ISSUE: November 2025

SPICE Simulation Of A Digitally-Compensated Buck Converter

by Christophe Basso, Future Electronics, Toulouse, France

The digital control of a switching converter offers many advantages over its analog counterpart. For instance, poles and zeros placed in the compensator no longer depend on passive components whose variability (tolerance, temperature, age) will affect the compensation strategy. Also, dynamically changing the positions of these poles and zeroes offers an interesting feature to ensure an optimum transient response over a wide variety of operating conditions.

For those of us coming from the analog world, the compensation can be thought out in the frequency-domain first, then translated into the discrete-time domain via an appropriate mapping process. Once the translation is done, it is worth validating the calculated coefficients and confirming the compensated converter delivers the expected transient response. To that end, LTspice offers a simple and efficient way to verify these calculations are correct before coding the compensator using a macro I've written to generate the filter coefficients.

However, LTspice can also help with other aspects of designing the digital compensator. For example, it can be used to extract the control-to-output transfer function, which is required to design the compensator. A recently added FRA function has made it easier to do so. Then, once the transfer function is obtained and a compensator design is completed, it can be simulated in the intended power circuit in LTspice to verify it gives the desired performance.

Then, using the previously mentioned macro, LTspice can generate the coefficients for the digital filter equivalent of a type 3 compensator. Designers can use this macro to simulate the performance of their digital filter before implementing it in software.

I demonstrate these capabilities of LTspice with a compensator design example for a buck converter. Along the way, I'll mention other methods for obtaining the transfer function, tips for compensating the converter, and pointers for testing the digital compensator.

Obtaining The Buck Converter Transfer Function

Before attempting to compensate a power stage—whether it is linear or switching—you must obtain the control-to-output transfer function of this power stage. By transfer function, I mean the small-signal ac response obtained by applying a stimulus to the control input—the feedback pin for example—and observing how this stimulus propagates in the circuit to generate the response, which could either be taken as an output voltage or current.

The complex ratio of the response to the stimulus is the transfer function of the network under study which will be displayed as a Bode plot. With this graph on hand, you can think of a compensation strategy to meet specific performance and stability goals.

This transfer function can be obtained in different ways:

• Small-signal modeling of the converter using an averaged model: you linearize your switching converter using the PWM switch model[1] for example, and carry out the analysis over the circuit. It can be long and tedious but, fortunately, literature abounds in which these expressions have already been determined for you.[2]

• Simulation with an averaged model: it is possible to implement an averaged model with a SPICE simulator and graph the transfer function in voltage- or current-mode control.[3] You can cover different scenarios (input/output conditions) and test the sensitivity to parasitics. An averaged model excludes the switching component and simulates fast.

• Simulation with a switching circuit: SIMPLIS is the typical tool suited for extracting the small-signal response of a switching converter. With its piece-wise linear modeling approach (PWL), the simulation speed is extremely fast and it works for all types of control, including resonant converters.[4]

• Bench measurements: you build a prototype and extract its small-signal response using a frequency-response analyzer or FRA. It can take time to collect the components, assemble the converter and make it work correctly. Nevertheless, this option is mandatory at the end of the development cycle, to validate your theoretical analysis. In recent years, making Bode plot measurements has gotten easier and more accessible with the introduction of low-cost FRAs and with the inclusion of the Bode measurement function in select oscilloscopes.

The transfer function of a buck converter operated in voltage-mode control (VM) is that of a second-order system whose quality factor Q depends on the circuit's efficiency. For instance, the ohmic paths are sensitive to the RMS currents and incur conduction losses: inductor and capacitor series resistance (respectively noted r_L and r_C), the transistor $R_{DS(ON)}$ and the free-wheel diode dynamic resistance r_d.

The switching process also brings losses with the transistor and the diode, if we consider its recovery process or parasitic capacitance affecting efficiency. Finally, magnetic losses and the load resistance R_L also affect the quality factor. The control-to-output transfer function for a VM buck operated in the continuous conduction mode (CCM) is given below:

$$\frac{V_{out}(s)}{V_{err}(s)} = H_0 \frac{1+\frac{s}{\omega_z}}{1+\frac{s}{\omega_0 Q}+\left(\frac{s}{\omega_0}\right)^2} \quad (1)$$

The dc gain H_0 is set by the input voltage V_{in} and the modulation ramp peak voltage V_p, if we neglect the inductor dc resistance r_L and the load resistance R_L:

$$H_0 \approx \frac{V_{in}}{V_p} \quad (2)$$

The zero is classically set by the output capacitor and its equivalent series resistance (ESR) r_C:

$$\omega_z = \frac{1}{r_C C_{out}} \quad (3)$$

while the double poles depend on the LC network and bring a resonant frequency ω_0 approximated as

$$\omega_0 \approx \frac{1}{\sqrt{L_{out} C_{out}}} \quad (4)$$

The quality factor Q is defined by a complicated expression which also simplifies when neglecting r_C and r_L:

$$Q \approx R_L \sqrt{\frac{C_{out}}{L_{out}}} \quad (5)$$

When the converter enters the discontinuous conduction mode (DCM), the system remains a second-order system but is heavily damped. Its dc transfer characteristic also changes and now involves the constitutive elements of the converter (inductance and load resistance), as well as the switching frequency.

However, in that regard, implementing synchronous rectification offers an advantage. Not only does it keep the same dc transfer characteristic ($V_{out} \approx NV_{in}$) in CCM and DCM, it also preserves the same control-to-output transfer function from one mode to the other, simplifying the control strategy.

Extracting The Transfer Function—An Example

As listed above, several options are available for this exercise. We can start with SIMPLIS which offers the fastest way to obtain an ac response from a switching cycle-by-cycle circuit. Fig. 1 describes a VM buck converter operated in open-loop[4] and it offers a simple way to extract the power stage ac response.

