Control Design with Buck Converter as an Example

Shivkumar V. Iyer

1 Abstract

This tutorial will examine the process of designing a controller for a power electronic converter. To begin with, power electronic converters differ vastly in topology and principle of operation which in turn results in the controllers implemented to achieve their desired operation to be drastically different. However, Proportional Integral (PI) controllers are the most commonly used controllers for most power electronic applications due to their simplicity of implementation and method of operation. In this tutorial, we examine the software implementation of a PI controller in an embedded controller such as DSP or FPGA rather than a hardware implementation using analog components. The purpose of this tutorial is to describe how the gains of the PI controller can be chosen by examining the output of the converter.

Usually, any control design will follow a mathematical process. The open loop transfer function of the converter is derived from which a frequency response characteristic is obtained. By deciding the type of controller (in this case a PI controller), the type of transfer function of the controller is known. Knowing the transfer function of the converter and the controller, the closed loop transfer function of the system is determined. The gains of the controller can be adjusted such that the frequency response characteristics of the closed loop system result in a stable system. Such a design process is highly recommended as this results in a rigorous design process that can be adjusted if the converter were to be scaled up or down in power rating. There are many commercial and free software that have in-built functions for designing controllers.

However, another method exists for choosing control gains - trial and error method. This method simply involves choosing different gains of the controller and observing the performance of the converter. Such a random method may seem to be inappropriate for designing any practical system. However, many systems in practice involve such a trial and error method to some extent. It should be noted that when a controller is designed, the parameters of the converter are never exactly equal to

what they would be when the converter is fabricated. Component tolerances, temperature and aging will cause the parameters to drift with time. Therefore, even if a controller is designed mathematically, it is recommended to use the trial and error method to change the gains close to the designed values to check for improved operation.

This tutorial will examine how a PI controller can be designed for a buck converter purely by trial and error method. By using simulation as a technique to design controllers and also learn about the method of operation of the controller, such a trial and error method does not have the risks of implementation in hardware. The objective of the tutorial is to understand how converter output, control output and actual converter operation can be used together to design a controller. Even after designing a controller by such a trial and error method, it is advisable to perform a frequency response analysis to check the stability margin of the system.

The outline of the tutorial is as follows:

1. Description of operation of a buck converter
2. Closed loop control scheme for a buck converter
3. Tuning of the PI controller

This tutorial is accompanied by a number of simulation cases. It is recommended to try out the simulation cases while reading the tutorial.

About the software: Python Power Electronics is a free and open source circuit simulator for power electronics and power systems professionals. It is available under the BSD 3.0 license agreement which is available with this tutorial and in every case as a LICENSE.txt file. Users of this tutorial are advised to read the license file. It should be noted that this tutorial is for learning and educational purposes. Implementation of the examples cited in this tutorial directly in a prototype cannot be guaranteed to work and the readers are responsible for taking precautions to prevent damage to their equipments and injury to themselves. For any queries, feel free to contact me at pythonpowerelectronics@gmail.com or visit the project website http://www.pythonpowerelectronics.com/.

