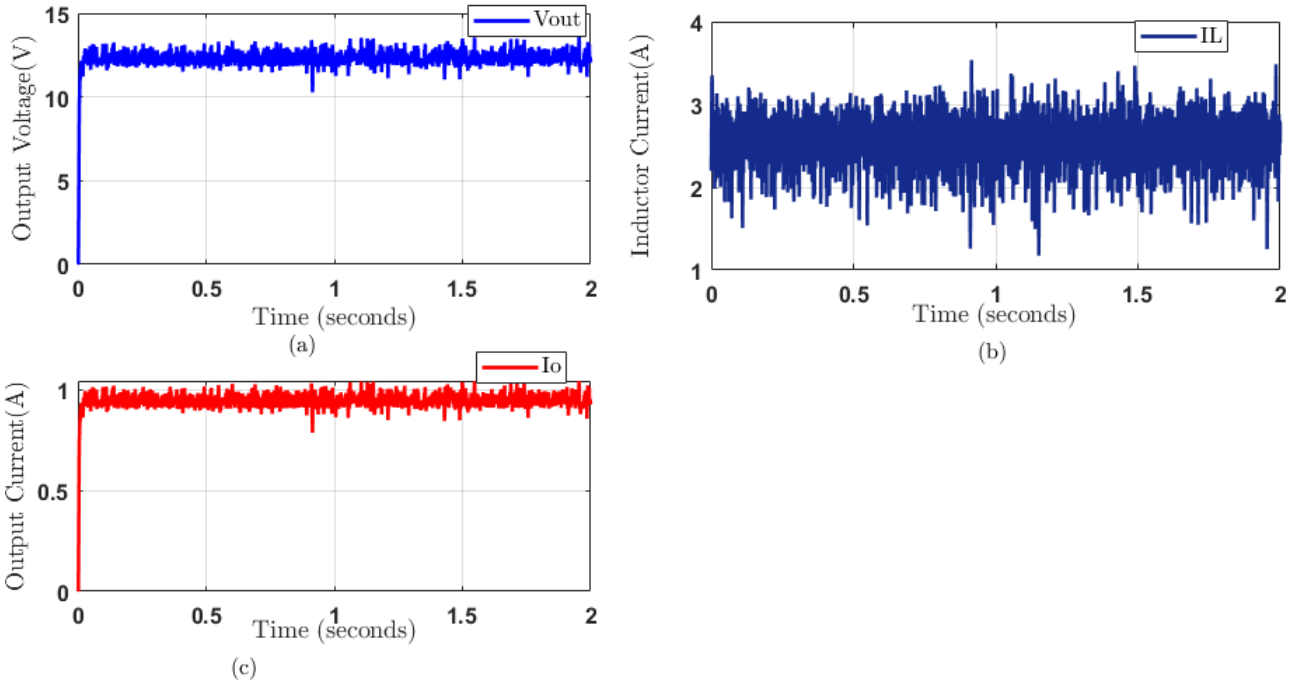

Figure 5.12: SMC Simulation of not using chattering rejection methods Outputs. (a). Output Voltage (b). Inductor Current and (c). Output Current

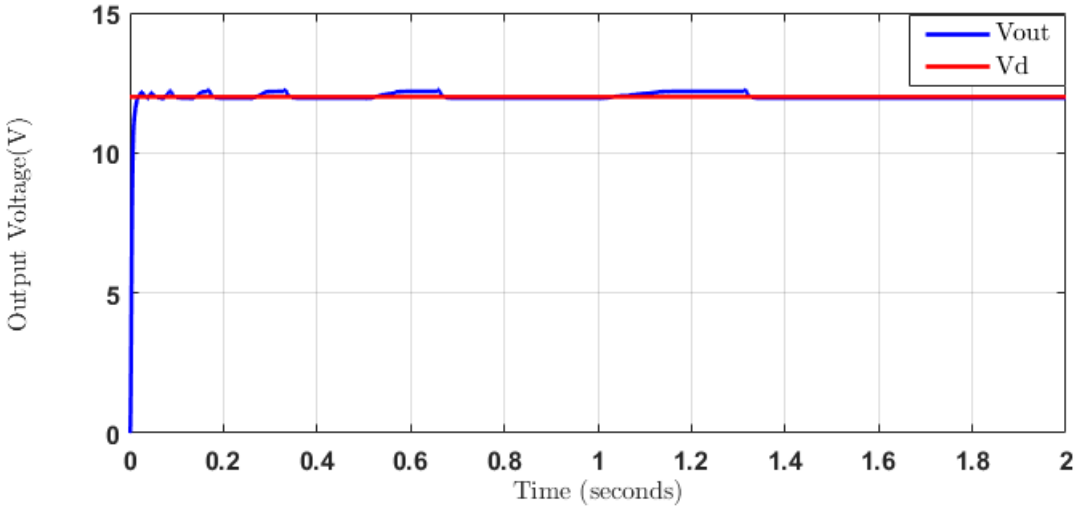

Figure 5.13: SMC Simulation of Output Voltage (V_{out})

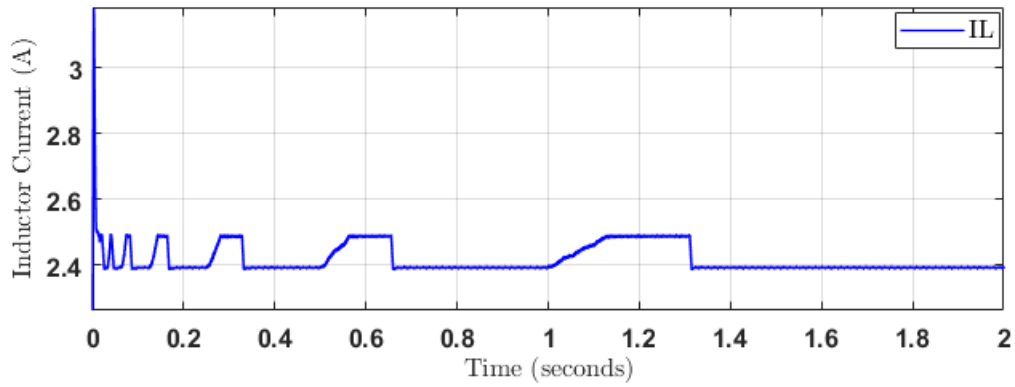

Figure 5.14: SMC Simulation of Inductor Current (I_L)

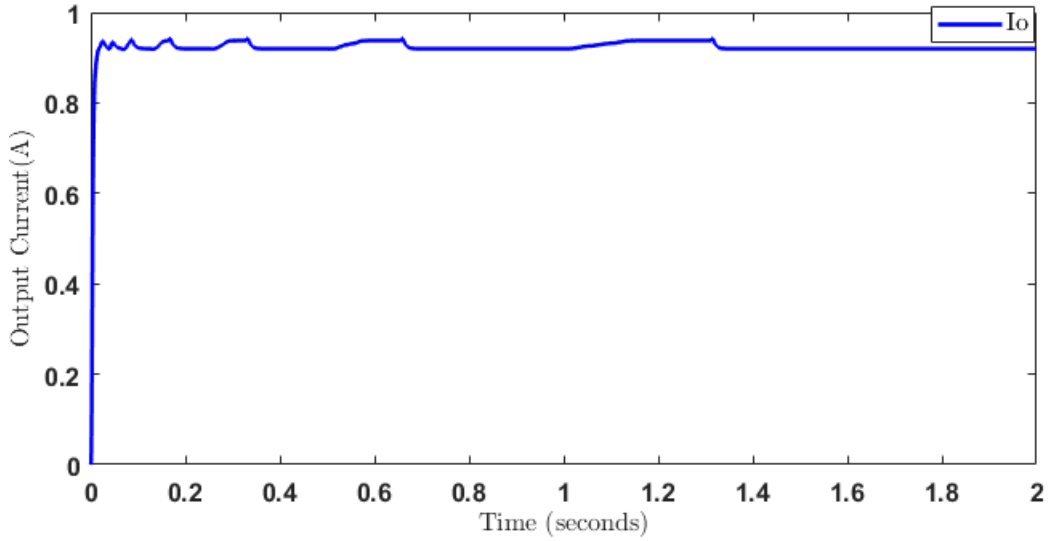

Figure 5.15: SMC Simulation of Output Current (I_{out})

