ECE 31033 Project #1: DC—DC Converter

ID Number: 229,506

Contents

Objective

The overall objective of this project was to simulate a step-down DC—DC converter, taking an 800V input and bringing it to a 400V output. In addition, the circuit has a load range of 50 to 250 kW with a switching frequency.

Using the information from above, we were tasked with analytically calculating the inductance and the minimum capacitance for the circuit. With these values, a Forward Euler integration algorithm was implemented to simulate the ideal circuit behavior by numerically solving the differential equations.

The simulation required the use of four MATLAB files, with two of them being functions that were implemented in the other files. The simulation plotted each circuit component's voltages and currents in the transient state and then repeated the process for the steady state, by looking at the last couple of periods. After the plotting was completed, the average value for each waveform was calculated. Finally, the efficiency of the circuit was calculated.

Brief Overview of the MATLAB Files

There were 4 required files for this project. Two of the files were functions and two of the files were scripts. The code for each file can be found after the results section of this document. Instead of pasting the code at the end of the document, each file was exported using a Visual Studio Code extension to preserve formatting and readability .

"sw.m"

This file was a function that created a Fourier series-based triangle wave which was compared to the duty cycle to determine the state of the transistor.

The inputs to this function were the duty cycle and a single instant in time. The output was a Boolean; 0 if the transistor if turned off or 1 if the transistor is turned on.

"buck.m"

This file was a script containing the Forward Euler integration algorithm. The algorithm consisted of a while loop that calculated the instantaneous voltage and current of each component in the circuit.

The script called the function "sw.m", where its output was used to determine if both the transistor and the diode were on or off.

"aver.m"

This file was a function that was used to calculate the average of a waveform by using a Reimann sum and taking the last period of the waveform.

The inputs to this function were the waveform, the period, and the period between the samples. The output is the average value of the waveform.

"buckproc.m"

The final file was a script that acted as a main function. This script contained each of the analytically solved component values while also calling the file "buck.m" to calculate the voltage and currents of each of the waveforms. After calling "buck.m", it was able to call "aver.m" to calculate the average value of each of the waveforms. The script then plotted each of the waveforms in both the transient and steady states using ideal and non-ideal conditions.

Analytical Calculations

Buck Converter Overview

The general schematic of a buck converter can be seen as follows:

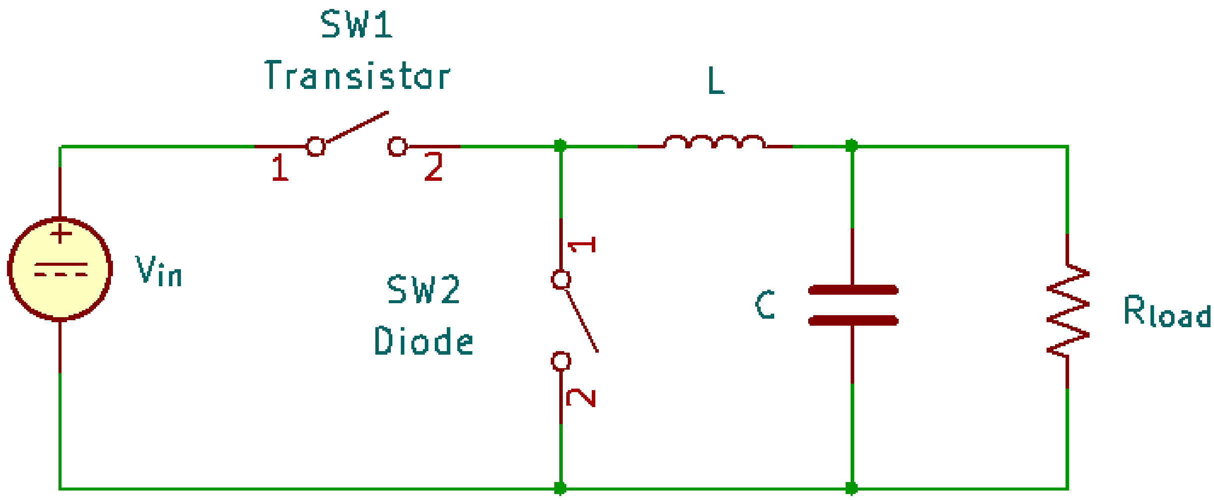

Figure 1: Standard circuit schematic for a buck converter.

The transistor switch and diode switch are opposites to each other; only one can be on at a time. Therefore, the circuit shown in Figure 1 has two configurations.

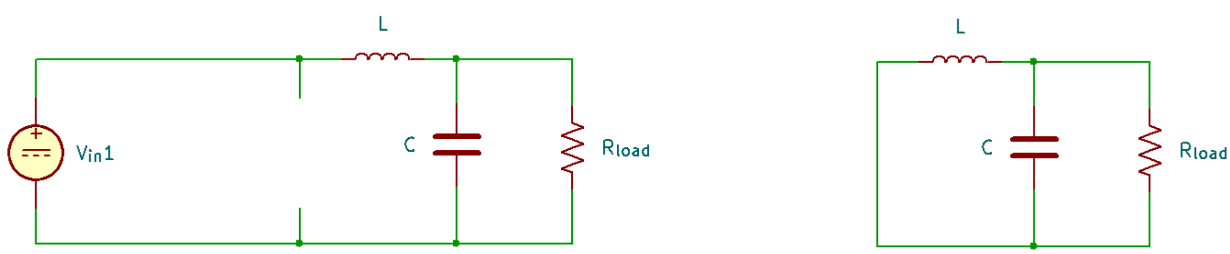

Figure 2: Buck converter schematic for when the transistor is on and for when the diode is on.

With each configuration, we can apply Kirchoff's Voltage and Current laws to determine the values of each of the circuit components.

Component Values

According to the project specifications, we were not given the values for both the inductor and the capacitor. I was able to determine that I would need to solve for the values of the inductor, capacitor, duty cycle, switching period, and finally the heavy and light load resistor.

To create an expression duty cycle, we can use the relationship between the load voltage and input voltage.

$$D = \frac{V_{load}}{V_{in}} = \frac{400}{800} = 0.5$$

Equation 1: Expression to find Duty Cycle.

The expression for switching period is simply the inverse of the switching frequency.

$$T_{sw} = \frac{1}{f_{sw}} = \frac{1}{10000} = 10^{-4}s$$

Equation 2: Expression to find Switching Period.

To calculate the load resistance, we can rearrange Ohm's law and isolate the resistance.

$$P_{load} = \frac{(V_{load})^2}{R_{load}} \rightarrow R_{load} = \frac{(V_{load})^2}{P_{load}}$$

Equation 3: Expression to find Load Resistance.

With this simplified expression, we can solve for both the heavy and light load resistance.

$$R_{load,light} = \frac{(V_{load})^2}{P_{load}} = \frac{(400)^2}{50000} = 3.2\Omega$$

Equation 4: Expression to find Light Load Resistance.

$$R_{load,heavy} = \frac{(V_{load})^2}{P_{load}} = \frac{(400)^2}{250000} = 0.64\Omega$$

Equation 5: Expression to find Heavy Load Resistance.

Next, we can determine an expression for critical inductance. Using this critical inductance value, we can solve for the actual inductor to use with this circuit.

$$L_{crit} = \frac{R_{load,light} \cdot (1 - D)}{2 \cdot f_{sw}} = \frac{(3.2) \cdot (1 - 0.5)}{2 \cdot 10000} = 8e^{-5}H$$

Equation 6: Expression to find Critical Inductance.

$$L = 1.1 \cdot L_{crit} = (1.1) \cdot (8e^{-5}) = 8.8e^{-5}H$$

Equation 7: Expression to find Inductance.

Finally, we can solve for the minimum capacitance, using some of the values that were given in the project specification in addition to ones that we calculated above.

$$C \geq \frac{1}{8L}\left({T_{sw}}^2 \cdot (1-D)\right) \cdot \frac{V_{load}}{\Delta V_{load}}$$

$$\geq \frac{1}{8 \cdot (8.8e^{-5})}\left((10^{-4})^2 \cdot (1-0.5)\right) \cdot \frac{400}{10}$$

$$\geq 2.84e^{-4}F$$

Equation 8: Expression to find minimum Capacitance.

