

# ECE 31033 Project #1: DC—DC Converter

ID Number: 229,506

## Contents

Contents.....	1
Objective .....	2
Brief Overview of the MATLAB Files .....	3
“sw.m” .....	3
“buck.m” .....	3
“aver.m” .....	3
“buckproc.m” .....	3
Analytical Calculations .....	4
Buck Converter Overview .....	4
Component Values.....	5
Fourier Series Triangle Wave .....	7
Forward Euler Integration.....	8
Switch 1 – The Transistor .....	8
Switch 2 – The Diode.....	8
Inductor.....	8
Capacitor .....	9
Resistor .....	9
Non-Ideal – Heavy Load .....	10
Circuit Differences.....	10
Duty Cycle .....	10
Efficiency .....	11
Maximum and Minimum Inductor Current .....	12
Results.....	13
Ideal – Heavy Load .....	13
Ideal – Light Load .....	15
Non-Ideal – Heavy Load .....	17
“aver.m” .....	19
“sw.m” .....	20
“buck.m” .....	21
“buckproc.m” .....	23

## 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 is 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:

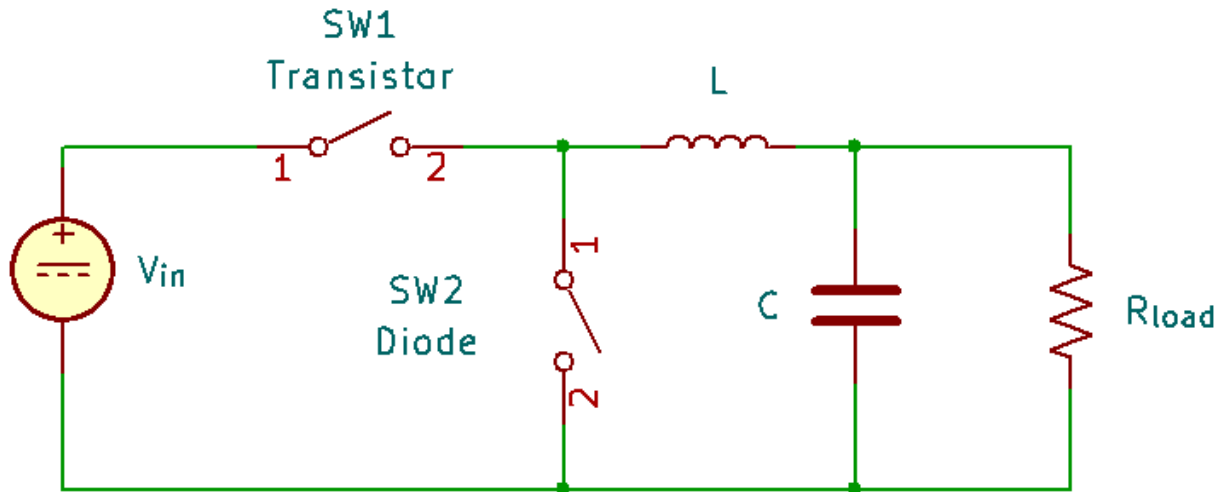


Figure 1: Standard circuit schematic for a buck converter.

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.

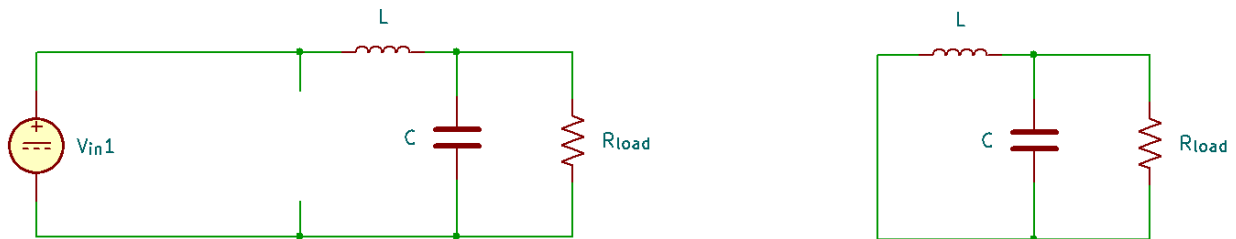


Figure 2: Buck converter schematic for when the transistor is on and for when the diode is on.

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.

$$\begin{aligned}
 C &\geq \frac{1}{8L} \left( T_{sw}^2 \cdot (1 - D) \right) \cdot \frac{V_{load}}{\Delta V_{load}} \\
 &\geq \frac{1}{8 \cdot (8.8e^{-5})} \left( (10^{-4})^2 \cdot (1 - 0.5) \right) \cdot \frac{400}{10} \\
 &\geq 2.84e^{-4} F
 \end{aligned}$$

*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.

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

*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.

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

*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.

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

*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.

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

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

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

*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.

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

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



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

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

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

*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.

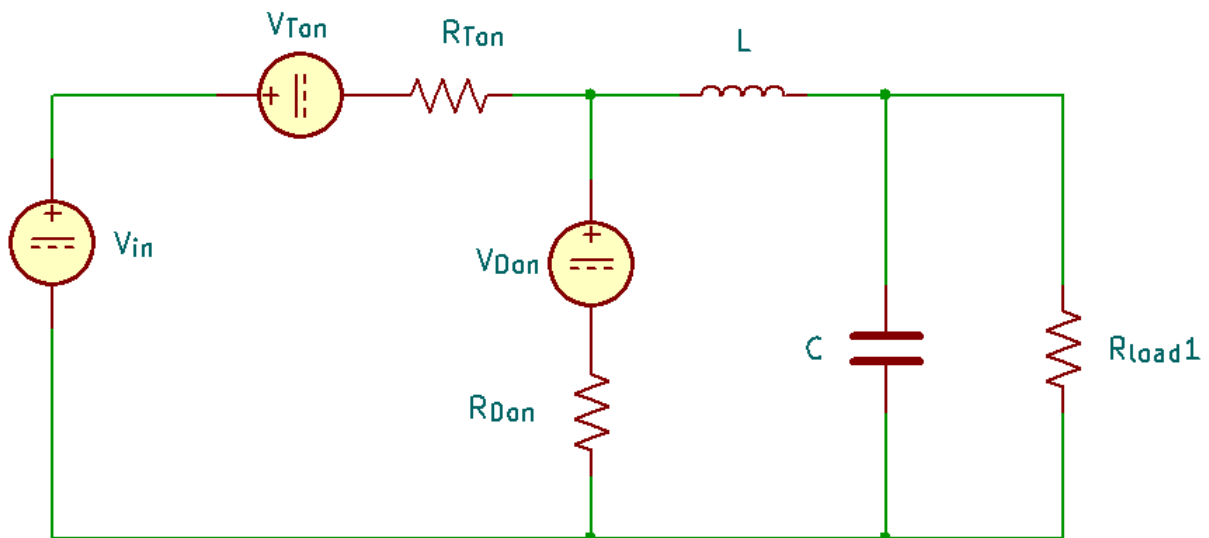


Figure 3: circuit schematic for a buck converter with non-ideal conditions.

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.