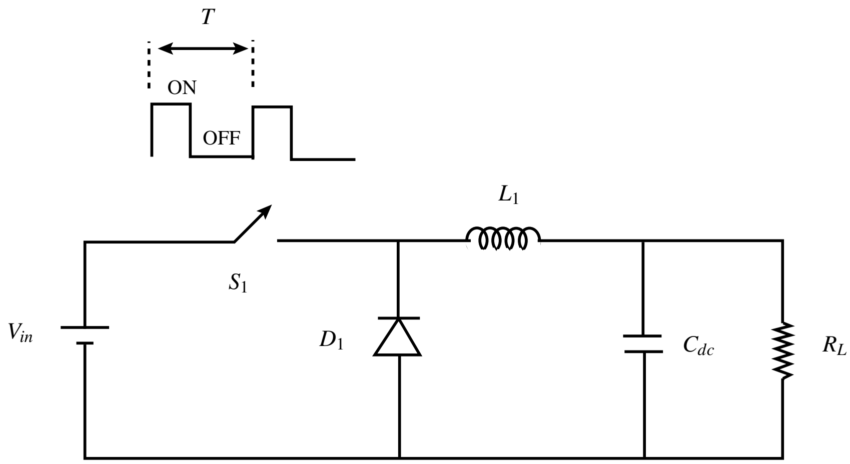


Figure 1: Buck converter

2 Buck converter

The above figure shows the topology of the buck converter which consists of a power device such as a MOSFET or IGBT represented as ideal switch S_1. The capacitor C_{dc} forms the output dc bus across which the load R_L is connected. Typically such a converter is operated at a constant switching frequency. This implies that in the time period T corresponding to the switching frequency, there is one turn-on and one turn-off of the device S_1. This is shown in the switching waveform above the switch S_1 in the figure. How this will be achieved comes later. This converter is called a buck converter because the output voltage across capacitor C_{dc} is always lower than the input voltage V_{in}. This is one of the simplest power converters which is why it is apt for examining control design as the operation of the converter is fairly simple as will be described next.

This tutorial will use the circuit file buck_conv.csv with parameters in buck_conv_params.csv. The control files are conv_control.py and modulator.py with their variables in conv_control_desc.csv and modulator_desc.csv respectively.