The values that were calculated above were used in the Forward Euler integration algorithm to find the instantaneous value of each circuit component.

Fourier Series Triangle Wave

To create a Fourier Series based triangle wave, I needed to calculate the values of the coefficient for each harmonic wave. Using the values for each coefficient, it can be put into a summation. Knowing that the triangle is an even function, we can use the following as the equation for the Fourier Series:

$$x(t) = a_0 + \sum_{k=1}^{\infty}(a_k + \cos(k\omega t))$$

Equation 8: General Fourier Series for an even function.

The coefficient, a_k, can be determined as:

$$a_k = \frac{2}{T_{sw}}\int_0^{T_{sw}} (x(t) \cdot \cos(k\omega t))dt$$

Equation 8: General expression to determine the Fourier Series coefficients.

Equation 8 can be broken into two integrals. The first integral accounts for when the triangle has a positive slope and the second integral accounts for when the triangle has a negative slope.

$$a_k = \frac{2}{T_{sw}}(\int_0^{T_{sw}} (t \cdot \cos(k\omega t))dt + \int_{DT_{sw}}^{T_{sw}} (-t \cdot \cos(k\omega t))dt)$$

Equation 9: General expression for the Fourier Series coefficients of a triangle wave.

The calculus and the algebra used to solve this equation can get quite out of hand. To simplify this expression, Equation 9 was inputted into an online calculator, leaving us with Equation 10.

$$a_k = 2 \cdot \frac{4 \cdot \cos(0.5k\omega t) - 2 \cdot \cos(k\omega t) - 2}{(k\omega t)^2}$$

Equation 10: Final expression for the Fourier Series coefficients of a triangle wave.

We can insert Equation 10 into the summation in Equation 8 to create the triangle wave.

Forward Euler Integration

In the buck converter circuit, there are 5 components: the transistor switch, the diode switch, the inductor, the capacitor, and the resistor. In the Forward Euler integration algorithm, the voltage and the current for each component was calculated.

Switch 1 – The Transistor

When the switch is on, the voltage of Switch 1 is equivalent to the input voltage. If the switch is off, there is no voltage running through the switch.

$$V_{Sw1} = V_{in} \cdot (switch\ state) = 800V \cdot (switch\ state)$$

Equation 11: Expression to find the voltage of Switch 1.

If the switch state is off, the value of the switch state in Equation 11 will become zero. If the switch state is on, the value of the switch state in Equation 11 will become 1. Therefore, the value of switch 1's voltage will be zero if the switch is off and 800 if the switch is on.

The current of the switch is calculated using the relationship between the inductor current and the switch state.

$$i_{Sw1} = i_L \cdot (switch\ state)$$

Equation 12: Expression to find the current of Switch 1.

Like for the voltage, if the switch state is off, the value of the switch state in Equation 12 will become zero. If the switch state is on, the value of the switch state in Equation 12 will become 1. Therefore, the value of switch 2's current will be zero if the switch is off and i_L if the switch is on.

Switch 2 – The Diode

Fundamentally, Switch 2 is like Switch 1 but differs slightly. Neither switch can be turned on or off at the same time. The values for current and voltage are similar.

$$V_{Sw2} = (-V_{load}) \cdot (switch\ state) = (-800) \cdot (switch\ state)$$

Equation 13: Expression to find the voltage of Switch 2.

$$i_{Sw2} = i_L \cdot (1 - switch\ state)$$

Equation 12: Expression to find the current of Switch 1.

Inductor

If we were to plot the voltage of the inductor, it would look reminiscent of a triangle wave. However, the voltage when the slope is positive and negative are different to each other.

$$V_L^{+\ slope} = V_{in} - V_{load} = 800 - 400 = 400$$

$$V_L^{-\ slope} = -V_{load} = -400$$

Equation 13: Expression to find the voltage of the inductor.

To determine the current for the inductor, the Forward Euler integration algorithm was implemented.

$$i_L(t + \Delta t) = i_L(t) \cdot \frac{(switch\ state) \cdot V_{in} - V_{load}(t)}{L}$$

Equation 14: Expression for the Forward Euler Integration to find inductor current.

Like with other components, the switch state either returns a 0 or a 1. Therefore, the numerator of the fraction in Equation 14 will resemble either the top or bottom line of Equation 13.

Capacitor

The voltage of the capacitor in this circuit is equivalent to the load voltage. Therefore, we can write the expression for capacitor voltage as:

$$V_C = V_{load}$$

Equation 15: Expression to find the capacitor voltage.

The current of the capacitor can be determined by taking the inductor voltage and subtracting the current through the load.

$$i_C = i_L - \frac{V_{load}}{R_{load}}$$

Equation 16: Expression to find the capacitor current.

Resistor

The value of the load's voltage was given in the project specification, so there was nothing to determine the voltage. To calculate the current through the load, we can rearrange Ohm's law to solve for current.

$$i_{load} = \frac{V_{load}}{R_{load}}$$

Equation 17: Expression to find the current of the load resistor.

The value R_{load} of depends on if the simulation is using heavy conditions or light conditions.

Non-Ideal – Heavy Load

Circuit Differences

In a non-ideal circuit, there are a couple differences to the circuit we saw in Figure 1. Specifically, for this project, each switch has a voltage drop and resistance. With this in mind, we can redraw the circuit to account for these non-idealities.

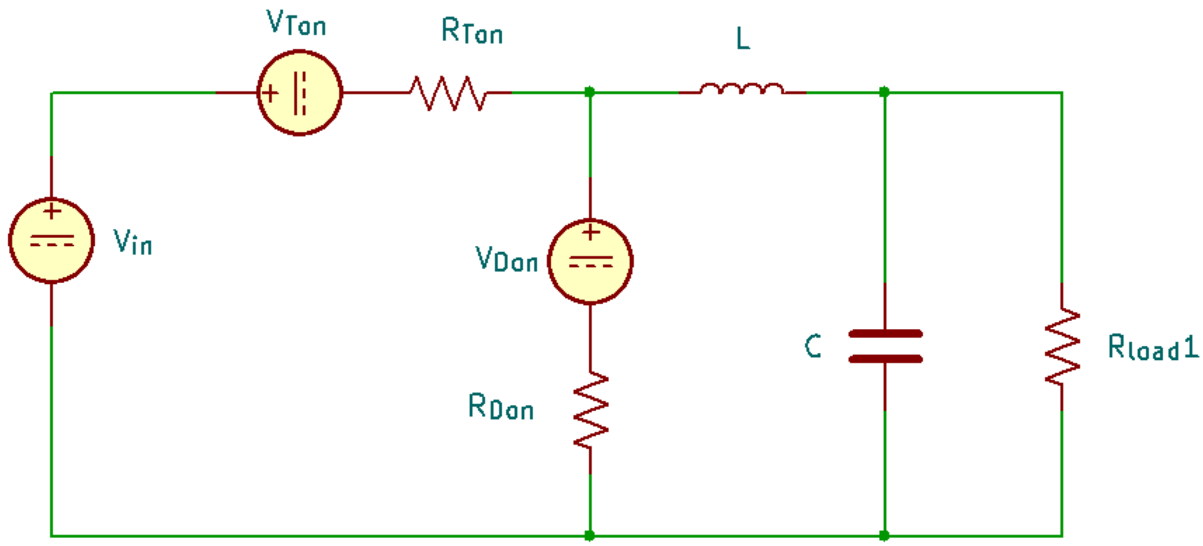

Figure 3: circuit schematic for a buck converter with non-ideal conditions.

Like the ideal conditions, only one switch can be on at a single point in time. Therefore, there are two smaller subcircuits that can be created using the circuit from Figure 3.