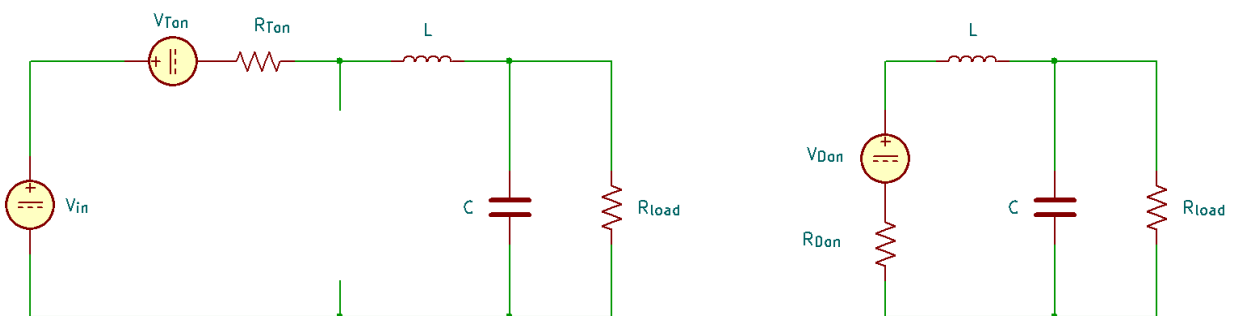


Figure 4: Non ideal Buck Converter schematics.

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.

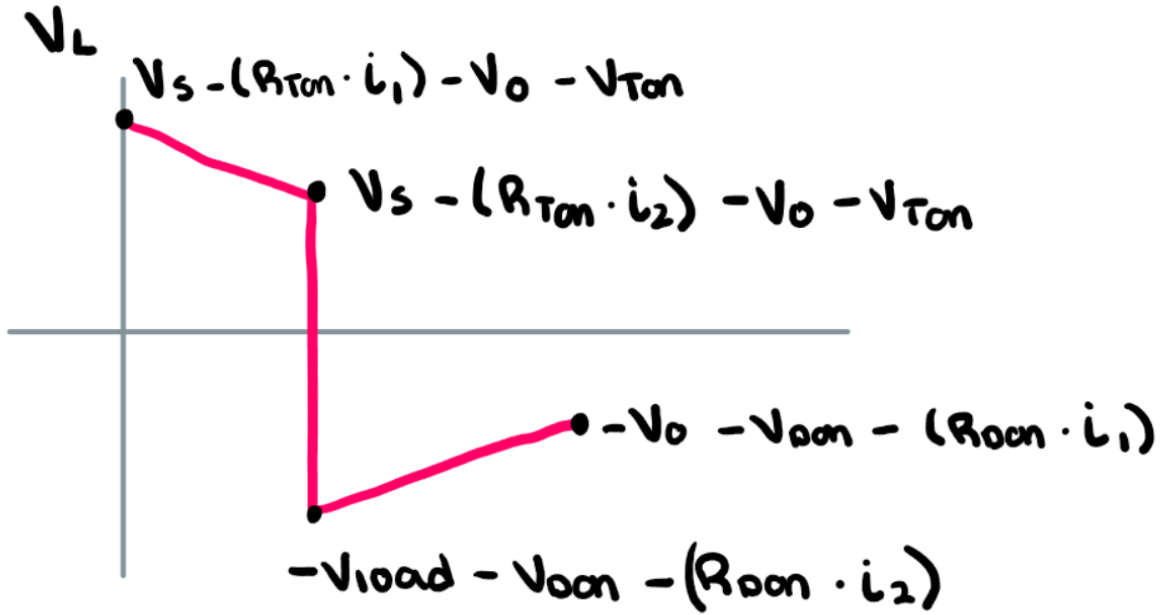


Figure 5: Plot for inductor voltage.

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

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

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  $i_{L,max}$ .

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

*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.

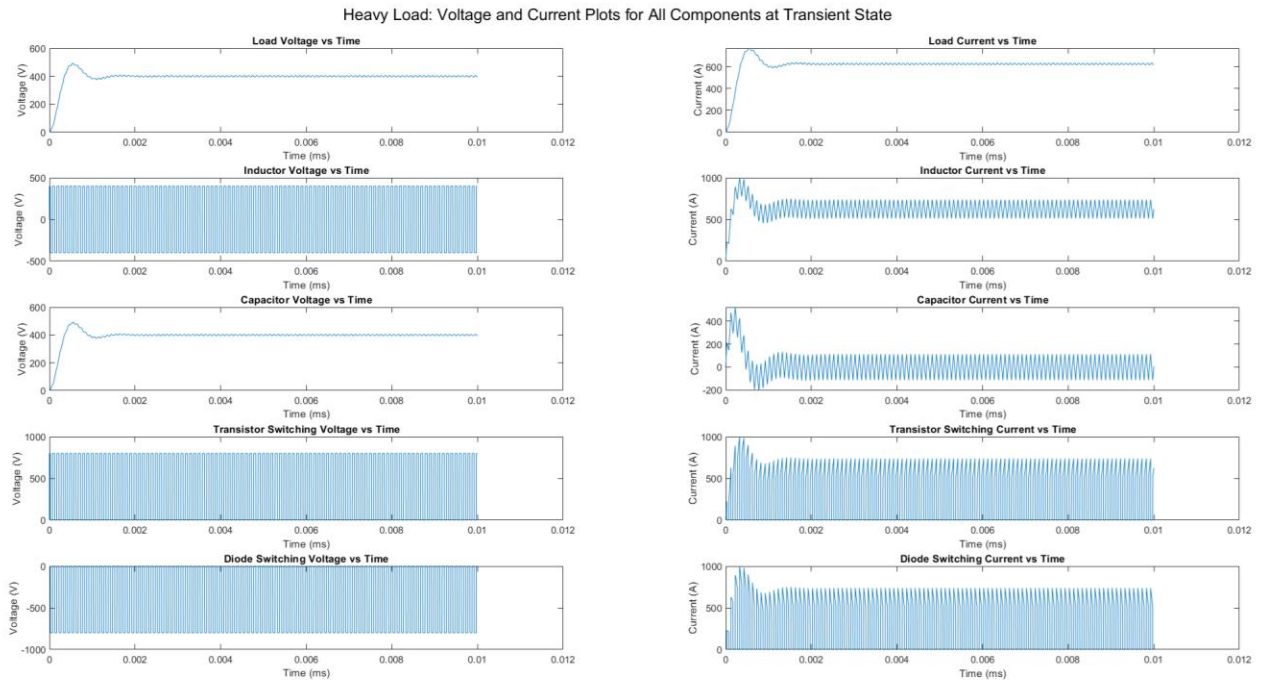


Figure 6: Voltage and Current graphs for each component at transient state.

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.

