Report

Implementation Details

The purpose of this project is to simulate the steady states of a commonly used circuit "Boost Converter". And as the following two figures show, there are two implementations of this circuit. The first circuit is the "Boost Converter" with an ideal switch, while the second circuit is the practical implementation.

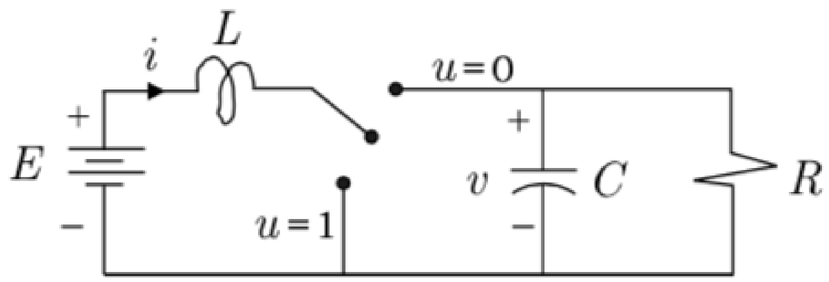


Figure 1. DC-to-DC Boost Converter with an ideal switch

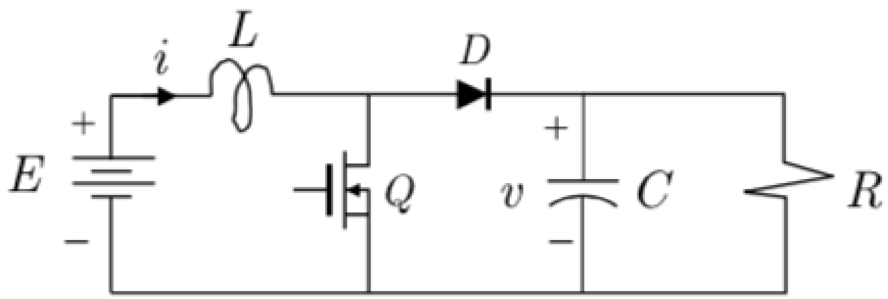


Figure 2. DC-to-DC Boost Converter with CMOS switch

We simulate these two models of "Boost Converter" respectively. For the first model, we simulate the model using C++. To implement the ideal switch, we transform the circuit into another form, as shown in Figure 3.

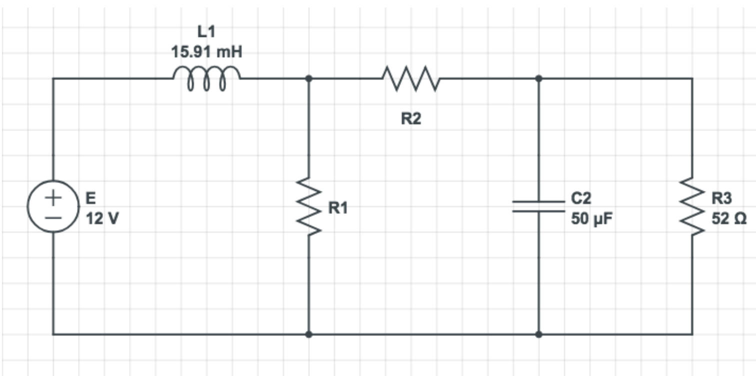


Figure 3. Transformed circuit of Boost Converter with an ideal switch

In this circuit, the ideal switch is replaced by two resistors. The values of these resistors will change according to the states of the ideal switch. When the switch connects to 1, the value of R_1 1Ω, while the value of R_2 will be $1M\Omega$. When the switch connects to 0, R_1 and R_2 will exchange their values. Thus, the circuit can be described using the following objective functions:

$$L\frac{di}{dt} = E - i'R_1$$

$$C\frac{dv}{dt} = -\frac{v}{R} + (i - i')$$

The variable i' represents the current passing the resistor R_1, and it satisfy the following equation:

$$i'R_1 = v + (i - i')R_2$$

For the second model, we use Java to simulate the model. We just use the EKV model to describe the behavior of the CMOS switch. The V_{GS} of the switch is given by a PWM, and V_{DS} of the switch is approximate to E. As for the diode, we also use a resistor R_1 to represent it. The value of it is 0 when the diode conducts current, and the value of it is $100M\Omega$ when the diode does not conduct current. Therefore, we will have the following two objective functions:

$$L\frac{di}{dt} = E - (i - i_d)R_1 - v$$

$$C\frac{dv}{dt} = i - i_d - \frac{v}{R_0}$$

Module Design

For the first model, there are mainly three modules: "main" module, "ODEhelper" module and "RK34" module. The "main" module is the entry of the program and it will call different modules to get the output. The "ODEhelper" includes all the helper functions needed to solve the ODEs. For example, to simulate the first model, we need to get the values of R_1 and R_2 and the helper functions "R0Helper" and "R1Helper" in the "ODEHelper" module will return the values of R_1 and R_2 according to time. In this case, we just use the RK34 with adaptive time step to simulate the model. Thus, the "RK34" module implement this algorithm and will be used to solve the ODEs with the help of "ODEHelper" module.

For the second model, the design is very similar to the design for the Assignment 4. There are three modules: "Main" module, "ODE_solver" module and "Simulator" module. The function of the "Main" module is to call the other modules and output the results of simulation. The "ODE_solver" module implements the RK34 with adaptive time step algorithm. And the third module “Simulator” is used to simulate the circuit with some parameters which include initial guess, start time, step size and simulation duration.

