# Write your own generic SPICE Power Supplies controller models

**Christophe BASSO**
November 1996
Manuscript for PCIM US

Simulating the switching behavior of a Switch Mode Power Supply (SMPS) is not always an easy task. This is especially true if the designer wants to use an exact SPICE model for the Pulse Width Modulator (PWM) controller which will be used in the design. The PWM model may exist, but its syntax may be incompatible with your simulator. If the model has not been created, the debate over whether or not to do the simulation is closed! The solution that is proposed in this article consists of writing your own generic model of the PWM controller and then adapting its intrinsic parameters to comply with the real one you are using. Fixed frequency Current Control Mode (CCM) and Voltage Control Mode (VCM) models will be thoroughly covered in this article, as well as the model translation between different simulators.

**The Berkeley B element, the standard behavior element**

An efficient PWM model that is easy to model must include functions that are generic. For instance, it would not be clever to model an internal current comparator with the complete transistor architecture of a LM311 or a LM193. Fortunately, there is a simple in-line equation that can describe the perfect comparison function. By adding some passive elements to incorporate various effects (propagation delay, input offset voltage, etc.) we can achieve the functionality we need without sacrificing the simulation speed.

The non-linear controlled source, or B element, is part of Berkeley SPICE3, which was released to the public domain in 1986. Depending on the compatibility of your SPICE 3 simulator, the corresponding syntax may vary significantly. B elements can be linear or non-linear current or voltage sources. Some vendors have expanded the B element syntax to include BOOLEAN and IF-THEN-ELSE functions. For INTUSOFT's IsSpice4 (San Pedro, CA) and CADENCE's Analog WorkBench Spice Plus (San-Jose, CA), the writing of I or V math equations using B elements is the same because both are SPICE 3 compatible. For example, current/voltage generators whose current depends on various nodes can be expressed as:

```
B1 1 0 I = V(5,8)*100*V(10)/(V(8)+V(12))          ; IsSpice or AWB current source
B2 2 0 V= V(9,8)*500*V(12)                        ; IsSpice or AWB voltage source
```

MICROSIM's PSpice (Irvine, CA) has departed from the Berkeley standard and uses a different syntax. PSpice modifies the standard calls for dependent voltage controlled sources (E and G elements). The equivalent PSpice examples are as follows:

```
G1 1 0 VALUE = { V(5,8)*100*V(10)/(V(8)+V(12)) }  ; PSpice current source
E2 2 0 VALUE = { V(9,8)*500*V(12) }               ; PSpice voltage source
```

**Implement your logical operations**

As stated in the above "paragraph, BOOLEAN and IF-THEN-ELSE expressions have become a part of most vendors' B elements. Their implementation also depends on the SPICE simulator. INTUSOFT exploits the concept of "binary" voltage, that is to say, a node value which is lower or higher than a user-defined threshold can be associated with 1 or 0. This threshold is driven by the keyword LTHRESH, whose value is set via a .OPTIONS line. Two other options, LONE and LZERO, will define the HI and LO electrical values which are delivered by a B element source when it is performing such BOOLEAN operations. A simple NAND equation between two nodes is simply expressed as:

```
BNAND 3 0 V= ~ ( V(1) & V(2) )          ; IsSpice complemented (~) AND operation between V(1) and V(2)
```

Because the other SPICE simulators do not directly support this syntax, it would be much easier to adopt a simpler expression in order to simplify any further translations. If we pass the logical thresholds directly into the equation, we obtain the following IsSpice IF-THEN-ELSE statement:

```
BNAND 3 0 V= (V(1)>800M) & (V(2)>800M)?  0V:  5V
```

In other words,
IF V(1) is greater than 800mV AND V(2) is greater than 800mV, THEN V(3,0)=0V; ELSE V(3,0)=5V

Now the translation to AWB and PSpice syntax is more straightforward:

```
E_BNAND 3 0 VALUE = { IF ( (V(1)>800M) & (V(2)>800M), 0V, 5V ) }    ; PSpice NAND gate
BNAND 3 0 V = IF ( (V(1)>800M) && (V(2)>800M), 0, 5 )               ; AWB NAND gate
```

Note that the AWB parser does NOT accept suffixes for the passed numerical values and the && symbol is doubled as in the C language to distinguish a logical AND from a binary AND. The diversity in implementing the B elements is only bound by the user's imagination. What we have shown above is only a small part of the possibilities offered by Behavioral Modeling via the B element.
If your simulator does not support B element modeling, the situation becomes complex. Some examples on how to model the logical functions with SPICE2 syntax are given at the end of this article.

Here's an application example, a simple voltage limiter which limits the differential voltage of nodes 1 and 2 between 100mV and 1V:

```
E1 3 0 TABLE {V(1)-V(2)}  100M,100M 1,1                              ; PSpice
B1 3 0 V = IF ( V(1,2) < 1, IF (V(1,2) < 100M, 100M, V(1,2)),1)     ; AWB (No suffixes!)
B1 3 0 V = V(1,2) < 100MV ? 100M : V(1,2) > 1 ? 1 : V(1,2)          ; IsSpice
```

In other words,
IF V(1,2) is less than 100mV, THEN V(3,0)=100mV; ELSE IF V(1,2) is greater than 1V, THEN V(3,0)=1V, ELSE V(3,0)=V(1,2)

B elements switch in essentially a zero time span. This characteristic may create convergence problems on transitions associated with these perfect sources. We recommend that you tailor the output switching times in a more realistic manner. A simple RC network is suitable for this purpose. A perfect comparator which accounts for these conditions is given below. We have included it in a SUBCIRCUIT in order to highlight the philosophy of constructing your own models:

```
.SUBCKT COMP          1     2     3
*                     (+)   (-)   OUT
E1 4 0 VALUE = { IF ( V(1) > V(2), 5V, 0 ) }    ; PSpice syntax; IsSpice Syntax: B1 4 0 V=V(1) > V(2) ? 5 : 0
RD 4 3 100                                      ; RC network
CD 3 0 100P                                     ; to slow down transitions
.ENDS COMP
```

Now that we have reviewed the basics of generating in-line equations, let's dip into the nitty-gritty of a constant frequency CCM PWM controller.