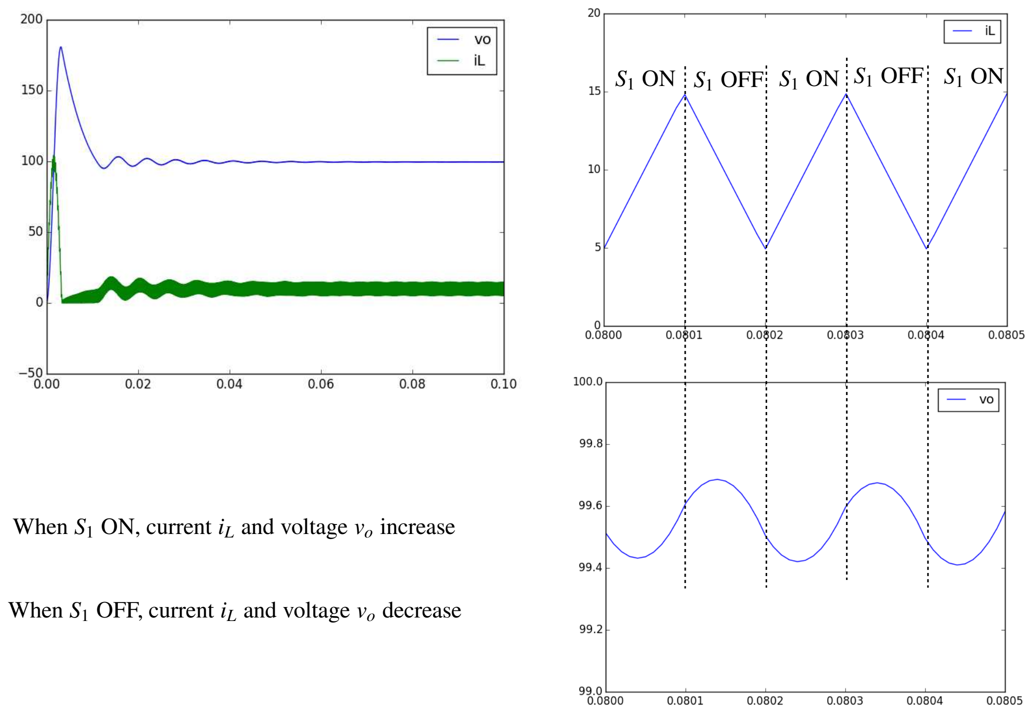


Figure 2: Operation of buck converter

The figure on top shows the method of operation through a simulation when the switch S_1 is ON for half the time period corresponding to the switching frequency. This is done by setting the variable duty_ratio to 0.5. When the device S_1 is turned on, the input voltage V_{in} appears across the the diode D_1, reverse biases it and therefore D_1 turns off. The difference between the input voltage and the output voltage appears across the inductor L_1 and so the current i_L through the inductor begins to rise. Since, the current i_L rises, the capacitor C_{dc} charges and the output voltage v_o rises. When the switch S_1 turns off, the current i_L through the inductor will need to continue flowing and therefore D_1 turns on so that the current i_L can freewheel through it. Since, the stored energy in the inductor is dissipating in this state, the current i_L decreases. Since, i_L decreases, the output voltage v_o also decreases since the load R_L is connected across the output and discharges the capacitor C_{dc}. The zoomed waveforms on the right show current i_L and voltage v_o during the on and off state of the switch. There is fairly vast amount of theory and literature related to this converter and the reader is encouraged to read. This tutorial will provide only the description needed to design the controller.

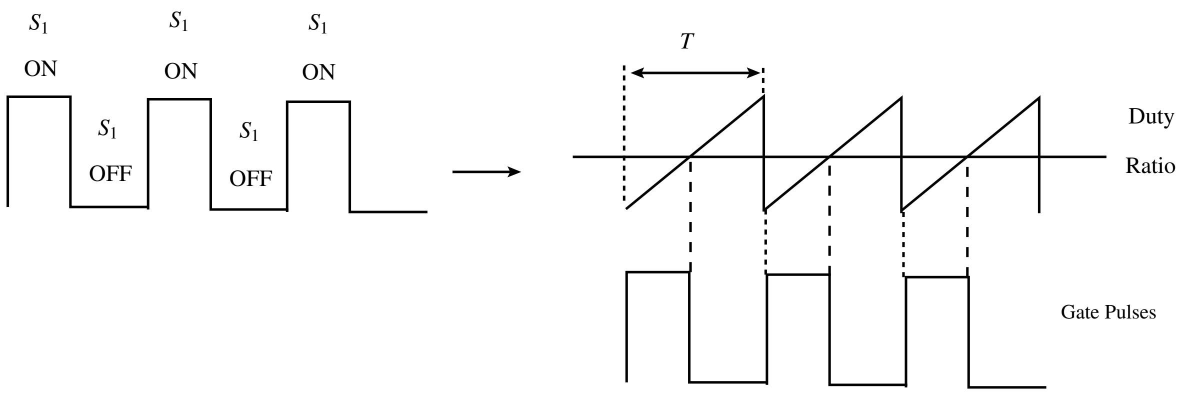


Figure 3: Pulse Width Modulation

From the previous description, it is evident that by increasing the ON time of the switch S_1, more energy is transferred from the input to the output and therefore the output voltage increases. As stated before, the switching frequency of the switch S_1 remains constant which means there is exactly one turn ON and one turn OFF during a switching time period (let us call this T). So, when the ON time of the switch increases, the OFF time of the switch decreases. The ratio of the ON time of the switch to the time period T is called the duty ratio d. This duty ratio will therefore be restricted between 0 and 1. The figure above shows how the ON time of the switch is varied. On the left is shown a pulse train that becomes the gate signal which controls the switch S_1. The right shows how this pulse train is obtained. Since, the switching frequency is constant, a sawtooth waveform on magnitude 1 and the same switching time period T is synthesized. This waveform is called a carrier waveform. This carrier waveform is compared with the duty ratio as shown. When the duty ratio is greater than the carrier, the switch S_1 is ON and when it is lower than the carrier, the switch S_1 is OFF. By increasing the duty ratio, the ON time of the switch will increase and by decreasing it, the ON time will decrease. This method of varying the ON time is called Pulse Width Modulation since the width of the gate pulses is varied by comparing a signal with a fixed frequency carrier waveform.

