

Modeling, Analysis, and Control of a Boost Converter Using Arduino and Simulink



**Cluster Innovation Centre
University of Delhi**

Abhay Pawar(152201) Anhad Mehrotra(152209)

April 2025

Semester Long Project

Submitted for the Paper *Control Systems*

Abstract

This project explores the modeling, analysis, and control of a DC-DC boost converter using an Arduino microcontroller integrated with Simulink. The study begins by developing a theoretical model of the circuit based on electrical principles and verifying its time-domain behavior through real-time switching experiments. The system's dynamic response is further examined by applying sinusoidally varying duty cycles to generate an experimental Bode plot, allowing for empirical frequency-domain analysis. Building upon these insights, a closed-loop feedback controller is designed to maintain a stable output voltage. The combined use of simulation and hardware experimentation provides a clear understanding of the working, behavior, and response characteristics of a boost converter in real-world conditions.

1 Introduction

A boost converter, also known as a step-up converter, is a type of DC-DC converter that increases the input voltage to a higher output voltage while reducing current. It plays a crucial role in modern electronics, particularly in battery-powered devices where stepping up voltage is essential for components that require higher power levels than the source provides. The converter operates by rapidly switching a transistor on and off, storing energy in an inductor during the ON phase and releasing it to the load during the OFF phase.

In the context of control systems, the boost converter is an ideal platform to study real-time dynamic behavior, switching control, and system response. Its non-linear nature and sensitivity to component variation make it a practical example for understanding time-domain and frequency-domain analysis, as well as feedback controller design. By analyzing the time response and frequency response of the system, one can develop accurate models and effective control strategies to maintain stable performance.

This project integrates theoretical modeling, simulation using Simulink, and physical implementation via Arduino to explore the dynamic operation of a boost converter. Through this hands-on approach, key control system principles such as time constant, steady-state gain, Bode plots, and closed-loop regulation are visualized and validated. The ultimate goal is to bridge the gap between theoretical understanding and practical application in power electronics and embedded systems.

2 Scope and Objectives

Scope

This project focuses on the modeling, simulation, and control of a DC-DC boost converter using a combination of theoretical analysis and real-time experimentation. The system is developed and tested using Arduino for switching control and Simulink for signal processing and visualization. The study encompasses both time-domain and frequency-domain analysis, as well as the design of a closed-loop feedback controller. The converter is analyzed under practical constraints such as hardware limitations and switching characteristics, making the project highly relevant to embedded control applications.

Objectives

- To model the boost converter using fundamental electrical principles and validate it using time-domain analysis.
- To experimentally generate a frequency response of the converter and plot the Bode diagram.
- To design and implement a feedback control system to regulate the output voltage.
- To gain hands-on experience with Arduino hardware and Simulink software for real-time control applications.

3 Prerequisites and System Overview

3.1 Boost Converter Principle of Operation

A boost converter is a type of DC-DC converter, specifically known as a step-up converter. Its purpose is to convert a lower DC input voltage to a higher DC output voltage. This is achieved through the controlled switching of components and energy storage elements like inductors and capacitors. Boost converters are widely used in power electronics applications, such as LED drivers, and renewable energy systems, where stepping up voltage efficiently is essential.

Below is a simplified version of a boost converter consisting of a DC voltage source e_i , an inductor L , a switch (typically a MOSFET), a diode, and a load resistor R_{load} . In practical applications, the load could be anything from a motor to a lighting system.

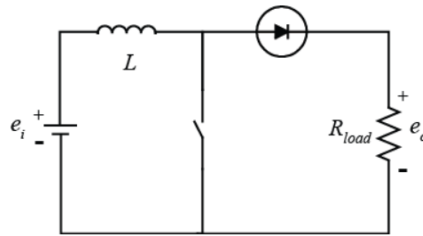
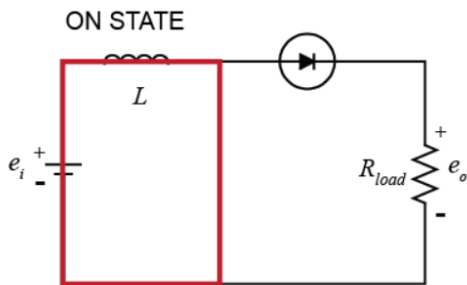
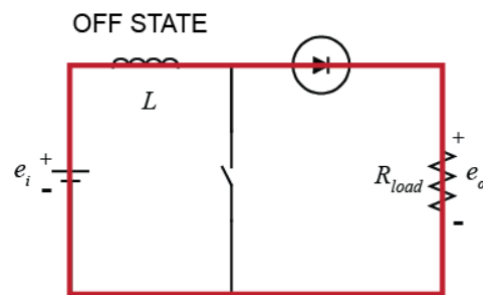


Figure 1: Circuit Diagram

When the switch is closed (ON state), the current flows through the closed switch due to lower path. In this state, the inductor stores energy in the form of a magnetic field. Very little or no current flows through the load during this phase.



(a) ON state of Boost Converter



(b) OFF state of Boost Converter

When the switch is opened (OFF state), the inductor's magnetic field collapses and the stored energy is released. The diode becomes forward biased, allowing the inductor's current to flow to the load. This increases the output voltage beyond the input supply level.

Thus, by rapidly switching the transistor between ON and OFF states, the boost converter stores energy in the inductor and then releases it to the load. This controlled switching results in a higher and more stable output voltage than the input.

3.2 Modeling from first principles

To better understand the boost converter's behavior, we develop a mathematical model based on fundamental electrical laws. This approach, called *modeling from first principles*, uses basic circuit elements and assumes ideal components. The transistor is modeled as a perfect switch (short when ON, open when OFF), and the diode is considered ideal (zero resistance in forward bias, infinite in reverse). We also include an equivalent series resistance (ESR) for the inductor to account for real-world losses.

Symbol	Description
R_{load}	Load resistance
R_{eq}	Inductor ESR
L	Inductance
e_i	Input voltage
e_o	Output voltage

Table 1: Notations Used

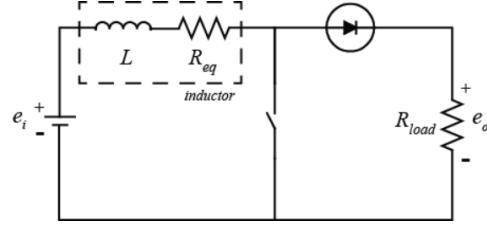


Figure 3: Boost Converter Circuit

ON State Working

We begin with the circuit in its **ON state** and assume that there is initially no current in the circuit ($i(0) = 0$). In this state, all of the current flows through the switch and direction of current will be clockwise as shown in the figure below.

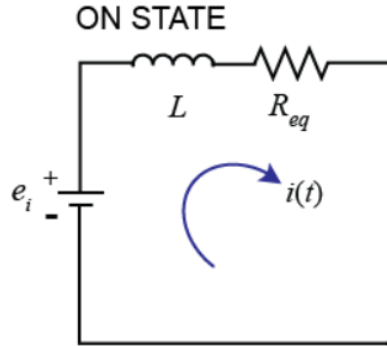


Figure 4: Boost converter circuit in ON state

Applying **Kirchhoff's Voltage Law (KVL)** around the closed loop, the sum of voltages must be zero. This gives us the following differential equation:

$$e_i - L \frac{di}{dt} - iR_{eq} = 0 \quad (1)$$

To solve for the time-domain behavior of the current, we take the Laplace transform (assuming $e_i(t) = E$ is constant):

$$\frac{E}{s} - L(sI(s) - i(0)) - I(s)R_{eq} = 0 \quad (2)$$

Assuming $i(0) = 0$, we solve for $I(s)$:

$$I(s) = \frac{E}{s(sL + R_{eq})} \quad (3)$$

Recognizing this as a standard first-order step response, we apply partial fractions:

$$I(s) = \frac{E/R_{eq}}{s} - \frac{EL/R_{eq}}{sL + R_{eq}} \quad (4)$$

Taking the inverse Laplace transform gives the time-domain current:

$$i(t) = \frac{E}{R_{eq}} (1 - e^{-R_{eq}t/L}), \quad t \geq 0 \quad (5)$$

This response is a first-order exponential rise with time constant $\tau = \frac{L}{R_{eq}}$ and steady-state gain of $\frac{E}{R_{eq}}$.