**Current mode controllers, a well known architecture**

**Figure 1**, the internal circuitry of a generic single output CCM PWM controller.

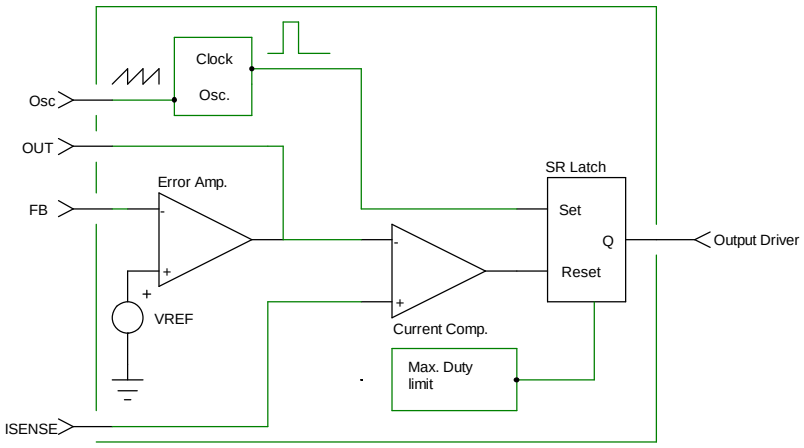


**Figure 1**

The modelling of such a block consists of: a) defining and testing each subcircuit individually, and b) assembling all of these domino-like circuits to form the complete generic model. All individual blocks should be tested before they are used within larger models. Below are some recommendations that will ease your task:

- Draw the symbol of your generic model with your favorite schematic capture tool. Once this is done, you won't have to worry about incorrect pin connections (as you would if you were creating a SPICE netlist with a text editor). Internal subcircuit testing is simplified since you may then access the connection pins directly in the schematic model, and the pin passing process (from schematic to netlist) is performed automatically.

- Place comments on every pertinent line, either with a "*" in column one (for a complete line), or a ";" immediately preceding a comment within a line. Also, use different commented header names for each section of code within the listing.

- Use descriptive names for the components you assemble in the subcircuit netlist, i.e. VCLOCK for a clock source, RDUM for a dummy load, etc.

- Use subcircuits whenever a function is called more than once. Even if the function is only called once, you can create a subcircuit and therefore simplify the netlist. This will also facilitate the writing of new models because the .SUBCKT functions are easily pasted into the netlist. Also, if required, the conversion process to another platform will be greatly simplified.

- Use realistic parameter values for primitive SPICE components such as the diode (D). These models may generate convergence problems since some of the default parameters are set to zero. For example,
  .MODEL DMOD D (TT=1N CJO=10P RS=100M)

- Use a main subcircuit pin number of up to 10 or 20 and use incremental decimal digit notation as you change the internal function. This is especially recommended for complex models in which the parent subcircuit may be large. Below is an arbitrary example, where nodes 7 through 19 are preserved in order to output test signals or add additional pins:

```
.SUBCKT    EXAMPLE          1 2 3 4 5 6
**** MAIN CLOCK ****
VCLOCK 20 21 PULSE          ; Main clock
ICHARGE  22 24 10MA         ; Current charge of capacitor C1
**** TRIGGER  ****
RTHRES 30 33 10K            ; Threshold high
CDEL 33 38 10NF             ; Propagation delay
**** COMPARATOR ****
RINP 40 42 10K              ; Input resistor
RFEED 45 49 120K            ; Feedback resistor
.ENDS EXAMPLE
```

**Writing the model step by step**

Let's start with the synchronization signals, Clock, Osc. and Max. duty cycle. The first one will initiate the on-time of the external switch by triggering the flip-flop latch. Its frequency sets the functioning period of the entire PWM circuit, and will therefore require user input (PERIOD). Osc. is provided for ramp compensation purposes. It delivers a signal which is equivalent to that which was delivered by the classical linear charge of the external oscillator RC network. Using the oscillator ramp is a possible option for ramp compensation, but there are others such as charging a RC network from the MOSFET driver output. If the capacitor voltage is kept at around 1 volt, it is possible to obtain a very low impedance linear ramp, without adversely affecting the PWM oscillator. Osc. will have the same period as Clock, but the user will select its peak amplitude (RAMP). Once the Clock pulse is issued, the Max. duty cycle must reset the latch after a specific period of time. This time is user-defined (DUTYMAX) and selects the maximum allowable duty cycle.

Parameters for IsSpice and PSpice are quite similar in format: the parameter, or any equations between parameters, is enclosed by curly braces. Our three generators are listed below. Arbitrary node numbers are used to simplify their understanding:

```
VCLK 1 0 PULSE 0 5 0 1N 1N 10N {PERIOD}
VRAMP 2 0 PULSE 0 {RAMP} 0 {PERIOD-2N} 1N 1N {PERIOD}
VDUTY 3 0 PULSE 0 5 {PERIOD*DUTYMAX} 1N 1N {(PERIOD-PERIOD*DUTYMAX)-2N} {PERIOD}
```

A quick simulation of this set of equations appears in figure 2, where a maximum duty cycle of 0.5 was selected.