As for validation, we need to validate the correctness of the ODE solver that we implement in this project. Since the RK34 with adaptive time step algorithm has already been implemented and tested in the Assignment 4. There is no need to use the same way to test the solver again. Thus, this time we directly test the results of our simulation with the help of an embedded model for Boost Converter simulation in Matlab.

Results and Discussion

For the first model, we used the parameters shown in the following table.

Table 1. Parameters used in simulating the first model

E	$12V$
L	$15.91mH$
C	$50\mu F$
R_3	52Ω
R_1	1Ω $(1M\Omega)$
R_2	$1M\Omega$ (1Ω)

We change the frequency of the switch and got different results. At the beginning, we set the frequency of the switch to 45KHz, and the result appears unstable. We tried to increase the frequency first, but we found that the result shows no tendency to become stable. Therefore, we decreased the frequency step by step, and the result became stable till the frequency decreases to 20KHz. And then we simulate the circuit with the same parameters on Matlab. The two results are combined and shown in the following figure. The yellow line is the result of simulation on Matlab while the blue line is result of our simulation.

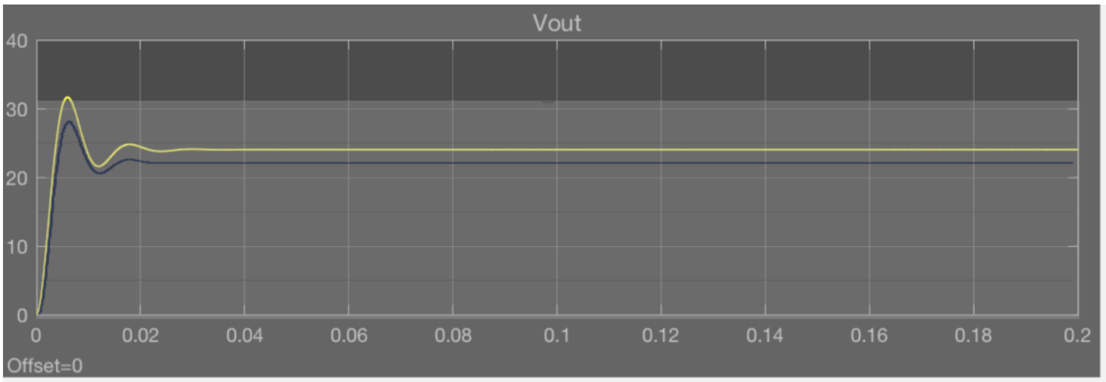
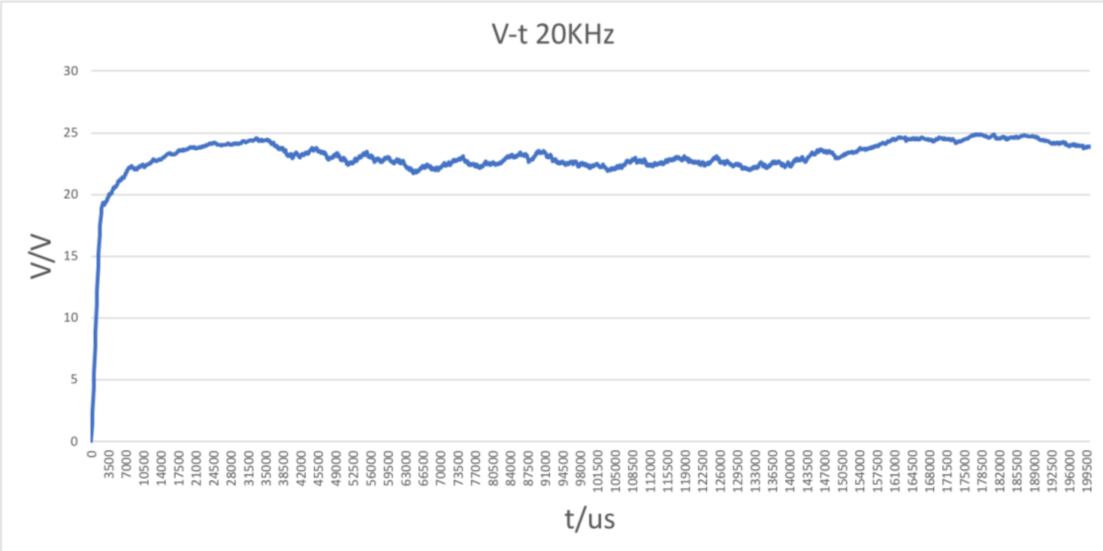


Figure 1. Simulations of the Boost converter with ideal switch

For the simulation of the circuit with CMOS switch, the parameters are the same for the same components of the two circuits. The value of the equivalent resistor of the diode has been introduced in the implementation detail part. As for the PWM input as the V_{GS}, we set the duty cycle of the signal to 50%, and the amplitude of the signal to 28V. The result of the simulation is shown in the following figure.



As you can see, the result that we got is not very stable. And found that the model that we built is very sensitive to the amplitude of the PWM. When the amplitude is less than 15V, the voltage on the load resistor will always get stuck to 12V. But when we continuously increasing the amplitude of PWM, the voltage on the load resistor tends to be stable at around the amplitude of the PWM. We have used the previous test example to test our ODE solver, and it turns out that the ODE solver can get the correct results. We have tried to change the step size of simulation but the results of simulation are still not satisfying. And the figure above is the best results that I have got. Thus, we think the problem of the model may lie in the objective functions that we built to describe the circuit. But till now we cannot find a better set of objective functions to simulate the circuit.