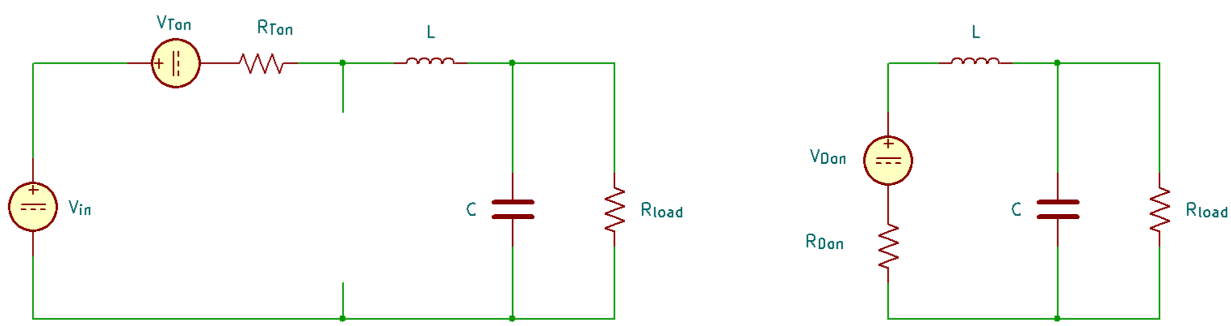

Figure 4: Non ideal Buck Converter schematics.

With these new circuit schematics, we can reapply Kirchoff's Laws to determine any changes to the values of the components or the duty cycle.

Duty Cycle

We know that the average inductor voltage is zero.

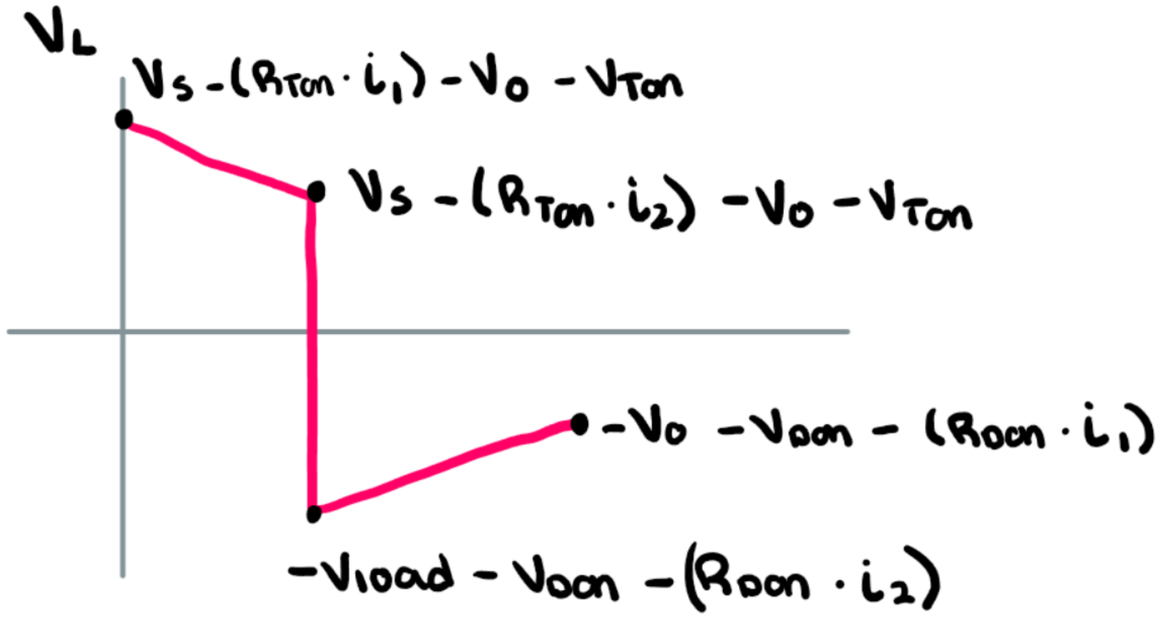

Figure 5: Plot for inductor voltage.

Using Figure 5, we can set up an expression for average inductor voltage.

$$< V_L > = 0 = \frac{1}{T_{sw}}\left(DT_{sw}\left(V_{in} - V_{T,ON} - i_{L2}R_{T,ON} - V_{load}\right) + \frac{DT_{sw}}{2}\left(-i_{L1}R_{T,ON} + -i_{C1}R_{T,ON}\right) + (1-D)T_{sw}\left(-V_{D,ON} - i_{L1}R_{D,ON} - V_{load}\right) + \frac{(1-D)T_{sw}}{2}\left(-i_{L2}R_{D,ON} + i_{L1}R_{D,ON}\right).\right.$$

Equation 18: Expression for average inductor voltage.

Using algebra, we can simplify the equation to eliminate some variables and solve for the duty cycle, D.

$$D = \frac{V_{D,ON} + V_{load} + \frac{R_{D,ON} \cdot V_{load}}{R_{load}}}{V_{in} - V_{T,ON} + V_{D,ON} + \frac{R_{D,ON} \cdot V_{load}}{R_{load}} - \frac{R_{T,ON} \cdot V_{load}}{R_{load}}} = \frac{1 + 400 + \frac{0.01 \cdot 400}{0.64}}{800} = 0.509625$$

Equation 19: Expression for non-ideal duty cycle.

Using this expression for non-ideal duty cycle, we can use it to solve for the voltages and the currents for each of the components.

Efficiency

Efficiency can be determined by dividing the output power by the input power.

$$\eta = \frac{P_{out}}{P_{in}}$$

Equation 20: Expression for finding the efficiency of a circuit.

Maximum and Minimum Inductor Current

To find $i_{L,max}$ and $i_{L,min}$, we must first solve for the load voltage. By rearranging Equation 19, we can determine an expression for load voltage.

$$V_{load} = \frac{D \cdot V_{in}}{\frac{D \cdot R_{T,ON}}{R_{load}} + \frac{(1-D) \cdot R_{D,ON}}{R_{load}}}$$

Equation 21: Expression to find load voltage under non-ideal conditions.

Like the ideal conditions, the graph for inductor current ranges resembles a triangle wave, with the lowest value being $i_{L,min}$ and the highest value being $i_{L,max}$. By applying Kirchoff's voltage and current laws, to Figure 4, we can determine values for the maximum and minimum inductor current, specifically by looking at the slope for each side of the triangle. This process leads to a system of equations which can be simplified using both calculus and basic algebra, but by using some linear algebra and an online matrix calculator, we can easily determine the forms for $i_{L,min}$ and $i_{L,max}$.

$$\begin{bmatrix} -e^{\frac{-R_{T,ON} \cdot D \cdot T_{sw}}{L}} & 1 \\ 1 & -e^{\frac{-R_{T,ON} \cdot (1-D) \cdot T_{sw}}{L}} \end{bmatrix} \begin{bmatrix} i_{L,min} \\ i_{L,max} \end{bmatrix} = [\begin{matrix} \frac{V_{in} - V_{load} - V_{T,ON}}{R_{T,ON}} \cdot (1 - e^{\frac{-R_{T,ON} \cdot D \cdot T_{sw}}{L}}) \\ [\frac{V_{in} - V_{load} - V_{T,ON}}{R_{T,ON}} \cdot (1 - e^{\frac{-R_{T,ON} \cdot (1-D) \cdot T_{sw}}{L}})] \end{matrix}$$

Equation 22: Expression to solve the minimum and maximum inductor current.

Results

Ideal – Heavy Load

Once the code for “buck.m” was complete, I was able to run “buckproc.m” to test my Forward Euler integration algorithm. This was done by plotting each of the voltages and currents for each component in both the transient state and steady state.