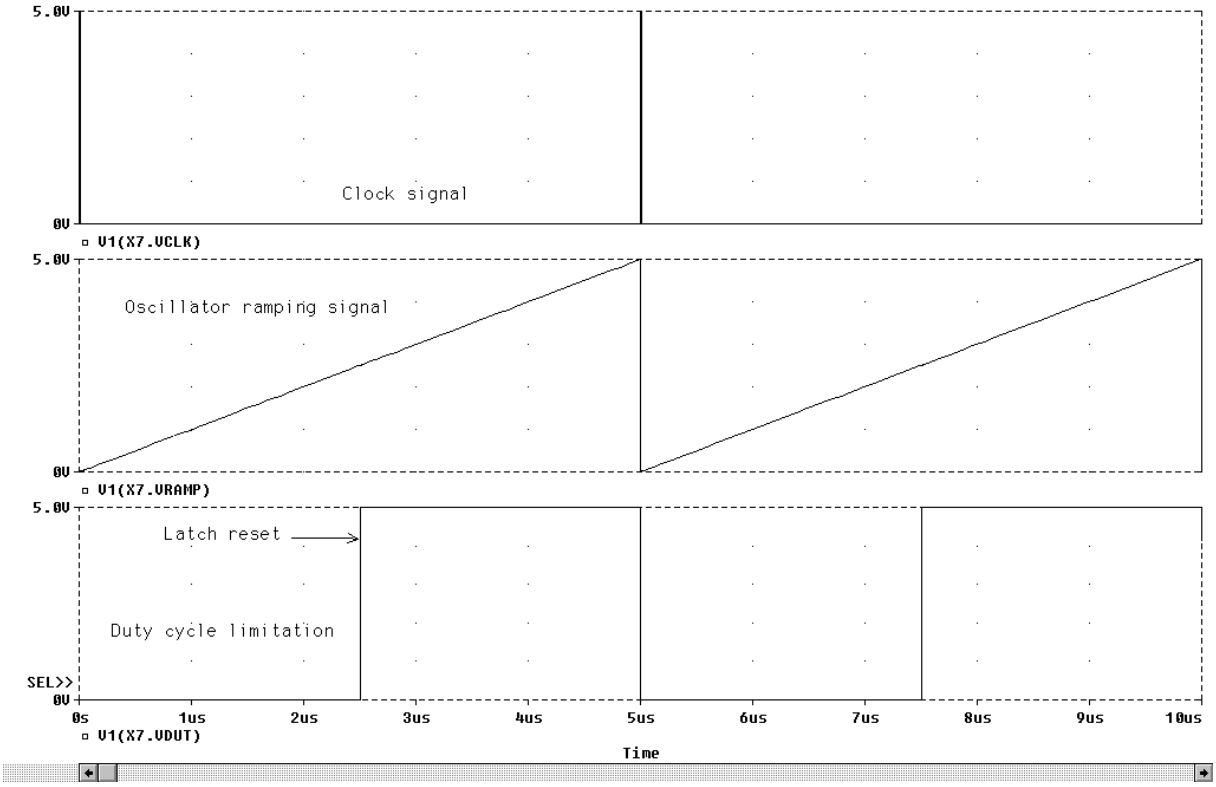



**Figure 2**

The current comparator requires a simple equation followed by a RC network which slows down its transitions. The model is the same as the one given in the example above.

The SR latch may be defined in many ways. We do not recommend the use of a proprietary flip-flop model. You can draw a classical RS flip-flop and add a couple of inverters in order to generate the required signals. Figure 3 shows the electrical circuit.

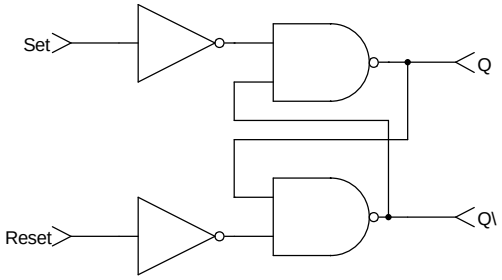



**Figure 3**

The subcircuit will appear as shown below, according to common AWB and IsSpice syntax rules:

```
.SUBCKT FFLOP 6   8  2   1
*                    S   R  Q  Q\
BQB   10 0 V=(V(8)<800M) & (V(2)>800M) ? 0 : 5     ; one input inverted two input NAND
BQ    20 0 V=(V(6)<800M) & (V(1)>800M) ? 0 : 5
RD1   10 1 100                                     ; delay elements
CD1   1 0 10p IC=5
RD2   20 2 100
CD2   2 0 10p IC=0
.ENDS FFLOP
```

The “IC” statements are mandatory in order to avoid conflicts when SPICE computes the DC operating point. You will then add the keyword “UIC” (Use Initial Conditions) at the end of the .TRAN statement.

**A simplified error amplifier macro-model**

There are an infinite number of ways to realize the error amplifier model. However, keep in mind that the simplest model yields the fastest simulation runs. We will use basic building blocks to create a model with the following parameters:

- DC open-loop gain: 90dB, or 31622 {GAIN}
- First pole: 30Hz {POLE}
- Maximum output voltage: {VHIGH}
- Minimum output voltage: {VLOW}
- Maximum sink current: {ISINK}
- Maximum source current: {ISOURCE}

The last two parameters correspond to the output current capability of the op-amp. Modeling its output current capability. instead of using a simple resistor $R_{out}$ in series with the final voltage source, yields more accurate results. This is because once the loop is closed, the dynamic output impedance of the amplifier is close to zero since the open-loop gain $T_{OL}$ is large: $R_{out} = R_{open\text{-}loop} / (1+T_{OL})$. However, this expression is only valid within the limit of the current capability of the amplifier. If the amplifier current exceeds this limit, the component is unable to maintain the proper voltage level, and the output plummets to zero (or rises, if the sink current limit is exceeded). A simple resistor cannot properly model this effect. The electrical schematics is given below:

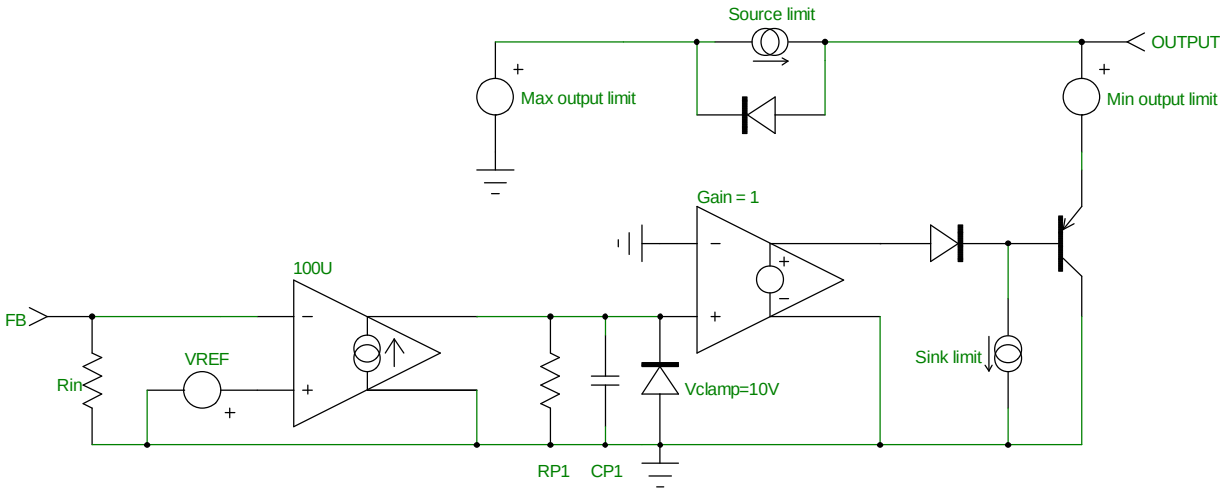



**Figure 4**

Voltage and current offset effects are not modeled, but can be easily added. Offset currents are important, especially when high value feedback networks are used (in the presence of high voltage regulated output, for instance). Output clipping of the voltage controlled sources is always a problem. It is alleviated by using a voltage controlled current source whose output is clipped by a diode and then buffered by a unity gain

