EEE 148 Buck Converter in Simulink

What is a digital signal, and what is digital signal control?

Everything we see and experience is analog. Although the technology we enjoy today like our smartphones, cameras, televisions, and computers may have been digitized, these are inherently analog to its core as the voltages and currents that power and control these devices are analog in nature.

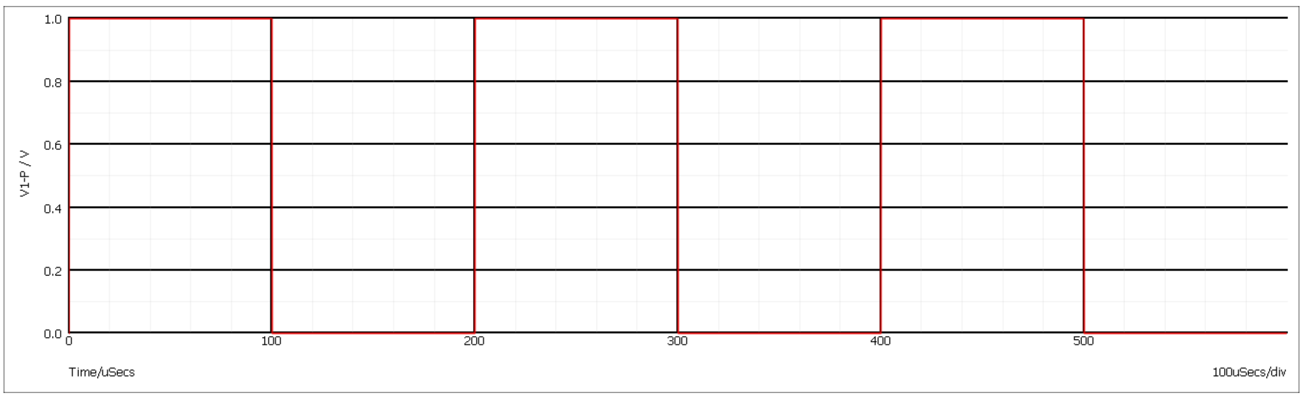

Figure 1: Square Wave with frequency of 5 kHz.

Looking at Figure 1, you may think this is a digital signal; but far from it, this is actually still analog. A digital signal is simply a representation of an analog signal for ease of processing and control. There are several ways to make an analog signal be digital, but for simplicity, let's just say that HIGH voltage represents 1 while LOW voltage means 0.

Question: What is the digital signal representation of the analog signal in Fig. 1? 1010? Maybe. However, this could easily be 11001100 or 111000111000 depending on the sampling frequency. This will be discussed further in the lab exercise for analog-to-digital conversion.

Representing an Analog Circuit as a System Block

In order to model a digitally-controlled system, it is necessary to represent our system in a way that can be understood by a digital system. To do this, we can represent each part of the system as blocks with a corresponding equation or transfer function. For example in Figure 2, we have a voltage divider circuit where we get the output across R2. We can see that the transfer function would be $V_{out}/V_{in} = G = R_2/(R_1 + R_2)$. In general, you want to do this whenever you are modeling a control system as it would be convenient to use functions to represent the circuits, instead of the actual circuits themselves.

Our digitally-controlled buck converter consists of several blocks: the buck converter, the PWM generator, and the compensator. As was discussed in the previous lab exercises, we will be using the PWM generator and the compensator to control our buck converter. In summary, the PWM generates pulses to the buck converter which then outputs a DC voltage. The compensator samples this DC voltage and compares it with a reference voltage to capture the difference between the actual output of our buck converter and the desired output. This is then used to vary the output of our PWM generator to vary the output of our buck converter.