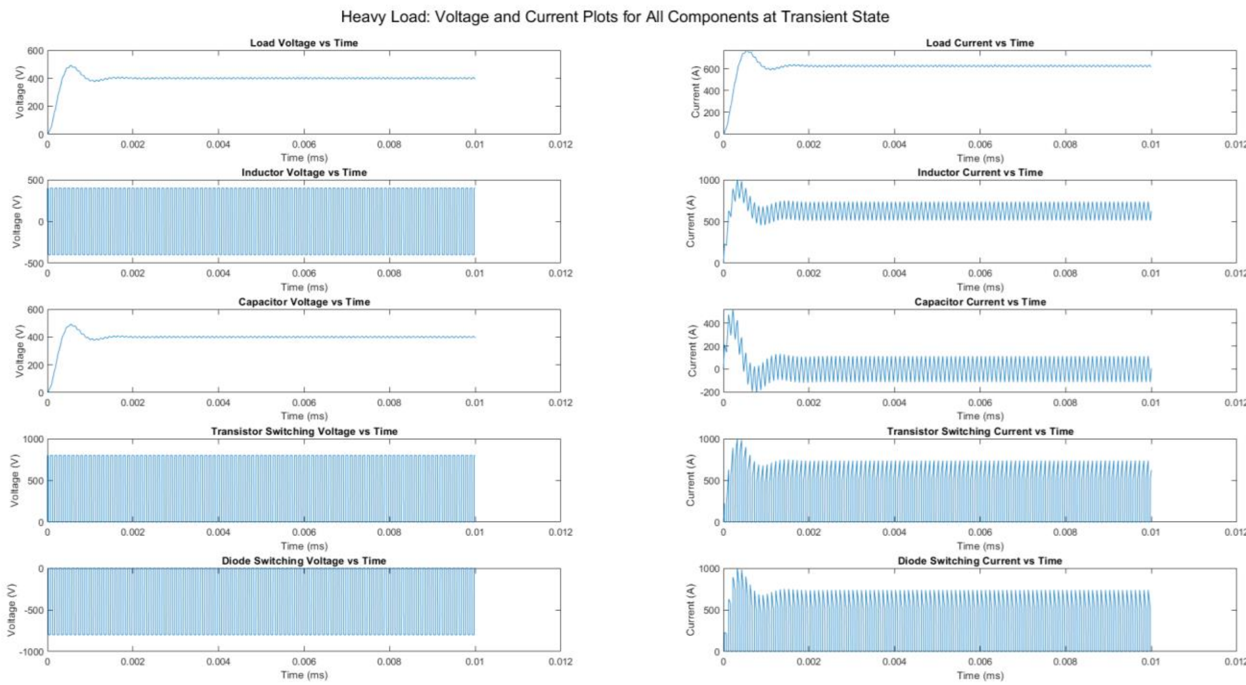

Figure 6: Voltage and Current graphs for each component at transient state.

From the graph for load voltage, load current, inductor current, and capacitor voltage, we can see that the values fluctuate for a small period before reaching steady state.

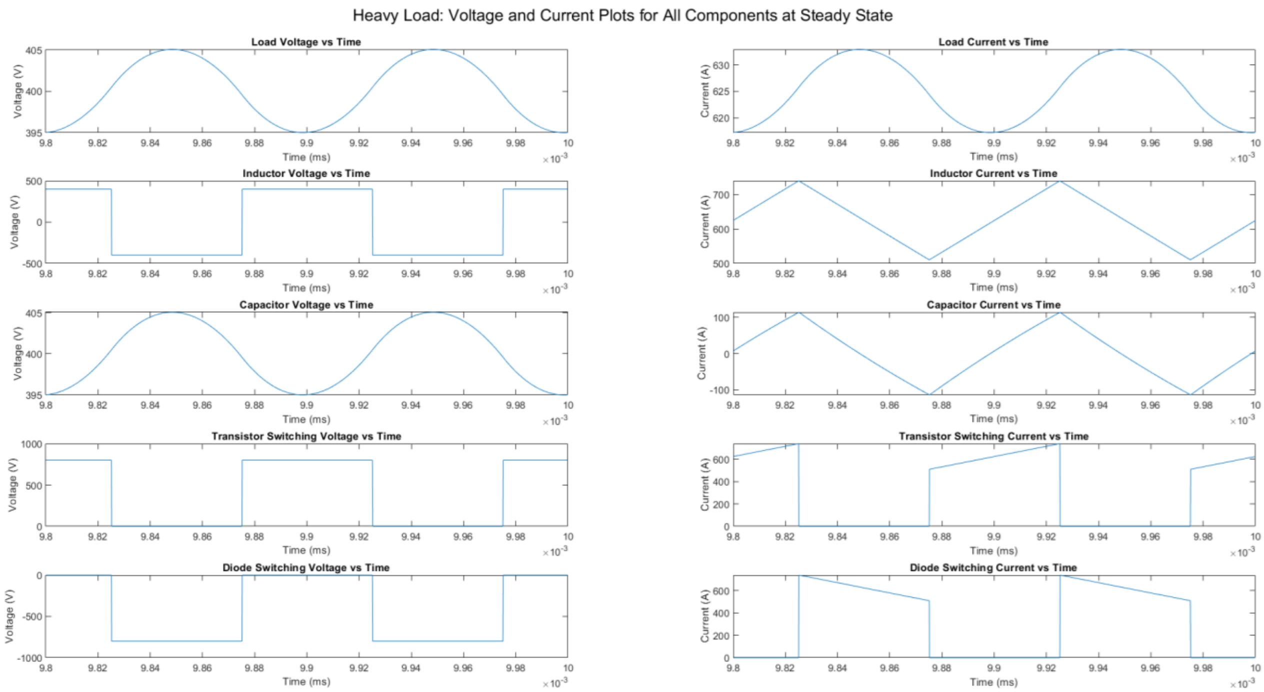

Figure 7: Voltage and Current graphs for each component at steady state.

In "buckproc.m", a separate section of code was written to plot the last two periods of each waveform. This was done to make each waveform's graph easier to read. With the number of periods being reduced, we can see where each waveform averages out to.

Instead of reading each of the graphs and estimating what the average of each waveform is, "buckproc.m" calls "aver.m" to display the average of each waveform onto the console.

Component	Voltage (V)	Current (A)
Load Resistor	400.0307	625.0479
Inductor	7.59E-15	625.0319
Capacitor	400.0307	-0.016048
Transistor	400	312.6551
Diode	-400	312.606

Table 1: Average Voltage and Current for each circuit component under heavy load.

If we use Equation 20 to solve the efficiency of the circuit, we find that the efficiency of circuit is 99.9657%.

Ideal – Light Load

When the simulation was complete for the heavy load, "buckproc.m" was able to focus on the light load. Like before, the script plotted each component's voltages and currents in the transient state.

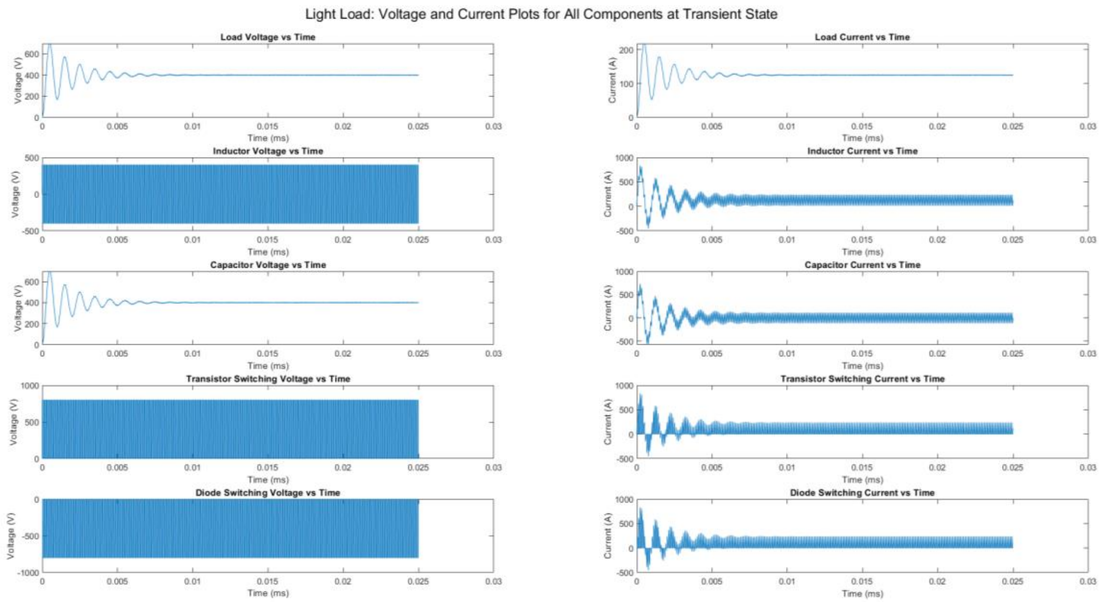

Figure 8: Voltage and Current graphs for each component at transient state.