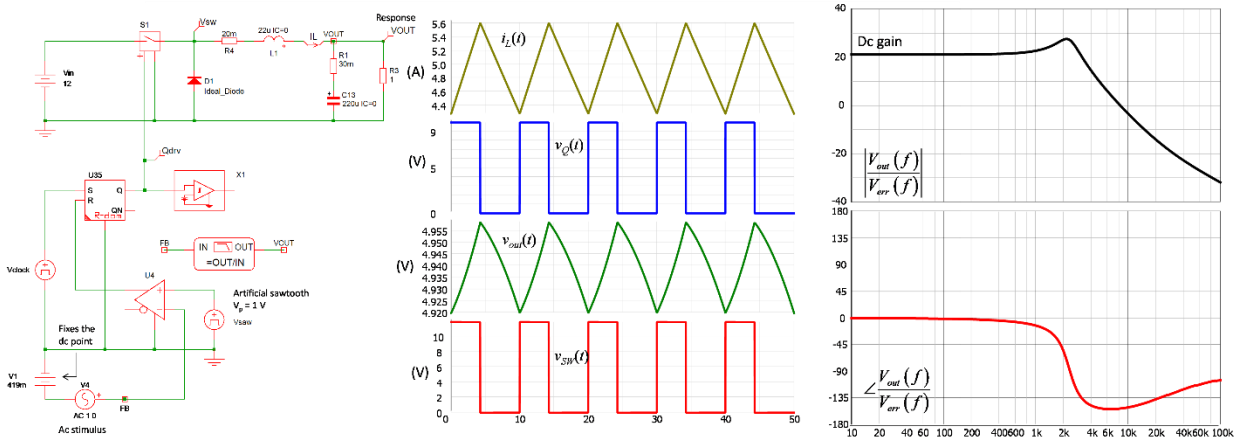


Fig. 1. A simple switching circuit featuring a pulse-width modulator and a perfect switch lets you extract the control-to-output transfer function in a few milliseconds with SIMPLIS.

When you start the simulation, SIMPLIS runs its periodic operating point (POP) to quickly converge towards a steady-state solution where all state variables have reached their final values, like ≈5 V for V_{out} in this example. Without the POP, you would need to run a cycle of several tens of milliseconds to let the converter stabilize, prior to performing the ac analysis.

In this example, the run took less than 1 s on my machine to complete the POP and the ac sweep of the converter. The Bode plot in the right side in Fig. 1 confirms a dc gain slightly above 20 dB – $20\log\left(V_{in}/V_p\right) \approx 21.6$ dB with a resonant frequency above 2 kHz. In the ac-sweep mode, the source amplitude is automatically adjusted by SIMPLIS during the simulation, ensuring the best signal-to-noise ratio.

A New Frequency Response Analyzer

For some time, LTspice has offered a way to ac-sweep switching circuits through the usage of a complicated set of macros placed on the simulated circuit. However, the developers have recently improved the process by including a neat frequency-response analyzer (FRA) symbol that you can tweak to meet your goals.

In this analysis mode, a sinusoidal source drives the control input of the converter—the feedback or control pin—to modulate the control variable (the duty ratio in VM or the peak inductor current in CM) and see how it propagates through the circuit to generate a response. Unlike with SIMPLIS, you have to care about the stimulus amplitude for several reasons.

First, you're dealing with a switching circuit and the modulation signal—varying from 10 to 100 kHz for example—is drowned in the noise and must be of sufficient amplitude for efficient extraction on the output. Secondly, injecting a signal into a network implies that the circuit remains linear during analysis and must not saturate. This imposes an upper limit on measures taken to satisfy the previous requirement.

Then, there's the variability of loop gain T. In a dc-closed-loop circuit, injecting a modulation signal is seen as a perturbation and the control circuit will oppose or reject the modulating signal, naturally reducing its presence on the output. However, the loop gain *T* is usually high at low frequency and reduces as the frequency approaches the crossover point, beyond which the loop gain drops below unity.

The ability to reject the perturbation depends on *T*: with a pole at the origin in the compensator (high loop gain at dc), you will have to inject a significant stimulus amplitude at low frequencies and reduce the level as f_C approaches. If not, either the low-frequency part of the Bode plot is polluted by large amplitude/phase swings because of insufficient level, or the graph exhibits strange shapes like jumps in magnitude or phase, which are artifacts caused by a saturated converter.

Finally, when the loop is *physically* open, there are fewer constraints on the modulation amplitude which can keep constant as long as the circuit operates in a linear way. You can check this fact by slightly adjusting the source amplitude and making sure the plot does not significantly change in shape. If it does, you are overmodulating the input and must reduce the amplitude.

Fig. 2 shows a typical LTspice setup[5] for the buck converter operated in voltage-mode control and using the FRA symbol. Source V2 provides the bias for setting the output at 5 V which corresponds to a duty ratio of 44%, considering the forward drop of the freewheel diode.

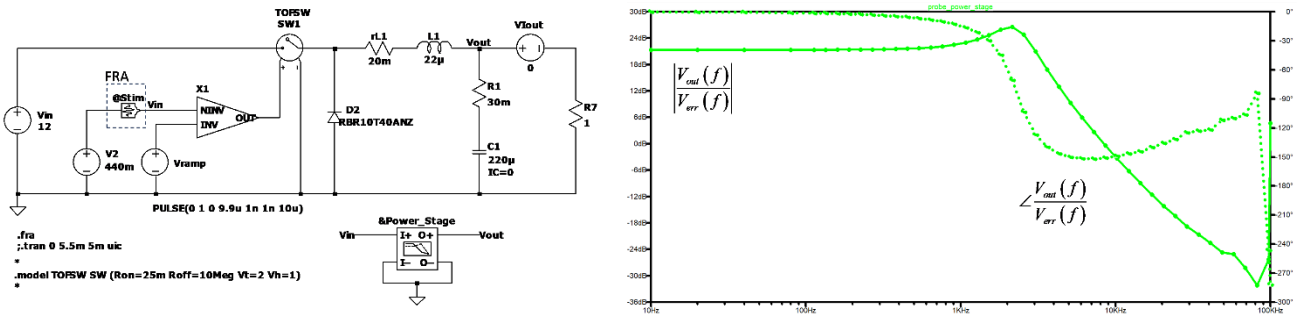


Fig. 2. The FRA symbol and its associated Bode box are installed in this simple buck converter modeled in LTspice on the left. The right-side picture is the control-to-output transfer function.

The ac response matches that of SIMPLIS and was delivered in 60 s, which is acceptable. The stimulus source is set at 100 mV while the 440-mV dc bias gives a 44% duty-ratio.