OFF State Working

We now analyze the circuit when the switch is **open**, i.e., the transistor is OFF. At this point, current previously flowing through the inductor continues to flow through the diode and into the load.

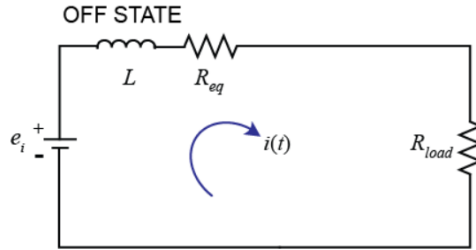


Figure 5: Boost converter circuit in OFF state

Applying Kirchhoff's Voltage Law (KVL), we get the following differential equation:

$$e_i - L \frac{di}{dt} - iR_{eq} - iR_{load} = 0 \quad (6)$$

This equation mirrors the ON state equation but includes the total resistance $R_{eq} + R_{load}$. Taking the Laplace Transform of this with an initial current $i(0) \neq 0$:

$$\frac{E}{s} - L(sI(s) - i(0)) - I(s)(R_{eq} + R_{load}) = 0 \quad (7)$$

Solving for $I(s)$:

$$I(s) = \frac{E}{s(sL + R_{eq} + R_{load})} + \frac{Li(0)}{sL + R_{eq} + R_{load}} \quad (8)$$

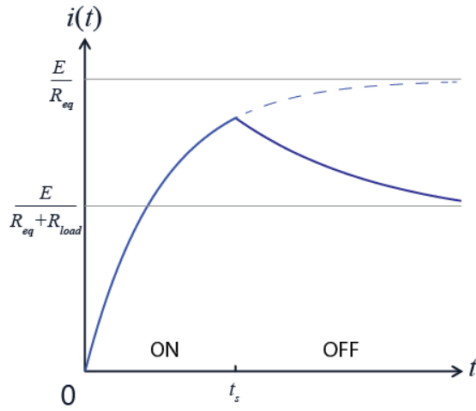
The first term represents the forced response due to the input, and the second term represents the natural response due to initial conditions. Taking inverse Laplace Transform:

$$i(t) = \frac{E}{R_{eq} + R_{load}} \left(1 - e^{-\frac{(R_{eq} + R_{load})t}{L}} \right) + i(0)e^{-\frac{(R_{eq} + R_{load})t}{L}}, \quad t \geq 0 \quad (9)$$

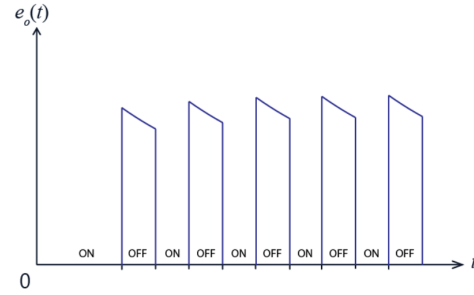
Considering the switch turns OFF at $t = t_s$, we time-shift the expression:

$$i(t) = \frac{E}{R_{eq} + R_{load}} \left(1 - e^{-\frac{(R_{eq} + R_{load})(t-t_s)}{L}} \right) + i(t_s)e^{-\frac{(R_{eq} + R_{load})(t-t_s)}{L}}, \quad t \geq t_s \quad (10)$$

If the ON-time was long enough, then at $t = t_s$, $i(t_s) > \frac{E}{R_{eq} + R_{load}}$. In such a case, the current gradually decays during the OFF state, transferring energy from the inductor to the load.



(a) Current in ON-OFF Switching Cycle



(b) Output Voltage Behavior

As the converter switches repeatedly between ON and OFF states, the current exhibits a ripple, yet on average supplies power to the load. The output voltage can be determined via Ohm's Law as:

$$e_o = i_R \cdot R_{load}$$

where i_R closely follows inductor current during the OFF state and is nearly zero during the ON state.

Hardware Requirements

To implement the boost converter experimentally and validate its theoretical performance, the following hardware components were used:

Inductor – 1 H	Capacitor – 2200 μ F
Resistor – 61 Ω	Diode – 1N4007
MOSFET – IRF540	Battery – 1.5 V, AA
Arduino Uno	Breadboard and Jumper Wires

The complete hardware wiring and the simulation model are shown below:

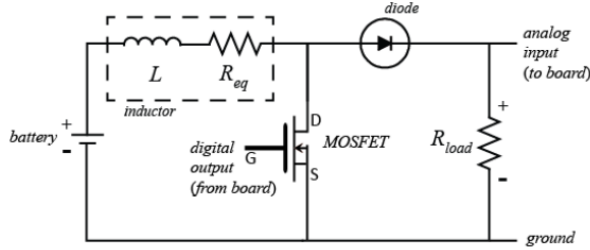


Figure 7: Overall Circuit Diagram

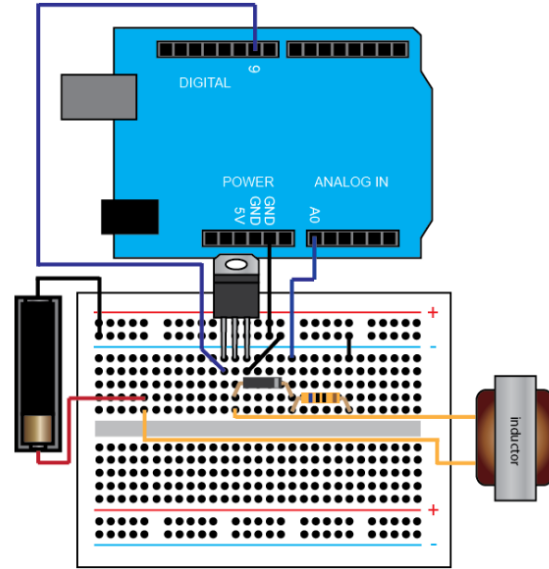


Figure 8: Circuit Schematic

4 Methodology

We begin our experimentation with the first task — analyzing the time-domain response of the boost converter to understand its dynamic switching behavior.

4.1 Time Response Experiment without Capacitor

The Time Response Experiment helps us understand how the boost converter reacts over time when the transistor is periodically switched ON and OFF. By analyzing the transient behavior and steady-state voltages, we gain insight into how effectively energy is transferred and stored within the circuit. This lays the foundation for understanding dynamic performance and guides future control design.

4.1.1 Hardware Setup

We have used the same circuit as given in the earlier figure. The circuit remains unchanged.

4.1.2 Software Setup

We use **Simulink** to control the switching of the transistor, to read the output voltage data from the board, and to plot the data in real time.

The **Pulse Generator** block toggles between 0 and 1 and is connected to the **Digital Write Block** to switch the transistor between its OFF state and its ON state, respectively. Referring to our previous analysis with the components identified above ($L = 1 \text{ H}$, $R_{eq} = 40\Omega$, $R_{load} = 61\Omega$),

the time constant of our circuit is approximately $\tau = 1/40 = 0.025$ seconds in its ON state and $\tau = 1/100 = 0.01$ seconds in its OFF state.

As it takes approximately 4 time constants for a first-order step response to reach 98 percent of its steady-state value, we set up the Pulse Generator block to have a period of 0.2 seconds with a duty cycle of 50 percent. In other words, the circuit is ON for 0.1 seconds and OFF for 0.1 seconds.

To clearly capture the transient response of the circuit, we choose our **sampling time** to be 0.01 seconds. The simulink model is given below:

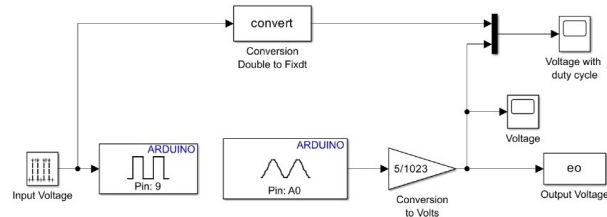


Figure 9: Simulink Model for Time Response Analysis

The given Simulink model then plots the recorded output voltage on a scope and also writes the output data to the MATLAB workspace for further analysis. Running the above model for 2 seconds generates the figure shown below of output voltage versus time.