voltage controlled source. Sink and source limits are associated with the output transistor; the sink limit is dependent of its static gain (default BF=100). The final netlist is as follows:

```
.SUBCKT ERRAMP 20    8     3       21
*                    +     -     OUT   GND
RIN 20 8 8MEG
CP1 11 21 {1/(6.28*(GAIN/100U)*POLE)} ; pole calculation
E1 5 21 11 21 1
R9 5 2 5
D14 2 13 DMOD
IS 13 21 {ISINK/100}                    ; sink limit, with BF=100
Q1 21 13 16 QPMOD
ISRC 7 3 {ISOURCE}                      ; source limit
D12 3 7 DMOD
D15 21 11 DCLAMP
G1 21 11 20 8 100U
V1 7 21 {VHIGH-0.6V}                    ; max output clipping
V4 3 16 {VLOW-38MV}                     ; min output clipping
RP1 11 21 {GAIN/100U}                   ; open loop gain calculation
.MODEL QPMOD PNP
.MODEL DCLAMP D (RS=10 BV=10 IBV=0.01)
.MODEL DMOD D (TT=1N CJO=10P)
.ENDS ERRAMP
```

The test of the amplifier confirmed the presence of the pole and the current output limits.

**Test of the complete current mode model**

Now that all of the individual elements have been defined and tested, it is time to place them within the final subcircuit model, PWMCM. The output driver model is simplified, and converts the latch levels to user-defined voltages which are associated with a resistor:

```
E_BOUT 15 0 VALUE = { IF ( V(10) > 3.5, {VOUTHI}, {VOUTLO} ) }      ; node 10 is the latch output
ROUT 15 1 {ROUT}
```

For editing convenience, the final PWMCM model will not be printed in this article, but may be downloaded from our BBS or our Internet Web site. The complete example .CIR files are available there also. They are available in PSpice, IsSpice and SPICE2 formats.

The test circuit is a buck converter which delivers 5V at 10A. All of the elements have been calculated using the new release of POWER 4-5-6 [2], which was developed by RIDLEY Engineering (http://members.aol.com/ridleyeng/index.html). Figure 5 depicts this switch-mode converter:

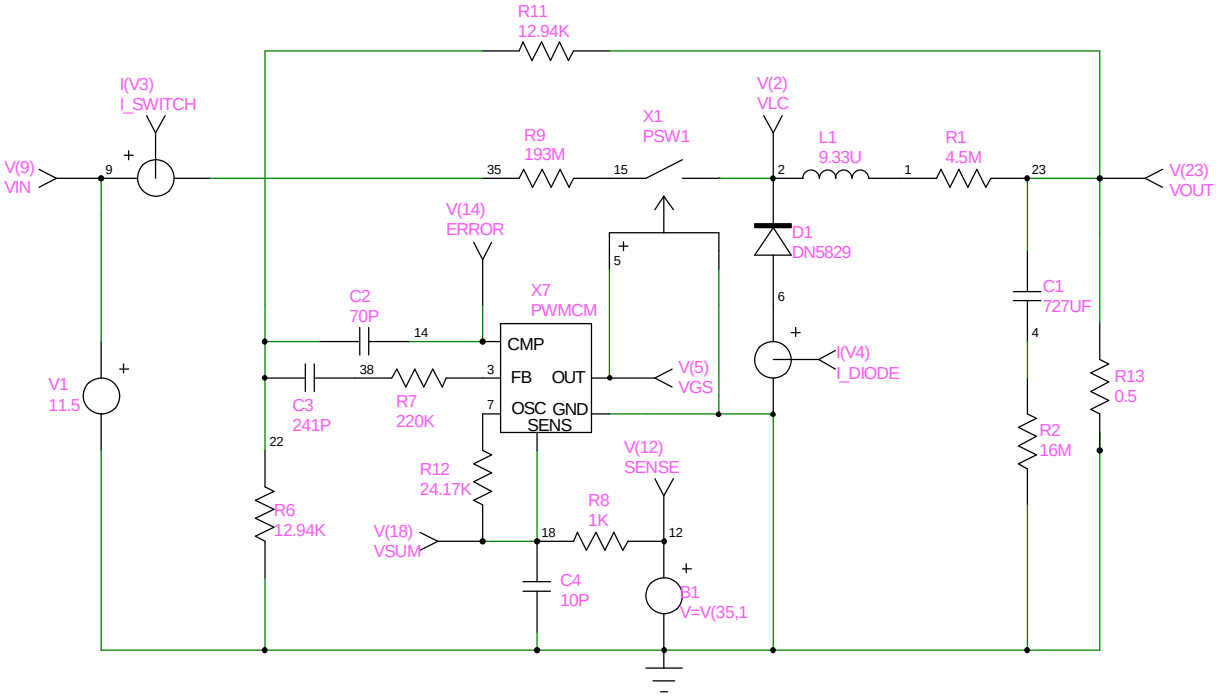



**Figure 5a**

The ramp compensation is accomplished by summing a fraction of the oscillator sawtooth and the current sense information. It has several beneficial effects that we will discuss later on. This circuit has been simulated in 25s upon a P120 machine for a 200µs transient run. With this simulation speed, output response to load or input step can be accomplished rather quickly. Figure 5b shows the start-up conditions before the output voltage has reached its final value. A switching frequency of 200kHz has been selected.