A Compensation Strategy

How are we going to place the poles and zeroes now that we have the power stage transfer function? We start with the crossover frequency selection. As described in reference [6], you select a crossover frequency based on the desired transient response but, very often, the topology imposes limits.

For instance, with a voltage-mode buck converter, you want the crossover frequency to be at least 3 to 5 times the resonant frequency defined by equation (4) ($\omega_0 \approx 1/\sqrt{L_{out}C_{out}}$). This is because you need loop gain for compensating oscillations naturally arising after a transient event over a peaky output impedance.

Should you *wrongly* choose to cross over before f_0, then dc regulation would be ensured, but *LC* oscillations would show up on the output as I illustrated in slide 143 of reference [7]. For a boost converter, also operated in voltage-mode control, you still need to cross over after f_0 (which now depends on the duty ratio) but the right-half-plane zero sets the upper limit to 20% of its lowest position.

Coming back to our case, because we run the buck converter in synchronous mode, there is no mode transition and the circuit remains in CCM even at no load (the average inductor current is 0 A in this case): the power stage transfer function does not change and the compensation is simpler. We will place two zeroes around f_0, a first pole at half the switching frequency and the second at a position where we meet the phase margin goal.

From the LTspice ac simulation, we read the Bode plot and see that for a resonant frequency around 2 kHz, a crossover point at 10 kHz looks like a possible choice, provided the op-amp offers a comfortable gain-bandwidth product (GBW). We now read the magnitude and phase at 10 kHz:

• |H (10 kHz)| = -3 dB

• ∠H (10 kHz) = -150°

The compensator will then be tailored to exhibit a 3-dB gain at 10 kHz and a phase boost[6] equal to:

$$boost = PM - PS - 90° = 70° + 150° - 90° = 130° \qquad (6)$$

To confirm this approach, I have added the compensation circuit with a synchronous rectifier, driven by a 150-ns dead-time generator. The circuit appears in Fig. 3 and you can see the automated macro on the right side.

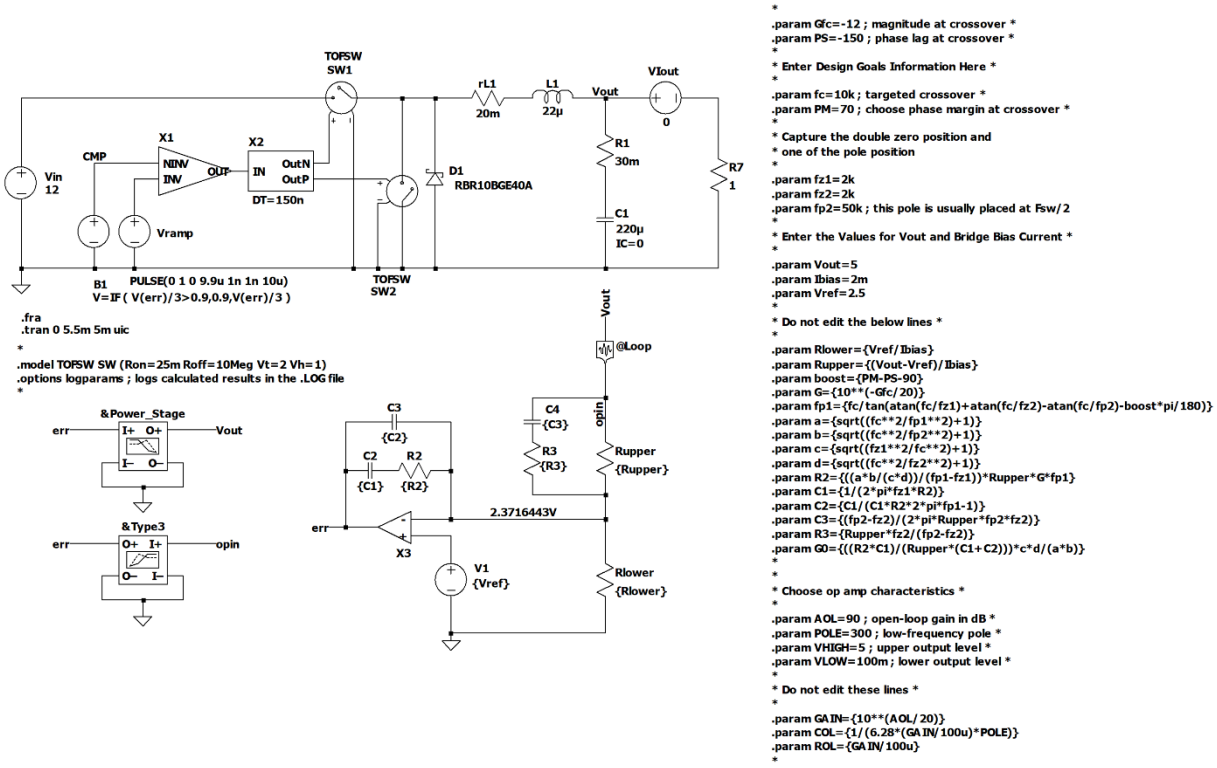


Fig. 3. The LTspice buck converter is supplemented with a type 3 compensator whose component values are calculated by the right-side macro.

The error amplifier voltage drives the pulse-width modulator via a divide-by-3 source, B_1, to force a larger dynamic on the op-amp output but also clamp the maximum voltage below 1 V (the sawtooth voltage peak V_p is 1 V). As a result, a 0.5-V bias voltage at the CMP node (e.g. for a 50% duty ratio) will imply an error voltage of 1.5 V, well within the linear range of the op-amp. This is a classic in design and you can also find this ratio in the UC384x controllers, supplemented by two diodes in series for shifting the op-amp bias point even higher.

The attenuation brought by B_1, implies a 9.5-dB decrease in the return path, bringing the new control-to-output transfer function magnitude at 10 kHz to -12.5 dB. The phase remains unchanged at 150°. These data are entered in the macro, together with a crossover goal of 10 kHz and a phase margin of 70°. All the component values are stored in the .log file generated after the simulation has run. The results are shown in Fig. 4.