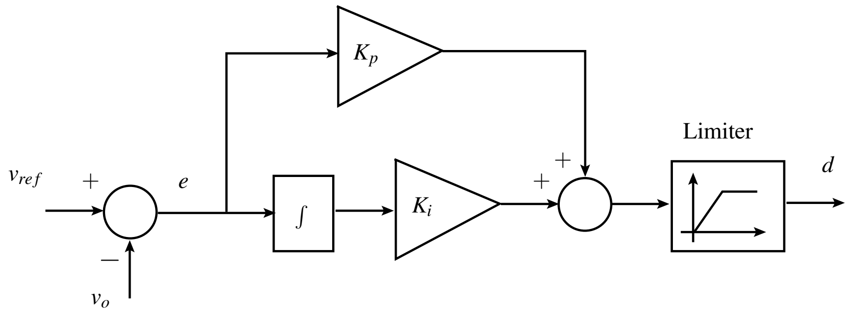


Figure 4: Voltage control

3 Closed loop control

In the previous simulation, we fixed the duty ratio to 0.5 which implies there was no control or rather "open-loop". The figure shows how this duty ratio is obtained when we close the loop and generate the duty ratio with respect to the measured output voltage v_o. We decide a reference voltage v_{ref}. This reference has to be lower than the input voltage V_{in} as it is a buck converter. The output voltage v_o is measured and fed back to be compared with the reference. The error e between the reference and the measured voltage $v_{ref} - v_o$ is calculated. This error is fed to a PI controller. One part of the controller simply multiplies the error by the proportional gain K_p. The other part integrates the error and then multiplies by the integral gain K_i. These two components are added together. The sum is then passed through a limiter since the duty ratio d should be restricted between 0 and 1.

If the output voltage v_o is lower than the reference v_{ref}, the error e is positive. The proportional component will yield a positive value whereas the integral component will yield an increasing value since the error is integrated. The result would be that the duty ratio d increases, which in turn will increase the ON time of the switch and therefore output voltage will increase. When output voltage is equal to the reference, the error is zero, the proportional component will be zero whereas the integral component will remain constant. The duty ratio which is their sum will be constant and steady state is achieved.

In order to be able to choose the control gains K_p and K_i, we need to understand the role they play in how the converter will behave as they change. The proportional component produces an output for any deviation of the output voltage v_o from the reference v_{ref}. The integral component integrates the error and generates an output. In any system, continuous changes in the input to the system (in this case the duty ratio) are not desirable as that results in jitter and oscillations. Therefore, the proportional gain should be kept as small as possible. The integral gain on the other hand will produce a relatively smooth output for errors in the output voltage. Therefore, the integral component should be dominant in a PI controller.

The integral component due to the integral action will always produce a delayed response to any error $v_{ref} - v_o$. If the proportional component is negligible, the control action to any error in the output voltage will be delayed. This will result in oscillatory behavior in steady state as there is no component that produces an immediate action to restore the output. The proportional component is responsible for this immediate action which damps oscillations.

In control theory, you will find reference to the Proportional-Integral-Derivative (PID) controller. The derivative component will decrease the oscillations and reduce the settling time of the closed loop system. However, using any derivative component in a power converter circuit must be done with caution. A power converter circuit always contains switching ripple and high frequency harmonics. If these components are fed to the derivative control, the derivative action will amplify high frequency signals. Therefore, if a derivative control is to employed, filtering must be performed to ensure that high frequency components are not fed to the derivative part. In most cases, a PI controller without any derivative is sufficient for adequate performance and this tutorial will examine the tuning of a PI controller.