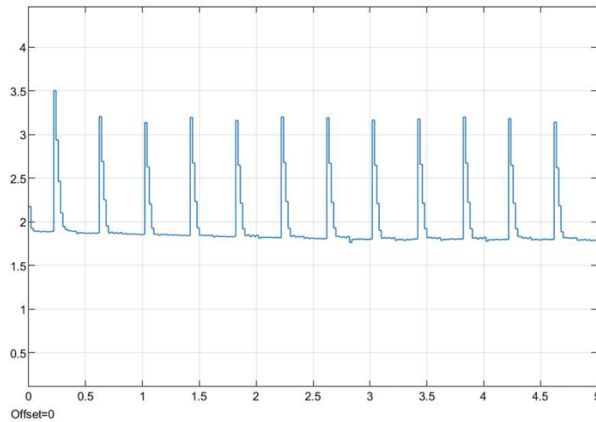


Figure 10: Output voltage response over time

The character of the displayed response matches what we predicted — the voltage across the load decays exponentially when the circuit is in its OFF state and the voltage (current) drops to near zero when the circuit is in its ON state.

Employing Ohm's law and our previous analysis, the maximum voltage across the load is predicted to be $e_o = R_{load}E/R_{eq} \approx 2.44$ Volts (where $E \approx 1.6$ Volts in this case). Therefore, the data shown above matches our expectations reasonably well.

Again employing Ohm's law and our previous analysis, the minimum voltage across the load when the circuit is in its OFF state is predicted to be $e_o = R_{load}E/(R_{eq} + R_{load}) \approx 0.96$ Volts. The actual

data demonstrates a limiting output voltage more in the neighborhood of 1.7 Volts, which while not exactly equal, is acceptable considering all of the simplifications we made in our analysis. Specifically, we treated the transistor, the diode, and the input and output channels of the board as ideal, which they of course are not.

The time constant of the circuit in the OFF state indicated by the above data also seems to be quite similar to the prediction of 0.01 seconds (though a bit larger).

4.2 Time Response Experiment with Capacitor

To prevent the output voltage from dropping significantly during the ON state of the boost circuit, a capacitor is typically added in parallel with the load, as shown in the circuit diagram below.

In this experiment, we include a capacitor in parallel with the load to observe its impact on the output voltage. The main objective is to reduce the ripple and obtain a more stable and continuous output voltage waveform.

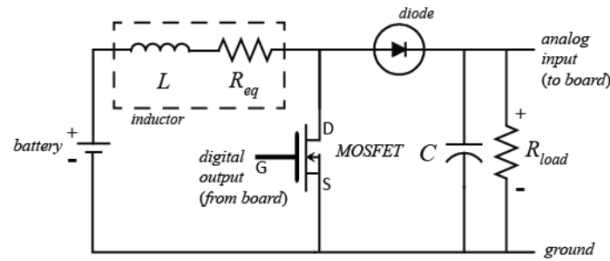


Figure 11: Circuit Diagram (with Capacitor)

Employing the same Simulink model from the previous time response experiment, we now add a 22000 μF capacitor in parallel with the load. A large capacitor value is selected to ensure that the circuit operates in *continuous conduction mode*, where the capacitor does not completely discharge during the ON phase. This slower dynamic response allows us to observe the gradual buildup of voltage and the capacitor's charging effect over time. The following graph has been obtained:

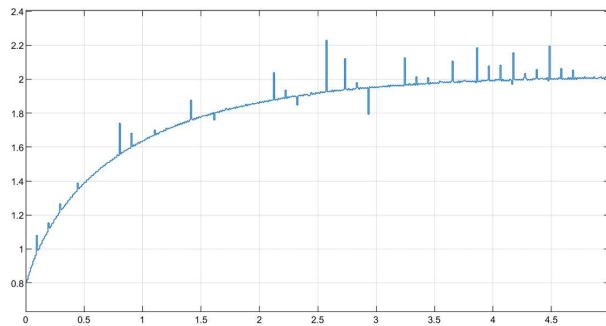


Figure 12: Output voltage response (with Capacitor)

4.2.1 Analytical Insight into Capacitor-Enhanced Boost Circuit

To better understand the dynamics introduced by the capacitor, we briefly analyze the boost circuit using complex impedances in the Laplace domain, assuming zero initial conditions ($i(0) = 0$).

When the switch is **OFF**, the inductor (L) and its ESR (R_{eq}) form a series impedance:

$$Z_1(s) = sL + R_{eq}$$

The capacitor (C) and the load resistor (R_{load}) are in parallel. Their equivalent impedance is:

$$Z_2(s) = \left[\frac{1}{1/Cs} + \frac{1}{R_{load}} \right]^{-1} = \frac{R_{load}}{R_{load}Cs + 1}$$

The circuit behaves like a voltage divider, and the transfer function between input voltage $E_i(s)$ and output voltage $E_o(s)$ is:

$$\frac{E_o(s)}{E_i(s)} = \frac{Z_2(s)}{Z_1(s) + Z_2(s)}$$

Substituting the values of $Z_1(s)$ and $Z_2(s)$, we obtain the second-order transfer function:

$$\frac{E_o(s)}{E_i(s)} = \frac{R_{load}}{LR_{load}Cs^2 + (R_{eq}R_{load}C + L)s + R_{eq} + R_{load}}$$

This expression shows that the addition of the capacitor transforms the system dynamics from first-order to second-order, enabling better ripple suppression and smoother voltage transitions.

In practical applications, boost converters are designed to respond faster and more smoothly. This is achieved by selecting optimized components (e.g., smaller capacitors) and using high-frequency PWM signals. In the next section, we explore the system's behavior under such high-speed switching conditions.

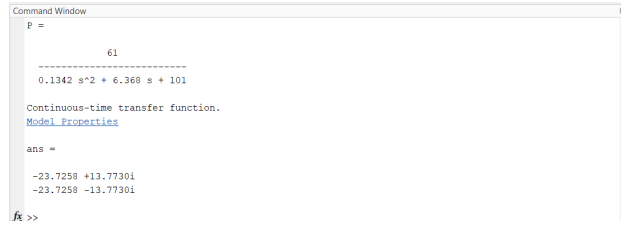
4.2.2 Poles of the Transfer Function

To analyze the stability of the boost converter system, we compute the poles of its open-loop transfer function using MATLAB. The poles provide insight into the system's natural response characteristics and help assess whether the system is stable under the given component values. The following code was used:

MATLAB Code for Bode Plot Generation

```
Rload = 61;
Req = 40;
L = 1;
C = 2200*10^-6;
s = tf('s');
P = Rload/(L*Rload*C*s^2 + (Req*Rload*C + L)*s + Req + Rload)
pole(P)
```

The result obtained was as follows-



```
Command Window
P =
      61
-----
0.1342 s^2 + 6.368 s + 101

Continuous-time transfer function.
Model Properties

ans =

-23.7258 +13.7730i
-23.7258 -13.7730i
fx >>
```

Figure 13: Result of code for calculating poles)

The poles of the transfer function indicate that the system is characterized by complex conjugate poles at $-23.7258 \pm 13.7730i$, which suggests an underdamped response. The real part of the poles, -23.7258 , indicates a stable system, with the magnitude of the real part determining the rate of decay, while the imaginary part, $\pm 13.7730i$, corresponds to the oscillatory behavior of the system. The natural frequency (ω_n) of the system is approximately 27.43 radians per second, and the damping ratio (ζ) is approximately 0.864, indicating that the system will exhibit oscillations that gradually decay over time, with the oscillatory behavior being moderately damped. This suggests that the system's response will involve oscillations that reduce in amplitude as time progresses.

4.3 Frequency Response Identification of the Circuit

The goal of this activity is to build further intuition about the behavior of a boost converter by experimentally determining its frequency response. Instead of relying solely on theoretical models, we identify how the output voltage responds to sinusoidal variations in the input (duty cycle), using real-time data to construct a Bode plot. This black-box approach is particularly useful for approximating the dynamics of a nonlinear system in a specific operating region.

