23

Computer Simulation of Power Electronics

Michael Giesselmann
Texas Tech University

23.1 Introduction

This chapter discusses the possibilities and limitations of computer simulations for power electronics systems. Obviously, advances in raw processing power for personal computers as well as the rapid development of electronic design software have influenced the field of power electronics. In this context, electronic design software means any software used for schematic capture, circuit board layout, electrical or thermal simulation, documentation, and other applications. From the very beginning, schematic capture and circuit board design software was used for power electronics systems. Of course, by their very nature, schematic capture and layout programs had graphical user interfaces. However, long before the advent of graphical user interfaces, electronic circuits were simulated by means of computers, mainly using variations of the circuit simulation code SPICE.

SPICE, an abbreviation for **S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis [14], was developed in the 1970s at the University of California at Berkeley. The initial motivation for the creation of the SPICE code was the simulation of analog electronic circuits to create integrated circuits (ICs). SPICE solves the fundamental differential equations governing electric circuits containing basic R, L, C elements and voltage (V) and current (I) sources, which can be fixed or dependent. Electronic parts, such as diodes, transistors, etc., are either implemented as native elements with equations appropriate to their nature or modeled via subcircuits containing basic and native electronic elements. Device equations are typically based on semiconductor theory and refined using semiempirical parameters.

However, the use of SPICE or similar codes for the simulation of power electronics systems proved to be difficult from the outset, because power electronics circuits typically operate in a highly discontinuous mode, with power semiconductor devices acting as almost ideal switches. The simulators typically could not follow the sudden switching transitions and would become unstable and crash. In addition, typically only transient (time domain) analyses could be performed. If the transient analysis was at all stable, it typically had to be run with very small time steps, resulting in long run times and huge output data files. Other types of analyses, such as AC (frequency domain) analysis, were not possible. For AC analysis, the circuit response is linearized around a bias point, and the small signal behavior is analyzed for a range of frequencies. Typical results are the well-known Bode plots, which have proved to be very useful for

the design of feedback control loops. Of course, if a normal power electronics system has several switches, which constantly turn on and off, a single bias point cannot be found and AC analysis will fail. To make matters worse, some circuit codes will perform AC analysis anyway and give totally erroneous results.

This chapter discusses techniques to overcome these problems. With these techniques, even complex power electronics circuits can be simulated, their behavior can be studied in both the time and frequency domain, and simulation can finally be fully integrated into the electronic design process.

In the following, the possibilities and limitations of simulation tools are discussed in more detail. Computer simulation of electronic circuits in general and power electronics circuits in particular has many obvious advantages, such as:

- New topologies can be quickly tested.
- New control strategies can be studied before implementation.
- Existing topologies can be analyzed for normal and fault conditions.
- Tests can be performed safely and quickly without risk of harm for personnel or equipment.

In addition, mechanical systems such as motors and mechanical attachments can be included in the simulation of power electronics systems, thus enabling the simulation of complete mechatronics systems.

Before proceeding farther it should be acknowledged that some limitations remain, which can only be overcome in special cases and with considerable effort involving extensive fine-tuning of models from experimental data. These limitations involve the details of the switching transitions. These details include the precise transients for voltages and currents in the switching devices, including peak voltage overshoot, etc. To model these transitions precisely, which typically occur in the nanosecond time frame, not only exact models for the semiconductors are necessary, but also the parasitic circuit elements, such as the inductance of the device packages and the circuit connections, must be known and accounted for in the simulation setup. Furthermore, the precise transient traces of the control signals in the nanosecond time regime must be known and implemented.

For the above-mentioned reasons, the author does not recommend use of a simulation to verify that, for example, a certain voltage stress level on an IGBT transistor in an inverter is not exceeded. Similarly, the precise amount of switching (unlike conduction) losses is difficult to predict from a simulation. This is better left to experimental work in the laboratory. However, with the exception of a very narrow time window around individual switching transitions, the response of the circuit is very realistic. The reader should recall that circuits are typically in transition for less than 1% of total time. Therefore, the voltage and current levels in all inductors and capacitors are typically within less than 1% of the real values.

In conclusion, simulation is a great tool to study the behavior of new and existing circuits including mechanical energy conversion devices and control systems with the possible exception of a very narrow window around the switching transitions.

23.2 Code Qualification and Model Validation

Before software of any kind is used as part of a design process or in support of a comprehensive analysis of an existing system, care should be taken to ensure that the software is working correctly for the intended application. It should be pointed out that most software will work correctly for the purpose that it was designed for, but sometimes software can easily be used (or misused) in ways or for applications for which it was not intended. To make matters worse, the fact that some software should not be used for a particular problem may not be so obvious to the user. The reader should be reminded that SPICE was initially created to support the design of integrated circuits. Therefore, all basic elements are ideal and zero dimensional, meaning that a resistor has no parasitic inductance associated with it and has no propagation delay. Similarly, an inductor has no losses and no propagation delay nor any parasitic capacitance. Nevertheless, SPICE turned out to be a code that could be used for general circuit analysis and for many applications not imagined at the outset. However, every prudent engineer or engineering supervisor should always try to evaluate a computer code using a typical example with known behavior

and carefully compare the simulation results with the known (measured) facts about the circuit. In this phase of code qualification, the engineer should also consult the accompanying documentation for background information about the code, its intended uses and limitations, and the internal workings of the simulation engine. This may often give important hints to the fidelity of the results of a particular application.

Close attention should also be paid to the device models that may be contained inside a particular code and their features and limitations. For example, it may be important to know if the model for a transformer uses nonlinear magnetics or not. If the code (as PSpice® and many others) allows it, custom models that have the properties needed for a given case can be added. However, in this case, the models should be carefully tested and validated before they are used, especially if critical engineering decisions are to be based on the results. The reader should also be cautioned that after an (ever more frequent) upgrade of a particular code, it is advisable to at least perform some sort of check to determine that the core of the simulation engine still behaves like before. For this purpose, the input files for a (not too simple) benchmark case should be retained along with a documentation of the output from previous versions of the code. It should also be mentioned that even "bug-fixes," "Internet-patches," or "code maintenance" can potentially cause a simulator to behave differently. (All of those things have happened to the author over the years). Sometimes the user may not even know that upgrades or the like have taken place, if software maintenance is performed by the information technology (IT) department of a company. In any event, it is always advisable to scrutinize the results of any simulation, compare them against known facts and expectations, and resolve any discrepancies.

23.3 Basic Concepts—Simulation of a Buck Converter

In the following, the simulation of a buck (step-down) converter is described to illustrate the concepts mentioned in the introduction. Good references for power electronics circuits in general are References 2, 6, and 9. The simulations have been performed using the SPICE implementation called PSpice, which was developed by MicroSim Corp., and is currently sold by Cadence, Inc [8]. As far as possible, the examples shown here can be run on the student version of the software. The examples are created using the "Schematics" editor, which provides a convenient graphical user interface.

Figure 23.1 shows the well-known topology of a switch mode, step-down (also called buck) converter. With the chosen values for the components, the converter is operating in the continuous-conduction mode (as referred to the inductor current), and the output voltage is equal to the input voltage multiplied by the duty cycle of the power MOSFET "M1."

This circuit uses only standard elements from the library of the student version. (In a real circuit a diode other than the 1N4002 would be used.) The key to a stable and quick simulation is the drive signal for the power MOSFET. The hierarchical block called "PWM-Generator" accomplishes this task. The input to this block is a voltage between 0 and 1 V, representing a duty cycle between 0 and 100%. The output

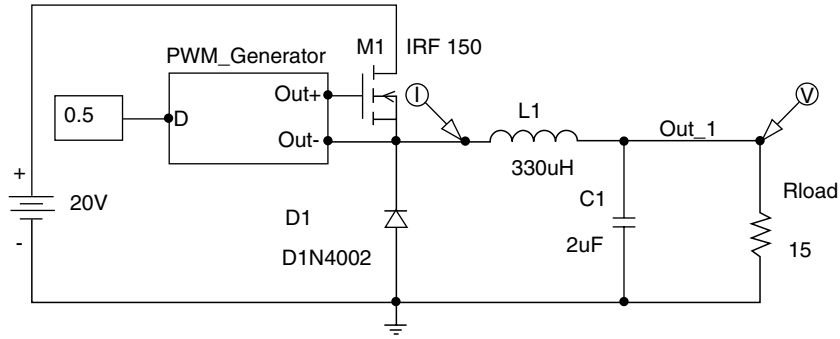

FIGURE 23.1 Schematic of a step-down converter.

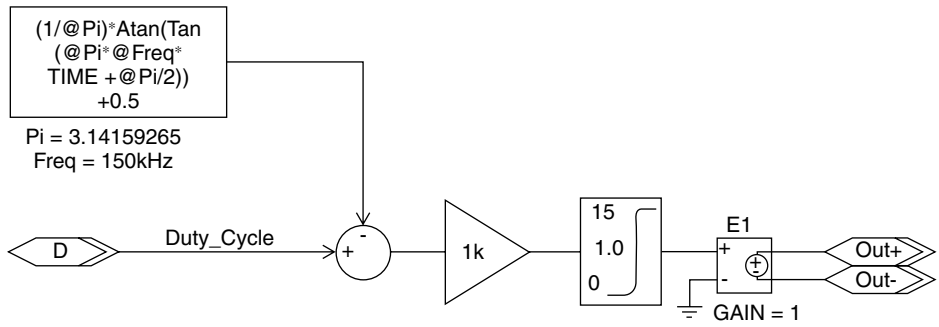

FIGURE 23.2 Schematic representing the "PWM Generator" hierarchical block of Fig. 23.1.

is a rectangular voltage, which is available between the outputs "Out+" and "Out–," which has an amplitude of 15 V, a duty cycle specified by the input and a repetition frequency, which can be freely chosen. In addition, the switching transitions of this rectangular waveform have controllable slopes with smooth edges to keep the simulator from crashing. Here is used a concept that was explained in the introduction, stating that in the interest of stable operation, short run times, and manageable output file size, it is not only permissible but recommended to replace the actual drive signal with one that is more suitable for simulation. Careful examination of the drive signals shows only very minute differences as the result of this substitution, but the advantages for stability and run times are enormous. Also as mentioned in the introduction, the total time that the circuit remains in transitions is very small. Therefore, the output voltage and the inductor current of this example are completely realistic.

For this example, the drive signals are generated entirely with so-called analog behavioral modeling (ABM) components, which have no counterpart in the real circuit. In Section 23.4 a realistic model for a real MOSFET driver circuit is presented, which also creates suitable gate drive signals.

Figure 23.2 shows the circuit that implements the PWM signals using the techniques discussed above. This circuit is an implementation of the carrier-based PWM generation method with PSpice® ABM parts. In the carrier-based PWM generation method, a voltage level, representing the duty cycle, is compared with a triangular or sawtooth-shaped carrier. A convenient way to generate such a carrier without resorting to mathematical functions with piecewise definition is to calculate the argument of a periodic trigonometric function. This is illustrated in Fig. 23.3. This figure was created using the MathCAD® [4] software package. The circuit in the upper half of Fig. 23.4 shows the implementation of a sawtooth function using basic ABM parts in PSpice in more detail.

In the lower part of Fig. 23.4, a more-compressed form with only one ABM part is shown, which generates the same output. Using the compressed form not only results in space savings on the "Schematics" page, but also reduces the total device count for the simulation. This can easily make the difference between being able to run a circuit within the limitations of the student version or not. The circuit shown in Fig. 23.2 compares the sawtooth signal with the duty cycle and amplifies the difference by a factor of 1000 (1 k) using a "Gain" device. The amplification factor controls the steepness of the transitions in the PWM signal. A soft limiter on the output of the amplifier limits the signal amplitude to the range of 0 to 15 V.