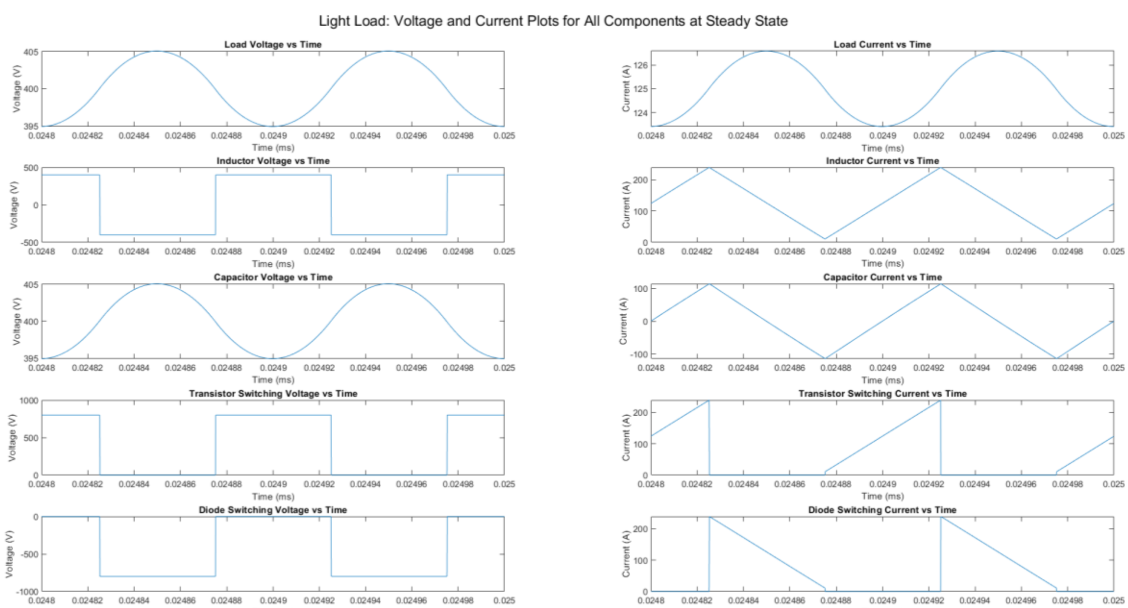

Figure 9: Voltage and Current graphs for each component at steady state.

Like the heavy load, the script plots the last two periods of each waveform to increase the readability.

Component	Voltage (V)	Current (A)
Load Resistor	400.0307	625.0479
Inductor	7.59E-15	625.0319
Capacitor	400.0307	-0.016048
Transistor	400	312.6551
Diode	-400	312.606

Table 2: Average Voltage and Current for each circuit component under heavy load.

The efficiency of the circuit under light load comes out to 99.8114%.

Non-Ideal – Heavy Load

The final section of "buckproc.m" completed the calculations and plotting for the non-ideal situation. Unlike the previous two sections, this section of the code only focused on calculating values for the load voltage, inductor current, in addition to the current, voltage, and power of each switch.

The code in "buck.m" had to be slightly altered in order to incorporate the non-idealities of the circuit, such as the voltage drop and inherent resistance for each switch.

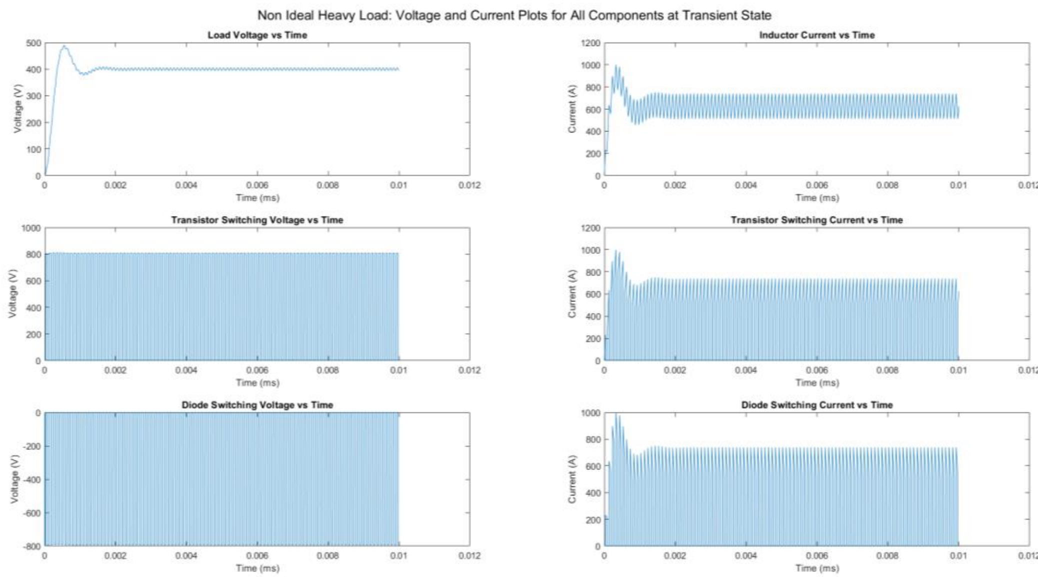

Figure 10: Voltage and Current graphs for each component at transient state.

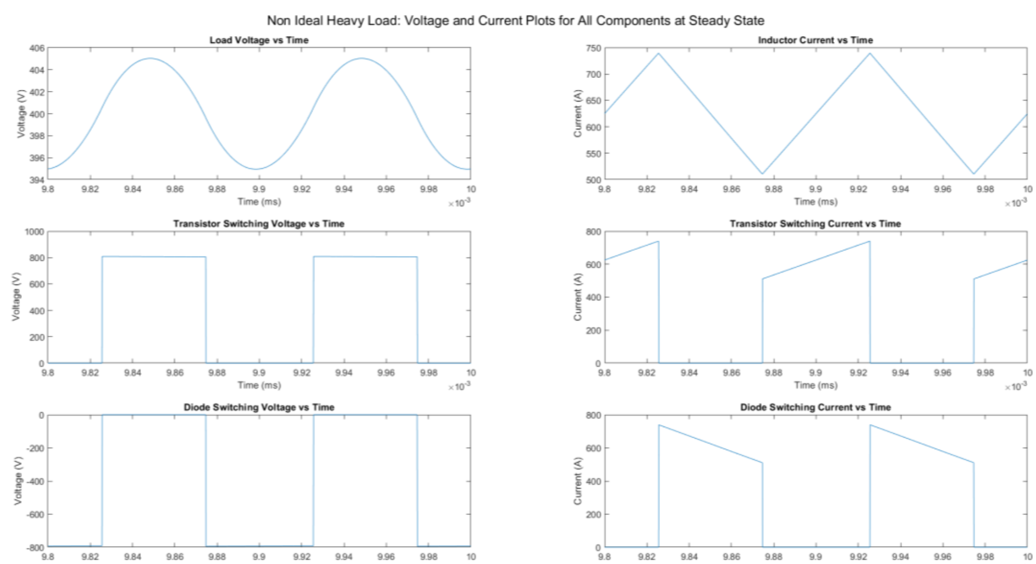

Figure 11: Voltage and Current graphs for each component at steady state.