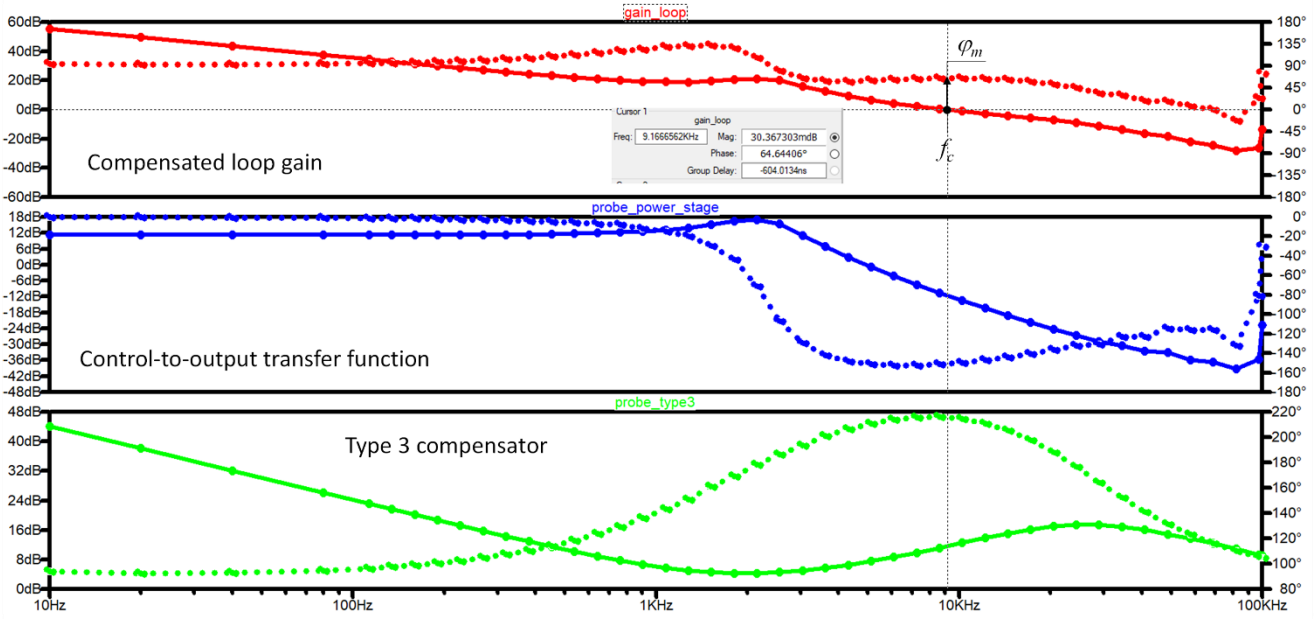


Fig. 4. The Bode plots confirm a crossover frequency approaching 10 kHz, with a comfortable phase margin.

The crossover is approaching our 10 kHz goal and the phase and gain margins are excellent. A transient test confirms the stability is good (Fig. 5).

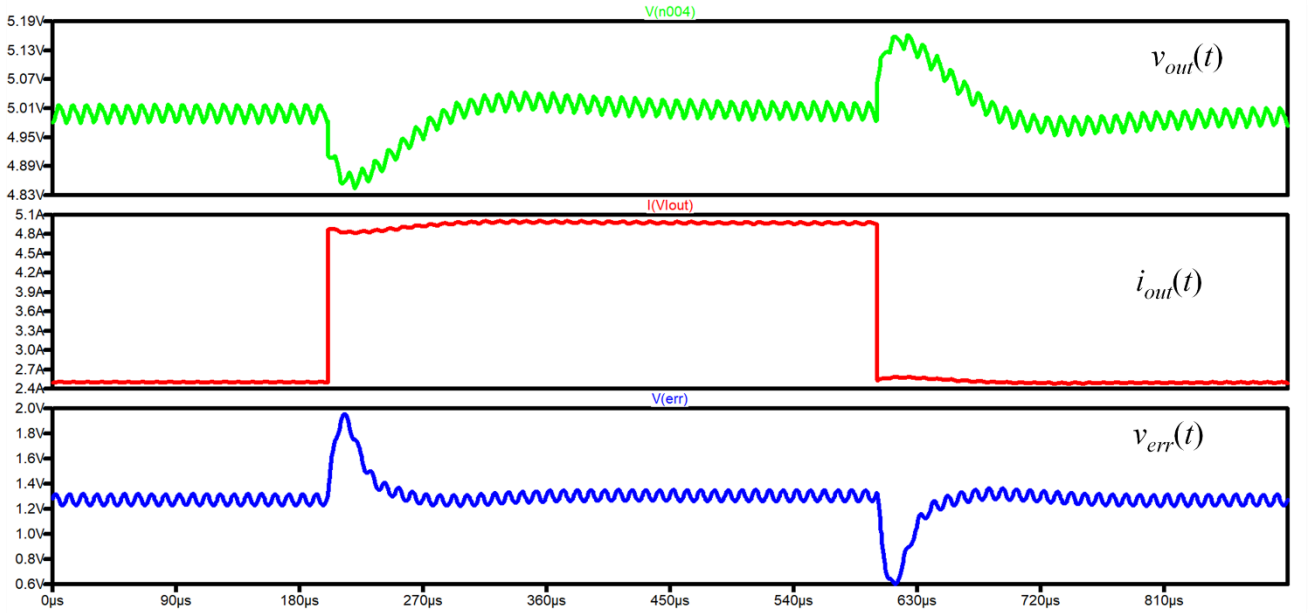
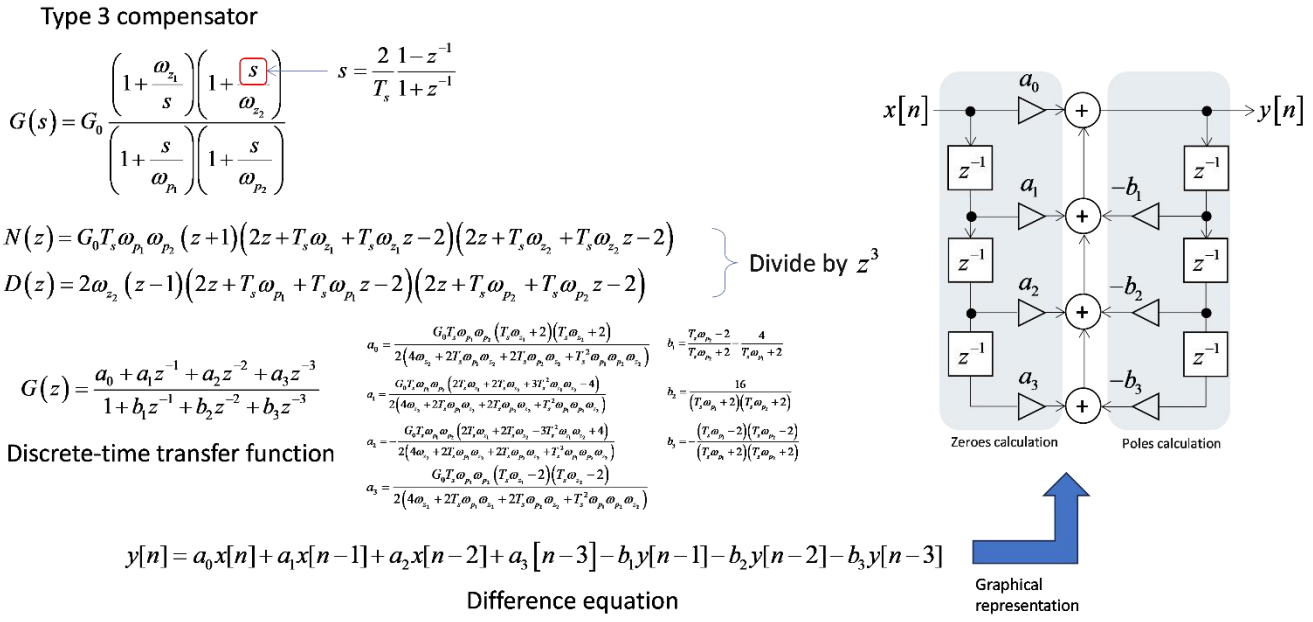