The soft limiter uses a hyperbolic tangent function to achieve its function. To illustrate this, Fig. 23.5 shows a MathCAD [4] plot of a hyperbolic tangent function for different steepness factors k. In fact, the steepness factors are just multipliers for the argument of the function. From Fig. 23.5 it is easy to see how a transition can be achieved that is steep but has rounded corners without abrupt slope changes at the same time. These signal properties are the key to a fast and stable operation of the simulator. The last element in Fig. 23.2, named "E1," is a voltage-controlled voltage source. It takes the output of the soft limiter, which is a voltage with respect to ground, and creates a voltage with a floating reference potential for driving high-side MOSFETs such as in buck converters.

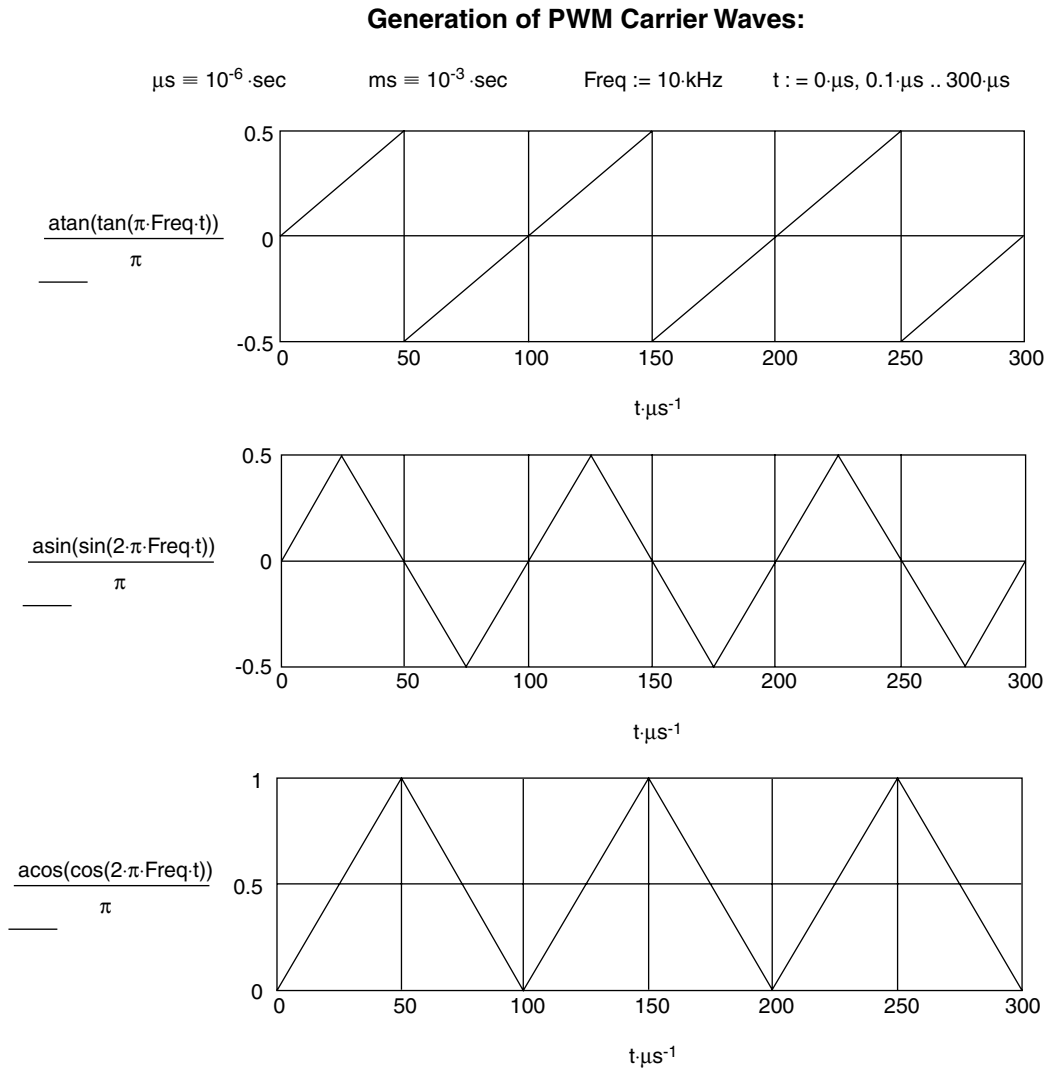

FIGURE 23.3 Illustration of the mathematical functions used for carrier wave generation.

Figure 23.6 shows the simulation results for the buck converter shown in Fig. 23.1. The simulation shows a start-up event, where a gate signal with a duty cycle of 50% is suddenly applied to MOSFET "M1" while both the inductor current as well as voltage on the output capacitor are zero. The upper half of Fig. 23.6 shows the trace of the output voltage, whereas the lower half of the graph shows the inductor current. It can be seen that both the output voltage (10 V) and the average output current (10 V/15 Ω) are represented correctly in Fig. 23.6. Since the input voltage is twice as high as the output voltage and the losses (occurring only in the MOSFET and the diode) are minimal, the average input current is half the output current. Because of the chosen gate signal generation, the simulation runs stable and fast, especially if the high switching frequency of 150 kHz is considered, which was chosen for this example.

Considering the fact that for the buck converter the ratio of the input and output voltages is proportional to the duty cycle D and the ratio of the average input and output currents is inversely proportional to D, the buck converter is acting as a transformer for DC. As in an AC transformer, the product of output voltage and the average output current is nearly identical to the product of the input voltage and the average input current. Of course, if no losses were present, the products would be precisely identical.

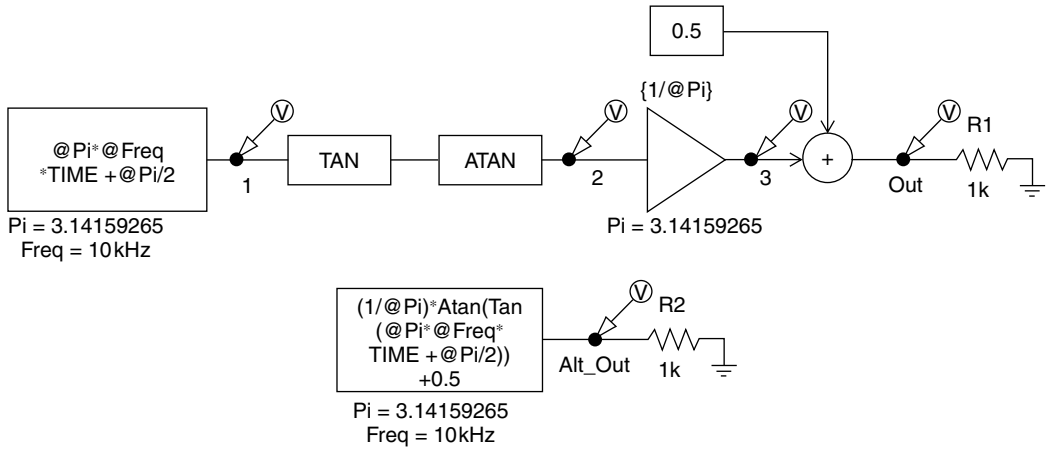

FIGURE 23.4 PSpice implementation of the sawtooth function.

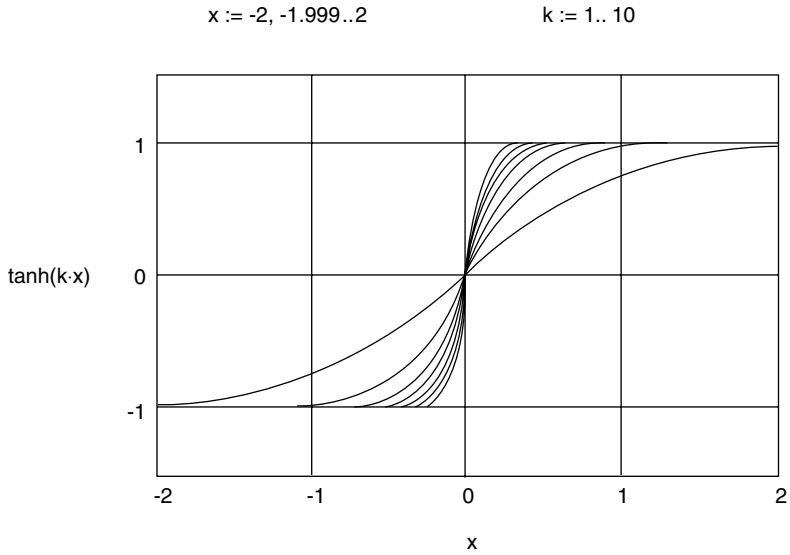

FIGURE 23.5 Hyperbolic tangent function with different steepness factors *k*.

This is true for both the continuous, as well as the discontinuous-conduction mode (referring to the current in the inductor), but in the latter case the dependence of the voltage and current ratio on the duty cycle *D* would be more complicated.

This behavior can be modeled in such a way that the switching elements in the circuit are replaced by an analog element, which is controlled by the duty cycle *D*. This element would create the same average voltages and currents that are present in the real circuit. However, since no actual switching takes place, the time step for the simulator can be increased dramatically, and the simulation could potentially run faster by a factor of 100 or more depending on the switching frequency of the original circuit. The reason is that, for a simulation of a circuit with switching elements, the time step (or, better, the time step ceiling, since the time step is adjusted dynamically in many simulators such as PSpice) must be small enough to ensure that the simulation can accurately follow the individual switching events. If the time step ceiling is too big, the simulator will try to finish the simulation run as fast as possible and internally select a time step that is just small enough so that the simulator remains stable. Remaining stable, however, does

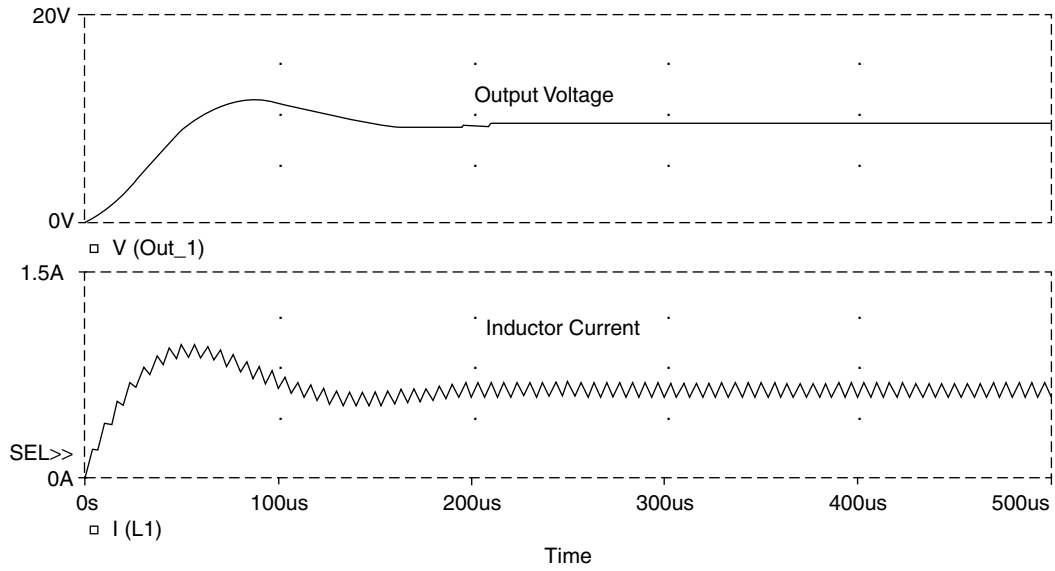

FIGURE 23.6 Simulation results for the buck converter shown in Fig. 23.1.