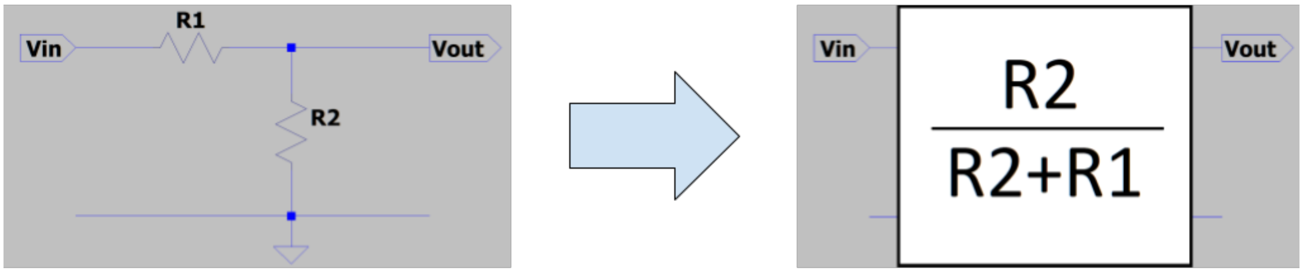

Figure 2: Representing a voltage divider circuit as a system block.

Buck Converter in Simulink

To represent the buck converter as a system block, we first look at the components inside the buck converter, and determine the input and output variables. The output of the buck would be vo which is the voltage across the load. The inputs are Vg, d, and io. However, why is io an input? Our buck converter should be designed to output a constant vo regardless of the load resistance. For that to happen, iout must be an input variable to do so.

We can treat the buck converter as a black box with 3 inputs and 2 outputs like the following:

$$v_o = f(V_s, d, i_{out}) \qquad\qquad i_L = f(V_s, d, i_{out})$$

where both $f()$ and $g()$ are solvable using concepts in analog circuits. The buck converter in Figure 3 can be digitally represented as shown in Figure 4. In this design, we set L to be 4.1uH and C to be 376 uF.

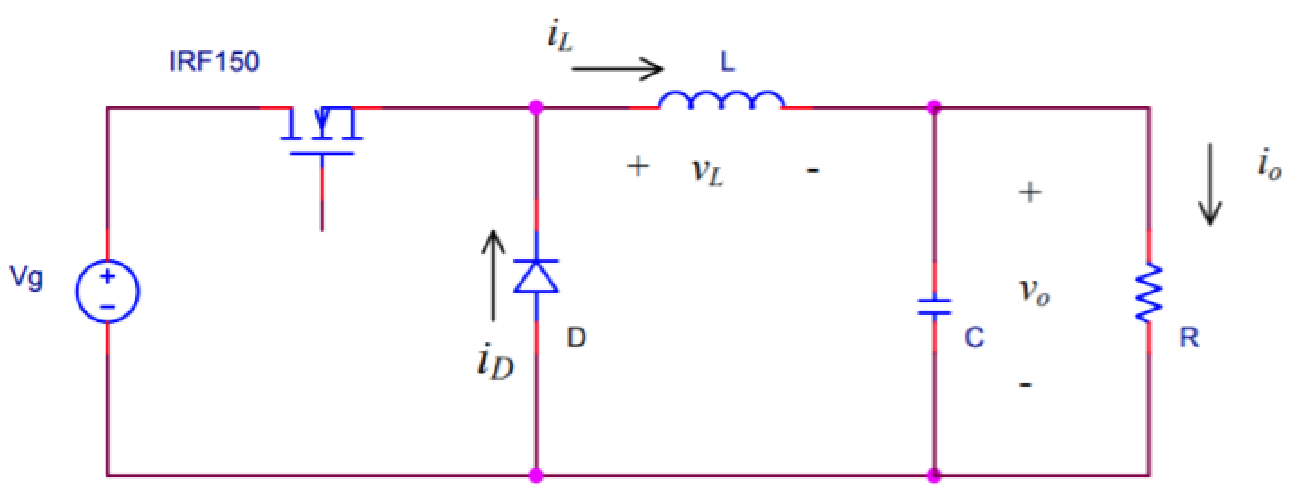

Figure 3: Buck Converter Circuit.

PWM in Simulink

For the PWM, the input is simply the duty cycle, and the output is a pulse train with the correct duty cycle. Frequency can also be used as an input, but for our purposes we will only make use of the duty cycle as the input. Previously, we designed a PWM generator using multivibrators, comparators, and/or op-amps. We varied the values of our components such as the resistors and capacitors in order to achieve the correct specifications.