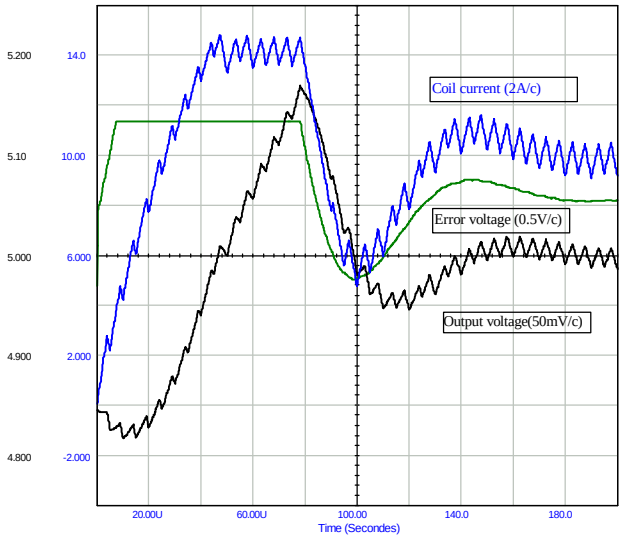



**Figure 5b**

**Current mode instabilities**

The control-to-output transfer function ($V_o/V_c$) of a continuous current mode control converter is a three pole system, as Raymond RIDLEY demonstrated in 1990 [1]: one low frequency pole, $\omega_p$, and two poles which are located at Fs/2. These poles move in relation to the duty cycle and the external ramp. The two high frequency poles present a Q that depends on the compensating ramp and the duty cycle. RIDLEY demonstrated that the Q becomes infinite at D=0.5 with no external ramp, which confirms the inherent instability of a current mode SMPS which has a duty cycle greater than 0.5. Q and $\omega_p$, which are part of the $V_o/V_c$ transfer function, are expressed as follows:

$$Q = \frac{1}{p \cdot (mc \cdot D' - 0.5)} \qquad \omega_p = \frac{1}{CR} + \frac{Ts}{LC} \cdot (mc \cdot D' - 0.5)$$

where $m_c = 1 + S_e / S_n$. $S_e$ is the external ramp slope, $S_n$ is the inductor on-time slope. D' = 1 - D

The low frequency pole, $\omega_p$, moves to higher frequencies as additional compensation ramp is injected. The addition of the external ramp will also damp the Q factor of the double Fs/2 poles, which are the result of the current sampling action, in continuous inductor current. The addition of more ramp will split the double pole into two distinct poles. The first one will move towards lower frequencies until it joins and combines with the first low frequency pole at $\omega_p$. At this point, the converter behaves as if it is operating in voltage mode.

Think of the current loop as an RLC network which is tuned to Fs/2. If we excite this network with a transient current step, it will ring like an RLC response [3], whose damping depends on the duty cycle and on the amount, if any, of ramp compensation. By increasing the duty cycle, we will raise the DC gain of the current loop until the phase margin at Fs/2 vanishes and makes the system unstable. When the duty cycle is greater than 0.5, the current gain curve crosses the 0dB point at Fs/2, and because of the abrupt drop in phase at this point, the converter oscillates. This sharp drop in phase at Fs/2 is created by the double RHP zeros which are engendered by the sampling action in the current information. These double RHP zeros which appear in the current gain are transformed into double poles when one calculates the $V_o/V_c$ transfer function. The Q of these poles is inversely proportional to the phase margin in the current gain at Fs/2. Compensating the system with an external ramp will oppose the duty cycle action by lowering the DC gain and increasing the phase margin at Fs/2, which will damp the high Q poles in the $V_o/V_c$ transfer function.

To highlight this phenomenon, let's open the voltage loop and place a fixed DC source at the right tail of R11 (node 23) in Figure 5a. R12 is elevated to 1MEG in order to suppress any ramp compensation. If we abruptly change the input voltage from 18V to 12.5V, the Fs/2 component appears (100kHz) and is damped after several switching periods, since the duty cycle is less than 0.5. Further stressing of the output would lengthen the damping time or produce a steady-state oscillation. The result is depicted in **Figure 5c**, where a filter has removed the main switching component from the coil current to allow the Fs/2 signal to be properly established.

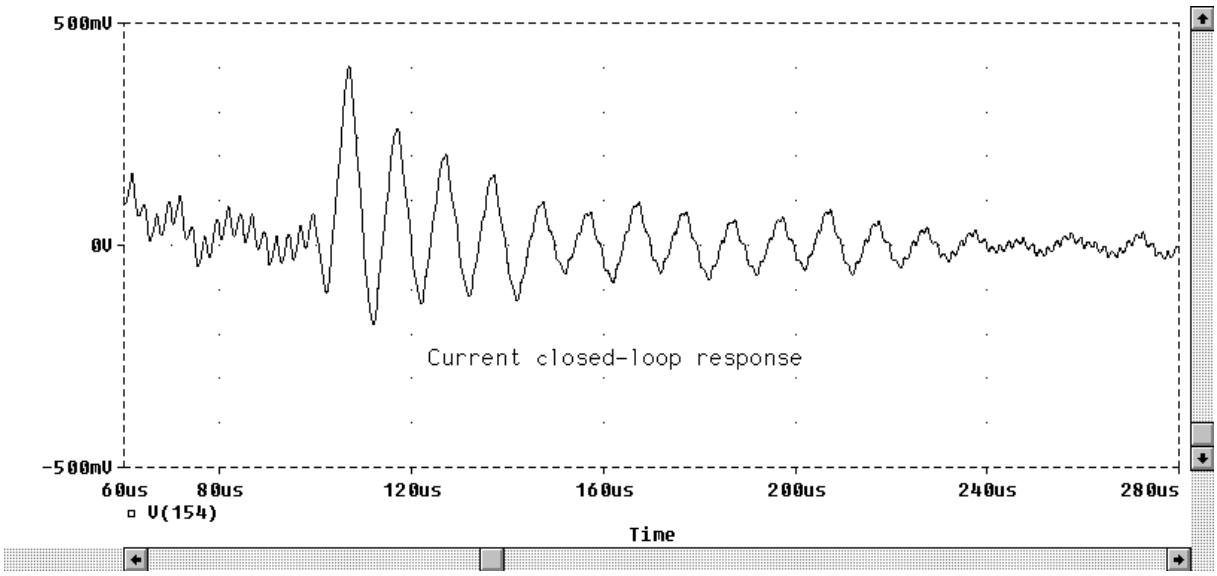



**Figure 5c**

If this network is now closed within a high gain outer loop, any current perturbation will make the entire system oscillate at Fs/2, even if the loop gain has a good phase margin at the 0dB crossover frequency. This so-called gain peaking is attributed to the action of the high-Q poles, which push the gain above the 0dB line at Fs/2, and produce an abrupt drop in phase at this point. If the duty cycle is smaller than 0.5, the oscillations will naturally cease, but if the duty cycle is greater, the oscillation will remain, as **Figure 5d** demonstrates with the FFT of the error amplifier voltage (Vin=11.5V). In conclusion, providing an external

ramp is a wise solution, even if your SMPS duty cycle will be limited to 0.5: the Fs/2 Q will be reduced, thereby preventing oscillations.