Fig. 5. The load is stepped from 50% to 100% with a 1-A/µs slope and the output voltage drops by ≈160 mV.

A Digital Compensator

Now that we have the correct set of poles and zeroes implemented in the analog way, we can translate them into the digital domain. Different structures exist to build the equivalent of a type 3 compensator. In the analog domain, the type 3 structure features two zeroes and three poles. When translated into the discrete world, this structure becomes a three-pole three-zero (3P3Z) filter which appears below (Fig. 6).



$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + a_3[n-3] - b_1y[n-1] - b_2y[n-2] - b_3y[n-3]$$

Fig. 6. The 3P3Z coefficients are computed after a conversion from the Laplace- to the z-domain via a bilinear transform.

As depicted in Fig. 6, you start from the type 3 transfer function expressed in the Laplace domain and apply the Tustin bilinear transform to obtain an equation in the discrete-time domain. From there, you can compute the coefficients in the numerator and the denominator, respectively denoted as a_n and b_n. I have built a macro that will generate these compensator coefficients automatically in LTspice. This macro will be shown in a later figure.

It is now interesting to test the validity of the computed coefficients and make sure they lead to the response expected from the compensator. Dedicated tools exist for this purpose, but using a SPICE simulator like LTspice is also a valid approach. Once the coefficients are confirmed to give the desired response, you can include them in your code and carry on with the project.

A delay line[8] lends itself very well to modeling the sampling delay, z^{-1}, as illustrated by the subcircuit schematic diagram in Fig. 7. This is the digital compensator modeled in LTspice. Please note the presence of the zero-order hold (ZOH) for reconstructing the time-domain error signal.

Before including this filter in a complete simulation, it is important to test its ac response and compare it with that of a type 3 built with an op-amp. The circuit in Fig. 8 imposes a 2.5-V output of the digital filter while an ac source sweeps the input. The coefficient values appear as bias points in the electrical diagram for illustration purposes as the *logparams* keyword would store them in the .log file generated by LTspice.

The response is shown in Fig. 9 and confirms the correctness of the type 3 compensator implemented in the digital filter.

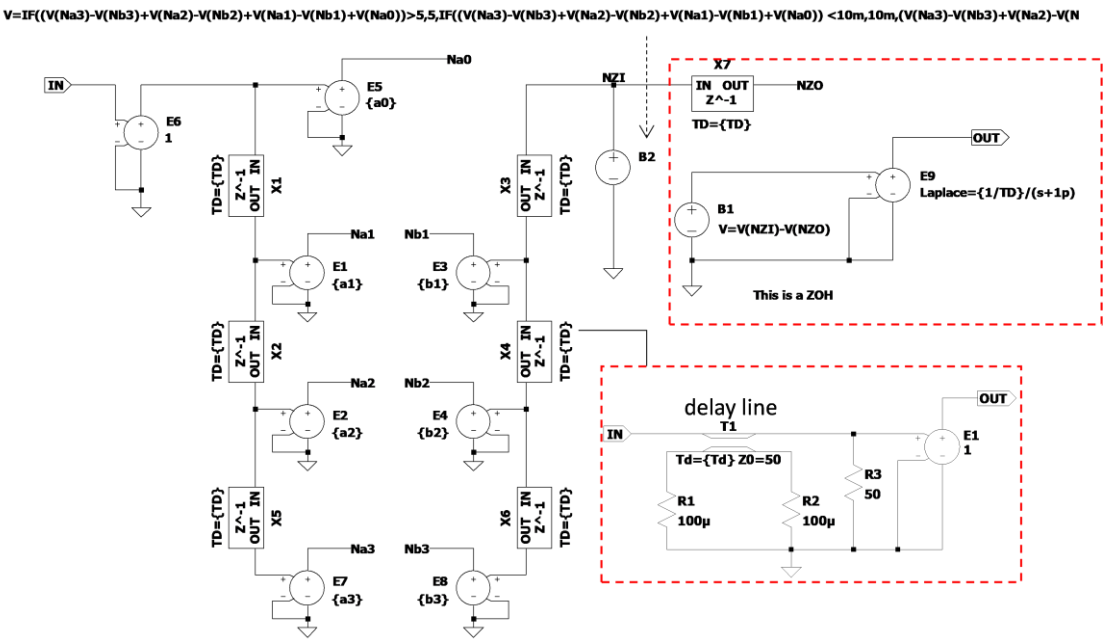


Fig. 7. Modeling the digital type 3 compensator in LTspice. The delay lines are assembled with voltage-controlled voltage sources for the coefficients.