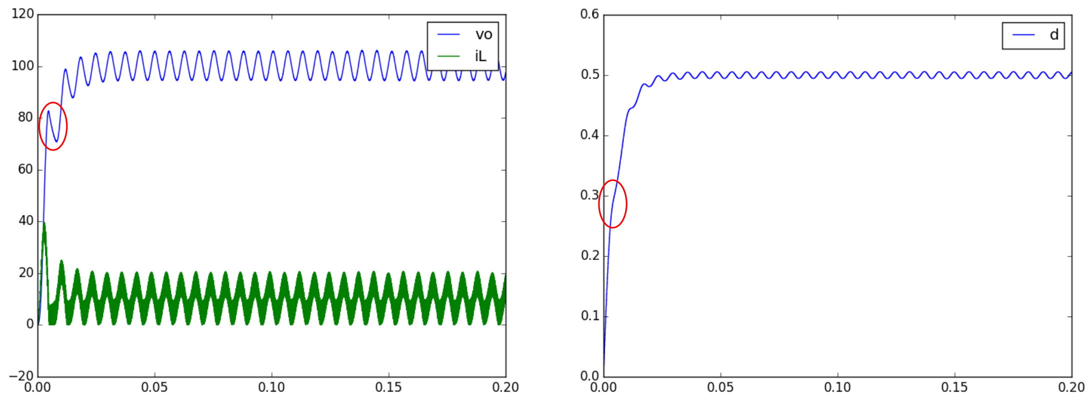


Figure 5: Control tuning $K_i = 1$, $K_p = 0$

4 Tuning the PI controller

To tune the controller, refer to the control file conv_control.py. A reference of 100 V is chosen with $V_{in} = 200$V. To start the tuning process, we will start with the integral gain K_i and set the proportional gain K_p to be 0. Since we have to start with a random number, let us start with $K_i = 1$. The figure above shows the simulation result with these control values. As expected, with only an integral control, the output voltage has oscillations and the output voltage in steady state does not settle at 100V. However, there is another inference that is not immediate. Look at the oscillations marked in red in the figure. These are the first oscillations and they occur below the reference of 100V. This implies that even with oscillations, the integral gain could be increased further so that the first oscillation occurs around 100V. The idea is that the integral gain should be so chosen initially that the output rises to the reference directly and then oscillates around it.

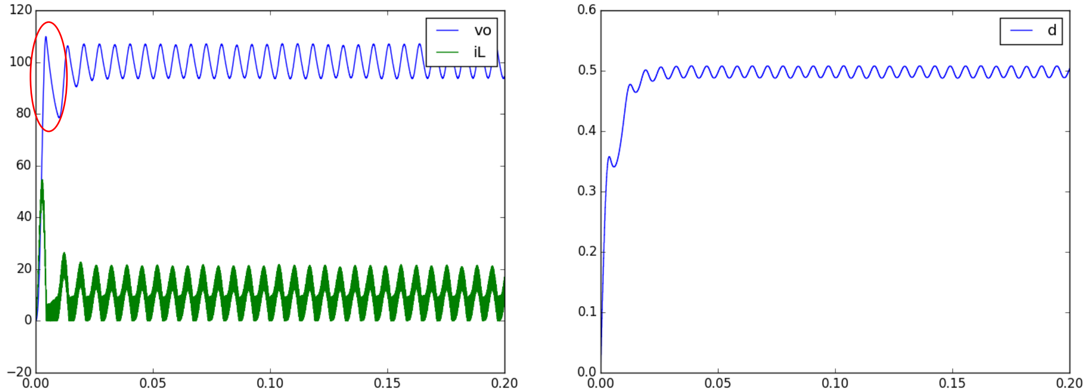


Figure 6: Control tuning $K_i = 1.5$, $K_p = 0$

The next step, let us increase the integral gain further. Let us choose $K_i = 1.5$ while proportional gain remains zero $K_p = 0$. The figure above shows the simulation result. With this integral gain, the first oscillation occurs around 100V with the output voltage rising up to the reference directly. Now, to tackle the other problem - oscillations in steady state. For this, we now add the proportional component in the controller. Typically, the proportional gain can be much smaller than the integral gain as the main purpose of K_p is to damp oscillations by providing immediate action to any error.