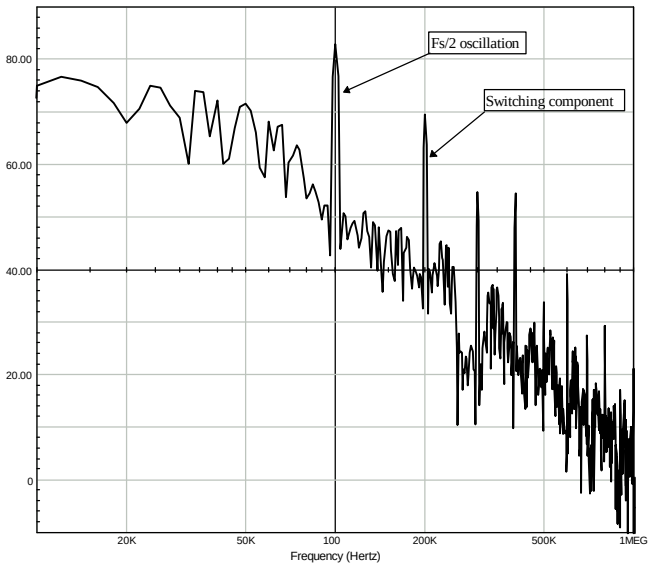
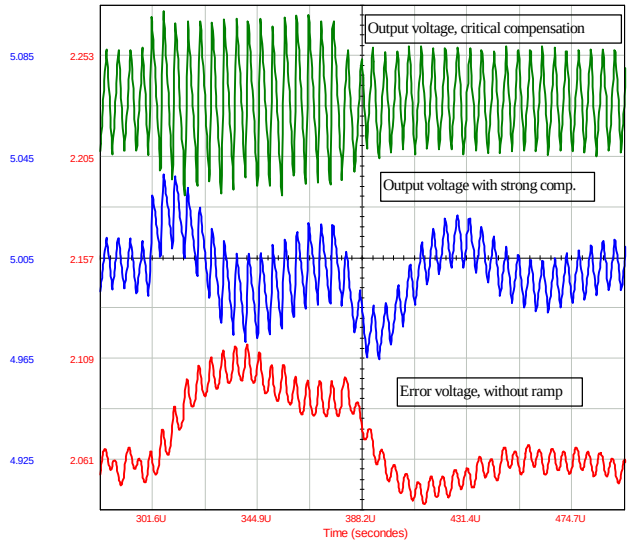



**Figure 5d**

The audio susceptibility is also affected by slope compensation. RIDLEY showed in his work that an external ramp whose slope is equal to 50% of the inductor downslope could nullify the audio susceptibility. As previously stated, excessive ramp compensation makes the converter behave as if it is in voltage mode, which degrades the audio susceptibility. Also, if minimal compensation or no ramp is provided, good input voltage rejection is achieved, the phase of the resulting audio susceptibility is negative; and an increase in input voltage will cause the output voltage to decrease. **Figure 5d** illustrates these behaviors when the input of the buck converter is stressed by a 6V variation. The upper curve depicts the output voltage for a critical ramp compensation. The voltage difference in the output envelope is only 10mV for a 6V input step, which leads to a (theoretical) $\Delta Vout/\Delta Vin$ of -55dB. The middle curve shows how the response starts to degrade as additional ramp is added. The lower curve represents the error amplifier response when a slight ramp compensation is added. The decrease in the output voltage is clearly revealed by the rise in the error voltage.




A close look at the error voltage response time leads to the closed-loop bandwidth of the SMPS. The measured rise time, $t_r$, is roughly 22µs, which gives a bandwidth of: $BW=1/\pi.t_r$ = 14.5kHz, which corroborates our initial design goal value which was passed to POWER 4-5-6.

**The voltage mode model, PWMVM**

The voltage mode generic controller will follow the description given in **Figure 6**.

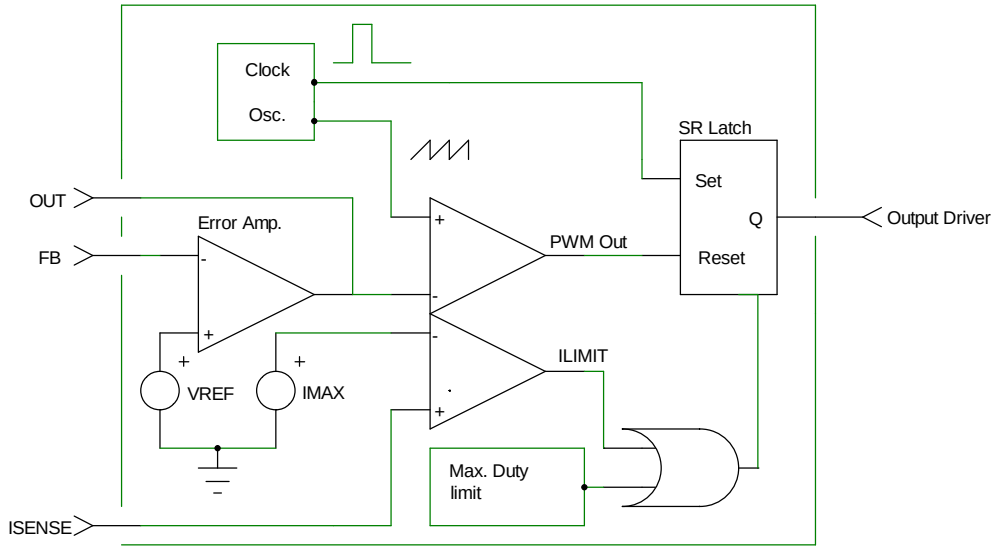



**Figure 6**

The architecture allows the inclusion of a current limitation circuit to reduce the on-time of the external power switch when its peak current exceeds a user-defined limit. This option is strongly recommended to make a rugged and reliable SMPS design that can safely handle input or output overloads. By simply connecting the ISENSE input to ground, you disable this option.