We use simple implementations of the boost converter circuit and apply sinusoidal variations in the duty cycle to experimentally derive the frequency response. This technique is crucial in control design where exact models are difficult to obtain due to switching behavior and component non-idealities.

4.3.1 Hardware Setup

We have used the same circuit as given in the earlier figure, except we will use different value components for the load in order to make the boost circuit respond faster. Specifically, we will employ a $1000\ \Omega$ resistor for the load. The updated circuit diagram is given below:

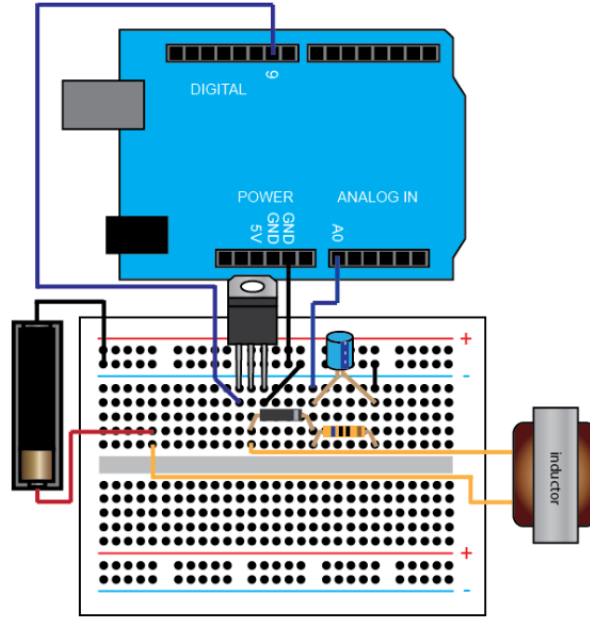


Figure 14: Circuit Diagram for Frequency Response Experiment

4.3.2 Software Setup

Our Simulink model for performing this frequency response experiment is similar to the model used in the Time Response experiment. However, we replace the **Pulse Generator** block with a **Sine Wave** block from the Sources library, since frequency response data is obtained from the system's reaction to sinusoidal inputs across a range of frequencies.

For the **Sine Wave** block, we continue to use a Sample time of $T_s = 0.01$ seconds. Furthermore, we set the **Bias** to 100 and the **Amplitude** to 40. As a result, the Sine Wave block generates a sinusoidal signal that oscillates between 60 and 140, corresponding to a duty cycle variation from approximately 24% to 55%. The frequency of the sine wave input is varied over the course of the experiment.

A depiction of the modified Simulink model is shown below:

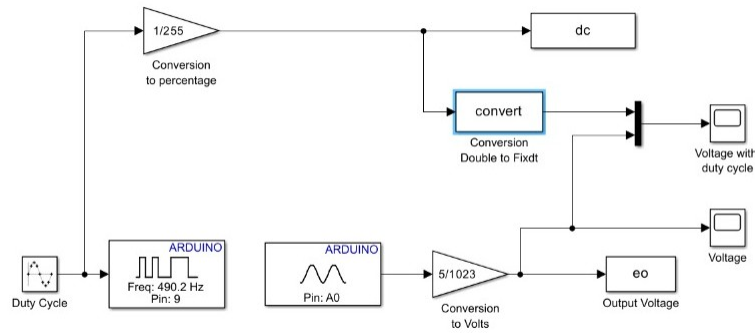


Figure 15: Simulink Model for Frequency Response

4.3.3 Bode Plot Generation

With frequency response analysis, we are interested in examining how a system responds to different sinusoidal input frequencies. For linear (or locally linear) systems, a sinusoidal input will produce a steady-state sinusoidal output at the same frequency but with altered amplitude and phase. This behavior is captured using a **Bode plot**, which represents the system's gain and phase shift versus frequency. In this experiment, we empirically determine this frequency response by applying sinusoidal duty cycle inputs and measuring the system's output.

We begin by setting the frequency of the Sine Wave block in our Simulink model to $\omega = 2\pi$ rad/sec (1 Hz). Running the model gives the following output voltage response for a 1.5V battery input.

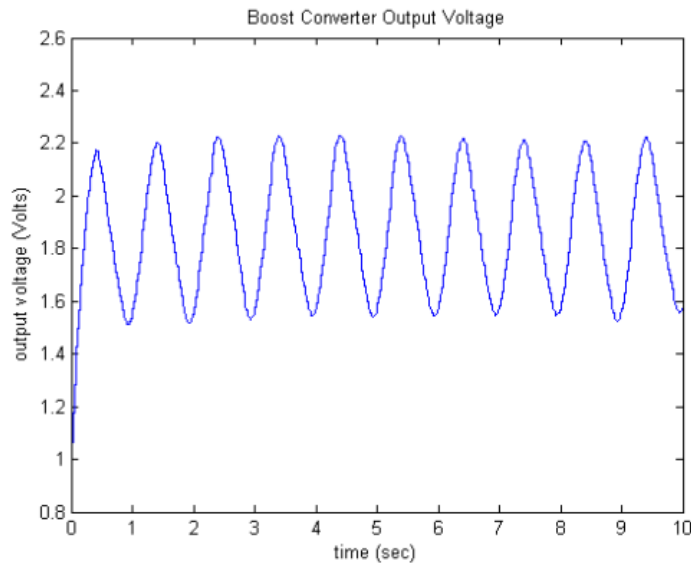


Figure 16: Output response to sinusoidal duty cycle input (1 Hz)

As shown, there is an initial transient caused by natural circuit dynamics. Once it settles, the system produces a sinusoidal output of the same frequency but delayed and scaled—demonstrating the expected steady-state behavior.

To calculate the gain (magnitude) and phase shift:

- The duty cycle amplitude is approximately $40/255 \approx 0.16$.
- Output voltage varies about 0.66 V peak-to-peak \Rightarrow amplitude = 0.33 V.
- Gain = $0.33/0.16 \approx 2.06 \Rightarrow 20 \log_{10}(2.06) \approx 6.28$ dB.
- Time lag = 0.16 s \Rightarrow Phase = $-0.16 \times 2\pi \approx -1$ rad $\approx -58^\circ$.

We repeat this process for nine frequencies ranging from 0.01 Hz to 5 Hz. The details for the same have been discussed later in the Results section.

This data can then be compiled in the form of a Bode plot using MATLAB:

MATLAB Code for Bode Plot Generation

```
freq = [0.0628 0.314 0.628 1.26 1.57 3.14 6.28 15.7 31.4]; % in rad/sec
mag = [17.29 16.38 16.76 16.86 16.03 16.27 16.12 15.80 15.36]; % in dB
phase = [87.06 52.86 1.89 -24.42 -114.93 -128.66 -148.39 -201.71 -241.50]; % in degrees

% Create Bode plot (2 subplots)
figure;

% Magnitude plot
subplot(2,1,1);
semilogx(freq, mag, 'o-', 'LineWidth', 1.5);
grid on;
xlabel('Frequency (rad/sec)');
ylabel('Magnitude (dB)');
title('Boost Converter Empirical Bode Diagram');

% Phase plot
subplot(2,1,2);
semilogx(freq, phase, 's-', 'LineWidth', 1.5);
grid on;
xlabel('Frequency (rad/sec)');
ylabel('Phase (degrees)');
```

In the next activity, we design a **feedback controller** to regulate the output voltage and improve dynamic performance based on the insights gained from the frequency response analysis.

4.4 Feedback Control of a Boost Converter Circuit

In this section, we implement feedback control for the boost converter circuit to regulate its output voltage. The control strategies include both static and dynamic compensation methods. A Proportional-Integral (PI) controller is used to improve the performance of the system. While static compensation applies a constant gain to correct steady-state errors, dynamic compensation introduces frequency-dependent elements to enhance the system's transient response and stability.

4.4.1 Hardware Setup

The hardware configuration used for implementing feedback control is largely the same as the setup used for frequency response identification in the previous activity. The only difference lies in the PWM output pin used on the Arduino. Instead of pin 9, we now use pin 6, which generates a PWM signal at 980 Hz, suitable for driving the boost converter under closed-loop control.