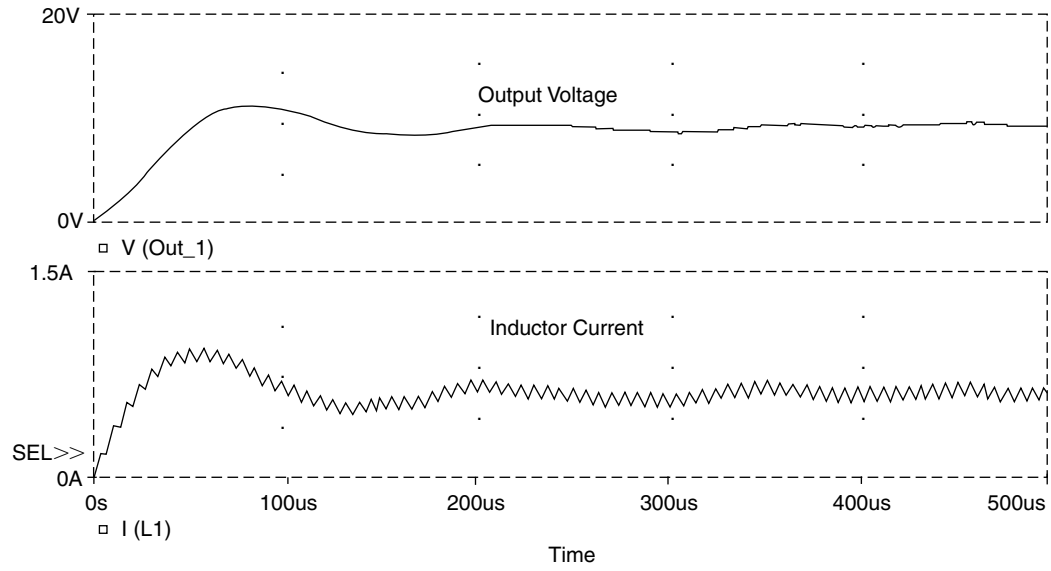

FIGURE 23.7 Incorrect simulation results for the buck converter due to improper time step settings.

not mean that the results are accurate. The size of the next time step is always predicted from the slope of the waveform just prior to the current time. If a step ceiling is set and the time step, which the simulator would choose by itself, is bigger than the time step ceiling, the time step ceiling is used instead. The choice of the proper time step ceiling requires some experience and experimentation. Figure 23.7 shows an example of a simulation that was run with a time step that is too large. It was obtained by rerunning the circuit shown in Fig. 23.1 with a different time step setting. Therefore, Fig. 23.7 can be directly compared with Fig. 23.6. It is obvious that the waveform for the inductor current in Fig. 23.7 is irregular and exhibits oscillations after the initial transient (after about 200 μs). These oscillations are caused by integration errors due to the wrong time step settings. Obviously in this example there is no reason for

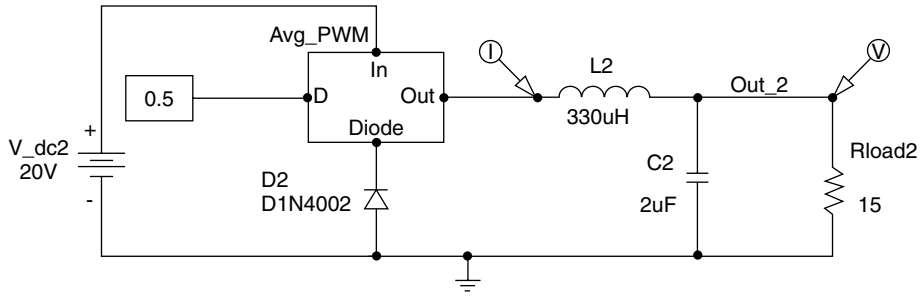

FIGURE 23.8 Buck converter with time-averaged PWM switch.

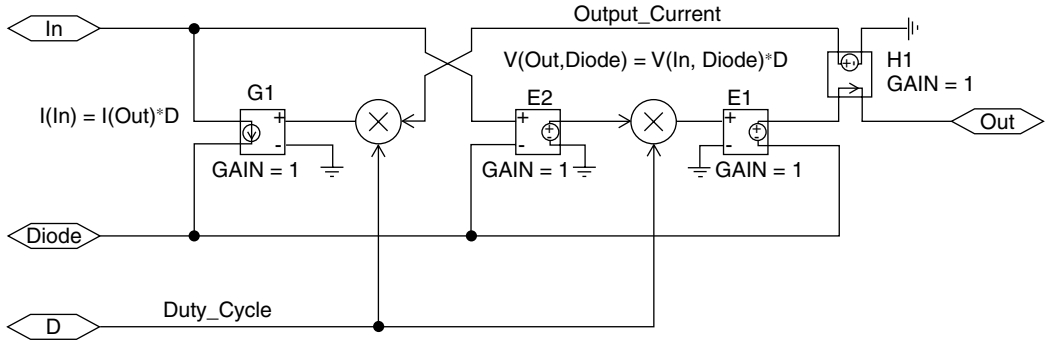

FIGURE 23.9 Subcircuit for "Avg_PWM" block.

any oscillation after the initial transient, since the circuit is run with a constant duty cycle. However, if an external feedback control system for the output voltage is present, such oscillations could occur as a result of the control action of the system, which constantly changes the duty cycle to keep the output voltage at a given value. In a case like this, considerable experience and good engineering judgment are required to avoid the wrong interpretation of the simulation results.

As mentioned above, the use of a simulation model with an analog switch replacement could be advantageous in such a case [1]. An example is shown in Fig. 23.8. A comparison with Fig. 23.1 shows that the power MOSFET "M1" has been replaced by a hierarchical block called "Avg_PWM." The associated subcircuit is shown in Fig. 23.9.

This subcircuit takes the voltage between the terminals "In" and "Diode" measured by the device "E2" and scales it with the duty cycle *D*. The output is provided between the terminals "Out" and "Diode." It should be noted that the "Diode" terminal is virtually at ground potential (about 0.7 V below due to the forward voltage of the diode) and the diode is not really needed for the operation of the circuit shown in Fig. 23.8.

The device "H1" measures the output current coming from the terminal "Out" and scales the value with the duty cycle *D*. The device "G1" will pull the scaled output current from the "In" terminal. This will implement the DC-transformer equations mentioned above. Figure 23.10 shows a comparison of the simulation output of the circuits shown in Figs. 23.1 and 23.8. It can be seen clearly, that the output of the circuit with the average PWM switch represents the "instantaneous average" (short-term average, taken over one switching cycle). In fact, if the switching frequency of the converter from Fig. 23.1 were raised high enough, the traces for both converters would be identical. This is already evident if the traces for the output voltage in Fig. 23.10 are compared since the output voltage of the switching converter has very little ripple at the chosen switching frequency of 150 kHz. Mohan [7] extends the DC-transformer approach for time-averaged modeling of H-bridge converters for motor drives.

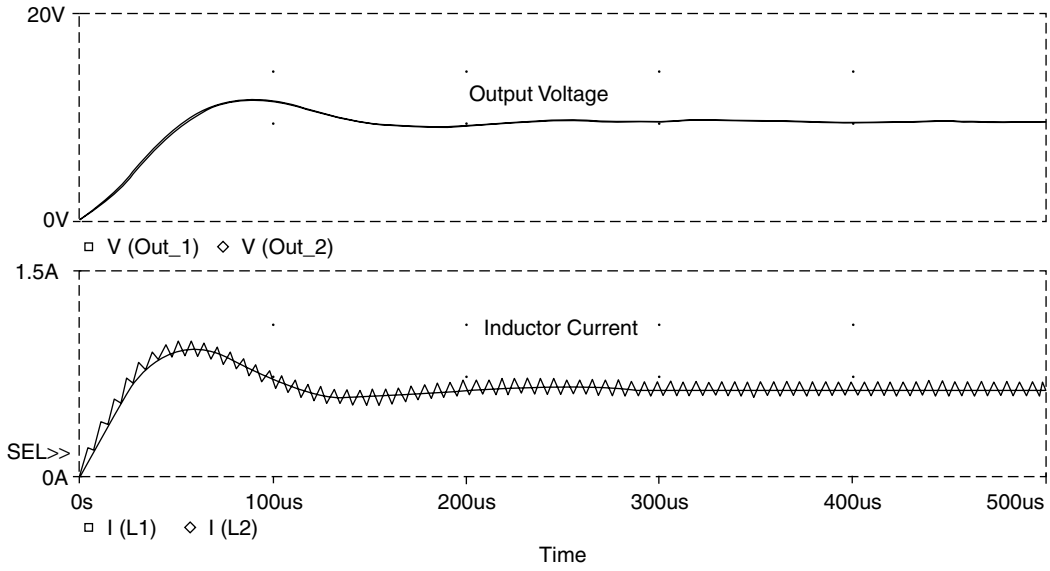

FIGURE 23.10 Combined simulation results for the buck converters from Figs. 23.1 and 23.8.

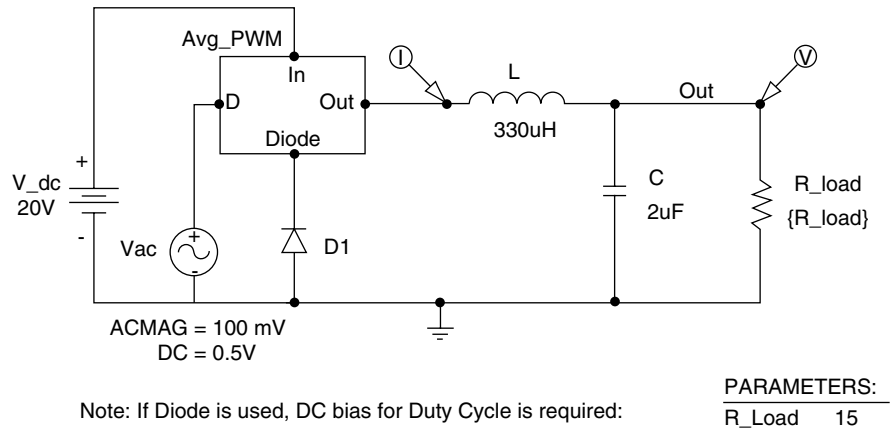

FIGURE 23.11 Simulation circuit for performing AC analysis for the buck converters from Fig. 23.8.

Besides the obvious benefit of faster simulation times, the added benefit of the buck converter with the average PWM switch is that AC or frequency domain analysis can be performed. A simulation setup for this is shown in Fig. 23.11. Here the buck converter is fed with a 50% duty cycle bias with a 10% (100 mV) AC component on top of it. The frequency of the AC component is swept from 100 mHz to 100 kHz for five different load resistors, 5W, 10W, 20W, 30W, and 40W. The result is shown in Fig. 23.12. In the upper portion of the diagram, the AC response of the output voltage is 2 V up to about 1 kHz (10% of 20 V input due to 10% AC amplitude). Above 1 kHz, the resonant peak of the LC-output filter is clearly visible for the 30-Ω load, which represents the smallest damping. Ref. 1 shows how the subcircuit in Fig. 23.9 can be used for other basic converters as well.