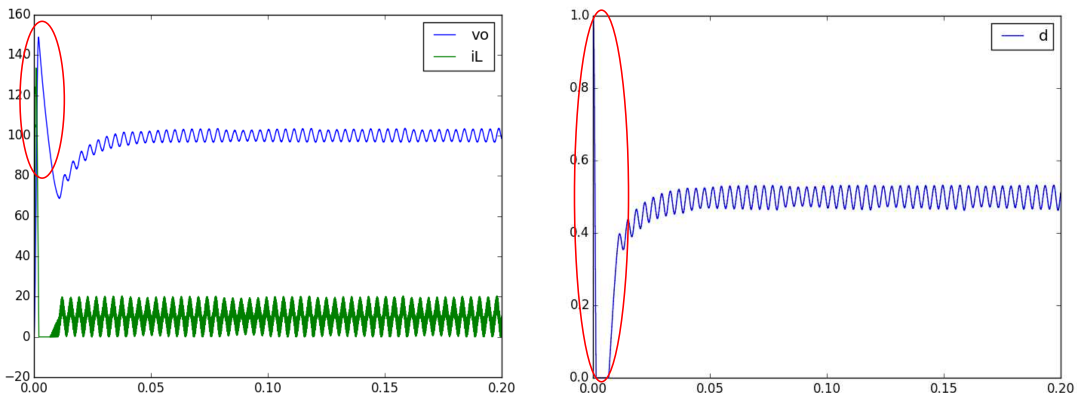


Figure 7: Control tuning $K_i = 1.5$, $K_p = 0.01$

Let us start with a proportional gain $K_p = 0.01$ which is much lesser than the integral gain of $K_i = 1.5$. The figure above shows the result. The converter behavior has changed. To begin with there is a large overshoot in the output voltage. When looking at the duty ratio, it can be seen to saturate at 1 in the beginning. The duty ratio then makes a nose dive and hits 0. This is a clear indication of too large a proportional gain and this means the proportional component has become dominant. So, we need to decrease the proportional gain and by quite an amount.

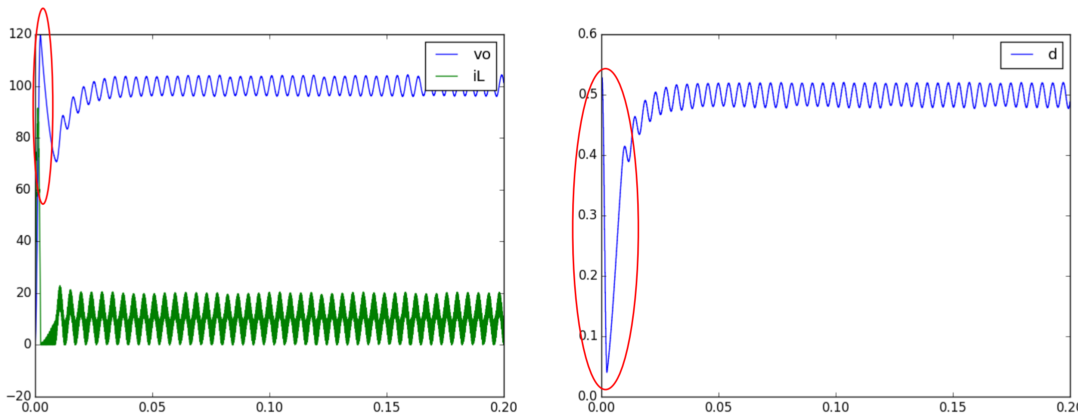
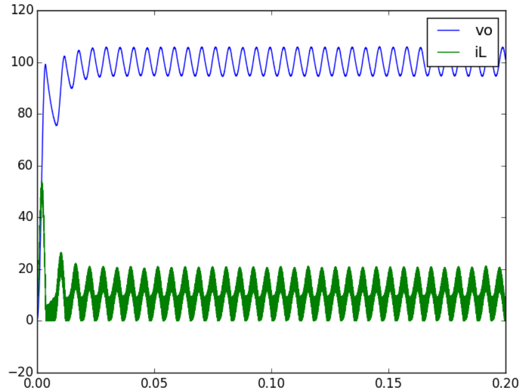