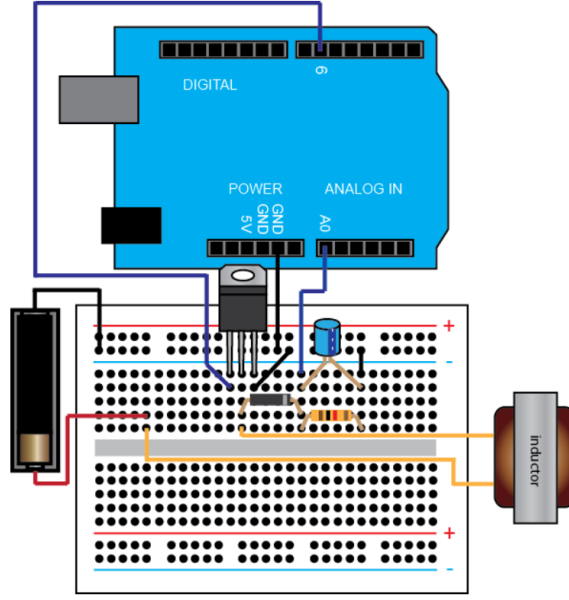


Figure 17: Circuit Schematic for Feedback Control

4.4.2 Static Compensator Design

Here, a static compensator is implemented using a simple proportional controller to regulate the output voltage of the boost converter. The control strategy is designed to respond to changes in the desired output voltage setpoint by adjusting the PWM duty cycle accordingly.

The proportional controller used here has a gain of $K_p = 0.5$. The reference voltage is initially set to 2.3 V and steps up to 3.2 V after 3 seconds. The controller calculates the error between the actual output voltage e_o and the nominal output voltage $\bar{e}_o = 1.8$ V, given by $\Delta e_o = e_o - \bar{e}_o$. This error signal is then multiplied by the proportional gain to generate a duty cycle correction Δdc .

Since the plant model (boost converter) is linearized around the nominal operating point, the controller only acts on deviations from this point. Therefore, the final duty cycle command is constructed by adding the nominal duty cycle \bar{dc} (which is approximately 39% or 100/255) to the output of the controller:

$$dc = \Delta dc + \bar{dc}$$

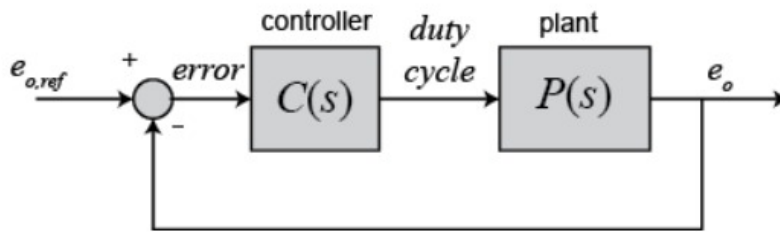


Figure 18: Boost converter control loop

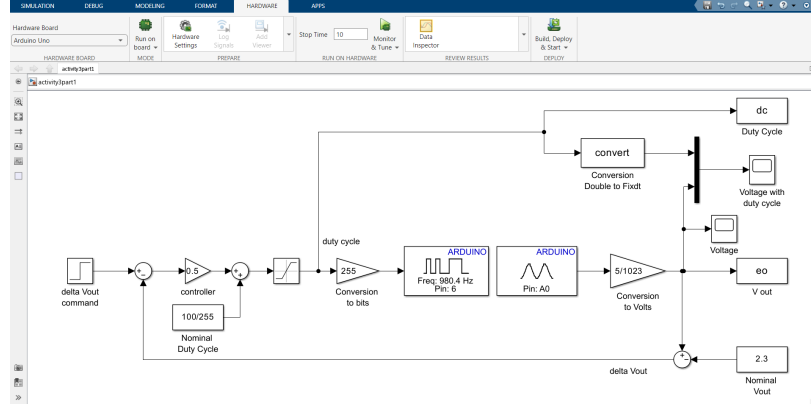


Figure 19: Simulink Model for Frequency Response

This method ensures that the controller fine-tunes the duty cycle only as needed to maintain or reach the desired output voltage, assuming that the system is already operating near the nominal point. The overall control loop operates by reading the output voltage via the Analog Read block, computing the error, generating the duty cycle correction, and sending the PWM signal back to the converter using the Analog Write block. This static compensator setup provides a basic form of regulation and serves as the foundation before introducing more advanced dynamic compensation.

4.4.3 Dynamic Compensator Design

In contrast to static compensators, which rely solely on the current input, dynamic compensators incorporate elements that depend on both past and future inputs. This is achieved through the inclusion of integrative and derivative components in the controller, represented by terms involving the Laplace variable s . Specifically, multiplication by s corresponds to differentiation, while division by s corresponds to integration. Consequently, the output of a dynamic compensator is influenced by the history and rate of change of the input signal, making it responsive to dynamic variations in the system.

The primary objectives in designing a dynamic compensator for the boost converter are:

- **Achieving Zero Steady-State Error:** Ensuring that the output voltage accurately tracks the desired setpoint over time.
- **Enhancing Phase Margin:** Improving system robustness and minimizing overshoot by targeting a phase margin of approximately 60° .
- **Increasing Gain Crossover Frequency:** Enhancing the speed of response by aiming for a gain crossover frequency of around 20 rad/s .

It is important that the chosen crossover frequency ω_c is significantly below the PWM switching frequency to avoid amplifying switching harmonics and ensure proper filtering. In this case, selecting $\omega_c = 20 \text{ rad/s}$ is appropriate, as it lies well below the PWM frequency and slightly above the resonant frequency of the plant, ensuring both speed and stability.

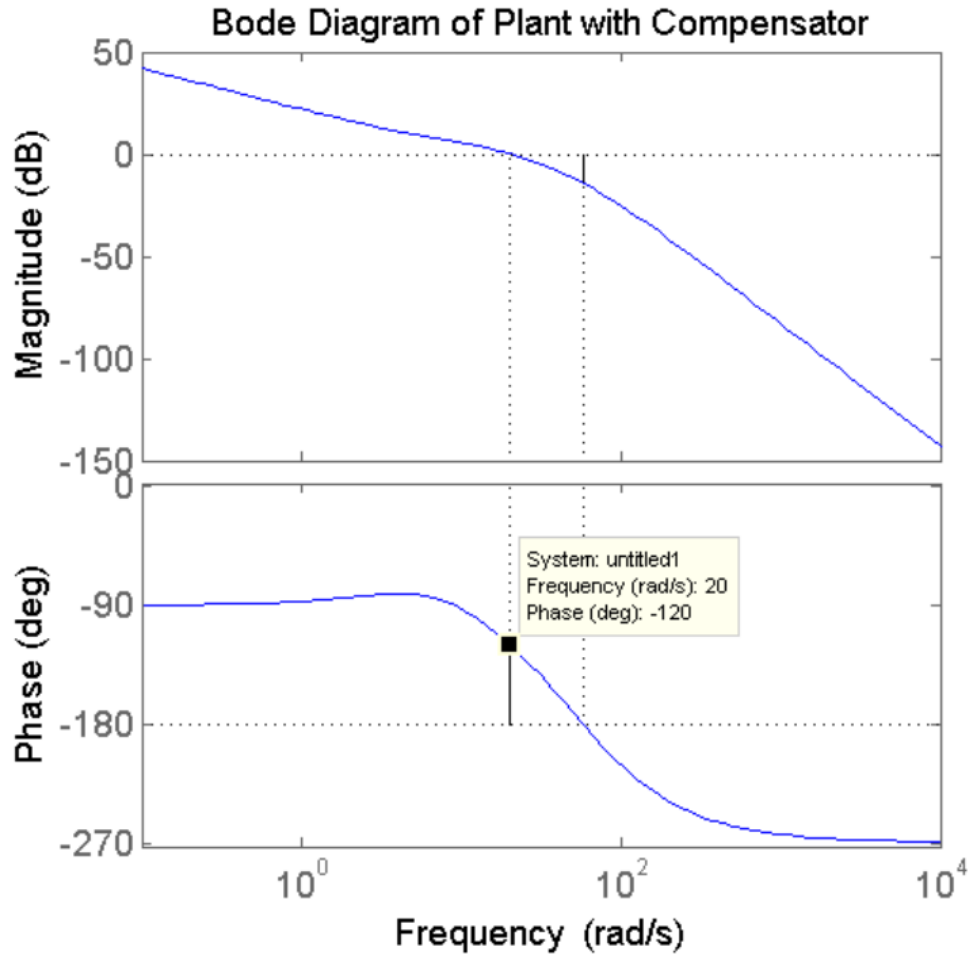


Figure 20: Bode Plot After Applying Dynamic Compensator

The Bode plot shown above demonstrates the improvements brought about by the dynamic compensator. Specifically:

- The phase margin is increased to approximately 60° , indicating improved damping and reduced likelihood of oscillations.
- The gain crossover frequency shifts to about 20 rad/s, signifying faster system response.
- Overall frequency response becomes more stable and predictable across a broader range of frequencies.