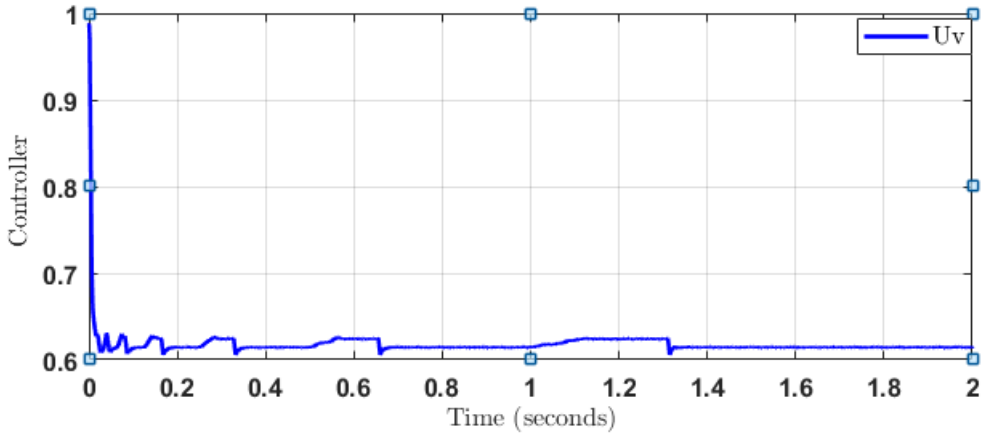

Figure 5.16: SMC Simulation of Voltage Controller (u_v)

5.2.2 SMC simulation performance at input voltage variation t=0.6 sec

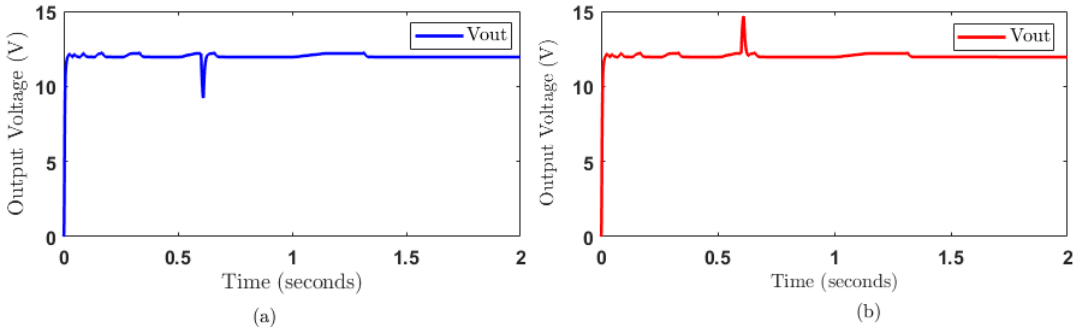

Figure 5.17: SMC Simulation of Output Voltage with input voltage variation (V_{out}),When (a) Vin=2.5V and (b) Vin=7.5V

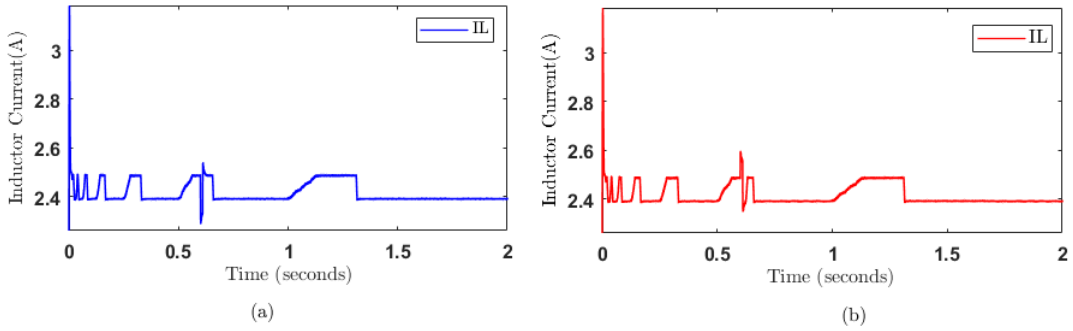

Figure 5.18: SMC Simulation of Inductor Current with input voltage variation (I_L),When (a) Vin=2.5V and (b) Vin=7.5V

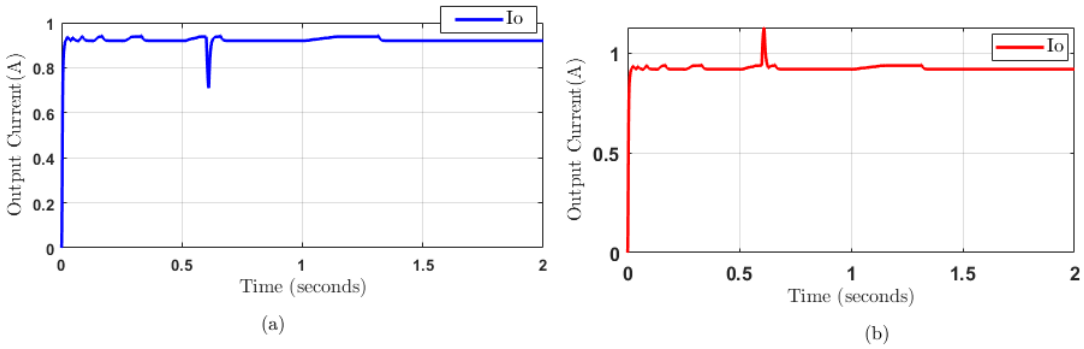

Figure 5.19: SMC Simulation of Output Current with input voltage variation (I_{out}),When (a) Vin=2.5V and (b) Vin=7.5V

5.2.3 SMC simulation performance at Load variation t=0.6 sec

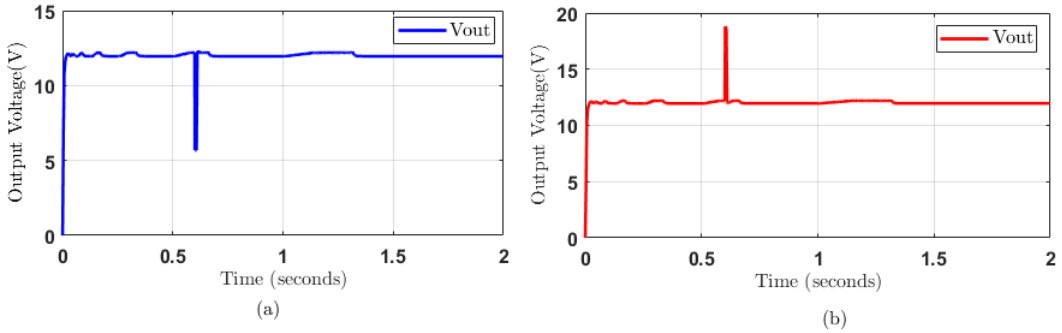

Figure 5.20: SMC Simulation of Output voltage with load variation (V_{out}),When (a) R=6.5Ω and (b) R=19.5Ω