To model a more complex circuit, such as an H-bridge with a DC motor connected to it, in the frequency domain, each half bridge can be modeled as a DC-transformer with a transformation ratio that is controlled by the duty cycle as described in Ref. 7. As an alternative, the complete H-bridge could be modeled as a linear gain-block with the duty cycle the input and the output voltage of the H-bridge the output. This is realistic for the design of feedback control systems. In fact, H-bridge inverters for

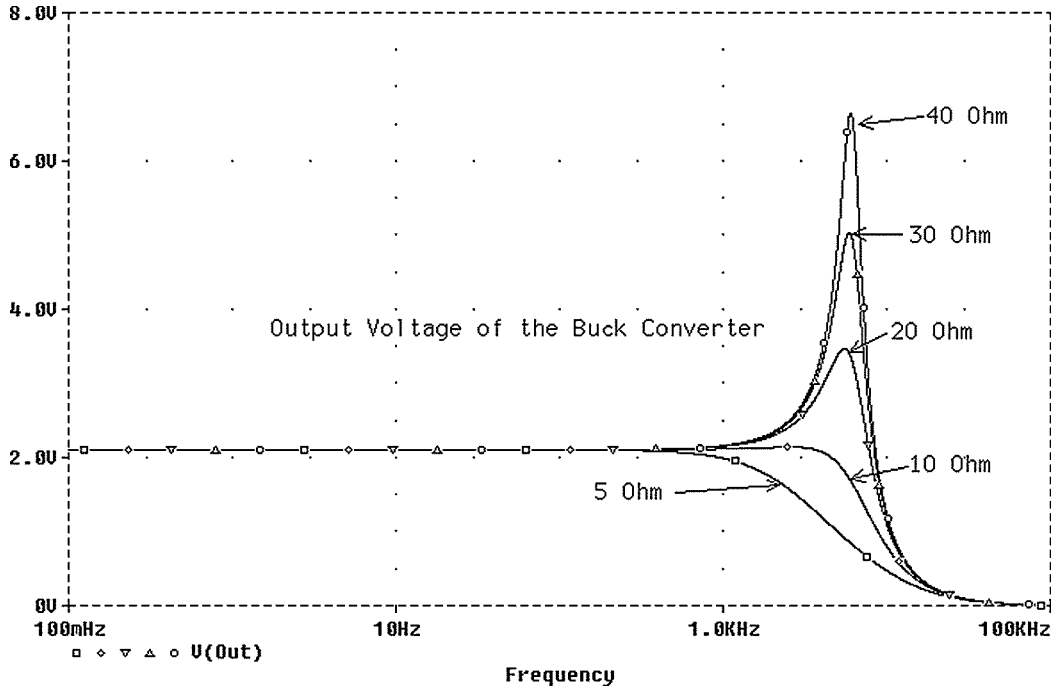

FIGURE 23.12 Simulation results for AC analysis of the buck converter from Fig. 23.11.

motor drives are often called servo-amplifiers for this reason. Some latency in the response of the amplifier could be included in the system model by adding a low-pass filter on the input.

23.4 Advanced Techniques—Simulation of a Full-Bridge (H-Bridge) Converter

In this section some advanced simulation techniques are shown using an H-bridge inverter with complementary MOSFETs. This example is part of one of the author's ongoing development projects. The goal is to build a small and efficient controller for low-voltage, high-current DC motors to be used in robotics applications. One of the design goals is the use of state-of-the-art surface-mount devices and to integrate circuit simulation into the overall design process. Figure 23.13 shows a first conceptual study for the H-bridge, which was realized entirely with parts from the PSpice student version. The upper MOSFETs ("M1" and "M3") are *p*-channel devices, whereas the lower ones ("M2" and "M4") are *n*-channel types. Therefore, and because the supply voltage is very low, no high-side drivers are needed for MOSFETs "M1" and "M3." The "PWM_Generator" is the same that was previously used, except the switching frequency is lower. Because of the well-formed signals from the PWM generator, the simulation is stable and fast. In this example bipolar switching is used, where the duty cycle controls both the polarity and the magnitude of the load current. In this mode, MOSFETs "M1" and "M4" are switched on alternating with MOSFETs "M2" and "M3."

The results of the simulation are shown in Fig. 23.14. The load, which in the final application is a DC motor with brushes, is acting as a low-pass filter for the output current. Therefore, the load current has only a relatively small amount of ripple even though the output voltage is an unfiltered PWM waveform, as shown on the upper half of Fig. 23.14. This simulation was performed to test the concept of driving both MOSFETs from the ground potential and without any special provisions for blanking time to prevent conduction overlap. The conclusion that blanking time is not needed is, however, somewhat risky, because of the previously discussed limitations for the precision of the results during switching transitions.

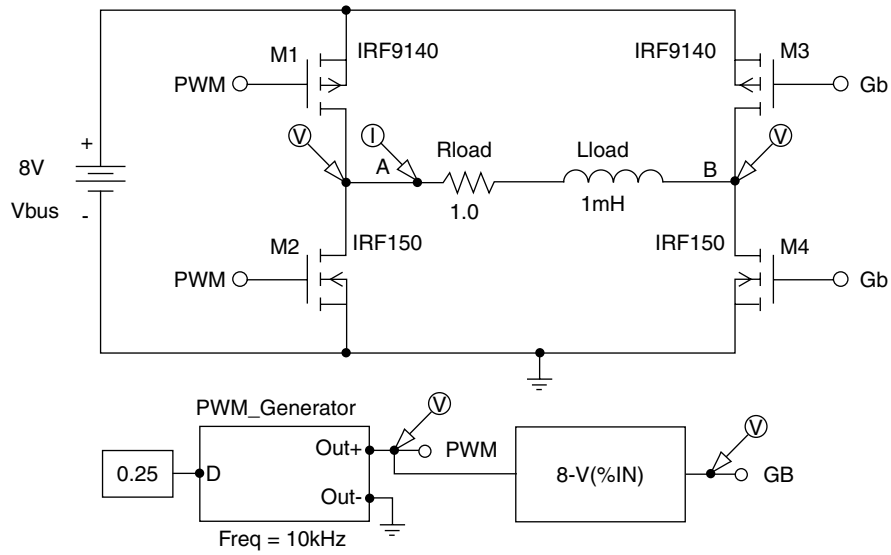

FIGURE 23.13 H-bridge with complementary MOSFETs for a low-voltage, high-current motor drive.

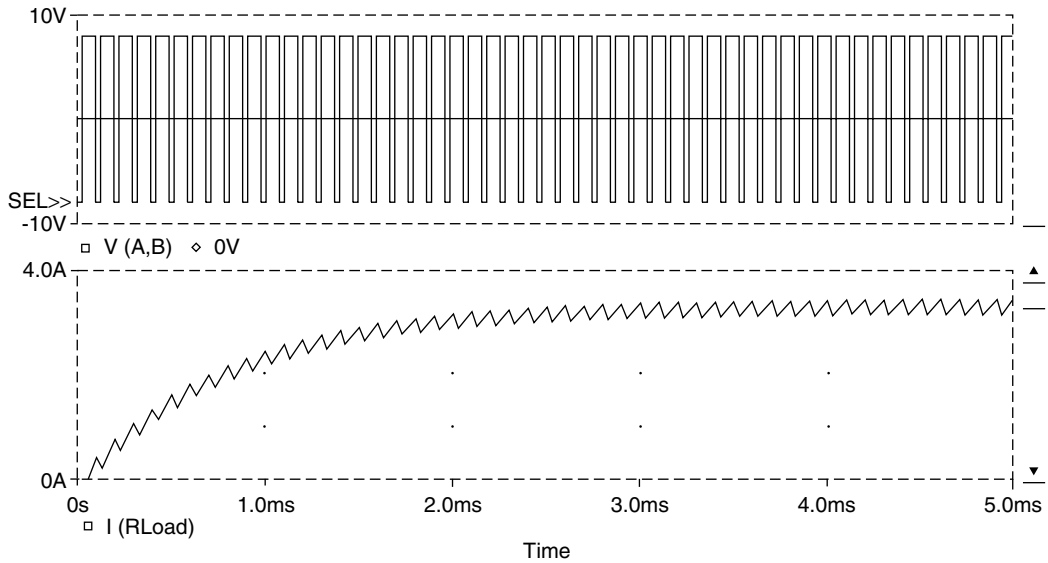

FIGURE 23.14 Diagonal voltage (upper trace) and load current (lower trace) from the circuit in Fig. 23.13.

Also, the MOSFETs used in this circuit are typically packaged in TO-220 style cases, which are not favored for the envisioned application. Therefore, to enhance the realism of the simulation study, MOSFET pairs "M1/M2" and "M3/M4" have been replaced by a custom part (NDS8858HCT from Fairchild), which represents a complementary half-bridge device packaged in a space-saving SO8 case. The new circuit is shown in Fig. 23.15. The custom part has all eight terminals of the real device and has a package definition for an SO8-type footprint, which is suitable for compact PC-board layout. In addition, the device has a "TEMPLATE" attribute, which makes it functional for simulation. During the process of "netlisting," which precedes the simulation, the "TEMPLATE" attribute generates a netlist entry. This netlist is then used as the actual input file for the simulator. Netlisting is therefore comparable with compilation of a program written in a high-level programming language. An alternative to creating a

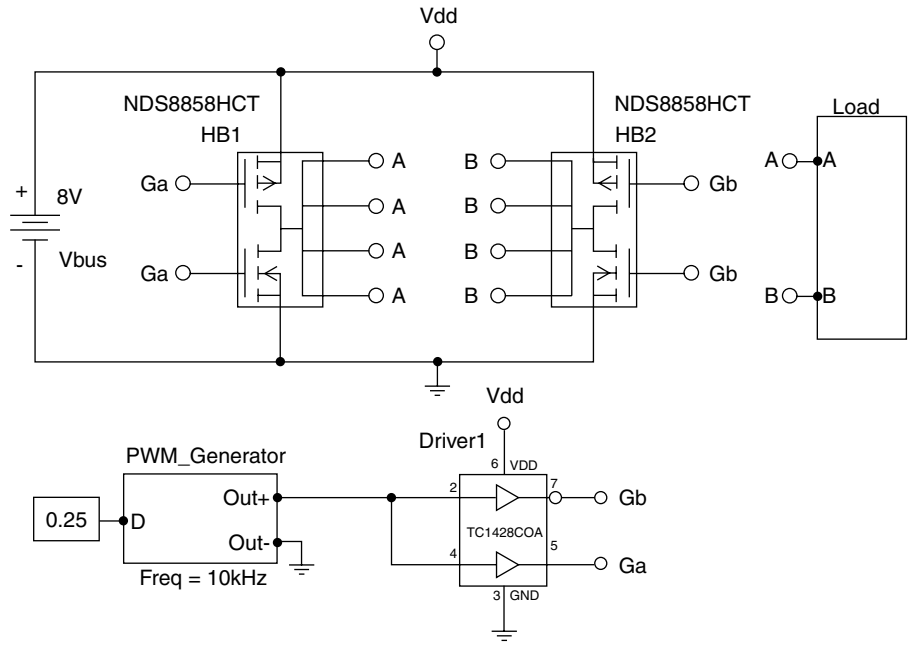

FIGURE 23.15 H-bridge circuit with custom parts for the half-bridge and the MOSFET driver.