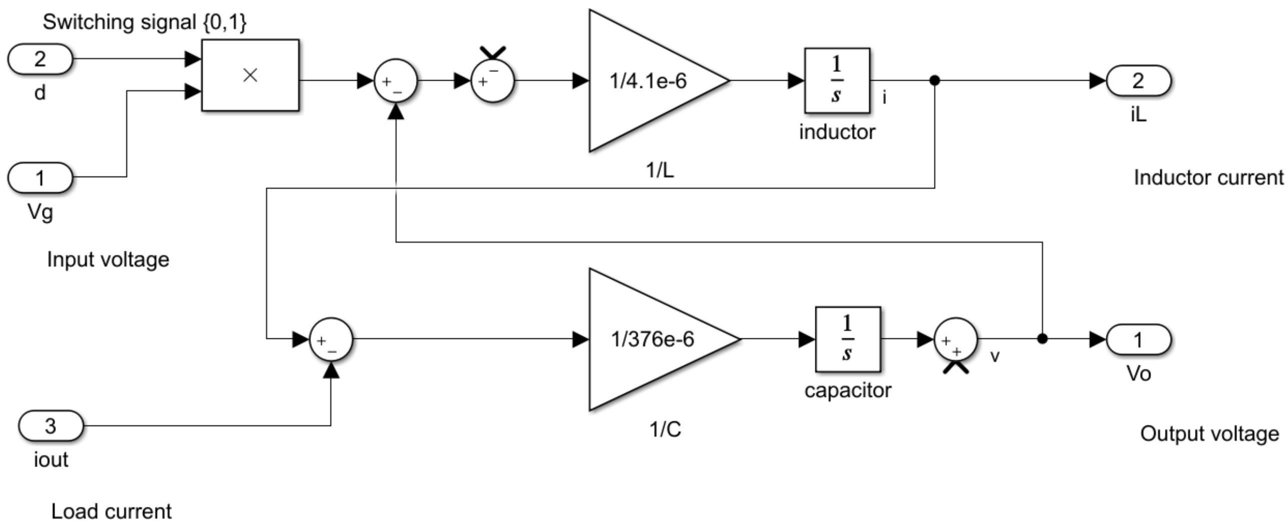

Figure 4: Digital Representation of a Buck Converter.

For a digital system, we can use a simplified representation of a PWM generator. We can treat it as a blackbox which takes in a value for the duty cycle and the frequency, and outputs a continuous pulse train based on that duty cycle. For our purpose, we add a bit more detail to our PWM module by representing the PWM comparator circuit in Lab 3. To design the PWM module, we will use a sawtooth generator for the ramp function, and a zero-crossing function for the comparator. The zero-crossing function compares the sawtooth output with the input duty cycle. We also use a fixed frequency, and only have the duty cycle as the input.

Question: If our duty cycle value ranges from 0 to 1 for 0% to 100%, what should be the parameters (e.g. amplitude, frequency, phase, offset, etc.) of our sawtooth generator such that it would map correctly to the desired PWM output?