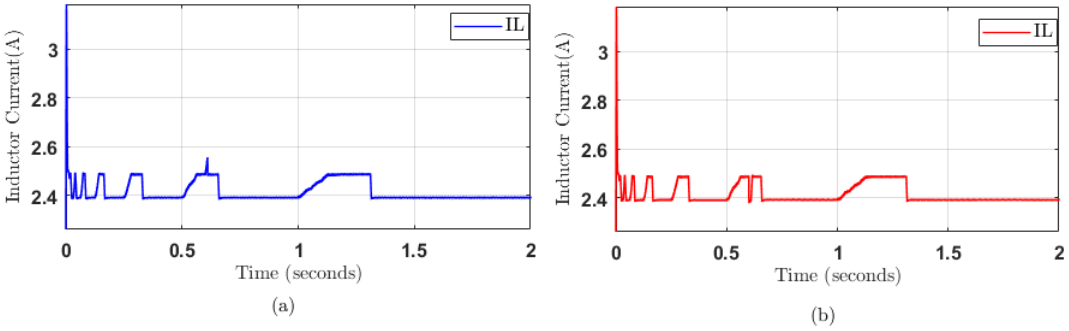

Figure 5.21: SMC Simulation of Inductor current with load variation (I_L),When (a) R=6.5Ω and (b) R=19.5Ω

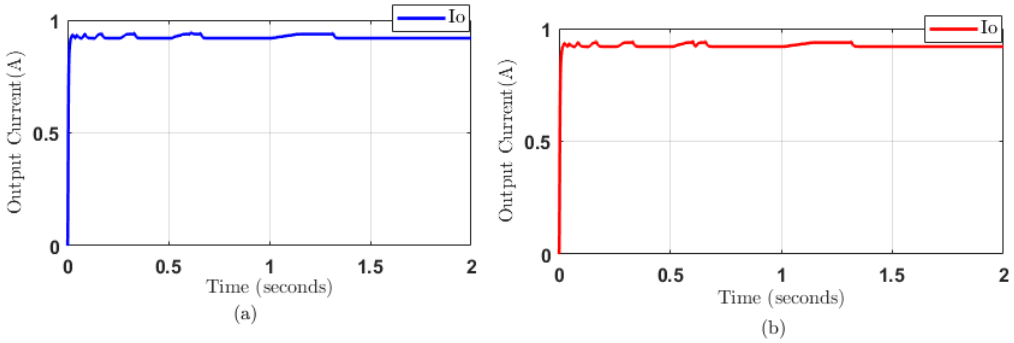

Figure 5.22: SMC Simulation of Output current with load variation (I_{out}),When (a) R=6.5Ω and (b) R=19.5Ω

5.2.4 SMC simulation performance at Controller variation at t=0.6sec

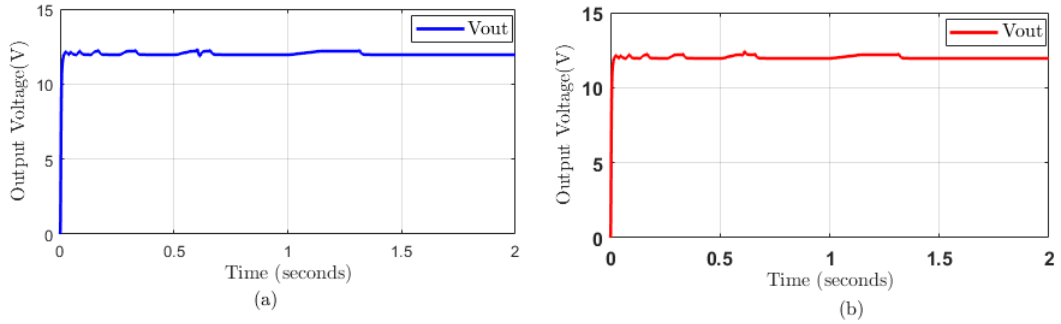

Figure 5.23: SMC Simulation of Output voltage with controller variation (V_{out}),When (a) u=0.29 and (b) u=0.87

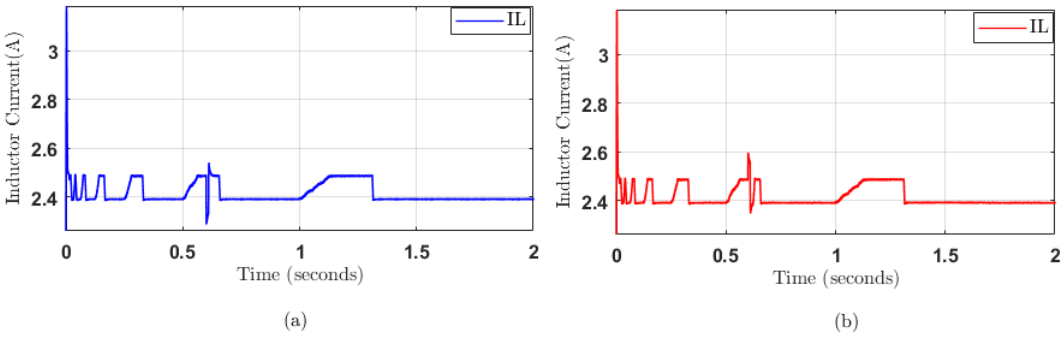

Figure 5.24: SMC Simulation of Inductor current with controller variation I_L,When (a) u=0.29 and (b) u=0.87

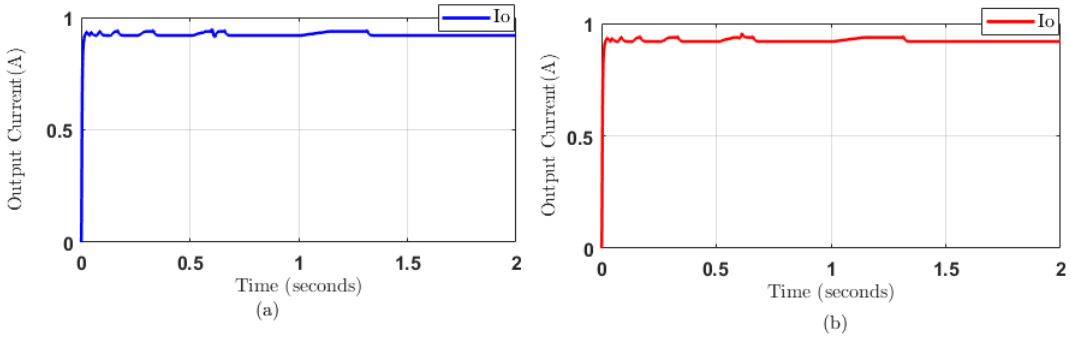

Figure 5.25: SMC Simulation of Output current with controller variation (I_{out}),When (a) u=0.29 and (b) u=0.87

5.3 Performance indices of PID and SMC simulation results

When evaluating the control performance of a boost converter, several performance index can be used, including ITAE, ISE, and IAE. Each index helps to assess different aspects of the control system's behavior, such as response time, error magnitude, and overall motion tracking accuracy.

- Integral of Time multiplied by Absolute Error (ITAE)

  The ITAE performance index places more weight on reducing the tracking error during

the early stages of the trajectory, which is especially important when the system needs to respond quickly. The ITAE is given by:

$$\text{ITAE} = \int_0^\infty t|e(t)|\,dt$$

where $e(t)$ is the error signal (the deviation from the desired output), and t is the time. In the context of the Boost Converter, this index helps assess the system's ability to minimize the error during the initial power-up phase and dynamic response. When using PID control, ITAE focuses on tuning the system to quickly converge to the desired output voltage with minimal early error, which is crucial for achieving a stable power supply. SMC may perform better in situations with more significant disturbances or uncertainties in the system, as it typically focuses on maintaining robustness and fast error correction.

- Integral of Squared Error (ISE)

  Minimizing the integral of the squared error over the whole trajectory is the main goal of the ISE index. It gives an indication of how far the system output deviates from the intended output overall. It is especially helpful when the goal is to minimize the error's overall magnitude over the course of the operation:

$$\text{ISE} = \int_0^\infty e^2(t)\,dt$$

  In the context of the Boost Converter, ISE is typically used when the objective is to minimize the cumulative squared voltage error, which is crucial for applications where voltage precision is a priority (e.g. battery charging systems). For PID control, the system may focus on reducing the error evenly over time, with a possible compromise in performance during fast transients. For SMC, the error reduction can be more aggressive, with the controller switching between different modes to minimize the squared error.