These changes confirm that the dynamic compensator effectively meets the desired performance and stability criteria, paving the way for reliable real-time control using the Arduino and Simulink framework.

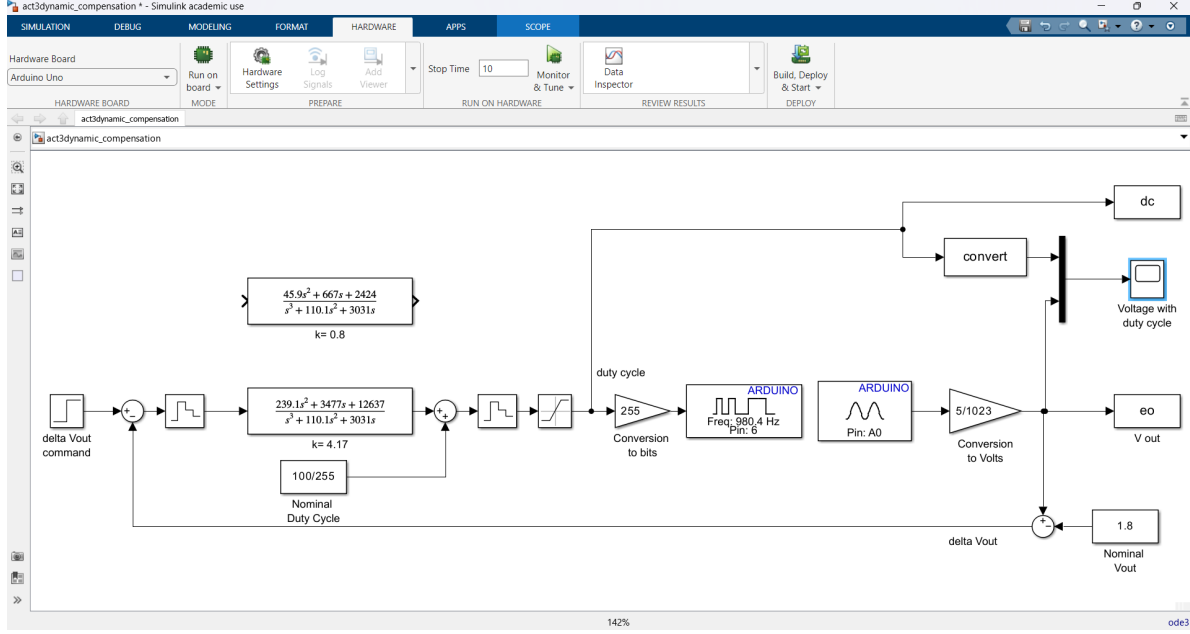


Figure 21: Simulink Implementation of the Dynamic Compensator for the Boost Converter

The figure above shows the Simulink model used to implement the dynamic compensator. It includes the plant model, compensator block, and the interface blocks for real-time communication with the Arduino.

Before executing the Simulink model, the sampling time was set to $T_s = 0.02\text{ s}$ to ensure appropriate discretization for real-time control.

The dynamic compensator was implemented as a Proportional-Integral (PI) controller with the following transfer function:

$$G_c(s) = K_p + \frac{K_i}{s}$$

Where:

- $K_p = 4.17$ is the proportional gain.
- $K_i = 0.8$ is the integral gain.

These gain values were selected based on the desired performance specifications, aiming for a phase margin of approximately 60° and a gain crossover frequency around 20 rad/s . The chosen integral gain of $K_i = 0.8$ strikes a balance between eliminating steady-state error and maintaining system stability without introducing excessive overshoot or oscillations.

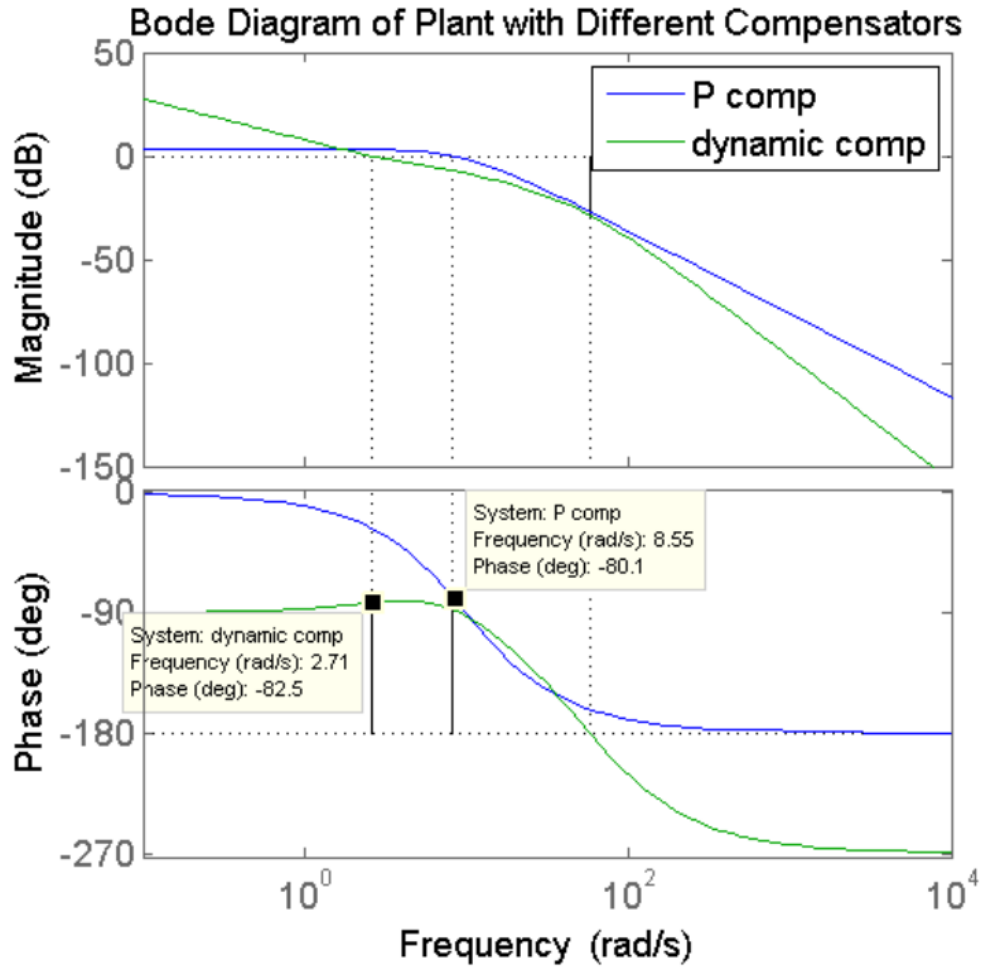
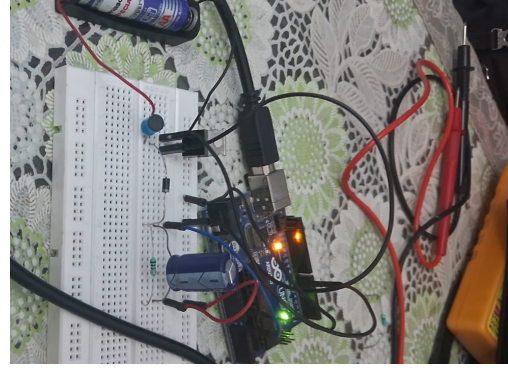
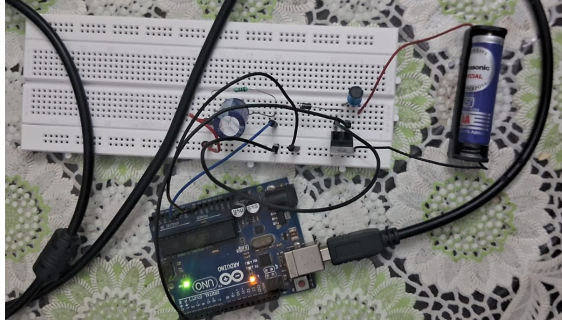