Each of the components that were plotted share a similar shape to both ideal cases. However, some of the values are slightly different from what were seen in the ideal cases.

Component	Voltage (V)	Current (A)	Power (W)
Load Resistor	399.9508	x	x
Inductor	x	624.9231	x
Transistor	396.3574	318.2797	2330.834
Diode	-403.5082	306.6434	2243.2617

Table 3: Average Voltage, Current, and Power for each circuit component under heavy load.

As we can see from Table 3, each component has slightly different values to the ones seen in both Table 1 and Table 2. These different values are most likely due to the non-idealities from both switches.

Using Equation 20, we find that the efficiency of the non-ideal circuit is 98.1559%.

From Table 3, we find that the power loss from the transistor is 2330.834 W and the power loss from the diode is 2243.2617 W.

Project #1\aver.m

```
% ID Number: 229,506
% ECE 31033 - Project #1
% aver.m

% The fourth file (aver.m) contains a function you create to compute the average of
% a waveform. Specifically, the function is of the form
%       function av = aver(x,T,dt)
% where x is the waveform to be averaged, T is its period, and dt is the period of time
% between samples. This function must use the last period of the input waveform to
% calculate the average.

function av = aver(x, T, dt)
    location = length(x);
    av = 0;
    time = 0;

    while (time <= T)
        av = av + dt * (x(location));
        time = time + dt;
        location = location - 1;
    end

    av = av / T;
end

```

Project #1\sw.m

```
% ID Number: 229,506
% ECE 31033 - Project #1
% sw.m

% The first file (sw.m) contains a function (sw) that accepts the duty cycle D, and a
% single instant of time as an input, and outputs the state (on/off) of the transistor
% at that time instant as an output. A Fourier series-based triangle wave that you
% create within this function should be compared with the duty cycle D to
% determine the state of the transistor. The output of the function is a 1 if the
% transistor is to be turned on. It is a value of 0 if it is turned off.

function state = sw(D, t)
    T_sw = 1 / 10000;

    w = 2 * pi / T_sw;

    a_k = 0;
    triangle_wave = 0.5;

    N = 200; % Number of Fourier terms.

    k = 1;
    while k <= N
        z = k * w * T_sw; % Temporary variable; to simplify code for the coefficient.

        a_k = (2 * (4 * cos(0.5 * z) - 2 * cos(z) - 2)) / (z^2);
        triangle_wave = triangle_wave + a_k * cos(k * w * t);
        k = k + 1;
    end

    if D >= triangle_wave
        state = 1;
    else
        state = 0;
    end
end
```

Project #1\buck.m

```matlab
% ID Number: 229,506
% ECE 31033 - Project #1
% buck.m

% The file (buck.m) contains the Forward Euler integration algorithm within a while
% loop (FOR LOOPS ARE NOT ALLOWED). buck is not a function. The file buck.m only
% contains a single while loop (i.e. while (t(k)<tend)) to solve for all circuit voltages
% and currents of your buck converter. Within the while loop, you will call the
% function sw at each time instant to determine the value of your transistor gate
% (on or off).  Voltages of currents and voltages of circuit components must be
% determined within the while loop.
while t_vec(k) < tend
    if (ideal_boolean) % If the circuit is ideal.
        switch_state(k) = sw(D, t_vec(k)); % calling sw.m

        % Inductor Current and Load Voltage Calculation
        i_L_vec(k+1) = i_L_vec(k) + dt * ((switch_state(k)) * V_in - V_load_vec(k)) / L; %i_L
        V_load_vec(k+1) = V_load_vec(k) + dt * ((i_L_vec(k) - (V_load_vec(k) / R_load)) / C); %V_load

        % Switch 1 and 2: Voltage and Current Calculations
        if(switch_state(k))
            V_switch1(k+1) = V_in;
            i_switch1(k+1) = i_L_vec(k+1) * switch_state(k);

            V_switch2(k+1) = 0;
            i_switch2(k+1) = 0;

            V_L_vec(k+1) = V_in - V_load_avg;
        else
            V_switch1(k+1) = 0;
            i_switch1(k+1) = 0;

            V_switch2(k+1) = -1 * V_in;
            i_switch2(k+1) = i_L_vec(k) * (1 - switch_state(k));

            V_L_vec(k+1) = -1 * V_load_avg;
        end

        % Capacitor: Voltage and Current Calculations
        i_C_vec(k+1) = i_L_vec(k) - (V_load_vec(k) / R_load);
        V_C_vec(k+1) = V_load_vec(k+1);

        % Load: Current Calculation
        i_load_vec(k+1) = V_load_vec(k+1) / R_load;

    else % If the circuit is non ideal.
        switch_state(k) = sw(D_non_ideal, t_vec(k)); % calling sw.m

        if(switch_state(k))
            %i_L_vec(k+1) = i_L_vec(k) + dt * ((V_in - V_T_on - V_load_vec(k) - (R_T_on * i_L_vec(k)))) / L;    %i_L
```

```
            i_L_vec(k+1) = i_L_vec(k) + dt * ((V_in - V_T_on - V_load_vec(k) - (R_T_on * i_L_vec(k))) / L);    %i_L

            % Switch 1: Voltage and Current Calculations
            V_switch1(k+1) = 0;
            i_switch1(k+1) = i_L_vec(k+1);
            P_switch1(k+1) = (R_T_on * i_L_vec(k+1) + V_T_on) * i_L_vec(k+1);

            % Switch 2: Voltage and Current Calculations
            V_switch2(k+1) = V_D_on + (R_D_on * i_L_vec(k+1)) - V_in;
            i_switch2(k+1) = 0;
        else
            i_L_vec(k+1) = i_L_vec(k) + dt * ((-1 * V_load_vec(k) - V_D_on - (R_D_on * i_L_vec(k)))) / L;

            % Switch 1: Voltage and Current Calculations
            V_switch1(k+1) = V_in + (R_D_on * i_L_vec(k+1)) + V_D_on;
            i_switch1(k+1) = 0;

            % Switch 2: Voltage and Current Calculations
            V_switch2(k+1) = 0;
            i_switch2(k+1) = i_L_vec(k+1);
            P_switch2(k+1) = (R_D_on * i_L_vec(k+1) + V_D_on) * i_L_vec(k+1);
        end

        V_load_vec(k+1) = V_load_vec(k) + dt * ((i_L_vec(k) - (V_load_vec(k) / R_load)) / C); %V_load
    end

    % Increment the time and index
    t_vec(k + 1) = t_vec(k) + dt;
    k = k + 1;
end
```