- Integral of Absolute Error (IAE)

  The IAE index is similar to the ISE, but instead of squaring the error, it uses the absolute error, which emphasizes the overall cumulative deviation from the desired output trajectory:

$$\text{IAE} = \int_0^\infty |e(t)|\,dt$$

This index is often employed when the objective is to control the magnitude of the error without overemphasizing larger deviations, as occurs in ISE with squared terms. In a Boost Converter, IAE is especially useful when trying to minimize fluctuations in output voltage while avoiding overly aggressive correction that may lead to instability. PID control often provides a smooth error reduction with a trade-off between speed and accuracy. On the other hand, SMC can offer a more robust performance by aggressively minimizing the absolute error, often at the cost of increased switching dynamics.

5.4 Voltage improvement comparison of PID and SMC

The percentage voltage improvement between PID and SMC controllers can be calculated using the following formula:

$$\text{Percentage voltage improvement} = \frac{(P_{\text{PID}} - P_{\text{SMC}})}{P_{\text{PID}}} \times 100 \tag{5.3}$$

Where,

- P_{PID} is the performance metric (ITAE, IAE, and ISE) that the PID controller produced.
- P_{SMC} is the SMC controller's performance metric (ITAE, IAE, ISE).

Table 5.2: Performance Indices at $t = 0.6$ seconds under Various Disturbances

Disturbance Type	**PID**			**SMC**			**Percentage Improvement**		
	ITAE	**IAE**	**ISE**	**ITAE**	**IAE**	**ISE**	**ITAE**	**IAE**	**ISE**
Nominal Condition	0.2594	0.6469	2.177	0.005526	0.06854	0.5263	97.87%	89.39%	75.8%
Input Voltage @ $V_{in} = 2.5$V	0.4038	0.8675	2.803	0.6352	0.5674	0.7501	-57.61%	34.64%	73.2%
Input Voltage @ $V_{in} = 7.5$V	0.4409	0.9214	3.006	0.7008	0.6204	0.8037	-58.73%	32.67%	73.2%
Load Variation @ $R = 6.5\Omega$	0.3781	0.833	2.88	0.7085	0.6403	1.13	-87.41%	23.73%	60.7%
Load Variation @ $R = 19.5\Omega$	0.3515	0.7936	2.741	0.1805	0.238	0.9545	48.56%	69.94%	65.2%
Controller @ $u = 0.29$	0.4986	1.006	0.3516	0.02557	0.08728	0.528	94.86%	91.33%	-50.6%
Controller @ $u = 0.87$	0.4363	0.9192	3.273	1.11	0.9092	1.044	-154.23%	0.42%	68.1%

5.5 Overview of PID and SMC Controller Comparison:

A comparison of the PID controller's performance improvement over SMC in different disturbance scenarios is given in the table 5.2.

5.5.1 At nominal Condition

- **PID vs. SMC:** Under nominal conditions,PID's performance is noticeably worse across all indices ITAE (0.2594), IAE (0.6469), and ISE (2.177) compared to SMC's ITAE (0.005526), IAE (0.06854), and ISE (0.5263).

- **Improvement:** SMC performs much better, with improvements of 97.87% in ITAE, 89.39% in IAE, and 75.8% in ISE, showing its superiority in steady-state conditions.

5.5.2 At input Voltage Disturbances

- **When Vin = 2.5V:** PID's performance worsens significantly (ITAE = 0.4038, IAE = 0.8675, ISE = 2.803), while SMC sees less deterioration (ITAE = 0.6352, IAE = 0.5674, ISE = 0.7501).
- **When Vin = 7.5V:** Similar trends—PID's performance drops further (ITAE = 0.4409, IAE = 0.9214, ISE = 3.006), while SMC still does better (ITAE = 0.7008, IAE = 0.6204, ISE = 0.8037).
- **Improvement:** SMC does show improvements here, but the percentages (about 30-35% in IAE) are lower compared to nominal conditions. Still, PID performs worse under voltage disturbances.

5.5.3 At load Variation Disturbances

- **When R = 6.5Ω:** PID does better in IAE and ISE, but SMC excels in ITAE (0.7085 vs. 0.3781).
- **When R = 19.5Ω:** SMC shines here, outperforming PID in all indices (ITAE = 0.1805, IAE = 0.238, ISE = 0.9545 compared to PID's ITAE = 0.3515, IAE = 0.7936, ISE = 2.741).
- **Improvement:** SMC shows major improvements, especially in IAE (69.94%) and ISE (65.2%) at R = 19.5Ω.

5.5.4 At controller gain variation:

- **When u = 0.29:** PID outperforms SMC by a large margin (ITAE = 0.4986, IAE = 1.006, ISE = 0.3516 vs. SMC's ITAE = 0.02557, IAE = 0.08728, ISE = 0.528), with improvements near 90% across all metrics.
- **When u = 0.87:** PID's performance drops significantly (ITAE = 0.4363, IAE = 0.9192, ISE = 3.273), and SMC does poorly as well (ITAE = 1.11, IAE = 0.9092, ISE = 1.044).

- **Improvement**

  PID clearly outperforms SMC at low controller gains (u = 0.29), but when the gain is higher (u = 0.87), both controllers struggle.

5.6 General Observations

- **PID Performance**

  PID tends to perform poorly when there are disturbances, particularly in terms of IAE and ISE. It struggles more with voltage and load variations compared to SMC.

- **SMC Performance**

  SMC generally shows a significant improvement over PID, especially under nominal conditions and during load variations. However, it can struggle when subjected to high controller gains (u = 0.87).

- **Improvements**

  SMC's improvements over PID are notably high in the nominal condition, with performance gains of nearly 98%. However, the improvements tend to decrease under disturbances, especially with higher voltages or controller gains.

Generally, SMC performs better than PID in standard and most disturbance conditions, especially with load variations. However, PID is more effective in certain situations, like when using low controller gains.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

This thesis involved the design and simulation of control strategies for a DC, DC boost converter. The average model was created from the switch, level circuit to represent the key dynamics of the converter. Since the system is described by a bilinear differential equation, it was linearized around a certain operating point to allow the application of linear control methods. Real-time simulations were performed in MATLAB/Simulink with the Simulink Desktop Real Time environment. Sliding Mode Control was executed in the connected I/O mode, and the model was running smoothly without any timing overruns. The real-time outcomes were in good agreement with the theoretical ones; thus, they not only validate the control approach's effectiveness but also confirm the robustness of the simulation setup. Some of the performance problems came to the fore during the simulations. The CPU was paced at times to keep the 20 μs real-time step, especially when more complex models were run, which resulted in small lags and some jitter in the output. The Simulink Desktop Real, Time monitoring instruments showed that the processor was working near its maximum during the heavy computational periods. However, the 2.42 GHz CPU was still noticeably better than lower, specification hardware in terms of performance and thus, it was able to provide smoother operations during normal conditions. Although more powerful processing would accelerate the speed of the highly detailed models, the system as a whole was a stable platform for real, time simulation. The research also included a comparison of PID control and Sliding Mode Control (SMC). The PID controller was easy to implement and managed small disturbances well, but it lost stability when sudden changes such as abrupt load variations happened unexpectedly. To overcome these limitations, SMC was brought

in and exhibited strong robustness characteristics, thus performance being quick and stable even under input voltage dips or varying load resistance conditions. The tuning of the SMC gains was done through Particle Swarm Optimization while PID tuning was done by trial and error. The standard performance indices ITAE, IAE, and ISE were used for the evaluation purpose. Overall, the comparison showed that while PID is suitable for mild disturbances, SMC offers superior resilience and responsiveness in more demanding and unpredictable operating scenarios.