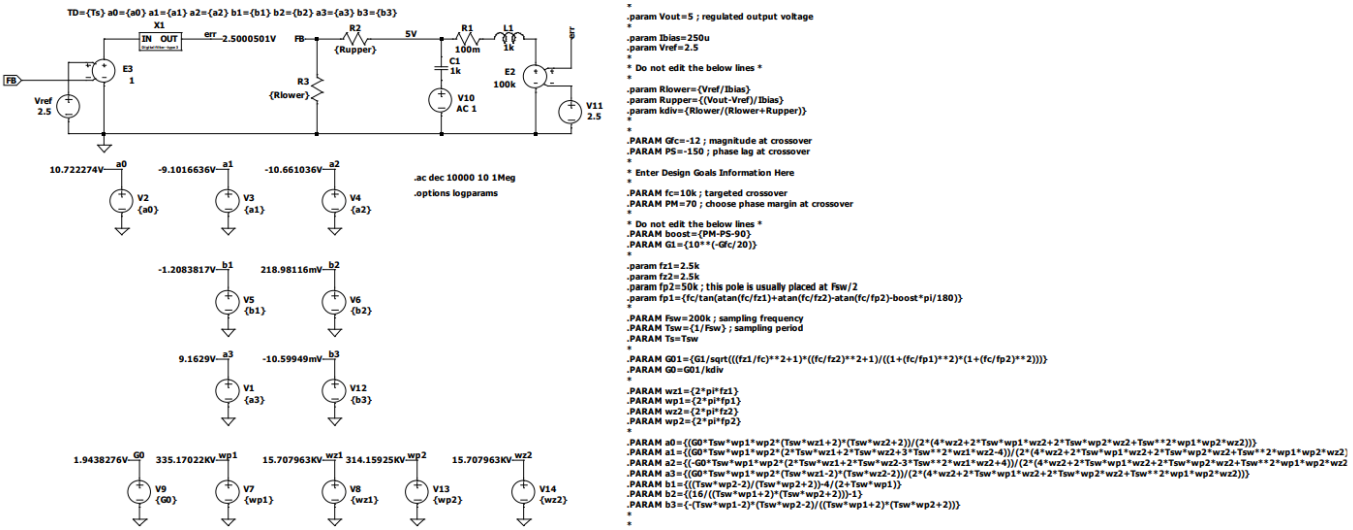


Fig. 8. This is a test setup to check the response of the digital type 3 compensator with the same specifications as in Fig. 3.

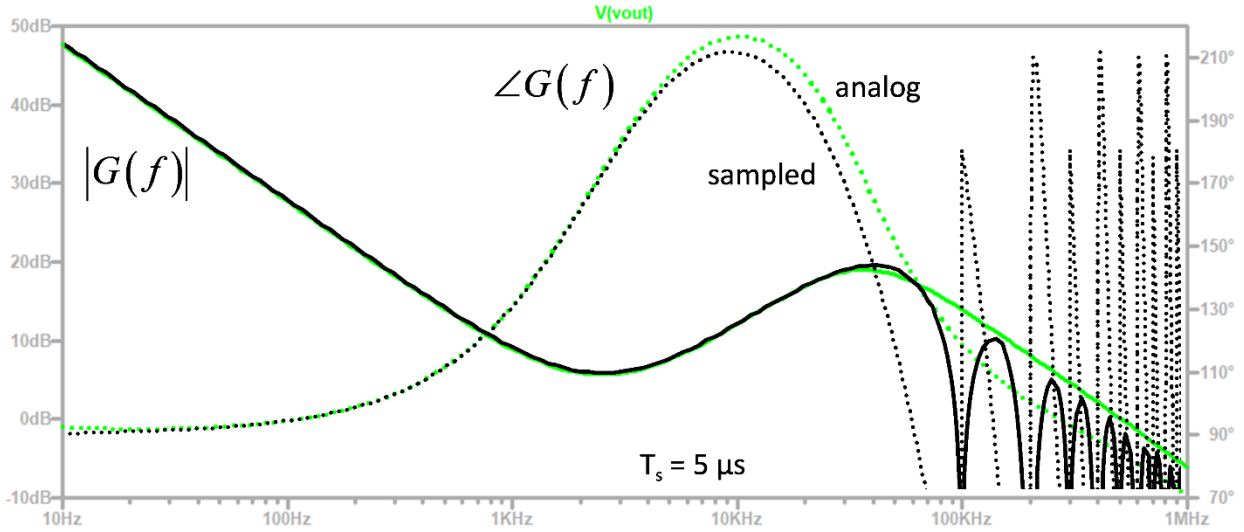


Fig. 9. The ac response of the digital type 3 matches that of its analog counterpart.

You can see the phase erosion brought by the sampling effects. Pushing the sampling frequency higher would reduce these effects but at the expense of selecting a faster computing core.

We can now exercise the compensator in a complete buck converter as shown in Fig. 10. The controller is fully synchronous, with its operating parameters programmed on the left side. There is no divide-by-three block inside the controller subcircuit, bringing the discrete type 3 gain target back to 3 dB. The macro content is available in Fig. 11 and includes the attenuation brought by the resistive divider used for regulation purposes.

Please note that this divider brings an attenuation while it does not in Fig. 3. In this figure, there is a virtual ground brought by the op amp for $s \neq 0$ and the low-side resistance does not play a role in the ac response. With the digital compensator, we do not have a virtual ground and the resistance is there in dc (for regulation) but also in ac. It attenuates the signal and must be compensated by a slightly larger gain, as underlined in the macro.