Figure 8: Control tuning $K_i = 1.5$, $K_p = 0.005$

Let us reduce the proportional gain to $K_p = 0.005$ while maintaining the integral gain to $K_i = 1.5$. The figure above shows the result. The duty ratio does not saturate at 1 and does not hit 0 after that but still makes a nose dive. This still implies that K_p is too large. The objective of control should be to achieve a smooth transition from 0 to 100V or for that matter from any level to any other level were the reference to change or the load to change. Therefore, a sign of a large jump in the control action which in this case is the duty ratio is a sign that the gain needs to be further decreased.



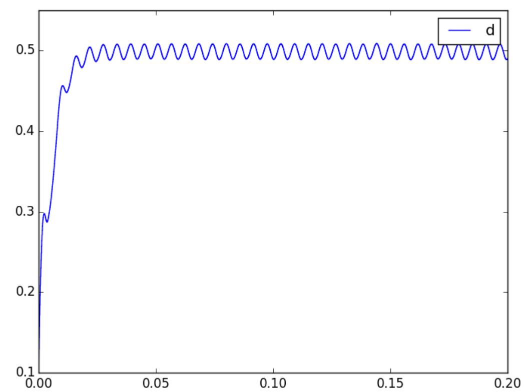


Figure 9: Control tuning $K_i = 1.5$, $K_p = 0.001$

Now we decrease the proportional gain to $K_p = 0.001$. The figure above shows the simulation result. Now the result shows that the drastic jumps and nose dives in the duty ratio are no longer present. The duty ratio climbs gradually and the output voltage does not have large overshoots. However, there is one problem that adding a proportional component has not solved - the problem of oscillations. Oscillations are lower in magnitude as compared to the purely integral control case when you compare with the first two simulation results. But the oscillations are still too large and noticeable to be acceptable. If we increase K_p, there will again be jumps in the duty ratio and the overshoots in output voltage. So what can be done?

In such a case, let us examine what is happening in the steady state. There are oscillations in both the output voltage and the duty cycle. In steady state, you would expect the control action to be smooth, which means the duty ratio will be almost constant. The output voltage will be regulated close to the reference. There will be a ripple in the output voltage but if the capacitor C_{dc} is large enough, the ripple will be negligible.

However, when there is a ripple in the control action in steady state, this usually means the controller is trying to do too much. If the controller should ignore minor fluctuations in output voltage, it is trying to produce a control action (change in duty ratio) to eliminate that fluctuation. And this is because the gains are too high. So the next question, which gain to decrease? Both gains. Because the ratio of K_i/K_p has produced the smooth response that we want. We need to decrease both gains by the same factor.