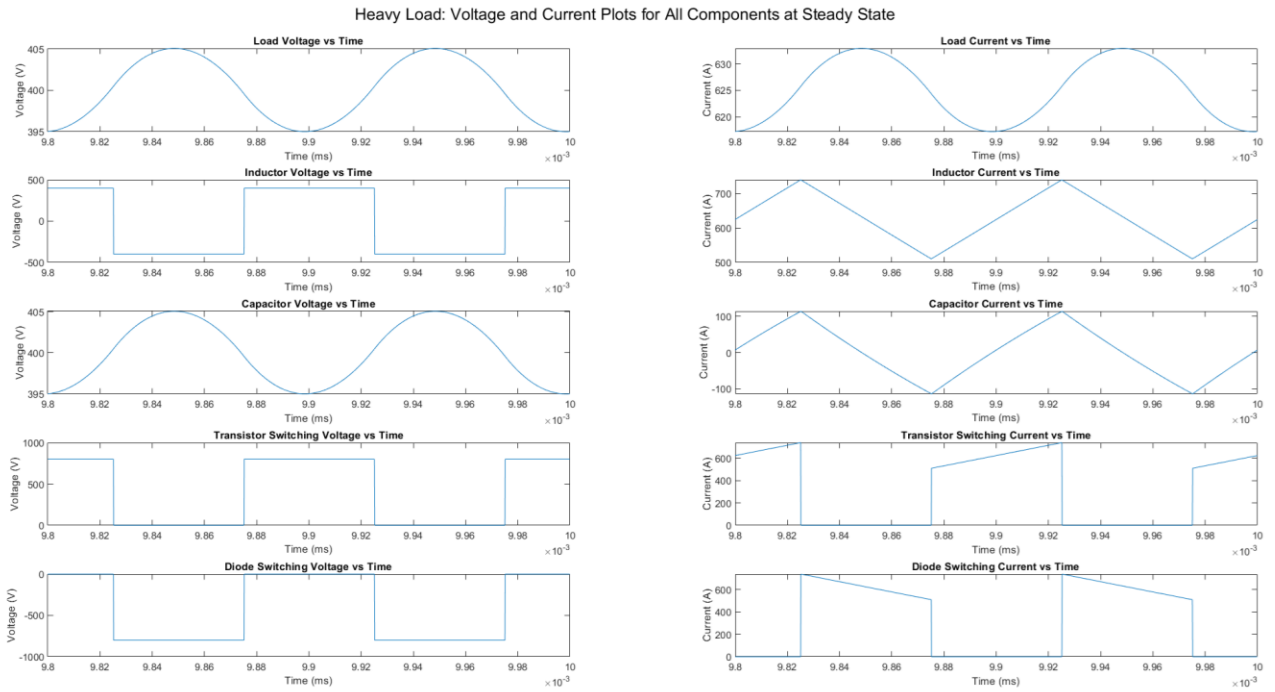


Figure 7: Voltage and Current graphs for each component at steady state.

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.

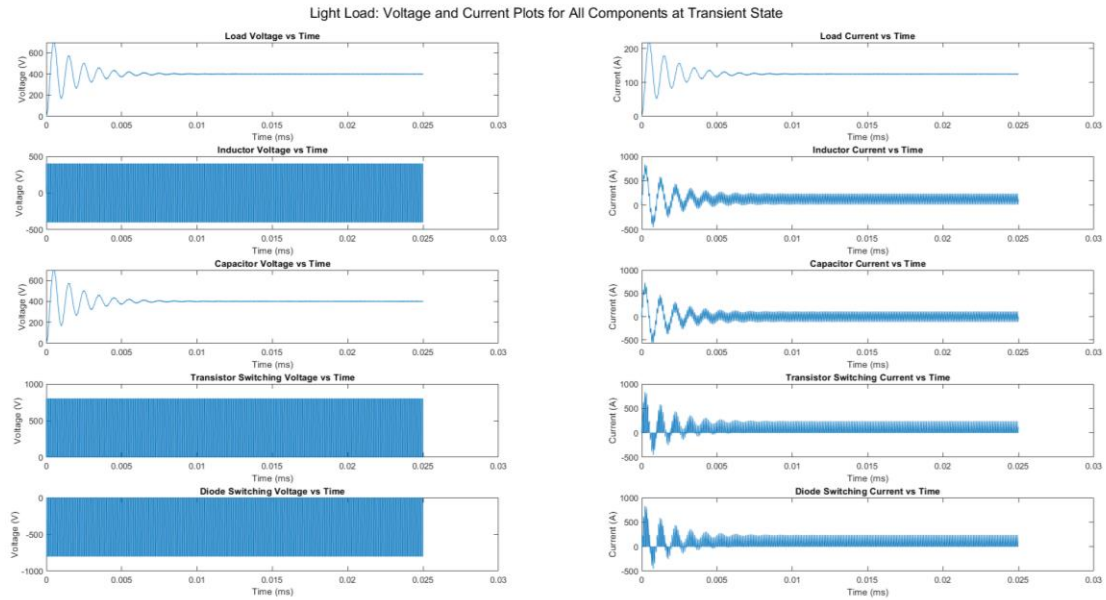


Figure 8: Voltage and Current graphs for each component at transient state.

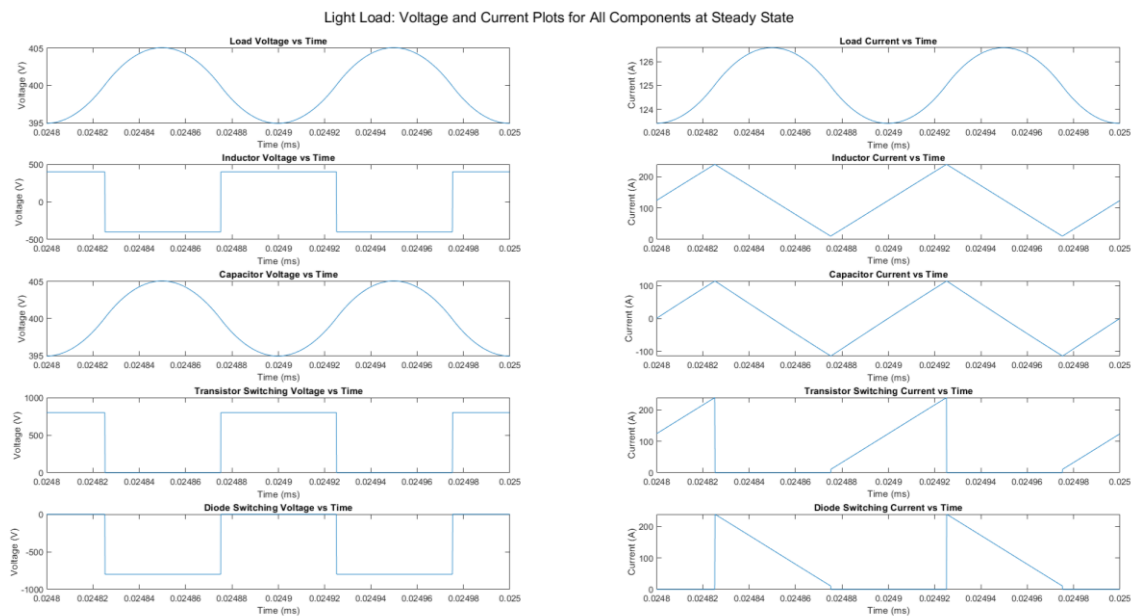


Figure 9: Voltage and Current graphs for each component at steady state.