I have not added a transport delay in this simulation setup but you could easily include one. A delay line is still an option, but I recognize that it slows down the analysis time in transient. Another option is to resort to an analog solution offering a unity gain but lagging the phase down to -180°. The transfer function of this extra circuit is defined as in reference [2] and its implementation appears just after Fig. 11.

```
*
.param Fsw=250k ; switching frequency
.param Tsw={1/Fsw}
.param Dutymax=0.8 ; max duty ratio
.param DT=150n ; deadtime between Q and Qb
.param VDRV=12 ; drive output voltage
.param Imax=1 ; current sense limit
*
*
* Internal PWM Model specifications
*
.param AOL=90 ; open-loop gain
.param GAIN={10**(AOL/20)}
.param pole=30 ; low-frequency pole
.param Isink=20m ; sink capability
.param Isource=500u ; source capability
.param VOL=100m ; low-level voltage
.param VOH=3 ; high-level voltage
*
.param Vv=0
.param Vp=1 ; sawtooth amplitude
.param Vref=2.5 ; internal voltage reference
.param Tss=1m ; internal soft-start duration
*
*
.tran 0 5m 0 100n

*
.model DBUCK D TT=10n Rs=10m Cjo=120p N=0.6 BV=150
.model TOFSW SW (Ron=25m Roff=10Meg Vt=2 Vh=1)
.options logparams ; will display the macro values in the .log file
*
```

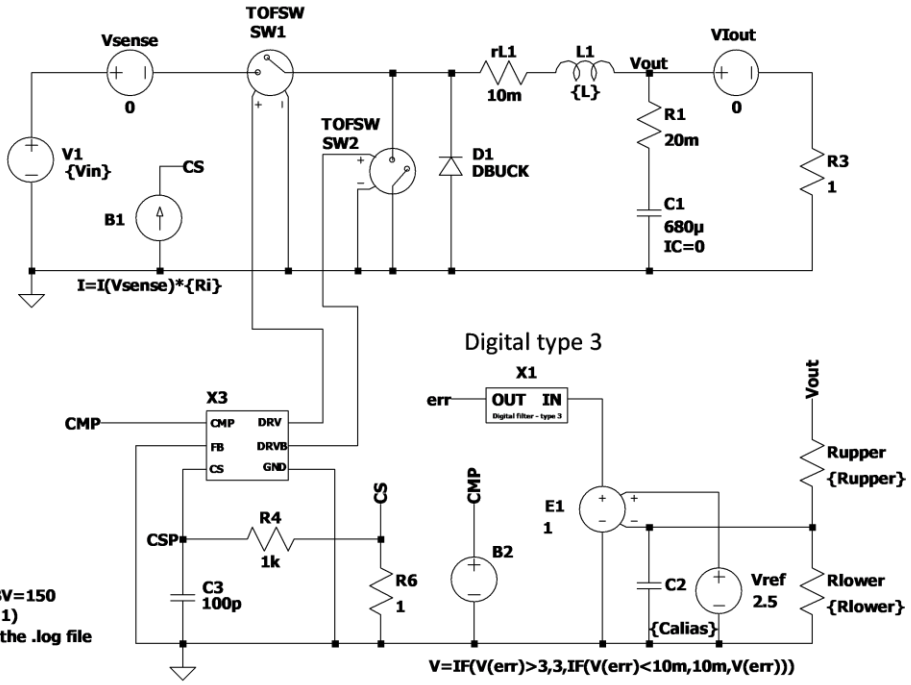


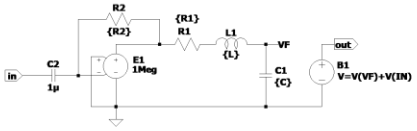
Fig. 10. The synchronous buck converter is now equipped with the digital filter (that is, the digital type 3 compensator).

```
*
.param Vin=12
.param Vout=5
.param L=22u
.param Ri=160m
*
.param Ibias=250u
*
* Do not edit the below lines *
*
.param Rlower={Vref/Ibias}
.param Rupper={(Vout-Vref)/Ibias}
.param kdiv={Rlower/(Rlower+Rupper)}
.param Rth={Rlower*Rupper/(Rlower+Rupper)}
.param Calias=1/(2*pi*Rth*(Fsw/2))
*
*
.PARAM Gfc=-3 ; magnitude at crossover
.PARAM PS=-160 ; phase lag at crossover
*
* Enter Design Goals Information Here
*
.PARAM fc=10k ; targeted crossover
.PARAM PM=70 ; choose phase margin at crossover
*
* Do not edit the below lines *
*
.PARAM boost={PM-PS-90}
.PARAM G1={10**(-Gfc/20)}
*
.PARAM fz1=2k
.PARAM fz2=2k
.param fp1={fc/tan(atan(fc/fz1)+atan(fc/fz2)-atan(fc/fp2)-boost*pi/180)}
.PARAM fp2=50k
*
```

The gain calculation accounts for the resistive divider attenuation.

```
*
.PARAM Ts=Tsw ; sampling period is Tsw
*
.PARAM G01={G1/sqrt(((fz1/fc)**2+1)*((fc/fz2)**2+1)/((1+(fc/fp1)**2)*(1+(fc/fp2)**2)))}
.PARAM G0=G01/kdiv
*
.PARAM wz1={2*pi*fz1}
.PARAM wp1={2*pi*fp1}
.PARAM wz2={2*pi*fz2}
.PARAM wp2={2*pi*fp2}
*
.PARAM a0={(G0*Tsw*wp1*wp2*(Tsw*wz1+2)*(Tsw*wz2+2))/(2*(4*wz2+2*Tsw*wp1*wz2+2*Tsw*wp2*wz2+Tsw**2*wp1*wp2*wz2))}
.PARAM a1={(G0*Tsw*wp1*wp2*(2*Tsw*wz1+2*Tsw*wz2+3*Tsw**2*wz1*wz2-4))/(2*(4*wz2+2*Tsw*wp1*wz2+2*Tsw*wp2*wz2+Tsw**2*wp1*wp2*wz2))}
.PARAM a2={(-G0*Tsw*wp1*wp2*(2*Tsw*wz1+2*Tsw*wz2-3*Tsw**2*wz1*wz2+4))/(2*(4*wz2+2*Tsw*wp1*wz2+2*Tsw*wp2*wz2+Tsw**2*wp1*wp2*wz2))}
.PARAM a3={(G0*Tsw*wp1*wp2*(Tsw*wz1-2)*(Tsw*wz2-2))/(2*(4*wz2+2*Tsw*wp1*wz2+2*Tsw*wp2*wz2+Tsw**2*wp1*wp2*wz2))}
.PARAM b1={((Tsw*wp2-2)/(Tsw*wp2+2))-4/(2+Tsw*wp1)}
.PARAM b2={(16/((Tsw*wp1+2)*(Tsw*wp2+2)))-1}
.PARAM b3={-(Tsw*wp1-2)*(Tsw*wp2-2)/((Tsw*wp1+2)*(Tsw*wp2+2))}
*
```

The polynomial coefficients are determined based on the poles-zeroes and gain values.




```
*
.param w0=3.14159/TD
.param Q={2/3.14159}
.param L=100u
.param C={1/(w0**2*L)}
.param R1={1/((({C}/(4*{L}))**0.5)*2*{Q})}
.param R2={TD/1u}
*
```

This is an analog delay that can be inserted after the compensator to model any transport delay in the loop. {TD} is the parameters in seconds.