6.2 Future Works

Even though this thesis has delved into an in-depth study of the boost converter with Sliding Mode Control (SMC) using MATLAB/Simulink and RTE in a desktop environment, it still leaves room for numerous intriguing ideas to be researched in the future. The following list presents some of the possible next steps in research:

Integration with MATLAB Real Time Environment (RTE) While this study was primarily centered on MATLAB/Simulink simulation, the following phase ought to be the integration of control algorithms in a real-time environment. With MATLAB RTE, the SMC can be directly tested on hardware platforms. This would allow actual world testing through hardware-in-the-loop (HIL) simulations. In this way, not only would it be possible to confirm the control strategies, but also to measure their effectiveness under conditions that are closer to reality.

Hardware Implementation and Testing

It is necessary to verify the SMC strategy beyond the simulated environment by implementing it in an actual boost converter circuit to check its performance under real conditions. Operating the system on real hardware such as microcontrollers would give the essential information about how the controller is functioning in reality. It may reveal issues like latency and performance limits that, as a result, can be resolved to make the device suitable for the practical application.

Multi-Phase or Multi-Output Boost Converters

Another potential research area is the use of the SMC scheme for multi-phase or multi-output boost converters. They are typically used in high power applications and pose certain challenges. By figuring out how SMC can be implemented in these complex systems, we might find ways to enhance stability and performance under a wider range of conditions.

Major improvements in the design and implementation of Boost Converter SMC using RTE can be achieved by diving deeper into these aspects. Such a move will not only advance the research to a level that is closer to real time but also pave the way for a wide range of practical applications in the field of power electronics. Consequently, it will become more and more relevant for the future development of the energy sector.

Bibliography

[1] B. Johansson, *DC-DC Converters-Dynamic Model Design and Experimental Verification.* 2005.

[2] C. Albea, *Control Design for Electronic Power Converters*. PhD thesis, Institut National Polytechnique de Grenoble-INPG; Universidad de Sevilla, 2010.

[3] Q. Zhang, R. Min, Q. Tong, X. Zou, Z. Liu, and A. Shen, "Sensorless predictive current controlled dc–dc converter with a self-correction differential current observer," *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 6747–6757, Dec. 2014.

[4] B. K. Bose, "Energy, environment, and advances in power electronics," *IEEE Transactions on Power Electronics*, vol. 15, pp. 688–701, Jul. 2000.

[5] B. Kurucs, A. Peschka, P. Stumpf, I. Nagy, and I. Vajk, "State space control of quadratic boost converter using lqr and lqg approaches," in *Proc. Intl Conf 2015 Intl Aegean Conf. Electrical Machines Power Electronics (ACEMP) Optimization of Electrical Electronic Equipment (OPTIM) 2015 Intl Symp. Advanced Electromechanical Motion Systems (ELECTROMOTION)*, pp. 642–648, Sep. 2015.

[6] Y. Jiao, F. L. Luo, and M. Zhu, "Voltage-lift-type switched-inductor cells for enhancing dc-dc boost ability: Principles and integrations in luo converter," *IET Proceedings on Power Electronics*, vol. 4, pp. 131–142, Jan. 2011.

[7] Y. Jiao and F. L. Luo, "N-switched-capacitor buck converter: Topologies and analysis," *IET Proceedings in Power Electronics*, vol. 4, pp. 332–341, Mar. 2011.

[8] Y. Jiao, F. L. Luo, and B. K. Bose, "Voltage-lift split-inductor-type boost converter," *IET Proceedings in Power Electronics*, vol. 4, pp. 353–362, Apr. 2011.

[9] Y. Jiao, F. L. Luo, and M. Zhu, "Generalized modeling and sliding mode control for n-cell cascade super-lift dc-dc converters," *IET Proceedings in Power Electronics*, vol. 4, pp. 532–540, May. 2011.

[10] S. Arulselvi, G. Uma, and M. Chidambaram, "Design of pid controller for boost converter with rhs zero," in *4th International Power Electronics and Motion Control Conference (IPEMC)*, (China), Aug. 2004.

[11] B. K. Bose, *Power Electronics and Motor Drives: Advances and Trends*. Burlington, MA: Academic Press, 2010.

[12] H. Mingzhi and X. Jianping, "Nonlinear pid in digital controlled buck converters," in *Applied Power Electronics Conference (APEC)*, (Anaheim, CA, USA), pp. 1461–1465, Feb./Mar. 2007.

[13] A. Foreyth and S. Mollov, "Modeling and control of dc-dc converters," *IEEE Power Engineering Journal*, vol. 12, no. 5, pp. 229–236, 1998.

[14] M. Rashid, *Power Electronics Handbook*. Academic Press, 2001.

[15] V. Racirah and P. Sen, "Comparative study of proportional- integral sliding mode, and fuzzy logic controllers for power converters," *IEEE Transactions On Industry Applications*, vol. 33, no. 2, pp. 518–524, 1997.

[16] W. Dai, "Output voltage ripple measurement and reduction for dc/dc voltage regulators," *Monolithic Power Systems*, 2023.

[17] J. Miller, *Modeling and Simulation of Dynamic Systems with MATLAB and Simulink*. Wiley, 2020.

[18] MathWorks, "Simulink real-time," 2023. `https://www.mathworks.com/products/simulink-real-time.html`.

[19] L. Mayer, *Real-Time Simulation and Control Systems*. Springer, 2018.

[20] The MathWorks, Inc., *Simulink® Desktop Real-Time™ User's Guide*. The MathWorks, Inc., 1 Apple Hill Drive, Natick, MA 01760-2098, 2024. `https://www.mathworks.com`.

[21] V. Utkin, *Sliding Modes and Their Applications on Variable Structure Systems*. Moscow: MIR Publisher, 1978.

[22] V. Utkin, *Sliding Modes in Control Optimization*. Springer-Verlag, 1981.

[23] H. Gueldemir, "Sliding mode control of boost converter," *Journal of Applied Science*, vol. 5, no. 3, pp. 588–592, 2005.

[24] Q. Zhang, R. Min, Q. Tong, X. Zou, Z. Liu, and A. Shen, "Sensorless predictive current controlled dc-dc converter with a self-correction differential current observer," *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 6747–6757, Dec. 2014.

[25] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942–1948, IEEE, 1995.

[26] Y. Cheng and C. Liu, "Boost converter design for high-efficiency power conversion," *IEEE Transactions on Power Electronics*, vol. 28, no. 7, pp. 3775–3783, 2015.

[27] Y. Li and Q. Wang, "Optimal design of boost converters for energy harvesting," *Journal of Power Electronics*, vol. 14, no. 3, pp. 605–611, 2014.

[28] Y. Li and Q. Wang, "Optimal design of boost converters for energy harvesting," *Journal of Power Electronics*, vol. 14, no. 3, pp. 605–611, 2014.

[29] P. K. Tewari and S. A. Soliman, "Analysis and design of high-efficiency boost converters," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3245–3253, 2008.

[30] M. H. Rashid, *Power Electronics: Circuits, Devices & Applications*. Pearson, 4th ed., 2013.

[31] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[32] Texas Instruments, "Understanding and designing mosfet gate driver circuits," Tech. Rep. SLUA618A, Texas Instruments, March 2013.

[33] S.-C. Tan, Y. Lai, C. K. Tse, L. M. Salmero, and C.-K. Wu, "A fast response sliding mode controller for boost type converter with a wide range of operating conditions," *IEEE Transactions On Industrial Electronics*, vol. 54, no. 6, 2007.

[34] M. Ahmed, M. Kuisma, and P. Silventoinen, "Comparison between pid control and sliding mode control for buck converter," in *Proceedings of the Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, (Capri, Italy), June 2004.

[35] G. Spiazzi, P. Mattavelli, and L. Rossetto, "Sliding control of dc-dc converters," in *4th Congresso Brasileiro de Eletrônica de Potência (COBEP)*, (Belo Horizonte, Brazil), pp. 59–68, December 1997.