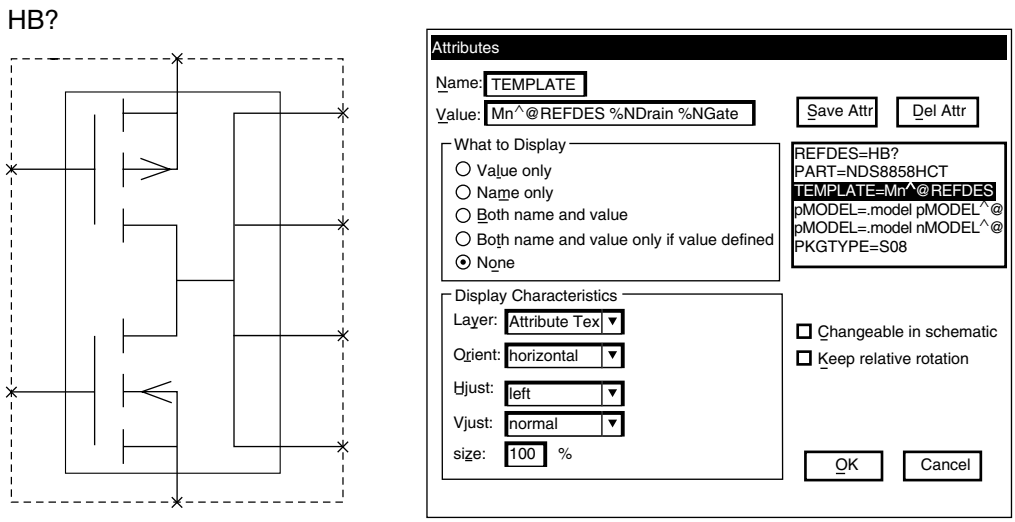

FIGURE 23.16 Symbol editor view of the NDS8858HCT complementary MOSFET half-bridge.

netlist entry via the "TEMPLATE" attribute is the creation of a subcircuit like the one shown in Figs. 23.2 or 23.9. However, if the custom part is also to be used for the generation of a PC-board, the "TEMPLATE" approach is better. Otherwise, each part in the simulation subcircuit must be listed as "SIMULATIONONLY" to prevent its inclusion on the PC-board.

```
Symbol NDS8858HCT
Definition Complementary MOSFET Half Bridge
@attributes
REFDES=HB?
PART=NDS8858HCT

TEMPLATE=Mn^@REFDES %NDrain %NGate %V-%V-nMODEL^@REFDES
\n@nMODEL

\nMp^@REFDES %PDrain %Gate %V+ %V+ pMODEL^@REFDES
\n@nMODEL

\nR1^@REFDES %Vout5  %Pdrain    1u
\nR1^@REFDES %PDrain %Ndrain    1u
\nR1^@REFDES %NDrain %Vout8     1u

pMODEL=.model pMODEL^@REFDES PMOS
\n+(Level=3 W=1 L=2u Vto=-1.8 Rd=65m Cbd=1n Cgso=1n Cgdo=1n)

nMODEL=.model nMODEL^@REFDES NMOS
\n+(Level=3 W=1 L=2u Vto=1.8 Rd=35m Cbd=1n Cgso=10n Cgdo=1n)

PKGTYPE=SO8
```

FIGURE 23.17 Attributes of the NDS8858HCT complementary MOSFET half-bridge.

In PSpice custom parts can easily be created using the built-in symbol editor. It is initiated from within the graphical "Schematics" editor. The custom symbols are saved in a file with an ".slb" extension. Figure 23.16 shows the view of the NDS8858HCT complementary MOSFET half-bridge part from the symbol editor. On the right side, the window for symbol attribute entry is shown. All the attributes for this part are listed in Fig. 23.17. The "TEMPLATE" attribute creates the simulation model for the part. It inserts five parts into the PSpice netlist, a *p*-channel and an *n*-channel MOSFET (Mn^@REFDES… and Mp^@REFDES…) as well as three "dummy" resistors to connect all four output terminals together. Note that the resistors have a value of 1 $\mu\Omega$ and are essentially short circuits. However, they are needed for the syntax of the model description if all four output terminals are to be connected to the center junction of the MOSFETs. In the expression "Mn^@REFDES," "M" stands for MOSFET, "n" designates the beginning of the name for the *n*-channel device, and "^@REFDES" inserts the path to the part including all subcircuit names as well as the name of the "Reference Designator." This is done to create a unique name for each part. The expression "\n" inserts a new line into the netlist. "\n+" inserts a new line into the netlist and places a "+" at the beginning of the new line to create a multiline expression. For clarity, a new line is inserted in the "TEMPLATE" listings in Figs. 23.17, 23.21, 23.25, and 23.27 whenever a new line designator is encountered.

The expressions for "pMODEL" and "nMODEL" specify the characteristics for the *p*- and *n*-channel MOSFETs. The data for the Rds_{on} value (Rd=) and turn on voltage (Vto=) were taken from the data sheet of the NDS8858HCT. For more advanced requirements of model fidelity, the "PARTS" program in the PSpice group can be used to determine the model parameters if they are not otherwise (Internet) available. Today, many parts manufacturers put SPICE models on their Web sites. Figure 23.18 shows a screen from the "PARTS" program using the IRF150 MOSFET as an example. In this case, the forward conduction characteristics are studied. The parameters, which are influencing the forward conduction performance (RD), are identified by an asterisk. The parameter(s) can be adjusted to match a given behavior and the performance can be displayed graphically. An example is shown in Fig. 23.19. This screen can be called by clicking on the icon under "Plot" in Fig. 23.18.

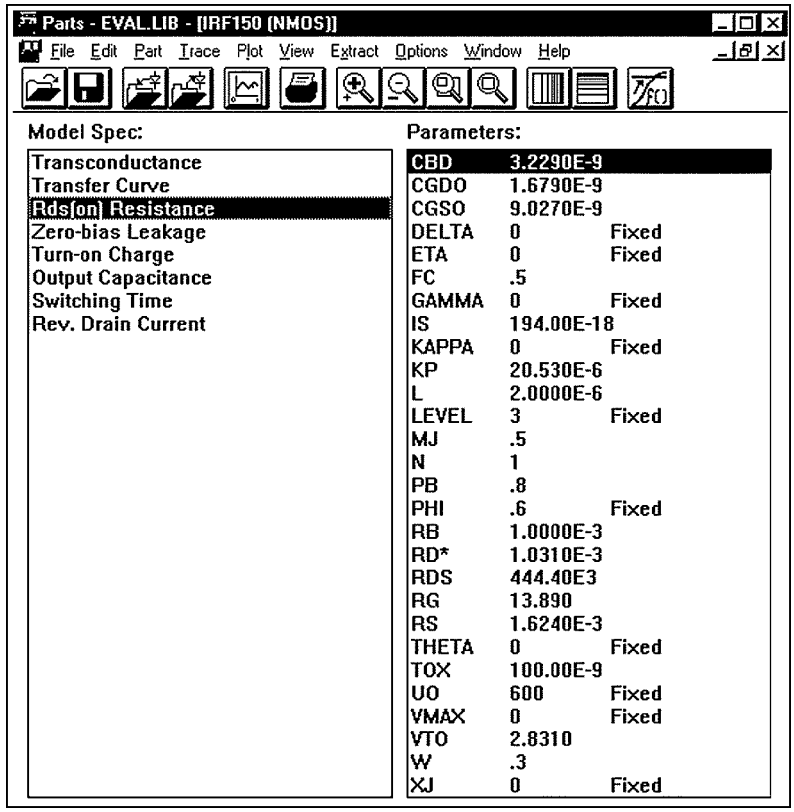

FIGURE 23.18 Screen from the "Parts" program for model parameters of the IRF150 MOSFET.

In addition to the custom part for the half-bridge, another custom part, implementing a MOSFET driver (TCI428COA from Telcom) packaged in an SO8 case, was created. For this part, a simulation model as well as a PC-board definition was included. The simulation model was created using ABM devices (basically voltage-controlled voltage sources with many options to define the control functions), to closely match the information in the data sheet. Again, the simulation model is contained in the "TEMPLATE" attribute. Figure 23.20 shows the view of the TCI428COA driver in the symbol editor. The pins 1 and 8 are marked as not included in the simulation (drawn with an interruption) since they are "not-connected" (NC) in the real part.

Voltage-controlled voltage sources, using a conditional statement—IF (Condition, Output for True, Output for False)—as their control function, have been used to implement a model for this part. The full set of attributes is shown in Fig. 23.21. The model compares the input voltage with the switching threshold (3 V) and switches between "GND" and "VDD" ±25 mV when it is crossed. A 10-Ω resistor models the output resistance for each channel. The threshold and saturation values as well as the output resistance were taken from the data sheet.

Figure 23.22 shows the simulation output of the H-bridge shown in Fig. 23.15 using the special parts described above. The output resistance of the driver in conjunction with the gate capacitance of the MOSFETs achieves smooth gate drive waveforms, which keep the simulator fast and stable and ensure realistic rise and fall times of the gate drive signals at the same time. In this circuit, the duty cycle controls both the direction and magnitude of the load current. This switching scheme is called bipolar switching. One of its disadvantages is that, without a control signal, the load (motor) is always driven at full power in one direction. Therefore, the circuit shown in Fig. 23.23 was developed. It uses a four-channel driver as

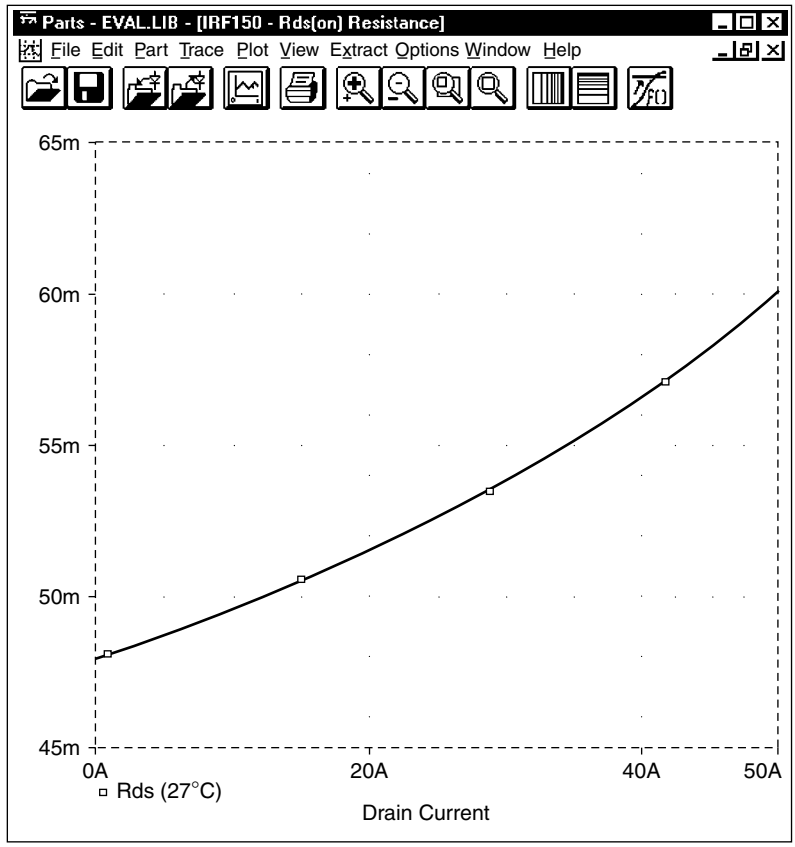

FIGURE 23.19 Graphical representation of the forward conduction of the IRF150 MOSFET.