Fig. 11. The macro automates the coefficients calculation for the digital filter and accounts for the resistive divider presence.

$$H(s) = 1 - \frac{s \cdot \tau}{1 + \frac{s}{\omega Q} + \left(\frac{s}{\omega_0}\right)^2} \qquad (7)$$

$$Q = \frac{2}{\pi} \text{ and } \omega = \frac{\pi}{\tau}$$

To test the stability of this converter, I have performed a load-step from 3.5 to 5 A with a 1-A/µs current slope. The graph from Fig. 12 confirms the good response obtained with this compensation strategy. The coefficients computed by LTspice appear in the figure and will have to be included in the coding of the 3P3Z compensator.

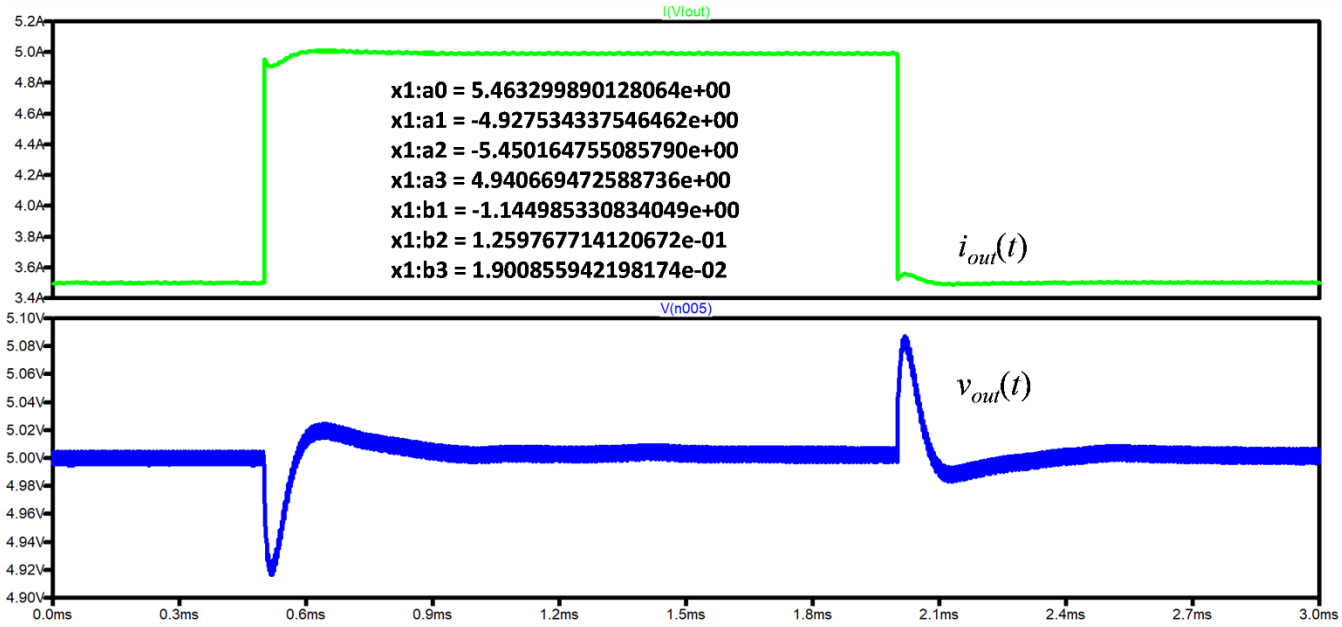


Fig. 12. The transient response is excellent and shows a clean output voltage waveform.

Conclusion

The compensation strategy for a buck converter operated in voltage-mode control, has been described with an analog compensator built around a type 3 architecture. Then, an equivalent version in the discrete-time domain has been presented, built around simple delay lines.

Capitalizing on LTspice-specific SPICE directives, I have built a macro which automates the coefficients of the 3P3Z compensator. This helps the designer test his strategy—and confirms it works as expected—before coding the coefficients into software lines. The simulation time is a bit slower than with the full analog version, nevertheless, it provides a valuable tool to explore different compensation strategies.

References

1. "Simplified analysis of PWM converters using model of PWM switch. Continuous conduction mode" by Vatché Vorpérian, IEEE Transactions on Aerospace and Electronic Systems, vol. 26, no. 3, May 1990.
2. *Transfer Functions of Switching Converters* by Christophe Basso, Faraday Press, June 2021.
3. *Switch-Mode Power Supplies: SPICE Simulations and Practical Designs* by Christophe Basso McGraw-Hill, 2nd edition, August 2014.
4. 150+ free templates for SIMPLIS: https://powersimtof.com/Downloads/Book/Book Collection.zip.

5. 150+ free templates for LTspice: https://powersimtof.com/Downloads/LTSpice/LTspice_SMPS.zip
6. *An Intuitive Guide to Compensating Switching Power Supplies*, Faraday Press, September 2024.
7. "Simulation and Analysis Applied to the Design of Buck Topologies," by Christophe Basso, APEC professional seminar, Anaheim, 2019.
8. "Generic average modeling and simulation of discrete controllers" by D. Adar and S. Ben-Yaakov, APEC, Anaheim, 2001.

About The Author

Christophe Basso is a business development manager with Future Electronics, a member of the power team and covering EMEA. Previously, he was a technical fellow with onsemi for 24 years where he originated numerous integrated circuits. SPICE simulation is also one of his favorite subjects and he has authored two books on the subject. Christophe's latest work is "An Intuitive Guide to Compensating Switching Power Supplies". Christophe received a BSEE-equivalent from the Montpellier University, France and an MSEE from the Institut National Polytechnique de Toulouse, France. He holds 25 patents on power conversion and often publishes papers in conferences and trade magazines.

For further reading on compensating dc-dc converters, see the How2Power Design Guide, locate the "Design Area" category and select "Stability". Also, see "Modeling and Simulation".