In this model, the duty cycle is no longer controlled by the current information (except in limitation mode). It is controlled by the PWM modulator, which compares the error voltage with the reference sawtooth. The error amplifier output swing will then define the duty cycle limits. Since this output swing is user dependent, the model will calculate the peak and valley voltages of the reference sawtooth such that the chosen duty cycle boundaries are not violated. **Figure 7** depicts the well-known naturally sampled PWM modulator.

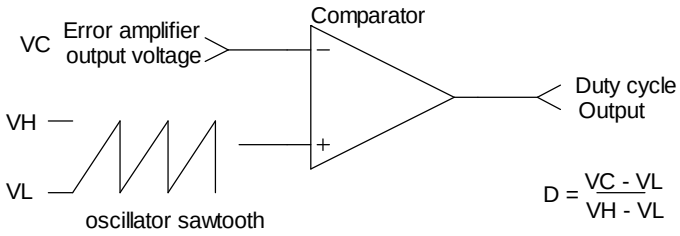



**Figure 7**

Since you will provide the main subcircuit with the duty cycle limits and the error amplifier output swing, it is possible to calculate the corresponding sawtooth peak values, $V_{valley}$ and $V_{peak}$. In MICROSIM's Pspice or INTUSOFT's IsSpice, it is easy to define some particular variables with a .PARAM statement. The reading of the remaining lines in the netlist is then considerably simplified:

```
.PARAM VP = { (VLOW*DUTYMAX-VHIGH*DUTYMIN+VHIGH-VLOW)/(DUTYMAX-DUTYMIN) }
.PARAM VV = { (VLOW-DUTYMIN*VP)/(1-DUTYMIN)}
```

The sawtooth source then becomes:

```
VRAMP 1 0  PULSE {VV} {VP} 0 {PERIOD-2N} 1N 1N {PERIOD}
```

The OR gate which routes the reset signal to the latch from the PWM or the limitation comparator requires a simple in-line equation, followed by the classical delay network:

```
.SUBCKT OR2 1 2 3
E_B1 4 0 VALUE = { IF ( (V(1)>800M) | (V(2)>800M), 5V, 0 ) }
RD 4 3 100
CD 3 0 10P
.ENDS OR2
```

Since the remaining elements have already been defined (comparators, error amplifier etc.), we are all set. As previously stated, the final PWMVM model will not be printed in this article, but can be downloaded from our BBS or our Internet Web site. The test circuit of **Figure 8a** is a forward converter which delivers 28V@4A from a 160V input source.

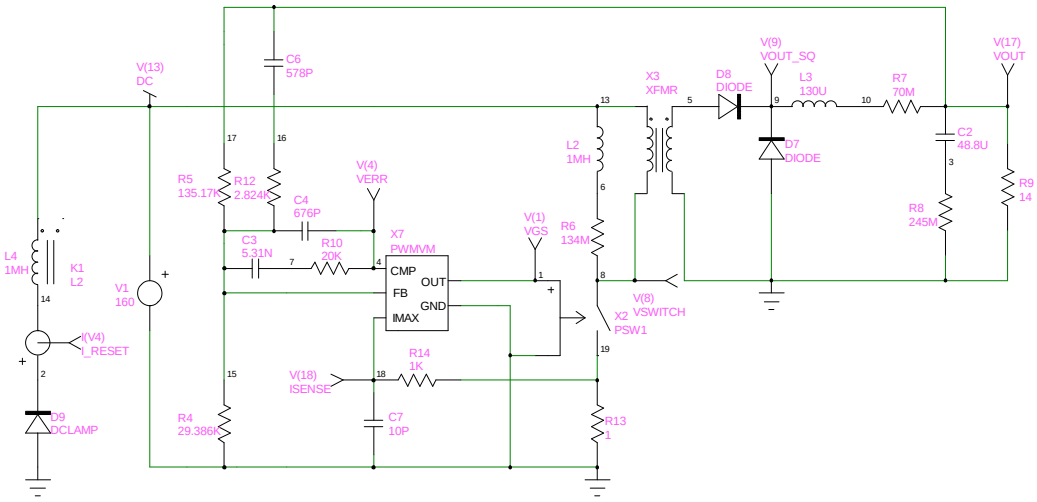



**Figure 8a**

The switching frequency is set at 200kHz, with a maximum duty cycle of 0.45 because of the forward structure. **Figure 8b** depicts the curves which are obtained at start-up. The power switch is modeled with a smooth transition element, as provided by MICROSIM and INTUSOFT. The error amplifier is pushed to its upper limit, and needs some time to recover this transient situation. This behavior is typical of the adopted compensation scheme for a voltage mode converter which is operating in continuous mode.