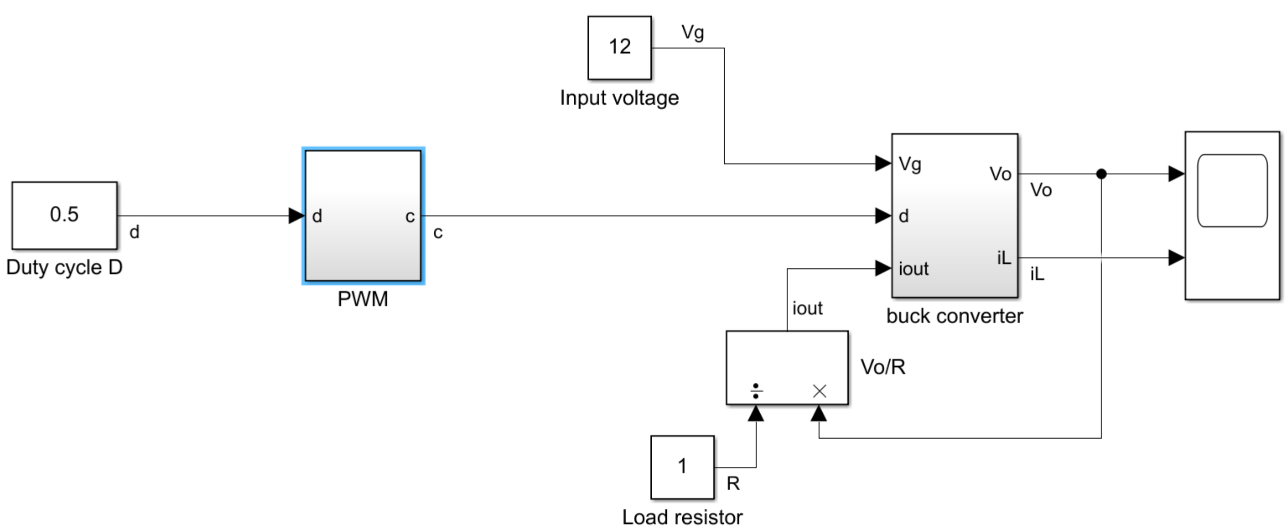

Figure 5: Open-Loop Buck Converter.

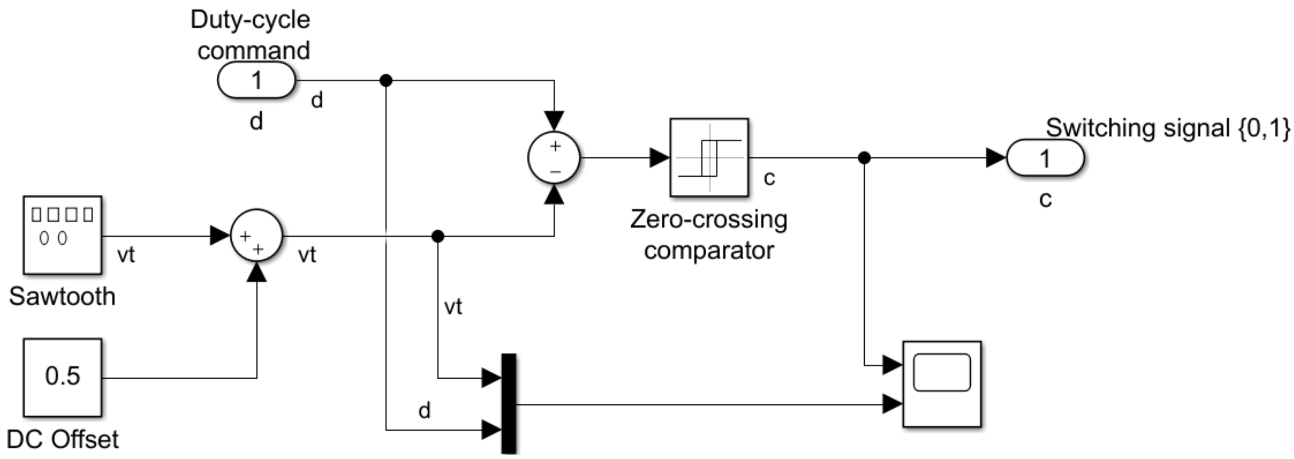

Figure 6: Pulse Width Modulation Module using a sawtooth generator and a zero-crossing function.

Learning Activity: Open-Loop Buck Converter

Figure 5 shows the diagram of an open-loop buck converter in simulink. We make use of an input duty cycle d to the PWM module which is then imputed to the buck converter. Our buck converter also takes in an input voltage V_g, and its own output current i_{out}. Notice how we sample the current from our buck converter. Since the output of the buck converter is a voltage, we can sample the current using a resistor to get the value of i_{out}.