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.

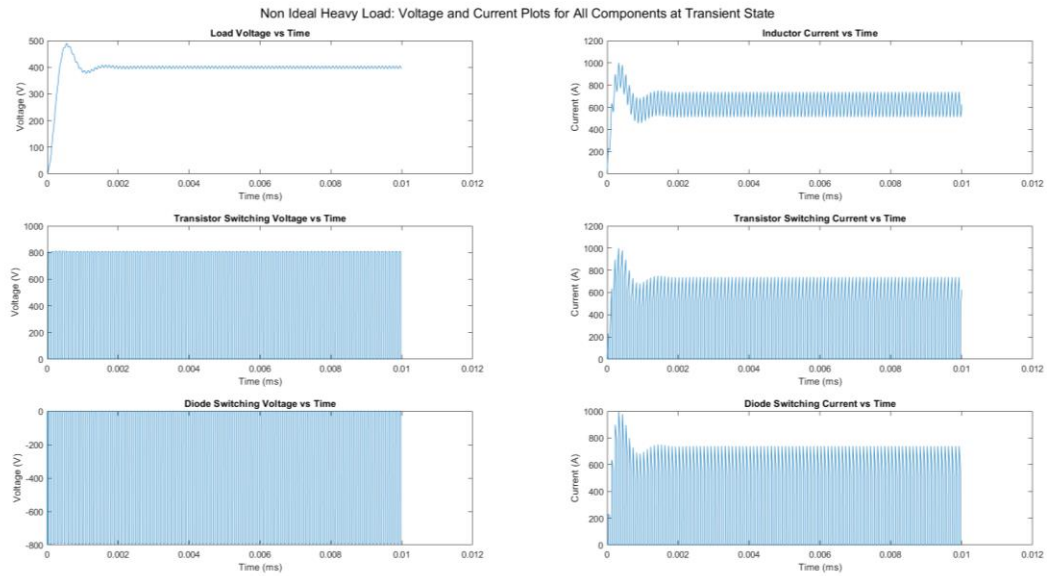


Figure 10: Voltage and Current graphs for each component at transient state.

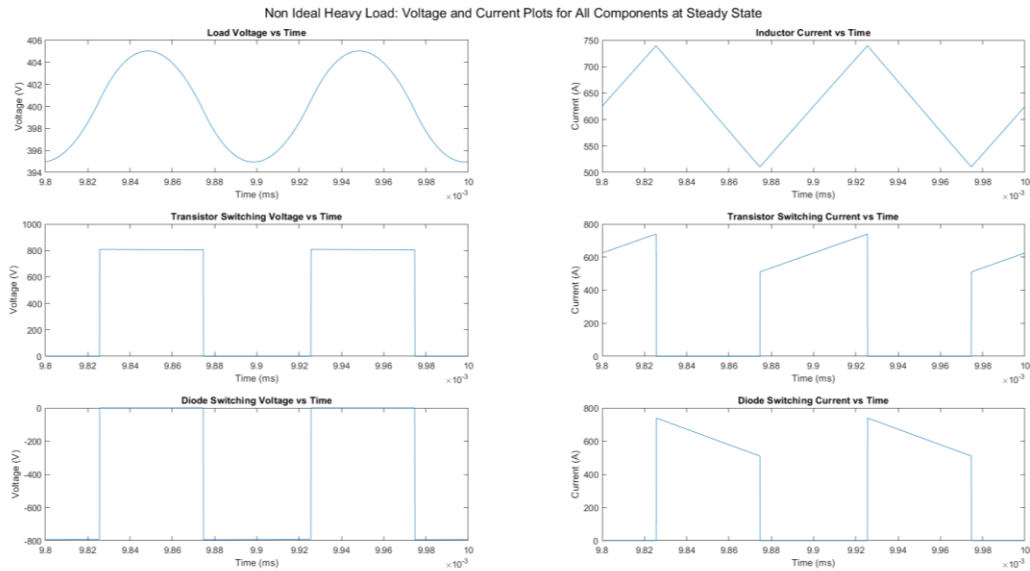


Figure 11: Voltage and Current graphs for each component at steady state.

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

```
1 % ID Number: 229,506
2 % ECE 31033 - Project #1
3 % aver.m
4
5 % The fourth file (aver.m) contains a function you create to compute the average of
6 % a waveform. Specifically, the function is of the form
7 %     function av = aver(x,T,dt)
8 % where x is the waveform to be averaged, T is its period, and dt is the period of time
9 % between samples. This function must use the last period of the input waveform to
10 % calculate the average.
11
12 function av = aver(x, T, dt)
13     location = length(x);
14     av = 0;
15     time = 0;
16
17     while (time <= T)
18         av = av + dt * (x(location));
19         time = time + dt;
20         location = location - 1;
21     end
22
23     av = av / T;
24 end
25
```

## Project #1\sw.m

```
1 % ID Number: 229,506
2 % ECE 31033 - Project #1
3 % sw.m
4
5 % The first file (sw.m) contains a function (sw) that accepts the duty cycle D, and a
6 % single instant of time as an input, and outputs the state (on/off) of the transistor
7 % at that time instant as an output. A Fourier series-based triangle wave that you
8 % create within this function should be compared with the duty cycle D to
9 % determine the state of the transistor. The output of the function is a 1 if the
10 % transistor is to be turned on. It is a value of 0 if it is turned off.
11
12 function state = sw(D, t)
13     T_sw = 1 / 10000;
14
15     w = 2 * pi / T_sw;
16
17     a_k = 0;
18     triangle_wave = 0.5;
19
20     N = 200; % Number of Fourier terms.
21
22     k = 1;
23     while k <= N
24         z = k * w * T_sw; % Temporary variable; to simplify code for the coefficient.
25
26         a_k = (2 * (4 * cos(0.5 * z) - 2 * cos(z) - 2)) / (z^2);
27         triangle_wave = triangle_wave + a_k * cos(k * w * t);
28         k = k + 1;
29     end
30
31     if D >= triangle_wave
32         state = 1;
33     else
34         state = 0;
35     end
36 end
```