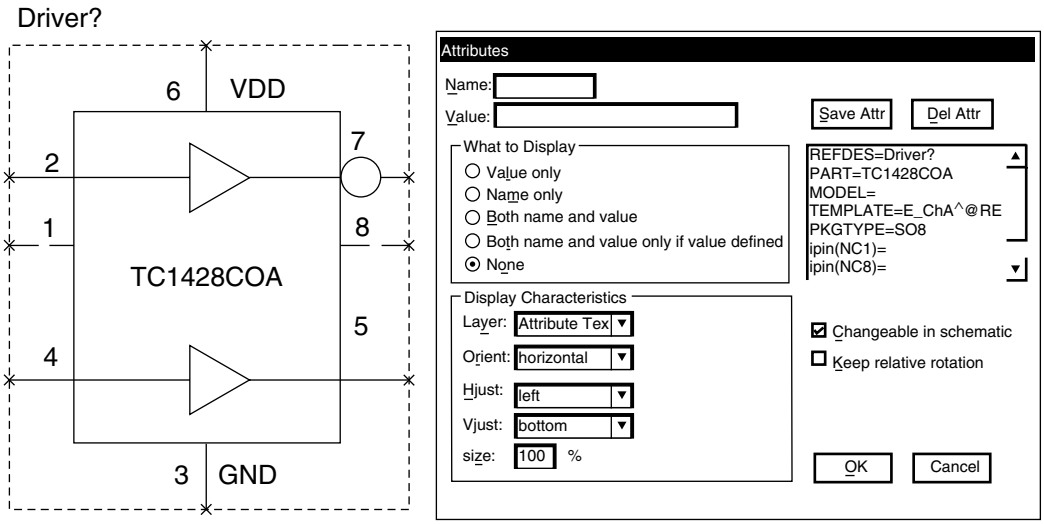

FIGURE 23.20 Symbol editor view of the TC1428COA MOSFET driver.

```
Symbol TC1428COA
Definition Complementary MOSFET Driver SO8
@attributes
REFDES=Driver?
PART=TC1428COA
MODEL=

TEMPLATE=E_ChA^@REFDES OutA^@REFDES,%GND VALUE {IF(V(%InA,%GND)>3.0V
V(%GND) + 25mV, V(%VDD,%GND)-25mV)}

\nE_ChB^@REFDES OutB^@REFDES,%GND VALUE {IF(V(%InB,%GND)>3.0V,
V(%VDD,%GND)-25mV, V(%GND)+25mV)}

\nR_OutA^@REFDES OutA^@REFDES,%OutA @Rout
\nR_OutB^@REFDES OutB^@REFDES,%OutB @Rout

PKGTYPE=SO8
ipin(NC1)=
ipin(Nc8)=
Rout=10.0
```

FIGURE 23.21 Attributes of the TC1428COA driver.

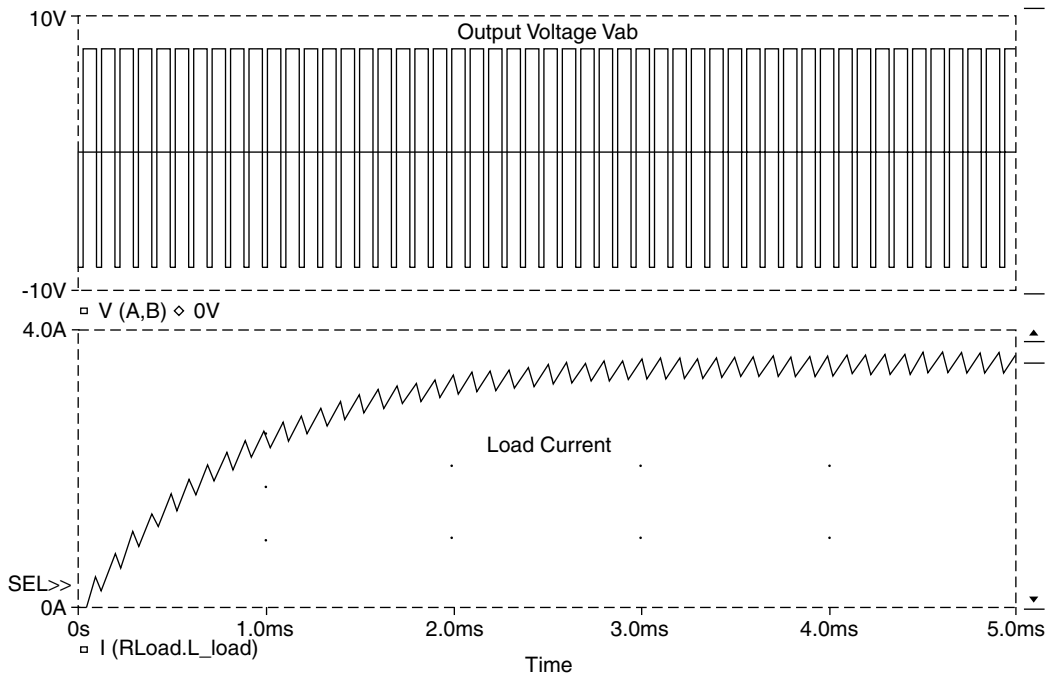

FIGURE 23.22 Simulation output for the circuit shown in Fig. 23.15.

well as an opto-coupler for isolation. In this circuit, one of the upper MOSFETs is turned on permanently for a particular direction, while the diagonally opposed (lower) MOSFET controls the speed. As an additional feature, when a lower MOSFET turns off, the MOSFET above it (which is normally not conducting for the selected direction) is turned on to shunt the current away from the body diode of the upper MOSFET and improve the efficiency. This is also referred to as synchronous rectification.

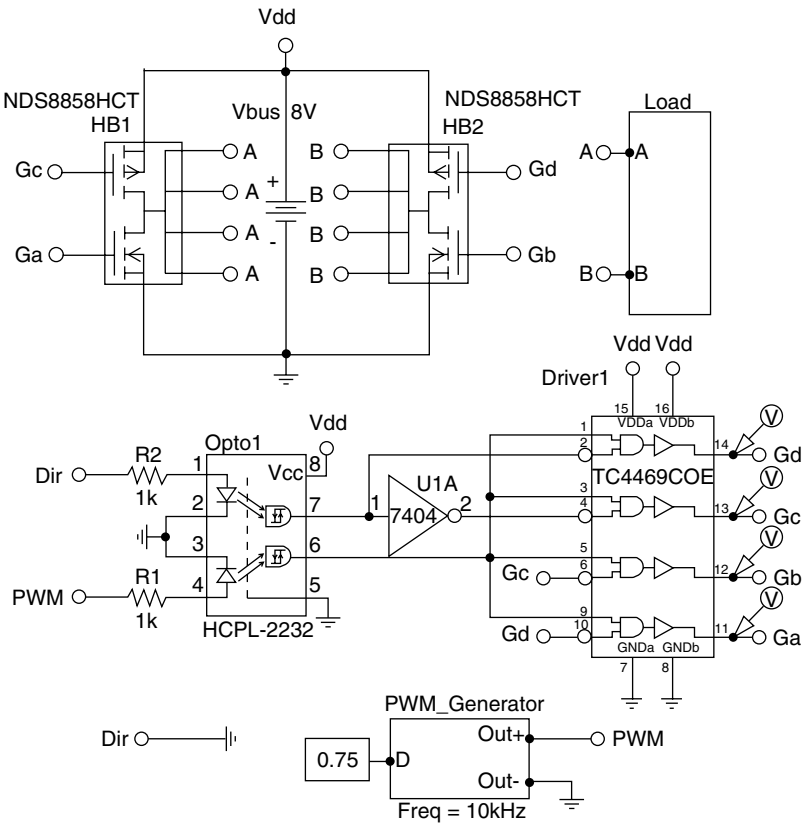

FIGURE 23.23 Advanced H-bridge circuit with synchronous rectification.

Figure 23.24 shows the view of the TC4469COE MOSFET driver from inside the symbol editor on the left side and the results of a test to check the device logic on the right side. The implementation of the simulation model is very similar to the implementation of the model for the TC1428COA driver. In fact, the "TEMPLATE" attribute from the TC1428COA was copied and then edited to implement the logical AND gates, which have one inverted and one noninverted input. The logical AND function was realized using the "&" symbol in the conditional statement inside of the IF function that controls the output voltage for each channel. Output resistors on all four outputs complete the model description. A complete list of all attributes, with the precise details of the implementation of the four-channel driver, is given in Fig. 23.25.

The correctness of the logic functions of the TC4469COE driver can be verified by inspection of the picture in the right half of Fig. 23.24. This picture was obtained by simulating a circuit containing one TC4469COE driver, a voltage source, and four termination resistors for each output. For clarity, only the TC4469COE driver is shown. In the simulation setup for this circuit, only the bias point calculation was selected. To check the logic, all possible combinations of "High" and "Low" signals were connected to the inputs. The result was made visible by selecting to display bias point voltages on the "Schematics" page. This test is another example of the model validation and verification process mentioned before.

As mentioned before, the circuit shown in Fig. 23.23 contains an opto-coupler. This particular device has an internal dielectric shield giving it a high immunity to dV/dt common mode swings on the output. Therefore, it is very suitable for use in power electronics applications. To be able to simulate the circuit containing this part, a simulation model for the opto-coupler was developed. However, in this specific case, the ability to run the simulation with the inclusion of the opto-coupler is more of a convenience than a necessity to check the performance of the circuit. In this application, the demands on the opto-coupler