Figure 22: Bode Plot of the System with PI Controller

The Bode plot demonstrates that the PI controller increases the phase margin to approximately 60° , indicating improved system stability. Additionally, the gain crossover frequency shifts to around 20 rad/s, suggesting a faster system response. These enhancements confirm that the PI controller effectively meets the desired performance and stability criteria, facilitating reliable real-time control using the Arduino and Simulink framework.

5 Results and Conclusion

Below are the physical circuits constructed for the boost converter experiments discussed in this activity. These setups were used to validate the theoretical and simulation-based findings in real hardware conditions

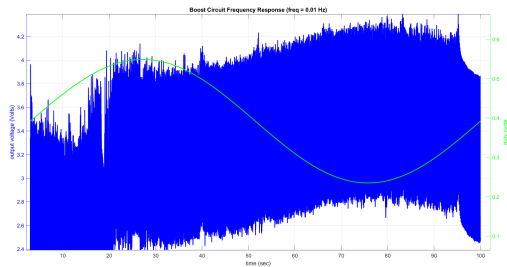


5.1 Frequency Response Identification

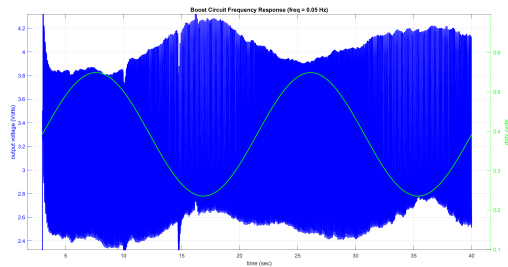
The frequency response of the boost converter system was evaluated by applying sinusoidal perturbations at various frequencies and measuring the corresponding output amplitude and phase shift. The table below summarizes the experimentally obtained data, including frequency values in Hz and rad/sec, magnitude in decibels (dB), phase shift in degrees, recommended run times, and the measured output voltage amplitude. Subsequently, the individual response curves at each test frequency are presented, followed by the overall Bode plots illustrating the amplitude and phase characteristics of the system.

Frequency (Hz)	Freq. (rad/sec)	Magnitude (dB)	Phase (deg)	Run Time (s)	Amplitude(V)
0.01	0.0628	22.97	-259.49	100	2.2092
0.05	0.314	23.01	-207.79	40	2.2190
0.10	0.628	22.81	-112.91	20	2.1676
0.20	1.26	23.36	-291.65	15	2.3094
0.25	1.57	22.73	-95.51	12	2.1469
0.50	3.14	22.84	-222.39	10	2.1744
1.00	6.28	22.76	-340.20	10	2.1542
2.50	15.7	22.64	-217.93	6	2.1249
5.00	31.4	22.77	-235.80	5	2.1570

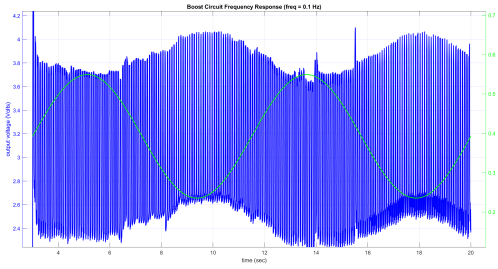
Table 2: Experimentally determined frequency response data with recommended run times



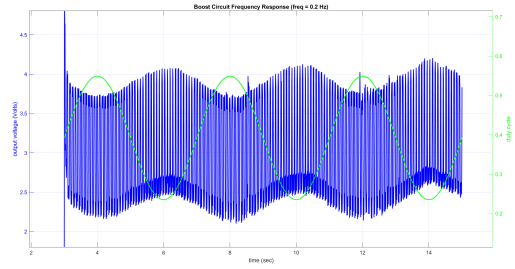
(a) Frequency Response (freq=0.01 Hz)



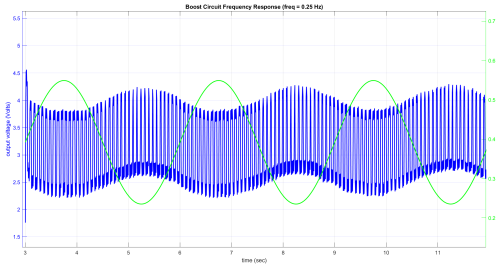
(b) Frequency Response (freq=0.05 Hz)



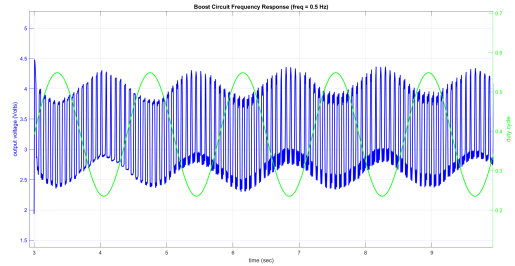
(a) Frequency Response (freq=0.1 Hz)



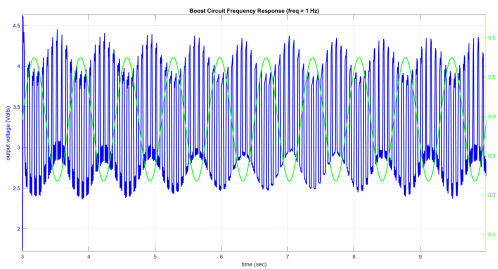
(b) Frequency Response (freq=0.2 Hz)



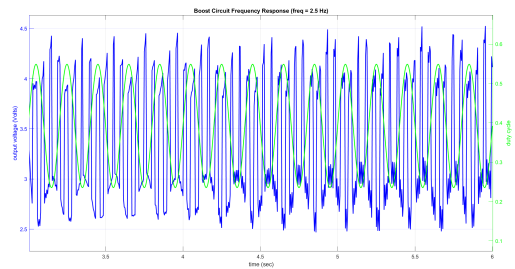
(a) Frequency Response (freq=0.25 Hz)



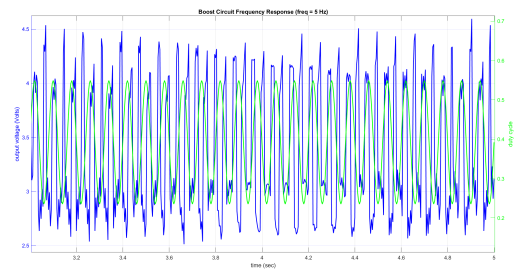
(b) Frequency Response (freq=0.5 Hz)



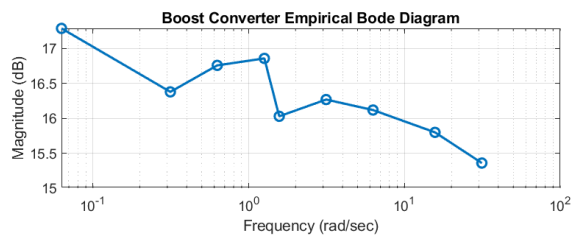
(a) Frequency Response (freq=1 Hz)



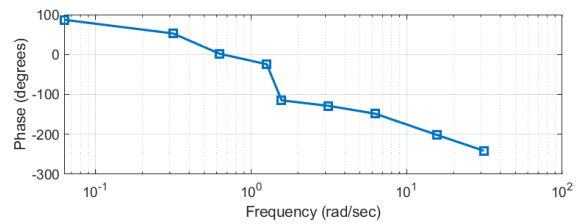
(b) Frequency Response (freq=2.5 Hz)



(a) Frequency Response (freq=5 Hz)



(a) Bode Plot showing amplitude response



(b) Bode Plot showing phase response

The experimentally obtained Bode plot aligns with the theoretical expectations of a **Type 0 second-order system**. This is evident from the *flat magnitude response at low frequencies*, the *roll-off slope suggesting two poles*, and the *phase shifting toward -180°* . Although the exact numerical behavior deviates from the ideal linear averaged model, the overall frequency response character is preserved.