## Project #1\buck.m

```
1 % ID Number: 229,506
2 % ECE 31033 - Project #1
3 % buck.m
4
5 % The file (buck.m) contains the Forward Euler integration algorithm within a while
6 % loop (FOR LOOPS ARE NOT ALLOWED). buck is not a function. The file buck.m only
7 % contains a single while loop (i.e. while (t(k)<tend)) to solve for all circuit voltages
8 % and currents of your buck converter. Within the while loop, you will call the
9 % function sw at each time instant to determine the value of your transistor gate
10 % (on or off). Voltages of currents and voltages of circuit components must be
11 % determined within the while loop.
12 while t_vec(k) < tend
13     if (ideal_boolean) % If the circuit is ideal.
14         switch_state(k) = sw(D, t_vec(k)); % calling sw.m
15
16         % Inductor Current and Load Voltage Calculation
17         i_L_vec(k+1) = i_L_vec(k) + dt * ((switch_state(k)) * V_in - V_load_vec(k)) / L;
18         V_load_vec(k+1) = V_load_vec(k) + dt * ((i_L_vec(k) - (V_load_vec(k) / R_load)) / C);
19
20         % Switch 1 and 2: Voltage and Current Calculations
21         if(switch_state(k))
22             V_switch1(k+1) = V_in;
23             i_switch1(k+1) = i_L_vec(k+1) * switch_state(k);
24
25             V_switch2(k+1) = 0;
26             i_switch2(k+1) = 0;
27
28             V_L_vec(k+1) = V_in - V_load_avg;
29         else
30             V_switch1(k+1) = 0;
31             i_switch1(k+1) = 0;
32
33             V_switch2(k+1) = -1 * V_in;
34             i_switch2(k+1) = i_L_vec(k) * (1 - switch_state(k));
35
36             V_L_vec(k+1) = -1 * V_load_avg;
37         end
38
39         % Capacitor: Voltage and Current Calculations
40         i_C_vec(k+1) = i_L_vec(k) - (V_load_vec(k) / R_load);
41         V_C_vec(k+1) = V_load_vec(k+1);
42
43         % Load: Current Calculation
44         i_load_vec(k+1) = V_load_vec(k+1) / R_load;
45
46     else % If the circuit is non ideal.
47         switch_state(k) = sw(D_non_ideal, t_vec(k)); % calling sw.m
48
49         if(switch_state(k))
50             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
```

```

51     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
52
53     % Switch 1: Voltage and Current Calculations
54     V_switch1(k+1) = 0;
55     i_switch1(k+1) = i_L_vec(k+1);
56     P_switch1(k+1) = (R_T_on * i_L_vec(k+1) + V_T_on) * i_L_vec(k+1);
57
58     % Switch 2: Voltage and Current Calculations
59     V_switch2(k+1) = V_D_on + (R_D_on * i_L_vec(k+1)) - V_in;
60     i_switch2(k+1) = 0;
61     else
62     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);
63
64     % Switch 1: Voltage and Current Calculations
65     V_switch1(k+1) = V_in + (R_D_on * i_L_vec(k+1)) + V_D_on;
66     i_switch1(k+1) = 0;
67
68     % Switch 2: Voltage and Current Calculations
69     V_switch2(k+1) = 0;
70     i_switch2(k+1) = i_L_vec(k+1);
71     P_switch2(k+1) = (R_D_on * i_L_vec(k+1) + V_D_on) * i_L_vec(k+1);
72     end
73
74     V_load_vec(k+1) = V_load_vec(k) + dt * ((i_L_vec(k) - (V_load_vec(k) / R_load)) / C);
%V_load
75     end
76
77     % Increment the time and index
78     t_vec(k + 1) = t_vec(k) + dt;
79     k = k + 1;
80 end

```

## Project #1\buckproc.m

```
1 % ID Number: 229,506
2 % ECE 31033 - Project #1
3 % buckproc.m
4
5 % The file buckproc.m first contains the circuit parameter values (i.e. L, C, fsw, time
6 % step, initial conditions etc.). Only the initial value of your circuit voltages and
7 % currents should be pre-established (i.e. Vload(1)=0). It then invokes buck. Finally,
8 % it performs your plotting and any post-processing calculations that are done
9 % using the simulated data (such as computing average values, efficiency, etc.).
10 %% Ideal - Given Values
11 V_in = 800;
12 V_load_avg = 400;
13 V_load_ripple = 10;
14 P_load_light = 50000;
15 P_load_heavy = 250000;
16 frequency = 10000;
17
18 ideal_boolean = 1; % = 0 if non ideal, = 1 if ideal; here, it is ON.
19
20 %% Ideal - Calculated Values
21 T_sw = 1 / frequency;
22 D = V_load_avg / V_in; % Duty Cycle
23
24 R_load_light = (V_load_avg^2) / P_load_light;
25 R_load_heavy = (V_load_avg^2) / P_load_heavy;
26
27 L_crit = (R_load_light * (1 - D)) / (2 * frequency);
28 L = L_crit * 1.1;
29
30 C = (V_load_avg / V_load_ripple) * (T_sw^2 * (1 - D)) / (8 * L);
31
32 I_load_light = V_load_avg / R_load_light;
33 I_load_heavy = V_load_avg / R_load_heavy;
34
35 i_L1_light = (V_load_avg / R_load_light) - (1 - D) * T_sw * V_load_avg / (2 * L);
36 i_L2_light = (V_load_avg / R_load_light) + (1 - D) * T_sw * V_load_avg / (2 * L);
37
38 i_L1_heavy = (V_load_avg / R_load_heavy) - (1 - D) * T_sw * V_load_avg / (2 * L);
39 i_L2_heavy = (V_load_avg / R_load_heavy) + (1 - D) * T_sw * V_load_avg / (2 * L);
40
41 %% Buck Initialization - Heavy Load
42 % Initializing Values
43 k = 1;
44 t = 0;
45 dt = 1e-7;
46
47 tend = 100 * T_sw;
48
49 % Zero Vectors (used in buck)
50 t_vec = [0];
51 switch_state = [0];
52
53 V_L_vec = [0];
```

```

54 i_L_vec = [0];
55
56 V_C_vec = [0];
57 i_C_vec = [0];
58
59 V_load_vec = [0];
60 i_load_vec = [0];
61
62 V_switch1 = [0];
63 i_switch1 = [0];
64
65 V_switch2 = [0];
66 i_switch2 = [0];
67
68 %% Running Buck - Using R_load_heavy
69 R_load = R_load_heavy;
70 disp('Running buck for heavy load.');
```