[36] D. Cortes and J. Alvarez, "Robust sliding mode control for the boost converter," in *Proceedings of VIII IEEE International Power Electronics Congress (CIEP 2002)*, pp. 208–212, October 2002.

[37] L. Guo, J. Hung, and R. M. Nelms, "Design and implementation of a digital pid controller for a buck converter," in *Proceedings of the 36th Intersociety Energy Conversion Engineering Conference*, vol. 1, pp. 187–192, Jul./Aug. 2001.

[38] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*. Boston, MA: Springer, 2nd ed., 2001.

[39] W. Jansen, E. Verreycken, A. Schenck, J.-E. Blanquart, C. Verhulst, N. Huebel, and J. Steckel, "Cosys-airsim: A real-time simulation framework expanded for complex industrial applications," *arXiv preprint arXiv:2303.13381*, 2023.

Appendix A

MATLAB Code

A.1 PSO Optimization Code

```
% Particle Swarm Optimization (APSO) for SMC of boost converter system
% model "SMC_of_Boost_Convertor_Model"
nVar = 6; % Number of Decision Variables
% Search space
ub = [100,70,70,100, 100, 100];% Upper Bound
lb = [0,0,0,0,0,0];% [kv_1,kc_1,ki_v,ki_c,lemda_v and lemda_c]Lower Bound
fobj = @PSO2; % Cost Function
% Define PSO's parameters
noP = 15;         % Number of particles for initialization
maxIter = 20;    % Maximum iterations
wMax = 0.9;       % Maximum value of the Inertia Weight
wMin = 0.4;       % Minimum value of the Inertia Weight
c1 = 2;           % Personal acceleration Coefficient (Cognitive constant)
c2 = 2;           % Global/Social acceleration Coefficient (Social constant)
% Random Velocity initialization
vMax = (ub - lb) .* 0.2;
vMin = -vMax;
% Initialize the Swarm structure
Swarm.Particles = struct('X', [], 'V', [], 'PBEST', struct('X', [], 'O', []));
Swarm.GBEST = struct('X', [], 'O', inf);

% Initialize particles
for k = 1:noP
Swarm.Particles(k).X = (ub - lb) .* rand(1, nVar) + lb; % Position
    initialization
Swarm.Particles(k).V = zeros(1, nVar);  % Initial velocity
```

```
    Swarm.Particles(k).PBEST.X = zeros(1, nVar); % Personal best position
    Swarm.Particles(k).PBEST.O = [inf, inf];  % Personal best objective (high
        initially)
end
% Main PSO loop
objectiveHistory = zeros(maxIter, 2); % Store both objective values (ITAE1 and
    ITAE2) for each iteration
for it = 1:maxIter
 % Calculate the objective value for each particle
 for k = 1:noP
currentX = Swarm.Particles(k).X;
Swarm.Particles(k).O = fobj(currentX); % Evaluate the objective function
% Update PBEST (personal best) for each particle
if Swarm.Particles(k).O(1) < Swarm.Particles(k).PBEST.O(1)
Swarm.Particles(k).PBEST.X = currentX;
Swarm.Particles(k).PBEST.O = Swarm.Particles(k).O;
        end
% Update GBEST (global best) for the swarm
if Swarm.Particles(k).O(1) < Swarm.GBEST.O(1)
Swarm.GBEST.X = currentX;
Swarm.GBEST.O = Swarm.Particles(k).O;
    end
    end
% Store the objective values for plotting
objectiveHistory(it, :) = Swarm.GBEST.O;
% Display iteration info
outmsg = ['Iteration#', num2str(it), ' Objective/Fitness Function (ITAE1,
   ITAE2) = ', num2str(Swarm.GBEST.O)];
disp(outmsg);
% Decrease inertia weight smoothly
w = wMax - (wMax - wMin) * (it / maxIter); % Smooth linear decay
for k = 1:noP
% Update velocity
Swarm.Particles(k).V = w .* Swarm.Particles(k).V + ...
c1 .* rand(1, nVar) .* (Swarm.Particles(k).PBEST.X - Swarm.Particles(k).X) +
   ...
c2 .* rand(1, nVar) .* (Swarm.GBEST.X - Swarm.Particles(k).X);
% Check and clamp velocity within bounds
Swarm.Particles(k).V = min(max(Swarm.Particles(k).V, vMin), vMax);
 % Update position
```

```
 Swarm.Particles(k).X = Swarm.Particles(k).X + Swarm.Particles(k).V;
 % Check and clamp position within bounds
Swarm.Particles(k).X = min(max(Swarm.Particles(k).X, lb), ub);
end
end
% Display the optimized gain values
disp('Optimized Gain Values:');
disp('--------------------------------');
disp(['ki_c: ', num2str(Swarm.GBEST.X(1))]);
disp(['ki_v: ', num2str(Swarm.GBEST.X(2))]);
disp(['lemda_c: ', num2str(Swarm.GBEST.X(3))]);
disp(['lemda_v: ', num2str(Swarm.GBEST.X(4))]);
disp(['kc_1: ', num2str(Swarm.GBEST.X(5))]);
%disp(['kc_2: ', num2str(Swarm.GBEST.X(6))]);
disp(['kv_1: ', num2str(Swarm.GBEST.X(7))]);
%disp(['kv_2: ', num2str(Swarm.GBEST.X(8))]);
disp('--------------------------------');
% Plotting the evolution of ITAE1 (the first cost)
figure;
plot(1:maxIter, objectiveHistory(:, 1), 'k', 'LineWidth', 1);
xlabel('Iteration#');
ylabel('Objective/Fitness Function ITAE1');
legend('Evolution of ITAE1 vs. iterations');
grid on;
% Plotting both ITAE1 and ITAE2
figure;
plot(1:maxIter, objectiveHistory(:, 1), 'k', 'LineWidth', 1, 'DisplayName', '
    ITAE1');
hold on;
plot(1:maxIter, objectiveHistory(:, 2), 'r', 'LineWidth', 1, 'DisplayName', '
    ITAE2');
xlabel('Iteration#');
ylabel('Objective/Fitness Function');
legend('ITAE1', 'ITAE2');
grid on;
function [cost] = PSO2(x)
 % Assign the variables to the base workspace
assignin('base', 'kv_1', x(1));
assignin('base', 'kc_1', x(3));
assignin('base', 'ki_v', x(5));
```

```
assignin('base', 'ki_c', x(6));
assignin('base', 'lemda_v', x(7));
assignin('base', 'lemda_c', x(8));
% Run the simulation
simOut = sim('SMC_of_Boost_Convertor_Model', 'ReturnWorkspaceOutputs', 'on');
    % Ensure output is returned
 % Fetch the ITAE values from the simulation workspace
ITAE1 = evalin('base', 'ITAE1');  % Get ITAE1 from the workspace
ITAE2 = evalin('base', 'ITAE2');  % Get ITAE2 from the workspace
% Check if ITAE1 and ITAE2 are valid
if isempty(ITAE1) || isempty(ITAE2)
error('ITAE1 or ITAE2 is empty! Ensure they are correctly output by the
    Simulink model.');
     end
% Return the final values of ITAE1 and ITAE2 as the cost
cost(1) = ITAE1(end);  % Last value of ITAE1
cost(2) = ITAE2(end);  % Last value of ITAE2
% Display cost for debugging
disp(['Current ITAE1: ', num2str(cost(1)), ', ITAE2: ', num2str(cost(2))]);
end
```