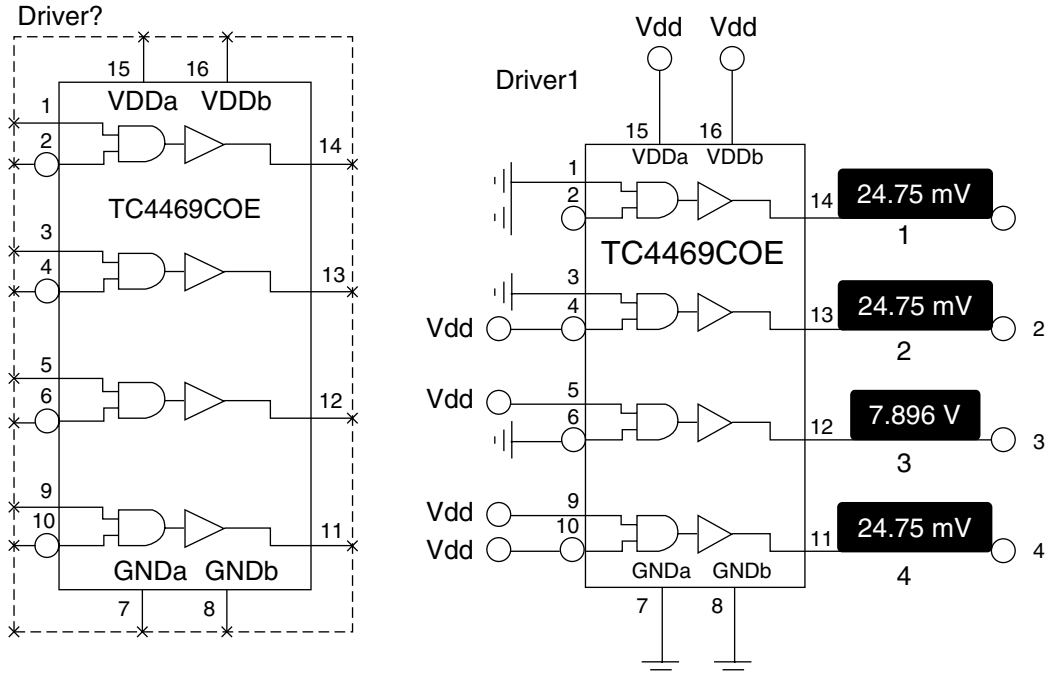

FIGURE 23.24 Symbol editor view and test output of the TC4469COE MOSFET driver.

```
Symbol TC4469COE
Definition Quad MOSFET Driver SOIC (Wide)
@attributes
REFDES=Driver?
PART=TC4469COE
MODEL=

TEMPLATE=E_Ch1^@REFDES Out1^@REFDES,%GNDa VALUE {IF(V(%1A,%GNDa)
>@Threshold & V(%1B,%GNDa)<@Threshold, V(%VDDa, %GNDa)-25mV, V(%GNDa)+25mV)}

\nE_Ch2^@REFDES Out2^@REFDES, %GNDa VALUE {IF(V(%2A,%GNDa)>@Threshold &
V(%2B,%GNDa)<@Threshold, V(%VDDa, %GNDa)-25mV, V(%GNDa)+25mV)}

\nE_Ch3^@REFDES Out3^@REFDES, %GNDa VALUE {IF(V(%3A,%GNDa)>@Threshold &
V(%3B,%GNDa)<@Threshold, V(%VDDa, %GNDa)-25mV, V(%GNDa)+25mV)}

\nE_Ch4^@REFDES Out4^@REFDES, %GNDa VALUE {IF(V(%4A,%GNDa)>@Threshold &
V(%4B,%GNDa)<@Threshold, V(%VDDa, %GNDa)-25mV, V(%GNDa)+25mV)}

\nR_Out1^@REFDES Out1^@REFDES,%1Y @Rout
\nR_Out2^@REFDES Out2^@REFDES,%2Y @Rout
\nR_Out3^@REFDES Out3^@REFDES,%3Y @Rout
\nR_Out4^@REFDES Out4^@REFDES,%4Y @Rout

PKGTYPE=SO16W
ROUT=10.0
Threshold=1.5V
```

FIGURE 23.25 Attributes of the TC4469COE driver.

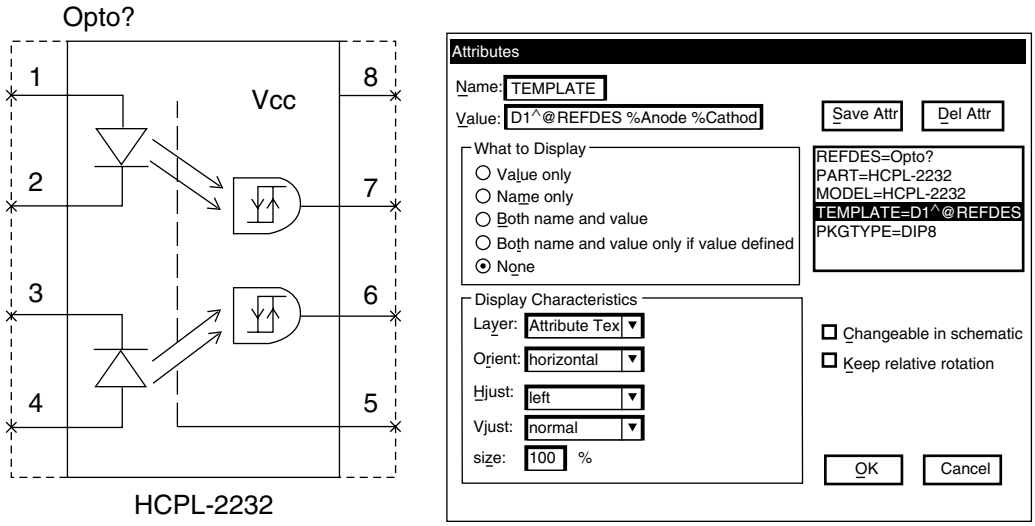

FIGURE 23.26 Symbol editor view of the TC4469COE MOSFET driver.

are quite low. The data throughput is much less than the actual bandwidth, the voltage stress is minimal, and the output does not have to drive a large load. Therefore, a rather simple model is adequate for this case. As a general rule, it is always recommended to evaluate how hard each circuit element is pushed to its limits. The harder a part is pushed, the more its behavior typically deviates from an ideal model. After this step, the choice of models becomes clearer. Complex and sophisticated models are typically needed for the parts that are pushed hard, whereas simpler models will be adequate for the parts that are operated well below their limits. It is also not recommended to use a complex model just because it is available when it is not needed. This is because the complex model typically slows the simulation down and creates stability problems.

With this said, the particular model used here for the opto-coupler can be discussed. The model comprises an input diode and a voltage-controlled voltage source for the output. Obviously, the LED current should be monitored and the output should respond to it. However, for a diode used in a simulator, the device characteristics are totally stable and voltage and current are perfectly correlated. Therefore, the model can monitor the forward diode voltage and switch the output voltage, whenever a certain level is crossed. It is also possible to include hysteresis into the model, but this is not done here. Obviously, in many other applications, in which the opto-coupler is pushed harder, a more complex model must be created. The symbol editor view of the dual channel opto-coupler is shown in Fig. 23.26. The details of the attributes including the "TEMPLATE" attribute, which defines the simulation model described above, are shown in Fig. 23.27. The easiest way to get an ASCII printout of all the attributes is to "Export" the symbol from within the symbol editor. This will create an ASCII file (extension ".sym") containing the full text of all attributes as well as textual descriptions of all the graphical elements in a format resembling postscript.

For clarity, the graphical descriptions for all custom parts have been omitted since they are better shown directly as in Fig. 23.26. The symbol file could, however, be e-mailed and be re-imported into an ".slb" library and thereby create a fully functioning part with all the graphical elements of the original.

The simulation results for the circuit shown in Fig. 23.23 are virtually the same as the results shown in Fig. 23.22 since the load is the same as in the circuit in Fig. 23.15. The load is a 1-Ω resistor in series with a 1-mH inductor as shown in Fig. 23.13. This simple load was replaced by a complete model for a DC motor in the circuit shown in Fig. 23.28. This circuit is suitable both for simulation and for creation of a PC-board. The reader may note a few more additions compared with the circuit shown in Fig. 23.23. One important addition is the block called "Cbuffer." This block contains a total of ten capacitors to

```
Symbol HCPL-2232
Definition Dual Logic Gate Optocoupler
@attributes
REFDES=Opto?
MODEL=HCPL-2232

TEMPLATE=D1^@REFDES %Anodel %Cathode Model^@REFDES
\nD2^@REFDES %Anode2 %Cathode2 Model^@REFDES
\n.model Model^@REFDES D
\nEout1^@REFDES %Vol, %Gnd VALUE {IF(V(%Anodel, %Cathodel)>0.65V, V(%Vcc,%Gnd), 25mV)}
\nEout2^@REFDES %Vo2, %Gnd VALUE {IF(V(%Anode2, %Cathode2)>0.65V, V(%Vcc,%Gnd), 25mV)}

PKGTYPE=DIP8
```

FIGURE 23.27 Attributes of the HCPL-2232 opto-coupler.

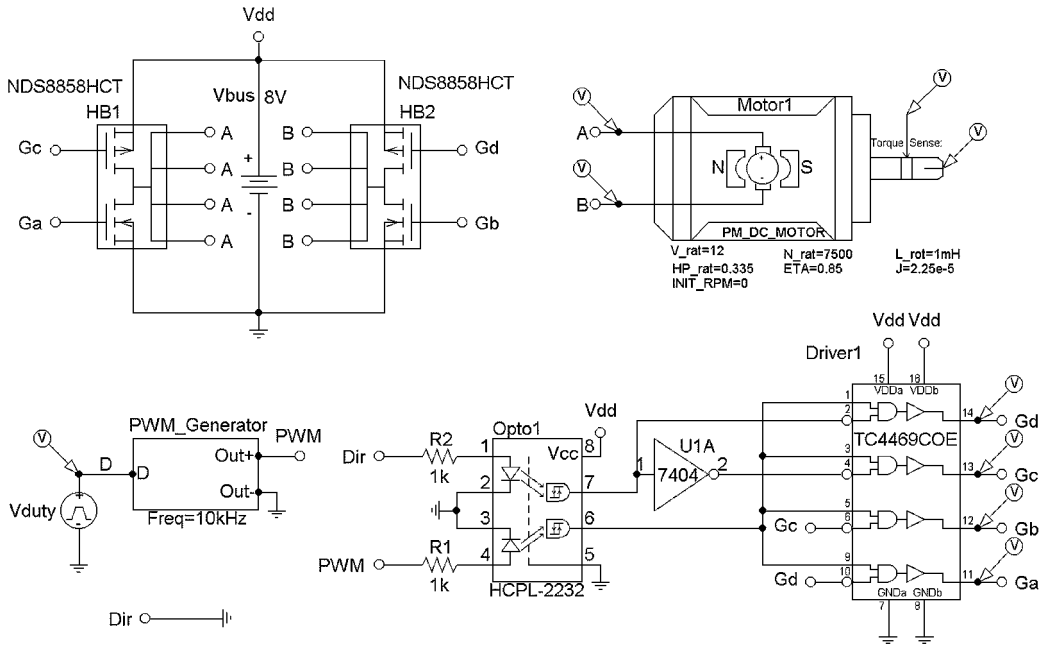

FIGURE 23.28 Advanced H-bridge circuit with motor load for simulation and PC-board layout.

buffer the DC bus. The associated subcircuit is shown in Fig. 23.29. The capacitors are not necessary for simulation, since the supply voltage "Vbus" is ideal, but needed for the real circuit.

To avoid start-up transients, all capacitors have been initialized to the bus voltage. For the PC-board layout, surface mount packages were chosen for all buffer capacitors.

In addition, parts "P1" and "P2" are screw terminals and "P6" is a 20-pin header. These parts are needed for connections to the PC-board but have no simulation model. Note also, that "Vss" is used for the ground on the primary side of the opto-coupler. Otherwise, the grounds on the PC-board would be joined and the isolation of the opto-coupler would be defeated. To be able to simulate the circuit anyway, "Vss" is tied to ground with a large resistor, which has a "SIMULATIONONLY" attribute to prevent it from being included in the circuit board layout. Also, the hidden power supply pins of the inverter "U1A" have been set to "Vdd" and "GND" so that they are connected on the PC-board.

Figure 23.30 shows the DC motor as viewed from the symbol editor. Because of the complexity of the simulation model it was not implemented using the "TEMPLATE" attribute. Instead, a subcircuit was used.

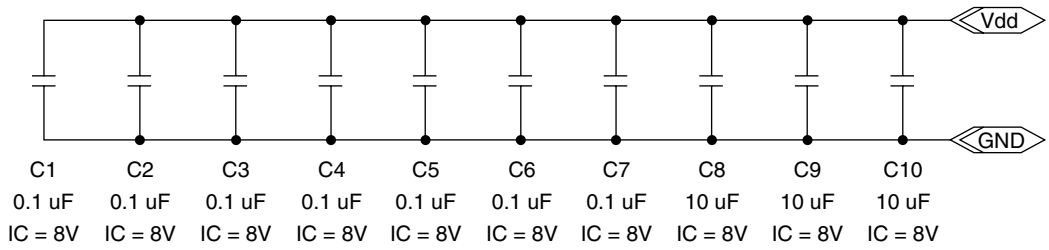

FIGURE 23.29 Subcircuit for the DC bus buffer caps.