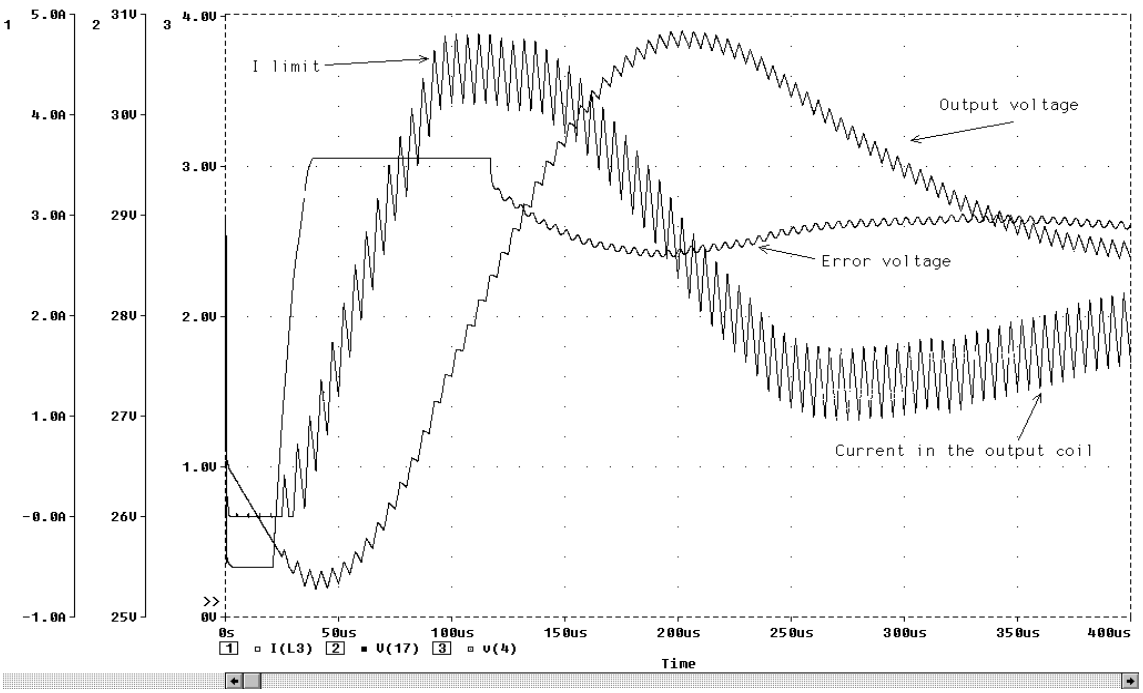



**Figure 8b**

**Modelling with SPICE2**

If you own a SPICE2 compatible simulator, you simply cannot use the B element syntax. To overcome this limitation, some equivalent (but more time consuming) circuits can be constructed. The first generic function which is called in our models is the perfect comparator. **Figure 9a** shows one solution. The unlabeled resistors provide a DC path to ground (10MEG).

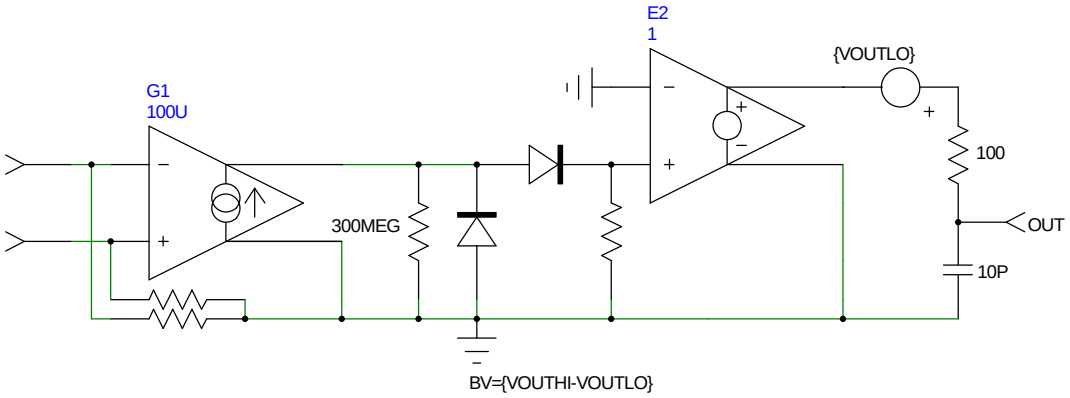



**Figure 9a**

The logical functions are less obvious, at least if you want to build something easily. The ideal gates in **Figure 9b** simulate quickly and converge well. They use the ideal SPICE2 voltage controlled switch or the PSpice/IsSpice smooth transition switch.

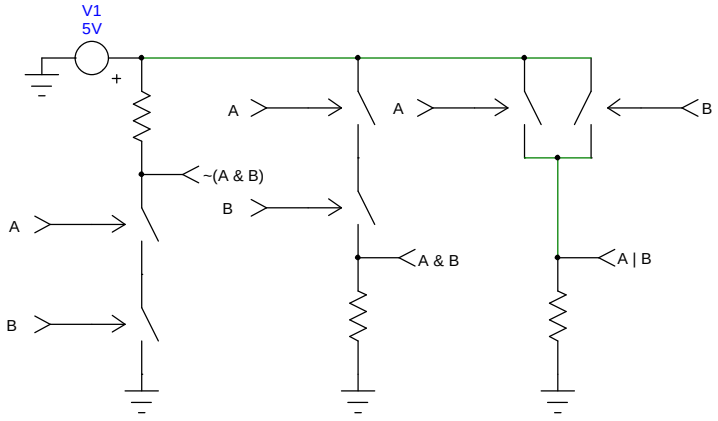



**Figure 9b**

The Flip-Flop is also translated to SPICE2 syntax, as the following lines explain.

```
**** FFLOP ****                  **** 2 INPUT NAND ****
.SUBCKT FFLOP 6 8 2 1            .SUBCKT NAND 1 2 3
*          S R Q Q\              RDUM1 1 0 10MEG
RDUM1 6 0 10MEG                  RDUM1 2 0 10MEG
RDUM2 8 0 10MEG                  S1 3 5 1 0 SMOD
XINVS 6 6 10 NANDS2 5 0 2 0 SMOD
XINVR 8 8 11 NAND                RL 3 4 100
XNAQB 11 2 10 NAND               CD 3 0 10P
XNAQ 10 1 20 NAND                VCC 4 0 5V
RD1   10 1 100                   .MODEL SMOD VSWITCH (RON=1 ROFF=1MEG VON=3 VOFF=100M)
CD1   1 0 10P IC=5               .ENDS NAND
RD2   20 2 100
CD2   2 0 10P IC=0
.ENDS FFLOP
****
```

The simulation of the buck converter circuit of figure 5a using SPICE2 syntax took 71seconds on our P120 machine, giving an increase in simulation time of 184%.

If you are interested in the SPICE2/SPICE3 macro modeling technique and the SPICE engine in general, consult “The SPICE Book”, written by Andrei VLADIMIRESCU [4] or “SMPS Simulation with SPICE3”, written by Steve SANDLER [5].

.

**Conclusion**

This article describes some guidelines which will help you in the process of writing your own generic models for the platform of your choice. The two models, PWMCM and PWMVM, simulate quickly and converge very well, therefore allowing the designer to readily implement the average model results to see if they are verified by the “real world” switched elements.

**Christophe BASSO**
e-mail: basso@esrf.fr

---

**References**

1. R. B. RIDLEY, “A new small-signal model for current-mode control“, PhD. dissertation, Virginia Polytechnic Institute and State University, 1990
   This document can be ordered from the VPEC Web site: http://www.vpec.vt.edu/cgi-bin/home.html
2. POWER 4-5-6 Technical manual, RIDLEY ENGINEERING (Battle-Creek, MI)
3. B. HOLLAND, “Modelling, Analysis and Compensation of the Current Mode Converter”, Powercon 11, 1984 Record, Paper H-2
4. A. VLADIMIRESCU, “The SPICE Book”, John Wiley & Sons, ISBN 0-471-60926-9
5. S. SANDLER, “SMPS Simulation With SPICE3”, McGraw-Hill, ISBN 0-07-913227-8

---

**Acknowledgements**