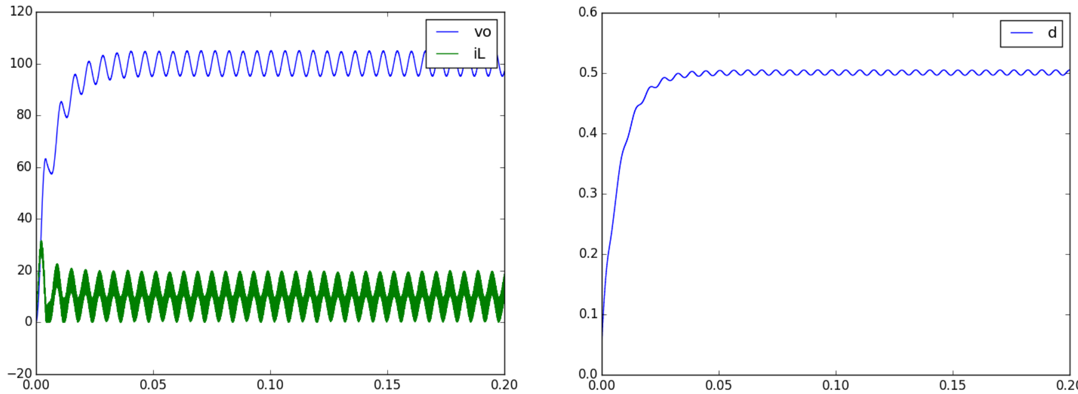


Figure 10: Control tuning $K_i = 0.75$, $K_p = 0.0005$

Let us decrease both K_p and K_i by half. So $K_i = 0.75$ and $K_p = 0.0005$. The figure above shows the result. The fluctuations in steady state can be seen to have decreased. However, the response is now sluggish as compared to the previous control gains. This the price we must pay for improved steady state - slower transient performance. Moreover, though the oscillations in steady state have decreased, they are still noticeable and not acceptable. We must decrease the control gains even further.

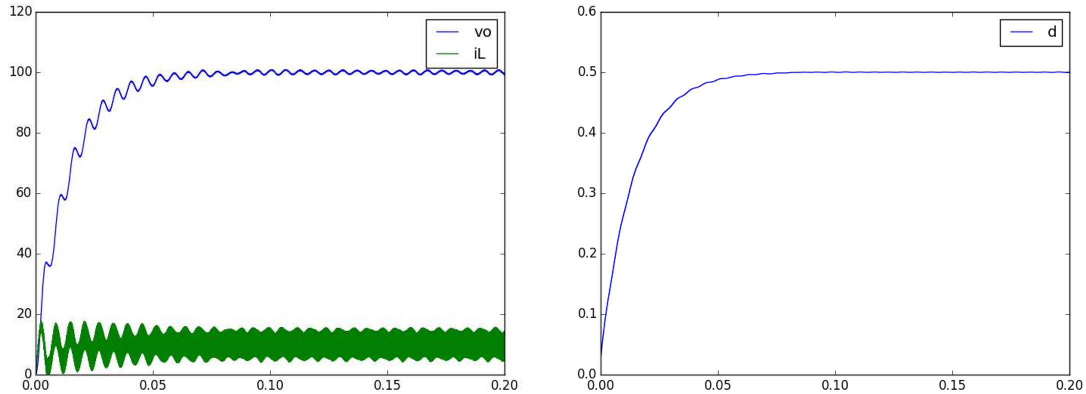


Figure 11: Control tuning $K_i = 0.375$, $K_p = 0.00025$

We halve the control gains again to $K_i = 0.375$ and $K_p = 0.00025$. The figure above shows the simulation result. The oscillations in the output voltage in steady state have drastically decreased. The duty ratio is almost a constant in the steady state which is what we were aiming for. And also the response is more sluggish that the previous case which is also expected from decreasing control gains. We have been changing control gains by a factor of 2. The reader is encouraged to further tune the control gains minutely to check for improved performance. This tutorial will end with this result.

It should be noted that the design of the converter is not addressed. This means the choice of switching frequency, value of inductor L_1, value of capacitor C_{dc}. The reader is encouraged to read separately on these topics. Moreover, a buck converter such as the one above has a continuous mode of operation and a discontinuous mode of operation. Specialized analytical techniques and control strategies have been developed for both methods of operation.

5 Conclusion

It is not at all necessary to follow steps one after the other as stated above. These are the steps that I intuitively follow if I were to design control gains by trial and error method. In the end, the readers need to develop their own “modus operandi” and you are always encouraged to experiment. Moreover, as the nature of the circuit changes, the transfer function of the open-loop system may be such that a PI controller may not work at all. For converters with higher-order filters, advanced multi-variable control strategies are sometimes used and these will described in the upcoming tutorials.

The main objective behind these tutorials is to be able learn the working of power converters through simulations. This tutorial therefore starts with a very simple buck converter. However, in practice, even for buck applications, other topologies may be used. It is important to map the operation of the converter with the control action produced to avoid tuning controllers arbitrarily and getting frustrated in the process. The reader is encouraged to plot the inductor current waveforms and zoom in on the transients to examine how the converter operates.