The theoretically predicted **non-minimum phase zero** is not observed within the tested frequency range, likely because its influence lies beyond the practical measurement bandwidth. High-frequency identification is inherently limited by factors such as *signal attenuation*, *sampling constraints*, and *quantization effects*. Nonetheless, these experimental insights provide valuable understanding of the converter's dynamic characteristics and help define the **effective frequency range** suitable for control applications.

The empirically-derived Bode diagram reflects a **Type 0 system** with two poles. A system is classified as Type 0 when there are no poles at the origin in its open-loop transfer function. In frequency response terms, this means the magnitude plot is flat (i.e., has a 0 dB/decade slope) at low frequencies, and the phase plot approaches 0 degrees. Such systems typically exhibit a steady-state error for step inputs and are less capable of rejecting low-frequency disturbances compared to higher type systems.

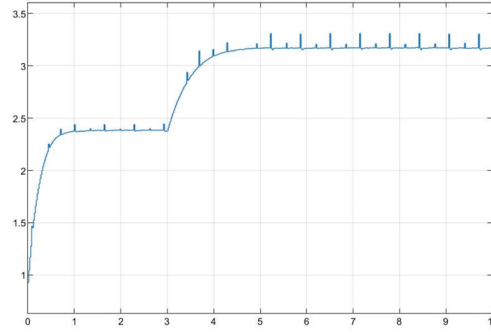
In the Bode magnitude plot, the low-frequency portion is approximately flat and approaches 9.5 dB, corresponding to a DC gain of about 3, since $20 \log 3 \approx 9.5$. The presence of two poles is indicated by the magnitude plot beginning to roll off at a slope of approximately -40 dB/decade, and the phase plot trending toward -180° . From visual inspection and trial-and-error estimation, it appears that the pole break occurs around 10 rad/s, which leads to an estimated natural frequency $\omega_n \approx 10$ rad/s.

Assuming that the poles are complex conjugates, and given that the frequency response lacks a resonant peak, it can be inferred that the damping ratio ζ is greater than 0.707. After several iterations, choosing a damping ratio of $\zeta \approx 0.9$ matches the experimental Bode data reasonably well. This assumption will be used in the modeling of the plant.

5.2 Feedback Control using Static Compensator

The performance of the boost converter under static compensation was evaluated by applying a step change to the reference voltage and observing the output response. The setup and implementation details were discussed in the methodology section. The following graph illustrates the system's output voltage tracking behavior when subject to this control strategy. It serves as a visual representation of how effectively the proportional controller maintains regulation around the desired setpoint.

From the graph, it can be observed that the output voltage initially rises to approximately 2.5 V in response to the first reference input. At around 3 seconds, a step change in the reference voltage is applied, and the output subsequently adjusts to track the new setpoint near 3.3 V. The system exhibits a smooth and relatively fast transient response with minimal overshoot and steady-state



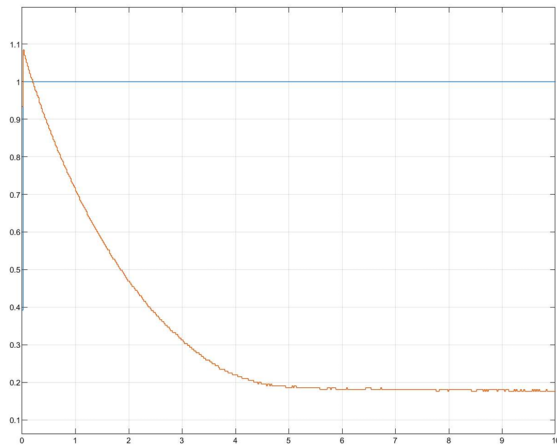
(a) Frequency Response (freq=5 Hz)

oscillations, indicating effective regulation.

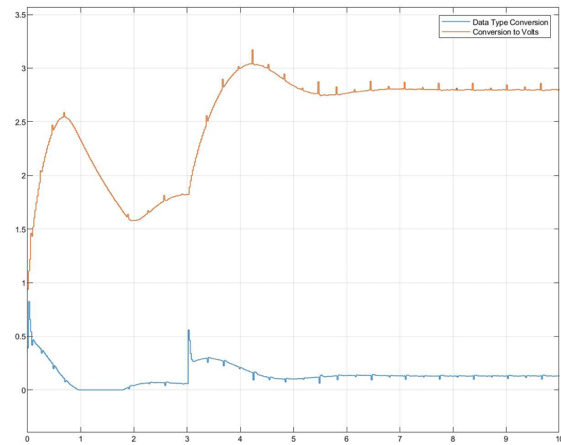
These results validate the proper implementation of the static compensator. The proportional control action is able to reduce the steady-state error significantly and ensure that the output closely follows the reference signal. The ripple observed is characteristic of switching converters and remains within acceptable bounds, demonstrating that the feedback mechanism successfully maintains output stability.

5.3 Feedback Control using Dynamic Compensator

In this subsection, the behavior of the system is analyzed under feedback control using a dynamic compensator. Two different gain values $K = 4.17$ and $K = 0.8$ are used to evaluate the system's performance. The following graphs illustrate the transient and steady-state response for each case, demonstrating the impact of the gain value on system dynamics.



(a) System response using dynamic compensator with $K = 4.17$



(b) System response using dynamic compensator with $K = 0.8$

The graphs illustrate the performance of the boost converter under dynamic compensator control with two different gain values, $k = 4.17$ and $k = 0.8$.

In the first graph ($k = 4.17$), the system output rapidly converges to the desired reference voltage with minimal steady-state error, indicating high accuracy and stability. The second graph ($k = 0.8$) shows a slower response with higher steady-state error, highlighting undercompensation and reduced controller authority.

These observations validate the simulation results, confirming that a higher gain in the dynamic compensator significantly improves the regulation performance of the boost converter.

6 Future Scope

There are some improvements and extensions which can be done to enhance the performance and applicability of the boost converter system. Some of them include:

- **Closed-loop PID control:** Future work can involve designing and tuning PID controllers for precise output voltage regulation under varying load and input conditions.
- **Digital control via PWM optimization:** Enhancing the control strategy using advanced digital techniques such as duty cycle linearization and high-frequency PWM for improved transient response.
- **Efficiency improvement and thermal analysis:** The system's efficiency can be optimized by studying component losses, implementing soft switching, and performing thermal profiling of MOSFETs and diodes.
- **Real-world deployment:** The converter can be embedded into renewable energy systems (like solar panels) or battery charging systems, integrating MPPT algorithms and low-power microcontrollers.

7 Appendix

To access all the Simulink models, MATLAB codes, and related resources for this project, you can visit the GitHub repository provided below:

<https://github.com/Anhad-01/Boost-Converter-Analysis-and-Control-using-Simulink>

8 References

The experiments and theoretical formulations in this project were primarily guided by resources provided by the Control Tutorials for MATLAB and Simulink (CTMS) at the University of Michigan. Detailed activities for modeling, time response, and frequency response of the boost converter are referenced below:

- **Boost Converter Activity A:** https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_BoostcircuitA
- **Boost Converter Activity B:** https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_BoostcircuitB
- **Boost Converter Activity C:** https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_BoostcircuitC

The component specifications were obtained from the following datasheets:

- **IRF540 MOSFET Datasheet:**
<https://www.vishay.com/docs/91021/irf540.pdf>
- **1N4007 Diode Datasheet:** <https://bit.ly/4d162KO>