buck

```

72
73 %% Post-processing Calculations (computing avg values, efficiency, etc)
74 disp("-----")
75 disp("Heavy Averages:")
76
77 V_load_avg_func_H = aver(V_load_vec, T_sw, dt);
78 disp("  V_load Average: " + V_load_avg_func_H);
79
80 i_load_avg_func_H = aver(i_load_vec, T_sw, dt);
81 disp("  i_load Average: " + i_load_avg_func_H);
82
83 V_L_func_H = aver(V_L_vec, T_sw, dt);
84 disp("  V_L Average: " + V_L_func_H);
85
86 i_L_func_H = aver(i_L_vec, T_sw, dt);
87 disp("  i_L Average: " + i_L_func_H);
88
89 V_C_func_H = aver(V_C_vec, T_sw, dt);
90 disp("  V_C Average: " + V_C_func_H);
91
92 i_C_func_H = aver(i_C_vec, T_sw, dt);
93 disp("  i_C Average: " + i_C_func_H);
94
95 V_sw1_func_H = aver(V_switch1, T_sw, dt);
96 disp("  V_sw1 Average: " + V_sw1_func_H);
97
98 i_sw1_func_H = aver(i_switch1, T_sw, dt);
99 disp("  i_sw1 Average: " + i_sw1_func_H);
100
101 V_sw2_func_H = aver(V_switch2, T_sw, dt);
102 disp("  V_sw2 Average: " + V_sw2_func_H);
103
104 i_sw2_func_H = aver(i_switch2, T_sw, dt);
105 disp("  i_sw2 Average: " + i_sw2_func_H);
106
107 P_out_H = (V_load_avg_func_H^2) / R_load;
108 P_in_H = V_in * i_sw1_func_H;
109 eff_H = P_out_H / P_in_H;
```



```

110
111 disp("Efficiency for Light Load: " + (eff_H * 100) + "%.");
112 disp("-----")
113
114 %% Plotting - Heavy Load - Transient
115 % Plots for the transient to steady state
116 figure;
117 sgtitle("Heavy Load: Voltage and Current Plots for All Components at Transient State");
118 % Plots for the Load
119 subplot(5,2,1);
120 plot(t_vec, V_load_vec);
121 title('Load Voltage vs Time');
122 xlabel('Time (ms)');
123 ylabel('Voltage (V)');
124
125 subplot(5,2,2);
126 plot(t_vec, i_load_vec);
127 title('Load Current vs Time');
128 xlabel('Time (ms)');
129 ylabel('Current (A)');
130
131 % Plots for the Inductor
132 subplot(5,2,3);
133 plot(t_vec, V_L_vec);
134 title('Inductor Voltage vs Time');
135 xlabel('Time (ms)');
136 ylabel('Voltage (V)');
137
138 subplot(5,2,4);
139 plot(t_vec, i_L_vec);
140 title('Inductor Current vs Time');
141 xlabel('Time (ms)');
142 ylabel('Current (A)');
143
144 % Plots for the Capacitor
145 subplot(5,2,5);
146 plot(t_vec, V_C_vec);
147 title('Capacitor Voltage vs Time');
148 xlabel('Time (ms)');
149 ylabel('Voltage (V)');
150
151 subplot(5,2,6);
152 plot(t_vec, i_C_vec);
153 title('Capacitor Current vs Time');
154 xlabel('Time (ms)');
155 ylabel('Current (A)');
156
157 % Plots for Switch 1
158 subplot(5,2,7);
159 plot(t_vec, V_switch1);
160 title('Transistor Switching Voltage vs Time');
161 xlabel('Time (ms)');
162 ylabel('Voltage (V)');
163
164 subplot(5,2,8);
165 plot(t_vec, i_switch1);

```

```

166 title('Transistor Switching Current vs Time');
167 xlabel('Time (ms)');
168 ylabel('Current (A)');
169
170 % Plots for Switch 2
171 subplot(5,2,9);
172 plot(t_vec, V_switch2);
173 title('Diode Switching Voltage vs Time');
174 xlabel('Time (ms)');
175 ylabel('Voltage (V)');
176
177 subplot(5,2,10);
178 plot(t_vec, i_switch2);
179 title('Diode Switching Current vs Time');
180 xlabel('Time (ms)');
181 ylabel('Current (A)');
182
183 %% Plotting - Heavy Load - Steady State
184 periods_to_plot = 2;
185
186 points_per_period = round(T_sw / dt); % Points per period
187 total_periods = floor(tend / T_sw); % Total number of periods in the simulation
188
189 start_index = max(1, (total_periods - periods_to_plot) * points_per_period + 1);
190 end_index = min(length(t_vec), total_periods * points_per_period);
191
192 range_to_plot = start_index:end_index;
193
194 %% Plot
195 figure;
196 sgtitle("Heavy Load: Voltage and Current Plots for All Components at Steady State");
197 % Plots for the Load
198 subplot(5,2,1);
199 plot(t_vec(range_to_plot), V_load_vec(range_to_plot));
200 title('Load Voltage vs Time');
201 xlabel('Time (ms)');
202 ylabel('Voltage (V)');
203
204 subplot(5,2,2);
205 plot(t_vec(range_to_plot), i_load_vec(range_to_plot));
206 title('Load Current vs Time');
207 xlabel('Time (ms)');
208 ylabel('Current (A)');
209
210 % Plots for the Inductor
211 subplot(5,2,3);
212 plot(t_vec(range_to_plot), V_L_vec(range_to_plot));
213 title('Inductor Voltage vs Time');
214 xlabel('Time (ms)');
215 ylabel('Voltage (V)');
216
217 subplot(5,2,4);
218 plot(t_vec(range_to_plot), i_L_vec(range_to_plot));
219 title('Inductor Current vs Time');
220 xlabel('Time (ms)');
221 ylabel('Current (A)');

```

```

222
223 % Plots for the Capacitor
224 subplot(5,2,5);
225 plot(t_vec(range_to_plot), V_C_vec(range_to_plot));
226 title('Capacitor Voltage vs Time');
227 xlabel('Time (ms)');
228 ylabel('Voltage (V)');
229
230 subplot(5,2,6);
231 plot(t_vec(range_to_plot), i_C_vec(range_to_plot));
232 title('Capacitor Current vs Time');
233 xlabel('Time (ms)');
234 ylabel('Current (A)');
235
236 % Plots for Switch 1
237 subplot(5,2,7);
238 plot(t_vec(range_to_plot), V_switch1(range_to_plot));
239 title('Transistor Switching Voltage vs Time');
240 xlabel('Time (ms)');
241 ylabel('Voltage (V)');
242
243 subplot(5,2,8);
244 plot(t_vec(range_to_plot), i_switch1(range_to_plot));
245 title('Transistor Switching Current vs Time');
246 xlabel('Time (ms)');
247 ylabel('Current (A)');
248
249 % Plots for Switch 2
250 subplot(5,2,9);
251 plot(t_vec(range_to_plot), V_switch2(range_to_plot));
252 title('Diode Switching Voltage vs Time');
253 xlabel('Time (ms)');
254 ylabel('Voltage (V)');
255
256 subplot(5,2,10);
257 plot(t_vec(range_to_plot), i_switch2(range_to_plot));
258 title('Diode Switching Current vs Time');
259 xlabel('Time (ms)');
260 ylabel('Current (A)');
261
262 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
263 %% Buck Initialization - Light Load
264 % Initializing Values
265 k = 1;
266 t = 0;
267 dt = 1e-7;
268
269 tend = 250 * T_sw;
270
271 % Zero Vectors (used in buck)
272 t_vec = [0];
273 switch_state = [0];
274
275 V_L_vec = [0];
276 i_L_vec = [0];
277