Exercise:

1. Open "open_buck_template.mdl". Implement the buck converter block using the block diagrams in Figures 4 and 5 in Simulink. Use duty cycle of 0.5. Show the following results:
   (a) v_o waveform. What is the value of v_o at steady-state?
   (b) i_{out} waveform.
2. In practical applications, our inductors and capacitors would have internal resistances. Modify the buck converter using the following values: $L = 5\mu H$, $R_L = 80m\Omega$, $C = 390\mu F$, $R_{esr} = 2m\Omega$.
   (a) Show the final block diagram of the buck converter with the correct values. (Hint: $V = IR$. You can treat R as a gain. R_L is the inductor resistance, R_{esr} is the capacitor resistance).
3. Connect the PWM to the buck converter. Connect $R_{load} = 1$. Set the duty cycle to 0.4.
   (a) Show the waveform of v_o? Get the value at steady state.
   (b) Show the waveform of i_{out}? Get the value at steady state.
4. Vary the duty cycle until the output of the buck converter (v_o) becomes 8V. Show the waveforms at this duty cycle.
   (a) What is the duty cycle d?
   (b) v_o waveform. What is the value of v_o at steady-state?
   (c) i_{out} waveform.

(d) What is your process for varying the duty cycle? How does the output of the buck converter affect your decision whether to increase or decrease the duty cycle?

5. Replace the load with $R_{load} = 10$ using the same duty cycle set in 3.

   (a) Show the new block diagram of the open-loop system.

   (b) What is the value of v_o? i_{out}? Show the waveforms. Get the values at steady state.

   (c) How does the load affect v_o and i_{out}? Explain.

   (d) How can you get back a desired value for v_o when using a different load?

Compensator in Simulink

The compensation block (discussed in Lab 3-4) is used to create a feedback connection in the system. It samples the output of the buck converter, compares it with a reference, then uses this to adjust the input to the PWM. This, then, corrects the output of the buck converter.

For now, let us create a compensation or feedback network. First, we need to sample the output for our buck converter. To sample an output voltage, we can use a voltage divider circuit which can be represented in our digital system as a gain similar to what we saw in Figure 2. This is then compared with a reference voltage and amplified by another gain block, similar to how the error amplifier circuit from Lab 3 works.

To compute for the reference voltage, we have to take note of the gain introduced by the voltage divide, then calculate the value such that the error would be zero as follows:

$$\begin{aligned} V_{ref} - v_{o,sampled} &= 0 \\ V_{ref} - G_{V divider} v_o &= 0 \\ V_{ref} &= G_{V divider} v_o \end{aligned}$$

where if our desired v_o is 5V, and $G_{V divider} = 0.4$, then $V_{ref} = 2$.

To complete the loop, we connect the output v_o to the voltage divider, then replace the constant d for the duty cycle with the output of our compensation network as shown in Figure 7.

Using Digital Devices

In some cases, we would want a more powerful device such as a microcontroller (or a computer) to control our equipment (e.g. buck converter). In such cases, we would need to use an analog-to-digital converter for our microcontroller to make use of the signals available. Most microcontrollers contain various functions such as ADCs, PWMs, timers, and even network connections which can be used in technologies like the smart devices, and Internet-of-Things. However, that is a topic we will not discuss here.

Instead of using a voltage divider to sample vo, we can use an ADC to sample vo, then program a microcontroller to process the digital signal, and output an appropriate signal for our buck converter system. This can be in the form of a voltage which represents the duty cycle D input for the PWM module. The microcontroller itself could also be the one outputting the PWM signal if it has that functionality.