A.2 SMC Code

```
% Voltage Error code
function e_v = fcn(x_2,x2_r)
    e_v = x2_r - x_2;
end
% Inductor current Error code
function e_c = fcn(x_1,x1_r)
    e_c = x1_r - x_1;
end
% Sliding surface of Voltage controller
function s_v = fcn(e_v, int_e_v)
    ki_v = 0.57;
    lemda_v = 1.6; % lemada_v is the sliding surface slope of inductor current
        (Vc)
    s_v = e_v + lemda_v * int_e_v + ki_v * int_e_v;
end
% Sliding surface of Current controller
function s_c = fcn(e_c, int_e_c)
```

```
    ki_c = 0.5;
    lemda_c = 1.7; % lemada_c is the sliding surface slope of inductor current
        (IL)
    s_c = e_c + lemda_c * int_e_c + ki_c * int_e_c;
end
% Voltage Controller
function u_v = fcn(x_1, x_2, s_v, x2_r)
    Rc = 0.01;
    R = 13;
    C = 0.001;
    lemda_v = 1.6;
    kv_1 = 0.001;
    % kv_2 = 0.001;
    ki_v = 0.57;
    u_v = 1 - (C * (R + Rc)) / (R * x_1) * (x_2 / (C * (R + Rc)) + lemda_v * (
        x2_r - x_2) + ki_v * (x2_r - x_2)) - (kv_1 * s_v); % u_v is the
        reference of inductor current i.e x1_r
end
% Current Controller
function u_c = fcn(x_2, x_1, x1_r, s_c)
    Rc = 0.01;
    Rd = 0.001;
    R = 13;
    VD = 0.5;
    L = 0.009;
    Vin = 5;
    RL = 0.05;
    Ron = 0.001;
    lemda_c = 1.7;
    kc_1 = 0.1;
    % kc_2 = 0.01;
    ki_c = 0.5;
    % Define numerator parameters for the Equivalent control of inner current
        loop
    v1 = (Ron * x_2) / (L * (R + Rc));
    v2 = -(Rd + RL) / L * x_1;
    v3 = -lemda_c * (x1_r - x_1) - ki_c * (x1_r - x_1) + Vin / L;
    % Define denominator parameters for the Equivalent control of inner
        current loop
    v4 = (R * Rc * (x_1)) / (L * (R + Rc));
```

```
    v5 = (Ron * x_1) / (L * (R + Rc));
    v6 = (R * x_1) / (L * (R + Rc));
    v7 = VD / L;
    u_c = 1 - (v1 + v2 + v3) / (v4 + v5 + v6 + v7) + ki_c * (x_1 - x1_r) - (
        kc_1 * s_c); % MOSFET switch controller
end
% PWM generator
function pwm_output = pwm_generator(u, t)
    % u: Duty cycle (value between 0 and 1)
    % t: Current simulation time (in seconds)
    % PWM Frequency: 50 kHz
    freq = 50000;
    % Calculate the period of the PWM signal in seconds
    period = 1 / freq;
    % Calculate how long the signal should stay high (in seconds)
    high_time = u * period;
    % Generate PWM signal (square wave)
    if mod(t, period) >= high_time
        pwm_output = 1;  % High signal (on)
    else
        pwm_output = 0;  % Low signal (off)
    end
end
```