```

```

278 V_C_vec = [0];
279 i_C_vec = [0];
280
281 V_load_vec = [0];
282 i_load_vec = [0];
283
284 V_switch1 = [0];
285 i_switch1 = [0];
286
287 V_switch2 = [0];
288 i_switch2 = [0];
289
290 %% Running Buck - Using R_load_light
291 R_load = R_load_light;
292 disp('Running buck for light load.');
```

buck

```

294
295 %% Post-processing Calculations (computing avg values, efficiency, etc)
296 disp("-----")
297 disp("Light Averages:")
298
299 V_load_avg_func_L = aver(V_load_vec, T_sw, dt);
300 disp("  V_load Average: " + V_load_avg_func_L);
301
302 i_load_avg_func_L = aver(i_load_vec, T_sw, dt);
303 disp("  i_load Average: " + i_load_avg_func_L);
304
305 V_L_func_L = aver(V_L_vec, T_sw, dt);
306 disp("  V_L Average: " + V_L_func_L);
307
308 i_L_func_L = aver(i_L_vec, T_sw, dt);
309 disp("  i_L Average: " + i_L_func_L);
310
311 V_C_func_L = aver(V_C_vec, T_sw, dt);
312 disp("  V_C Average: " + V_C_func_L);
313
314 i_C_func_L = aver(i_C_vec, T_sw, dt);
315 disp("  i_C Average: " + i_C_func_L);
316
317 V_sw1_func_L = aver(V_switch1, T_sw, dt);
318 disp("  V_sw1 Average: " + V_sw1_func_L);
319
320 i_sw1_func_L = aver(i_switch1, T_sw, dt);
321 disp("  i_sw1 Average: " + i_sw1_func_L);
322
323 V_sw2_func_L = aver(V_switch2, T_sw, dt);
324 disp("  V_sw2 Average: " + V_sw2_func_L);
325
326 i_sw2_func_L = aver(i_switch2, T_sw, dt);
327 disp("  i_sw2 Average: " + i_sw2_func_L);
328
329 P_out_L = (V_load_avg_func_L^2) / R_load;
330 P_in_L = V_in * i_sw1_func_L;
331 eff_L = P_out_L / P_in_L;
332
333 disp("Efficiency for Light Load: " + (eff_L * 100) + "%.");
```

```

334 disp("-----")
335 %% Plotting - Light Load - Transient
336 % Plots for the transient to steady state
337 figure;
338 sgtitle("Light Load: Voltage and Current Plots for All Components at Transient State");
339 % Plots for the Load
340 subplot(5,2,1);
341 plot(t_vec, V_load_vec);
342 title('Load Voltage vs Time');
343 xlabel('Time (ms)');
344 ylabel('Voltage (V)');
345
346 subplot(5,2,2);
347 plot(t_vec, i_load_vec);
348 title('Load Current vs Time');
349 xlabel('Time (ms)');
350 ylabel('Current (A)');
351
352 % Plots for the Inductor
353 subplot(5,2,3);
354 plot(t_vec, V_L_vec);
355 title('Inductor Voltage vs Time');
356 xlabel('Time (ms)');
357 ylabel('Voltage (V)');
358
359 subplot(5,2,4);
360 plot(t_vec, i_L_vec);
361 title('Inductor Current vs Time');
362 xlabel('Time (ms)');
363 ylabel('Current (A)');
364
365 % Plots for the Capacitor
366 subplot(5,2,5);
367 plot(t_vec, V_C_vec);
368 title('Capacitor Voltage vs Time');
369 xlabel('Time (ms)');
370 ylabel('Voltage (V)');
371
372 subplot(5,2,6);
373 plot(t_vec, i_C_vec);
374 title('Capacitor Current vs Time');
375 xlabel('Time (ms)');
376 ylabel('Current (A)');
377
378 % Plots for Switch 1
379 subplot(5,2,7);
380 plot(t_vec, V_switch1);
381 title('Transistor Switching Voltage vs Time');
382 xlabel('Time (ms)');
383 ylabel('Voltage (V)');
384
385 subplot(5,2,8);
386 plot(t_vec, i_switch1);
387 title('Transistor Switching Current vs Time');
388 xlabel('Time (ms)');
389 ylabel('Current (A)');

```

```

390
391 % Plots for Switch 2
392 subplot(5,2,9);
393 plot(t_vec, V_switch2);
394 title('Diode Switching Voltage vs Time');
395 xlabel('Time (ms)');
396 ylabel('Voltage (V)');
397
398 subplot(5,2,10);
399 plot(t_vec, i_switch2);
400 title('Diode Switching Current vs Time');
401 xlabel('Time (ms)');
402 ylabel('Current (A)');
403
404 %% Plotting - Light Load - Steady State
405 periods_to_plot = 2;
406
407 points_per_period = round(T_sw / dt); % Points per period
408 total_periods = floor(tend / T_sw); % Total number of periods in the simulation
409
410 start_index = max(1, (total_periods - periods_to_plot) * points_per_period + 1);
411 end_index = min(length(t_vec), total_periods * points_per_period);
412
413 range_to_plot = start_index:end_index;
414
415 %% Plot
416 figure;
417 sgtitle('Light Load: Voltage and Current Plots for All Components at Steady State');
418 % Plots for the Load
419 subplot(5,2,1);
420 plot(t_vec(range_to_plot), V_load_vec(range_to_plot));
421 title('Load Voltage vs Time');
422 xlabel('Time (ms)');
423 ylabel('Voltage (V)');
424
425 subplot(5,2,2);
426 plot(t_vec(range_to_plot), i_load_vec(range_to_plot));
427 title('Load Current vs Time');
428 xlabel('Time (ms)');
429 ylabel('Current (A)');
430
431 % Plots for the Inductor
432 subplot(5,2,3);
433 plot(t_vec(range_to_plot), V_L_vec(range_to_plot));
434 title('Inductor Voltage vs Time');
435 xlabel('Time (ms)');
436 ylabel('Voltage (V)');
437
438 subplot(5,2,4);
439 plot(t_vec(range_to_plot), i_L_vec(range_to_plot));
440 title('Inductor Current vs Time');
441 xlabel('Time (ms)');
442 ylabel('Current (A)');
443
444 % Plots for the Capacitor
445 subplot(5,2,5);

```

```

446 plot(t_vec(range_to_plot), V_C_vec(range_to_plot));
447 title('Capacitor Voltage vs Time');
448 xlabel('Time (ms)');
449 ylabel('Voltage (V)');
450
451 subplot(5,2,6);
452 plot(t_vec(range_to_plot), i_C_vec(range_to_plot));
453 title('Capacitor Current vs Time');
454 xlabel('Time (ms)');
455 ylabel('Current (A)');
456
457 % Plots for Switch 1
458 subplot(5,2,7);
459 plot(t_vec(range_to_plot), V_switch1(range_to_plot));
460 title('Transistor Switching Voltage vs Time');
461 xlabel('Time (ms)');
462 ylabel('Voltage (V)');
463
464 subplot(5,2,8);
465 plot(t_vec(range_to_plot), i_switch1(range_to_plot));
466 title('Transistor Switching Current vs Time');
467 xlabel('Time (ms)');
468 ylabel('Current (A)');
469
470 % Plots for Switch 2
471 subplot(5,2,9);
472 plot(t_vec(range_to_plot), V_switch2(range_to_plot));
473 title('Diode Switching Voltage vs Time');
474 xlabel('Time (ms)');
475 ylabel('Voltage (V)');
476
477 subplot(5,2,10);
478 plot(t_vec(range_to_plot), i_switch2(range_to_plot));
479 title('Diode Switching Current vs Time');
480 xlabel('Time (ms)');
481 ylabel('Current (A)');
482
483 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
484 %% Non Ideal - Given Values
485 V_T_on = 1;
486 V_D_on = 1;
487 R_T_on = 0.01;
488 R_D_on = 0.01;
489
490 ideal_boolean = 0; % = 0 if non ideal, = 1 if ideal; here, it is OFF.
491
492 %% Non Ideal - Calculated Values
493 R_load = R_load_heavy;
494 D_non_ideal = (V_D_on + V_load_avg + (R_D_on * V_load_avg / R_load));
495 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));
496
497 i_L1_NI = (V_load_avg / R_load) - (((1 - D_non_ideal) * T_sw * V_load_avg) / (2 * L));
498 i_L2_NI = (V_load_avg / R_load) + (((1 - D_non_ideal) * T_sw * V_load_avg) / (2 * L));
499
500 %% Buck Initialization - Heavy Load