Project #1\buckproc.m

```
% ID Number: 229,506
% ECE 31033 - Project #1
% buckproc.m

% The file buckproc.m first contains the circuit parameter values (i.e. L, C, fsw, time
% step, initial conditions etc.). Only the initial value of your circuit voltages and
% currents should be pre-established (i.e. Vload(1)=0). It then invokes buck. Finally,
% it performs your plotting and any post-processing calculations that are done
% using the simulated data (such as computing average values, efficiency, etc.).
%% Ideal - Given Values
V_in = 800;
V_load_avg = 400;
V_load_ripple = 10;
P_load_light = 50000;
P_load_heavy = 250000;
frequency = 10000;

ideal_boolean = 1; % = 0 if non ideal, = 1 if ideal; here, it is ON.

%% Ideal - Calculated Values
T_sw = 1 / frequency;
D = V_load_avg / V_in;  % Duty Cycle

R_load_light = (V_load_avg^2) / P_load_light;
R_load_heavy = (V_load_avg^2) / P_load_heavy;

L_crit = (R_load_light * (1 - D)) / (2 * frequency);
L = L_crit * 1.1;

C = (V_load_avg / V_load_ripple) * (T_sw^2 * (1 - D)) / (8 * L);

I_load_light = V_load_avg / R_load_light;
I_load_heavy = V_load_avg / R_load_heavy;

i_L1_light = (V_load_avg / R_load_light) - (1 - D) * T_sw * V_load_avg / (2 * L);
i_L2_light = (V_load_avg / R_load_light) + (1 - D) * T_sw * V_load_avg / (2 * L);

i_L1_heavy= (V_load_avg / R_load_heavy) - (1 - D) * T_sw * V_load_avg / (2 * L);
i_L2_heavy= (V_load_avg / R_load_heavy) + (1 - D) * T_sw * V_load_avg / (2 * L);

%% Buck Intialization - Heavy Load
% Initializing Values
k = 1;
t = 0;
dt = 1e-7;

tend = 100 * T_sw;

% Zero Vectors (used in buck)
t_vec = [0];
switch_state = [0];

V_L_vec = [0];
```

```
i_L_vec = [0];

V_C_vec = [0];
i_C_vec = [0];

V_load_vec = [0];
i_load_vec = [0];

V_switch1 = [0];
i_switch1 = [0];

V_switch2 = [0];
i_switch2 = [0];

%% Running Buck - Using R_load_heavy
R_load = R_load_heavy;
disp('Running buck for heavy load.');
buck

%% Post-processing Calculations (computing avg values, efficiency, etc)
disp("---------------------")
disp("Heavy Averages:")

V_load_avg_func_H = aver(V_load_vec, T_sw, dt);
disp("  V_load Average: " + V_load_avg_func_H);

i_load_avg_func_H = aver(i_load_vec, T_sw, dt);
disp("  i_load Average: " + i_load_avg_func_H);

V_L_func_H = aver(V_L_vec, T_sw, dt);
disp("  V_L Average: " + V_L_func_H);

i_L_func_H = aver(i_L_vec, T_sw, dt);
disp("  i_L Average: " + i_L_func_H);

V_C_func_H = aver(V_C_vec, T_sw, dt);
disp("  V_C Average: " + V_C_func_H);

i_C_func_H = aver(i_C_vec, T_sw, dt);
disp("  i_C Average: " + i_C_func_H);

V_sw1_func_H = aver(V_switch1, T_sw, dt);
disp("  V_sw1 Average: " + V_sw1_func_H);

i_sw1_func_H = aver(i_switch1, T_sw, dt);
disp("  i_sw1 Average: " + i_sw1_func_H);

V_sw2_func_H = aver(V_switch2, T_sw, dt);
disp("  V_sw2 Average: " + V_sw2_func_H);

i_sw2_func_H = aver(i_switch2, T_sw, dt);
disp("  i_sw2 Average: " + i_sw2_func_H);

P_out_H = (V_load_avg_func_H^2) / R_load;
P_in_H = V_in * i_sw1_func_H;
eff_H = P_out_H / P_in_H;
```

```
disp("Efficiency for Light Load: " + (eff_H * 100) + "%.");
disp("---------------------")

%% Plotting - Heavy Load - Transient
% Plots for the transient to steady state
figure;
sgtitle("Heavy Load: Voltage and Current Plots for All Components at Transient State");
% Plots for the Load
subplot(5,2,1);
plot(t_vec, V_load_vec);
title('Load Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,2);
plot(t_vec, i_load_vec);
title('Load Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Inductor
subplot(5,2,3);
plot(t_vec, V_L_vec);
title('Inductor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,4);
plot(t_vec, i_L_vec);
title('Inductor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Capacitor
subplot(5,2,5);
plot(t_vec, V_C_vec);
title('Capacitor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,6);
plot(t_vec, i_C_vec);
title('Capacitor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 1
subplot(5,2,7);
plot(t_vec, V_switch1);
title('Transistor Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,8);
plot(t_vec, i_switch1);
```

```
title('Transistor Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 2
subplot(5,2,9);
plot(t_vec, V_switch2);
title('Diode Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,10);
plot(t_vec, i_switch2);
title('Diode Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

%% Plotting - Heavy Load - Steady State
periods_to_plot = 2;

points_per_period = round(T_sw / dt);  % Points per period
total_periods = floor(tend / T_sw);    % Total number of periods in the simulation

start_index = max(1, (total_periods - periods_to_plot) * points_per_period + 1);
end_index = min(length(t_vec), total_periods * points_per_period);

range_to_plot = start_index:end_index;

%% Plot
figure;
sgtitle("Heavy Load: Voltage and Current Plots for All Components at Steady State");
% Plots for the Load
subplot(5,2,1);
plot(t_vec(range_to_plot), V_load_vec(range_to_plot));
title('Load Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,2);
plot(t_vec(range_to_plot), i_load_vec(range_to_plot));
title('Load Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Inductor
subplot(5,2,3);
plot(t_vec(range_to_plot), V_L_vec(range_to_plot));
title('Inductor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,4);
plot(t_vec(range_to_plot), i_L_vec(range_to_plot));
title('Inductor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');
```

```

% Plots for the Capacitor
subplot(5,2,5);
plot(t_vec(range_to_plot), V_C_vec(range_to_plot));
title('Capacitor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,6);
plot(t_vec(range_to_plot), i_C_vec(range_to_plot));
title('Capacitor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 1
subplot(5,2,7);
plot(t_vec(range_to_plot), V_switch1(range_to_plot));
title('Transistor Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,8);
plot(t_vec(range_to_plot), i_switch1(range_to_plot));
title('Transistor Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 2
subplot(5,2,9);
plot(t_vec(range_to_plot), V_switch2(range_to_plot));
title('Diode Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,10);
plot(t_vec(range_to_plot), i_switch2(range_to_plot));
title('Diode Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Buck Intialization - Light Load
% Initializing Values
k = 1;
t = 0;
dt = 1e-7;

tend = 250 * T_sw;

% Zero Vectors (used in buck)
t_vec = [0];
switch_state = [0];

V_L_vec = [0];
i_L_vec = [0];