Appendix B

MATLAB Script

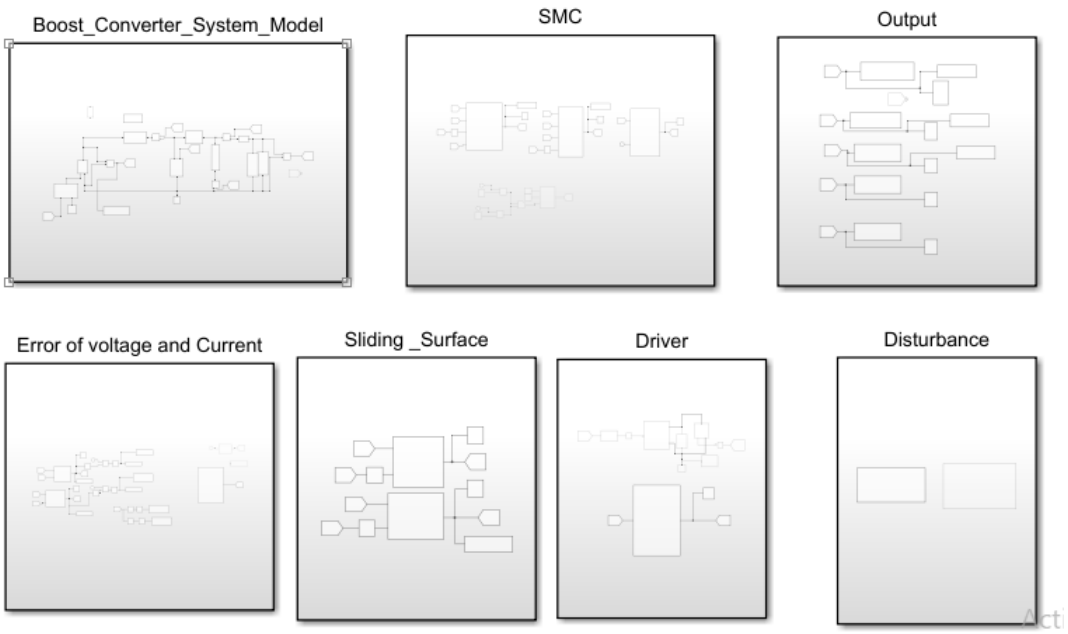

Figure B.1: General System Design in MATLAB/Simulink Software

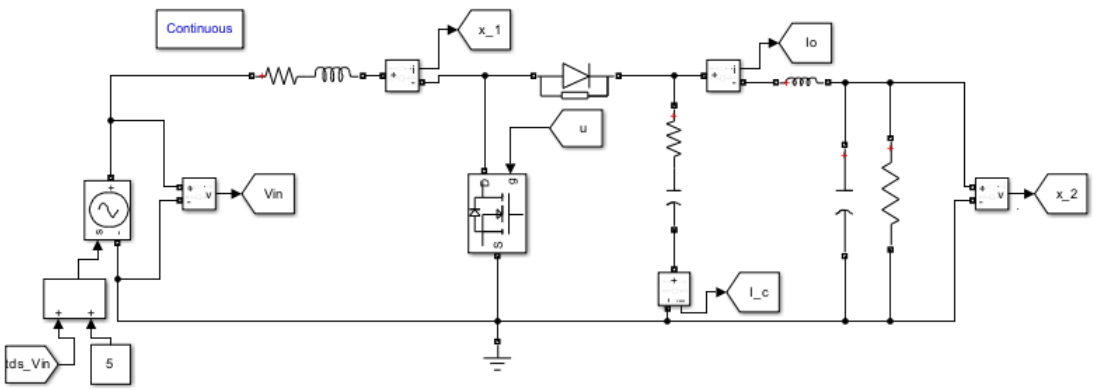

Figure B.2: System Design in Matlab/Simulink Software

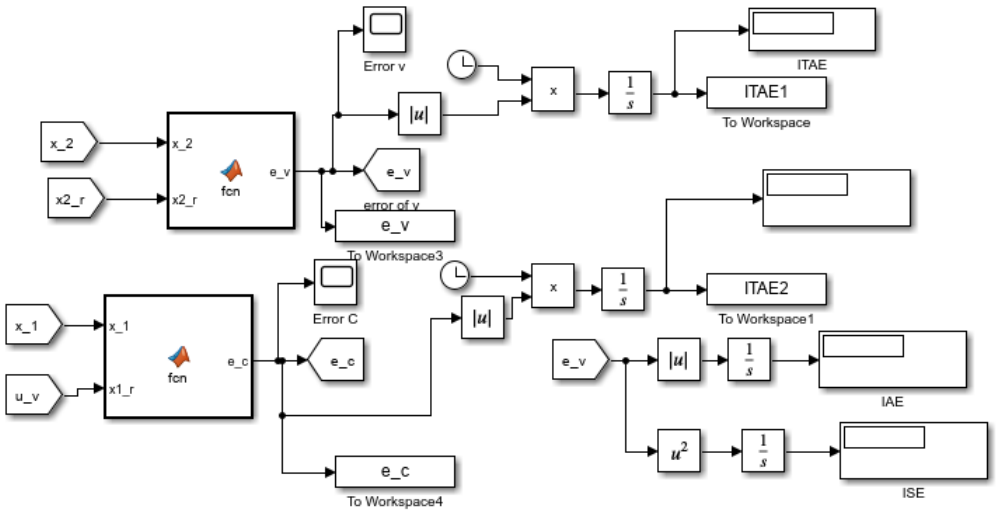

Figure B.3: Error Design in Matlab/Simulink Software

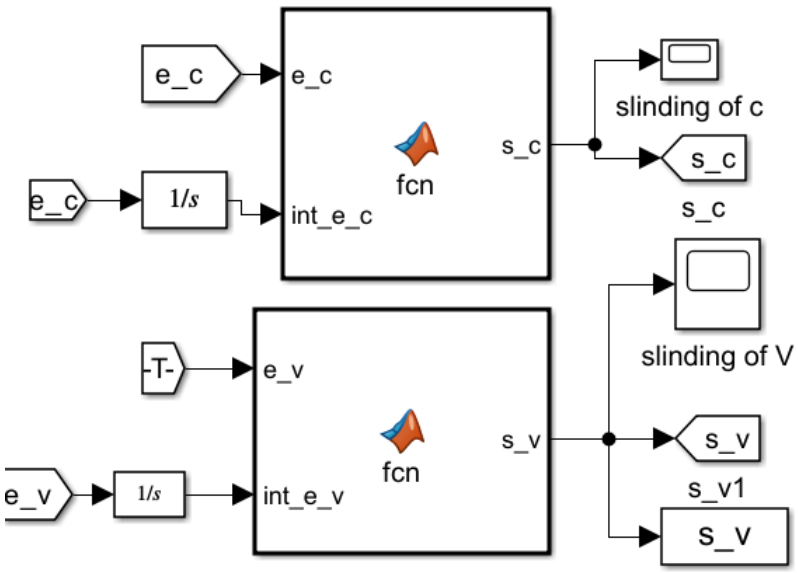

Figure B.4: Sliding Surface Design in Matlab/Simulink Software

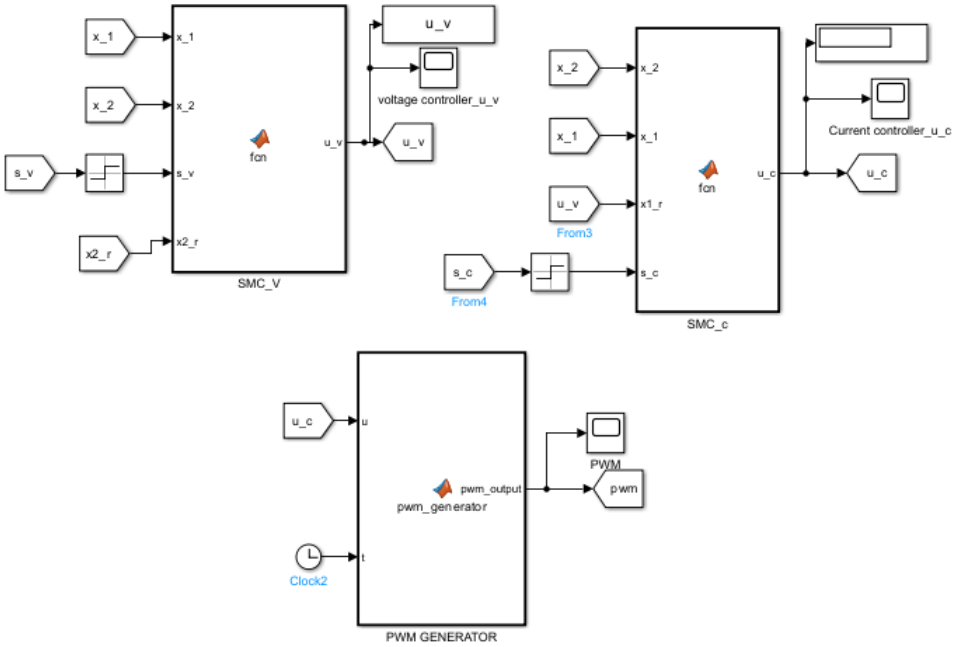

Figure B.5: SM Controller Design in Matlab/Simulink Software

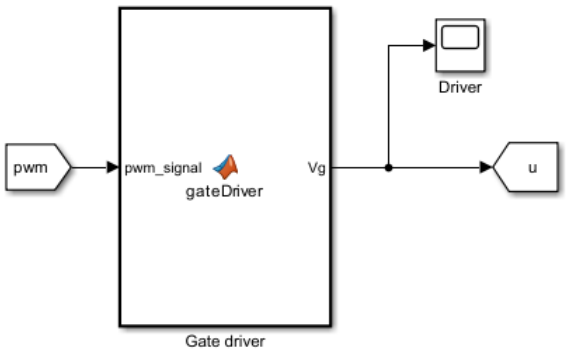

Figure B.6: MOSFET Circuit Gate Driver in Matlab/Simulink Software

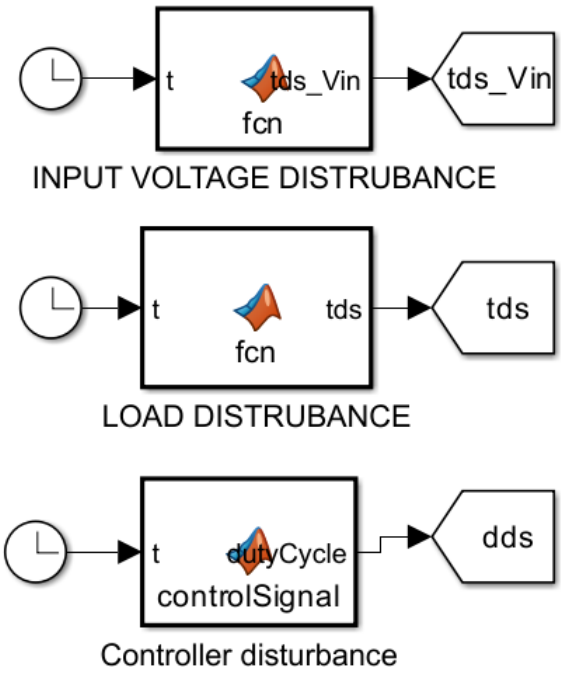

Figure B.7: Disturbance Design in Simulink/Matlab Software

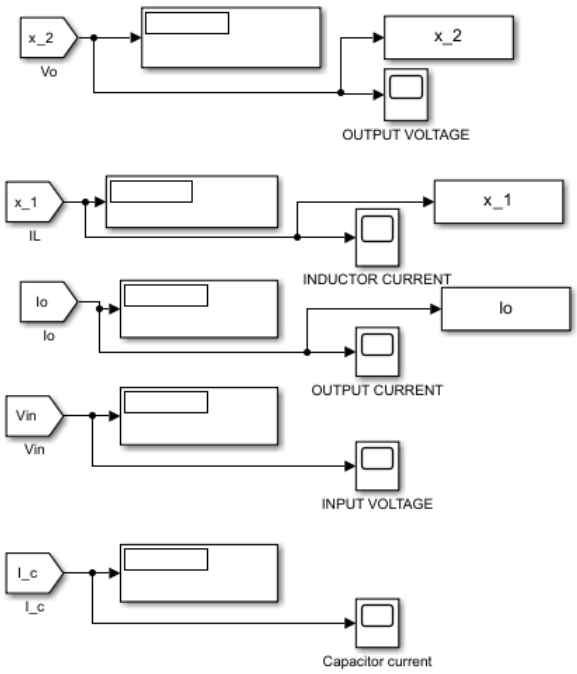

Figure B.8: Output Displays in Matlab/Simulink Software