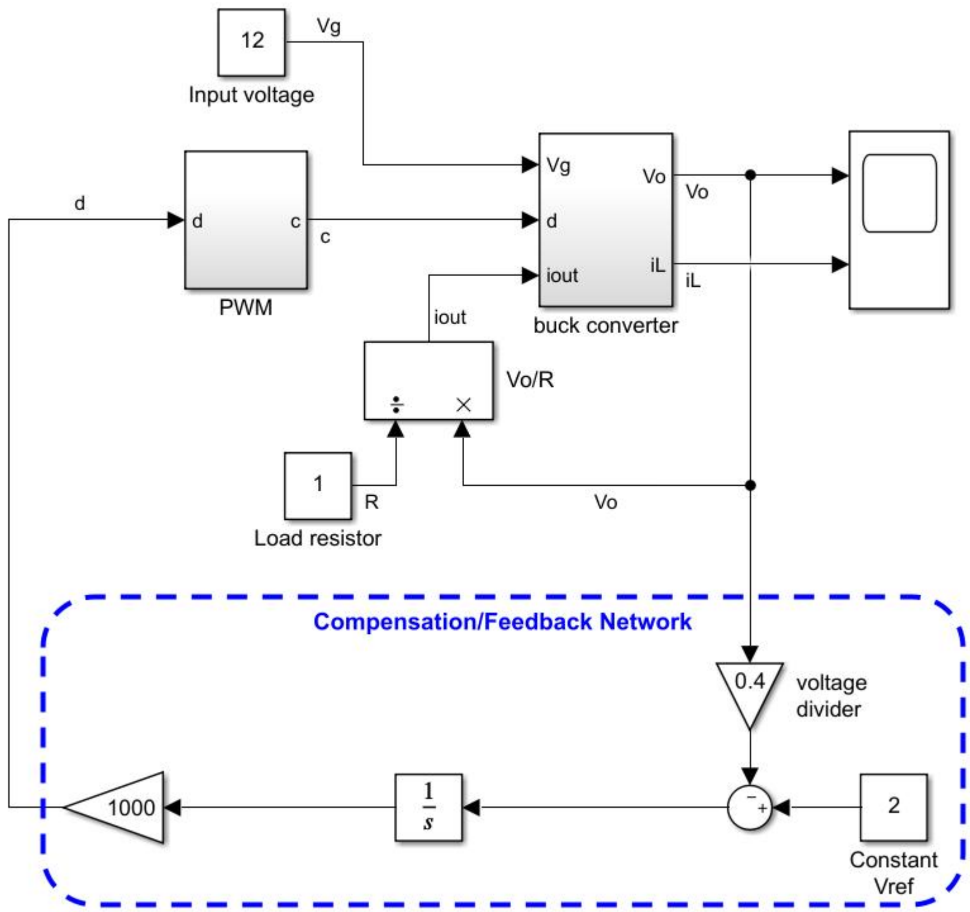

Figure 7: Closed-loops buck converter with a simple compensation network.

Take note here that the digital part of the system is mainly the compensation network, and, possibly, the PWM. The buck converter itself is not digital. It is still the same buck converter which we designed in previous experiments. The buck converter in our system diagram is merely a representation of a buck converter. In actual use, the buck converter still produces an analog output voltage which is then interfaced with an ADC to allow a digital device to process the signal. This digital device then outputs a signal to the PWM outputs back an analog signal which will be inputted to the transistor that controls the buck converter.

Learning Activity: Closed-loop Buck Converter

Exercise:

1. Modify your open-loop system to create the closed-loop system shown in Figure 7.
   (a) v_o waveform. What is the value of v_o at steady-state?
   (b) i_{out} waveform.
2. Modify only one value in the system such that the buck converter would have a steady-state output of 8V instead of 5V.

(a) Show the final block diagram of the buck converter with the correct values.

(b) What variable must be changed? What should be its value?

(c) Explain how this works. Include both equations and explanations.

(d) Probe the value of d. What is the value at steady-state? Compare this with the duty cycle value you got in the open-loop exercise to also get $v_o = 8V$. Explain.

3. How can you implement V_{ref} without using another voltage source? Note that we only have a single input voltage source V_g available to us.

4. Replace the load with $R_{load} = 10$.

   (a) Show the new block diagram of the system.

   (b) What is the value of v_o? i_{out}? Get the values at steady state.

   (c) How does the load affect v_o and i_out? Explain.

   (d) Probe the value of the duty cycle d. What is the value at steady-state? Did the value change compared to when $R_{load} = 1$? Why or why not?

5. Explain how an open-loop system would get a desired v_o, and how it differs from a closed-loop system.

6. Explain the advantages and disadvantages of using an open-loop vs. using a closed-loop system.

Other Notes

- Use at least a stop time of 0.01 unless otherwise stated.
- When showing waveforms, include important cursor measurements in the screenshot/diagram.
- When explaining answers, you can include diagrams and equations to support your explanations.