```

```
V_C_vec = [0];
i_C_vec = [0];

V_load_vec = [0];
i_load_vec = [0];

V_switch1 = [0];
i_switch1 = [0];

V_switch2 = [0];
i_switch2 = [0];

%% Running Buck - Using R_load_light
R_load = R_load_light;
disp('Running buck for light load.');
buck

%% Post-processing Calculations (computing avg values, efficiency, etc)
disp("---------------------")
disp("Light Averages:")

V_load_avg_func_L = aver(V_load_vec, T_sw, dt);
disp("  V_load Average: " + V_load_avg_func_L);

i_load_avg_func_L = aver(i_load_vec, T_sw, dt);
disp("  i_load Average: " + i_load_avg_func_L);

V_L_func_L = aver(V_L_vec, T_sw, dt);
disp("  V_L Average: " + V_L_func_L);

i_L_func_L = aver(i_L_vec, T_sw, dt);
disp("  i_L Average: " + i_L_func_L);

V_C_func_L = aver(V_C_vec, T_sw, dt);
disp("  V_C Average: " + V_C_func_L);

i_C_func_L = aver(i_C_vec, T_sw, dt);
disp("  i_C Average: " + i_C_func_L);

V_sw1_func_L = aver(V_switch1, T_sw, dt);
disp("  V_sw1 Average: " + V_sw1_func_L);

i_sw1_func_L = aver(i_switch1, T_sw, dt);
disp("  i_sw1 Average: " + i_sw1_func_L);

V_sw2_func_L = aver(V_switch2, T_sw, dt);
disp("  V_sw2 Average: " + V_sw2_func_L);

i_sw2_func_L = aver(i_switch2, T_sw, dt);
disp("  i_sw2 Average: " + i_sw2_func_L);

P_out_L = (V_load_avg_func_L^2) / R_load;
P_in_L = V_in * i_sw1_func_L;
eff_L = P_out_L / P_in_L;

disp("Efficiency for Light Load: " + (eff_L * 100) + "%.");
```

```
disp("---------------------")
%% Plotting - Light Load - Transient
% Plots for the transient to steady state
figure;
sgtitle("Light Load: Voltage and Current Plots for All Components at Transient State");
% Plots for the Load
subplot(5,2,1);
plot(t_vec, V_load_vec);
title('Load Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,2);
plot(t_vec, i_load_vec);
title('Load Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Inductor
subplot(5,2,3);
plot(t_vec, V_L_vec);
title('Inductor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,4);
plot(t_vec, i_L_vec);
title('Inductor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Capacitor
subplot(5,2,5);
plot(t_vec, V_C_vec);
title('Capacitor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,6);
plot(t_vec, i_C_vec);
title('Capacitor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 1
subplot(5,2,7);
plot(t_vec, V_switch1);
title('Transistor Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,8);
plot(t_vec, i_switch1);
title('Transistor Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');
```

```
% Plots for Switch 2
subplot(5,2,9);
plot(t_vec, V_switch2);
title('Diode Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,10);
plot(t_vec, i_switch2);
title('Diode Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

%% Plotting - Light Load - Steady State
periods_to_plot = 2;

points_per_period = round(T_sw / dt);  % Points per period
total_periods = floor(tend / T_sw);    % Total number of periods in the simulation

start_index = max(1, (total_periods - periods_to_plot) * points_per_period + 1);
end_index = min(length(t_vec), total_periods * points_per_period);

range_to_plot = start_index:end_index;

%% Plot
figure;
sgtitle("Light Load: Voltage and Current Plots for All Components at Steady State");
% Plots for the Load
subplot(5,2,1);
plot(t_vec(range_to_plot), V_load_vec(range_to_plot));
title('Load Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,2);
plot(t_vec(range_to_plot), i_load_vec(range_to_plot));
title('Load Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Inductor
subplot(5,2,3);
plot(t_vec(range_to_plot), V_L_vec(range_to_plot));
title('Inductor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,4);
plot(t_vec(range_to_plot), i_L_vec(range_to_plot));
title('Inductor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for the Capacitor
subplot(5,2,5);
```

```
plot(t_vec(range_to_plot), V_C_vec(range_to_plot));
title('Capacitor Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,6);
plot(t_vec(range_to_plot), i_C_vec(range_to_plot));
title('Capacitor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 1
subplot(5,2,7);
plot(t_vec(range_to_plot), V_switch1(range_to_plot));
title('Transistor Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,8);
plot(t_vec(range_to_plot), i_switch1(range_to_plot));
title('Transistor Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 2
subplot(5,2,9);
plot(t_vec(range_to_plot), V_switch2(range_to_plot));
title('Diode Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(5,2,10);
plot(t_vec(range_to_plot), i_switch2(range_to_plot));
title('Diode Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Non Ideal - Given Values
V_T_on = 1;
V_D_on = 1;
R_T_on = 0.01;
R_D_on = 0.01;

ideal_boolean = 0; % = 0 if non ideal, = 1 if ideal; here, it is OFF.

%% Non Ideal - Calculated Values
R_load = R_load_heavy;
D_non_ideal = (V_D_on + V_load_avg + (R_D_on * V_load_avg / R_load));
D_non_ideal = D_non_ideal / (V_in - V_T_on + V_D_on + (R_D_on * V_load_avg / R_load) - (R_T_on * V_load_avg / R_load));

i_L1_NI = (V_load_avg / R_load) - (((1 - D_non_ideal) * T_sw * V_load_avg) / (2 * L));
i_L2_NI = (V_load_avg / R_load) + (((1 - D_non_ideal) * T_sw * V_load_avg) / (2 * L));

%% Buck Intialization - Heavy Load
```

```
% Initializing Values
k = 1;
t = 0;
dt = 1e-7;

tend = 100 * T_sw;

% Zero Vectors (used in buck)
t_vec = [0];
switch_state = [0];

V_L_vec = [0];
i_L_vec = [0];

V_C_vec = [0];
i_C_vec = [0];

V_load_vec = [0];
i_load_vec = [0];

V_switch1 = [0];
i_switch1 = [0];

V_switch2 = [0];
i_switch2 = [0];

P_switch1 = [0]; % Power loss across the transistor; new for non-ideal calculations.
P_switch2 = [0]; % Power loss across the diode; new for non-ideal calculations.

%% Running Buck - Using R_load_heavy
R_load = R_load_heavy;
disp('Running buck for heavy load and non ideal conditions.');
buck

%% Post-processing Calculations (computing avg values, efficiency, etc)
disp("----------------------")
disp("Non Ideal Averages:")

V_load_avg_func_NI = aver(V_load_vec, T_sw, dt);
disp("  V_load Average: " + V_load_avg_func_NI);

i_L_func_NI = aver(i_L_vec, T_sw, dt);
disp("  i_L Average: " + i_L_func_NI);

V_sw1_func_NI = aver(V_switch1, T_sw, dt);
disp("  V_sw1 Average: " + V_sw1_func_NI);

i_sw1_func_NI = aver(i_switch1, T_sw, dt);
disp("  i_sw1 Average: " + i_sw1_func_NI);

V_sw2_func_NI = aver(V_switch2, T_sw, dt);
disp("  V_sw2 Average: " + V_sw2_func_NI);

i_sw2_func_NI = aver(i_switch2, T_sw, dt);
disp("  i_sw2 Average: " + i_sw2_func_NI);
```

```
P_sw1_func_NI = aver(P_switch1, T_sw, dt);
disp("  P_sw1 Average: " + V_sw2_func_NI);

P_sw2_func_NI = aver(P_switch2, T_sw, dt);
disp("  P_sw2 Average: " + i_sw2_func_NI);

P_out_NI = (V_load_avg_func_NI^2) / R_load;
P_in_NI = V_in * i_sw1_func_NI;
eff_NI = P_out_NI / P_in_NI;

disp("Efficiency for Non-Ideal: " + (eff_NI * 100) + "%.");
disp("Transistor Power Loss: " + P_sw1_func_NI);
disp("Diode Power Loss: " + P_sw2_func_NI);
disp("---------------------")
%% Plotting - Non Ideal Heavy Load - Transient
% Plots for the transient to steady state
figure;
sgtitle("Non Ideal Heavy Load: Voltage and Current Plots for All Components at Transient State");
% Plots for the Load
subplot(3,2,1);
plot(t_vec, V_load_vec);
title('Load Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(3,2,2);
plot(t_vec, i_L_vec);
title('Inductor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 1
subplot(3,2,3);
plot(t_vec, V_switch1);
title('Transistor Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(3,2,4);
plot(t_vec, i_switch1);
title('Transistor Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 2
subplot(3,2,5);
plot(t_vec, V_switch2);
title('Diode Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(3,2,6);
plot(t_vec, i_switch2);
title('Diode Switching Current vs Time');
xlabel('Time (ms)');
```

```
ylabel('Current (A)');

%% Plotting - Heavy Load - Steady State
periods_to_plot = 2;

points_per_period = round(T_sw / dt);  % Points per period
total_periods = floor(tend / T_sw);    % Total number of periods in the simulation

start_index = max(1, (total_periods - periods_to_plot) * points_per_period + 1);
end_index = min(length(t_vec), total_periods * points_per_period);

range_to_plot = start_index:end_index;

%% Plot
figure;
sgtitle("Non Ideal Heavy Load: Voltage and Current Plots for All Components at Steady State")
;
% Plots for the Load
subplot(3,2,1);
plot(t_vec(range_to_plot), V_load_vec(range_to_plot));
title('Load Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(3,2,2);
plot(t_vec(range_to_plot), i_L_vec(range_to_plot));
title('Inductor Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 1
subplot(3,2,3);
plot(t_vec(range_to_plot), V_switch1(range_to_plot));
title('Transistor Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(3,2,4);
plot(t_vec(range_to_plot), i_switch1(range_to_plot));
title('Transistor Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');

% Plots for Switch 2
subplot(3,2,5);
plot(t_vec(range_to_plot), V_switch2(range_to_plot));
title('Diode Switching Voltage vs Time');
xlabel('Time (ms)');
ylabel('Voltage (V)');

subplot(3,2,6);
plot(t_vec(range_to_plot), i_switch2(range_to_plot));
title('Diode Switching Current vs Time');
xlabel('Time (ms)');
ylabel('Current (A)');
```