```

```

501 % Initializing Values
502 k = 1;
503 t = 0;
504 dt = 1e-7;
505
506 tend = 100 * T_sw;
507
508 % Zero Vectors (used in buck)
509 t_vec = [0];
510 switch_state = [0];
511
512 V_L_vec = [0];
513 i_L_vec = [0];
514
515 V_C_vec = [0];
516 i_C_vec = [0];
517
518 V_load_vec = [0];
519 i_load_vec = [0];
520
521 V_switch1 = [0];
522 i_switch1 = [0];
523
524 V_switch2 = [0];
525 i_switch2 = [0];
526
527 P_switch1 = [0]; % Power loss across the transistor; new for non-ideal calculations.
528 P_switch2 = [0]; % Power loss across the diode; new for non-ideal calculations.
529
530 %% Running Buck - Using R_load_heavy
531 R_load = R_load_heavy;
532 disp('Running buck for heavy load and non ideal conditions. ');
533 buck
534
535 %% Post-processing Calculations (computing avg values, efficiency, etc)
536 disp("-----")
537 disp("Non Ideal Averages:")
538
539 V_load_avg_func_NI = aver(V_load_vec, T_sw, dt);
540 disp(" V_load Average: " + V_load_avg_func_NI);
541
542 i_L_func_NI = aver(i_L_vec, T_sw, dt);
543 disp(" i_L Average: " + i_L_func_NI);
544
545 V_sw1_func_NI = aver(V_switch1, T_sw, dt);
546 disp(" V_sw1 Average: " + V_sw1_func_NI);
547
548 i_sw1_func_NI = aver(i_switch1, T_sw, dt);
549 disp(" i_sw1 Average: " + i_sw1_func_NI);
550
551 V_sw2_func_NI = aver(V_switch2, T_sw, dt);
552 disp(" V_sw2 Average: " + V_sw2_func_NI);
553
554 i_sw2_func_NI = aver(i_switch2, T_sw, dt);
555 disp(" i_sw2 Average: " + i_sw2_func_NI);
556

```



```

557 P_sw1_func_NI = aver(P_switch1, T_sw, dt);
558 disp(" P_sw1 Average: " + V_sw2_func_NI);
559
560 P_sw2_func_NI = aver(P_switch2, T_sw, dt);
561 disp(" P_sw2 Average: " + i_sw2_func_NI);
562
563 P_out_NI = (V_load_avg_func_NI^2) / R_load;
564 P_in_NI = V_in * i_sw1_func_NI;
565 eff_NI = P_out_NI / P_in_NI;
566
567 disp("Efficiency for Non-Ideal: " + (eff_NI * 100) + "%.");
568 disp("Transistor Power Loss: " + P_sw1_func_NI);
569 disp("Diode Power Loss: " + P_sw2_func_NI);
570 disp("-----")
571 %% Plotting - Non Ideal Heavy Load - Transient
572 % Plots for the transient to steady state
573 figure;
574 sgtitle("Non Ideal Heavy Load: Voltage and Current Plots for All Components at Transient
State");
575 % Plots for the Load
576 subplot(3,2,1);
577 plot(t_vec, V_load_vec);
578 title('Load Voltage vs Time');
579 xlabel('Time (ms)');
580 ylabel('Voltage (V)');
581
582 subplot(3,2,2);
583 plot(t_vec, i_L_vec);
584 title('Inductor Current vs Time');
585 xlabel('Time (ms)');
586 ylabel('Current (A)');
587
588 % Plots for Switch 1
589 subplot(3,2,3);
590 plot(t_vec, V_switch1);
591 title('Transistor Switching Voltage vs Time');
592 xlabel('Time (ms)');
593 ylabel('Voltage (V)');
594
595 subplot(3,2,4);
596 plot(t_vec, i_switch1);
597 title('Transistor Switching Current vs Time');
598 xlabel('Time (ms)');
599 ylabel('Current (A)');
600
601 % Plots for Switch 2
602 subplot(3,2,5);
603 plot(t_vec, V_switch2);
604 title('Diode Switching Voltage vs Time');
605 xlabel('Time (ms)');
606 ylabel('Voltage (V)');
607
608 subplot(3,2,6);
609 plot(t_vec, i_switch2);
610 title('Diode Switching Current vs Time');
611 xlabel('Time (ms)');

```

```

612 ylabel('Current (A)');
613
614 %% Plotting - Heavy Load - Steady State
615 periods_to_plot = 2;
616
617 points_per_period = round(T_sw / dt); % Points per period
618 total_periods = floor(tend / T_sw); % Total number of periods in the simulation
619
620 start_index = max(1, (total_periods - periods_to_plot) * points_per_period + 1);
621 end_index = min(length(t_vec), total_periods * points_per_period);
622
623 range_to_plot = start_index:end_index;
624
625 %% Plot
626 figure;
627 sgtitle("Non Ideal Heavy Load: Voltage and Current Plots for All Components at Steady State")
628 ;
629 % Plots for the Load
630 subplot(3,2,1);
631 plot(t_vec(range_to_plot), V_load_vec(range_to_plot));
632 title('Load Voltage vs Time');
633 xlabel('Time (ms)');
634 ylabel('Voltage (V)');
635
636 subplot(3,2,2);
637 plot(t_vec(range_to_plot), i_L_vec(range_to_plot));
638 title('Inductor Current vs Time');
639 xlabel('Time (ms)');
640 ylabel('Current (A)');
641
642 % Plots for Switch 1
643 subplot(3,2,3);
644 plot(t_vec(range_to_plot), V_switch1(range_to_plot));
645 title('Transistor Switching Voltage vs Time');
646 xlabel('Time (ms)');
647 ylabel('Voltage (V)');
648
649 subplot(3,2,4);
650 plot(t_vec(range_to_plot), i_switch1(range_to_plot));
651 title('Transistor Switching Current vs Time');
652 xlabel('Time (ms)');
653 ylabel('Current (A)');
654
655 % Plots for Switch 2
656 subplot(3,2,5);
657 plot(t_vec(range_to_plot), V_switch2(range_to_plot));
658 title('Diode Switching Voltage vs Time');
659 xlabel('Time (ms)');
660 ylabel('Voltage (V)');
661
662 subplot(3,2,6);
663 plot(t_vec(range_to_plot), i_switch2(range_to_plot));
664 title('Diode Switching Current vs Time');
665 xlabel('Time (ms)');
666 ylabel('Current (A)');

```