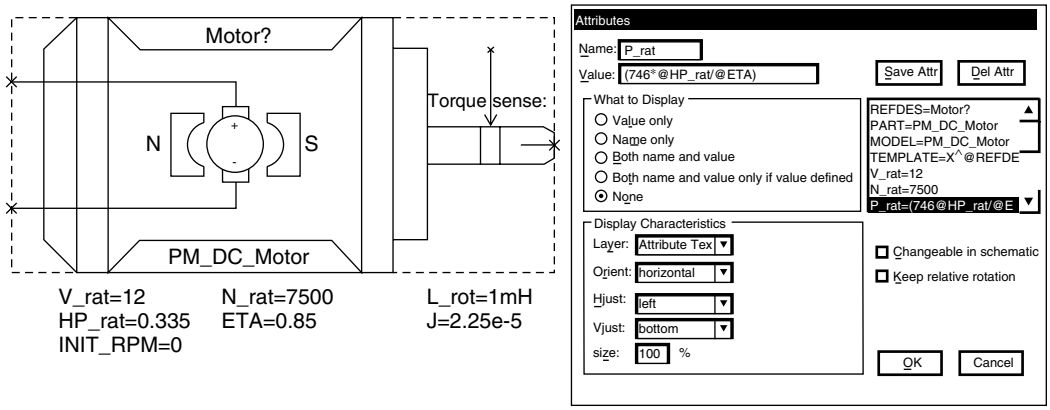

FIGURE 23.30 View of the DC motor symbol from the symbol editor.

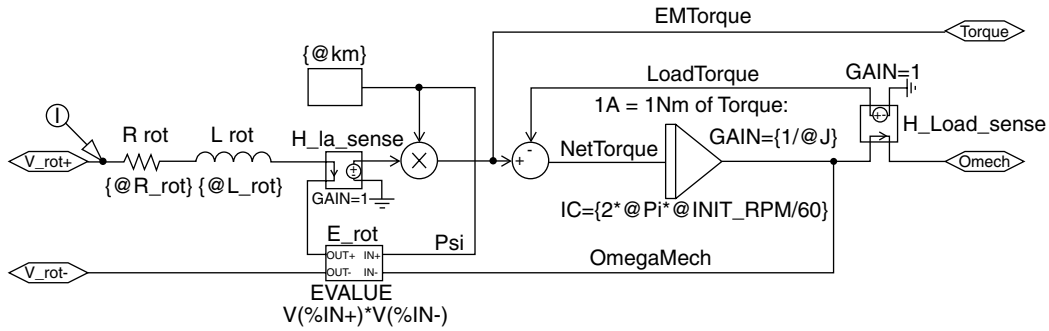

FIGURE 23.31 Schematic for the model of the DC motor.

This subcircuit is shown in Fig. 23.31. All parts are marked as "SIMULATIONONLY." The pins of the motor symbol are connected to the interface ports with the same names. This subcircuit implements both the electrical behavior of the motor and the mechanical acceleration and deceleration. The electrical circuit consists of the resistor "Rrot" and the inductor "Lrot" and the rotational voltage "E_rot". The induced voltage is equal to the motor constant "Km" multiplied with the mechanical speed. The voltage on the pin on the shaft of the machine ("Omech" terminal) is directly equivalent to the mechanical speed according to the relation: 1 V = 1 rad/s. The model also calculates the internally developed torque and makes the value available at a monitoring pin. The torque is calculated as the product of the machine constant "Km" and the rotor current, which is monitored by "H_Ia_sense". The current coming from the "Omech" terminal represents the load torque according to the relation 1 A = 1 Nm. After subtraction of the load torque from the internally generated torque, the mechanical speed is calculated by integrating the net torque weighted by the moment of inertia "J". For more details on machine theory, the reader is referred to Ref. 3.

```
Symbol PM_DC_MOTOR
DefinitionDC_Motor, Permanent Magnet
@attributes
REFDES=Motor?
PART=PM_DC_Motor
MODEL=PM_DC_Motor
TEMPLATE=X^@REFDES %V_rot+%V_rot-%Omech%Torque @MODEL
V_rat=12
N_rat=7500
P_rat=(746*@HP_rat/@ETA)
R_rot=((@V_rat-@Km*@Omega_rat)/@I_rat)
L_rot=1mH
J=2.25e-5
Pi=3.14159265
Omega_rat=(2*@Pi*@N_rat/60)
Km=(@Torque_rat/@I_rat)
HP_rat=0.335
ETA=0.85
Torque_rat=(746*@HP_rat/@Omega_rat)
INIT_RPM=0
I_rat=(@P_rat/@V_rat)
@views
DEFAULT=PM_DC_Sub.sch
```

FIGURE 23.32 Complete list of attributes for the DC motor symbol.

The subcircuit shown in Fig. 23.31 has no actual model parameters within it and is therefore truly generic. The model parameters are associated with attributes of the motor symbol itself and are passed on to the subcircuit via the "@" symbol. A complete list of all attributes of the motor symbol is shown in Fig. 23.32. The information for this figure was extracted from an exported symbol (".sym") file.

Figure 23.33 shows the results of the simulation of the most advanced H-bridge circuit shown in Fig. 23.28. Shown are the motor current and the increasing mechanical speed.

As mentioned before, the circuit shown in Fig. 23.28 can be used not only as an input for the PSpice simulator, but also as a schematic entry page for a PC-board layout to implement the circuit. This makes simulation an integrated part of the design and analysis process. Figure 23.34 shows a view of the circuit board layout program, which uses the layout netlist from the circuit shown in Fig. 23.28. It should be mentioned that the layout netlist is different from the simulation netlist (these are different files) because it was used for a different purpose. The view of Fig. 23.34 shows the footprints for all parts and their relative placement as well as the connections between them before the trace routing process. The logical connections between the parts are often referred to as "ratsnest."

23.5 Conclusions

In this chapter, the simulation of power electronics circuits has been discussed starting from principal issues to advanced techniques, including the integration of the simulation into the overall design process. The proper uses of simulation techniques, as well as limitations and potential pitfalls, have been discussed in detail. In addition, the generation of custom symbols, which can be used both for simulation as well as for PC-board layout has been shown in a number of examples. The philosophy behind the creation of simulation models for custom parts has been discussed. It should also be acknowledged, that there are many more excellent software packages for circuit simulation available. For some of the more popular

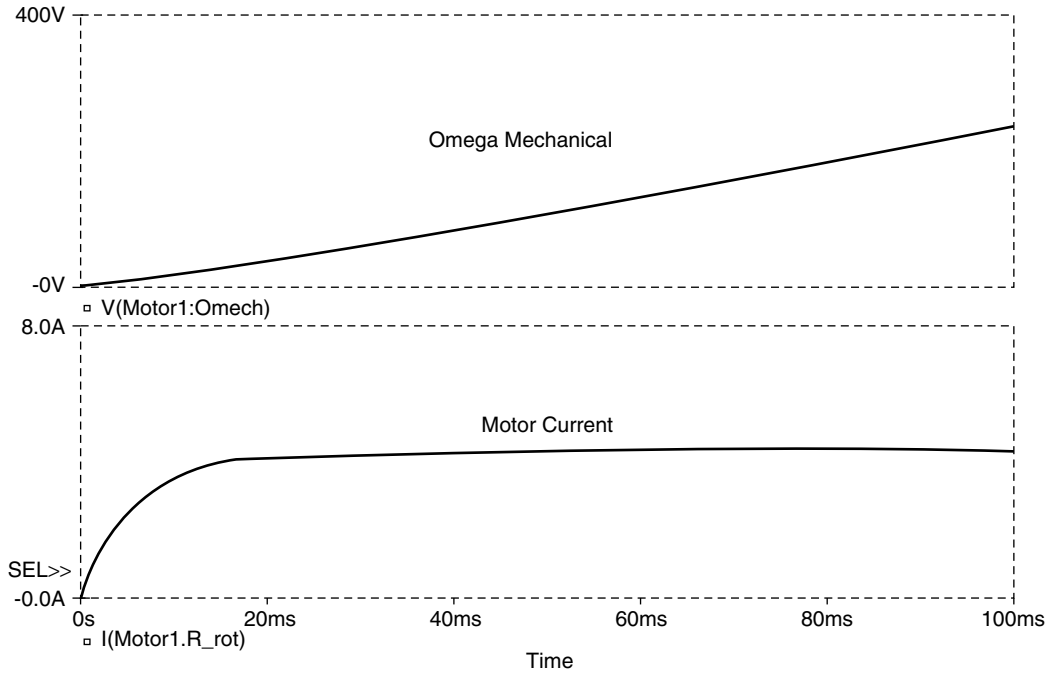

FIGURE 23.33 Simulation results for the circuit in Fig. 23.28.

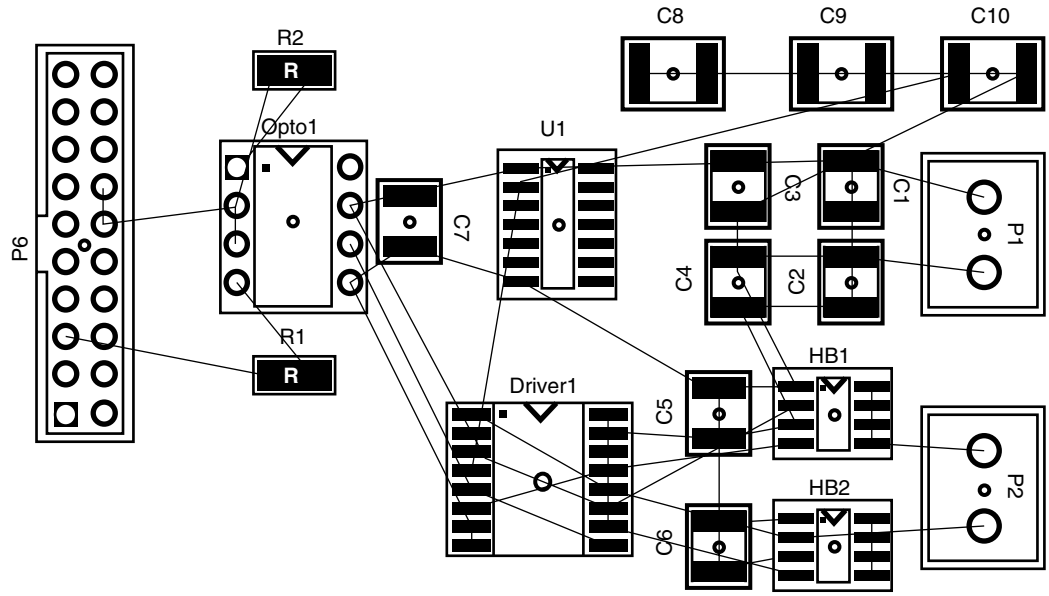

FIGURE 23.34 View of the PC-board layout for the circuit shown in Fig. 23.28.

ones, the reader is referred to Refs. 5, 12, and 13. In addition, Refs. 10, 11, and 14 give further information on simulation techniques for power electronics.

References

1. Giesselmann, M. G., Averaged and cycle by cycle switching models for buck, boost, buck-boost and cuk converters with common average switch model, in *Proceedings of the 32nd Intersociety Energy Conversion Engineering Conference,* IECEC-97, Honolulu, HI, Jul. 27–Aug. 01, 1997.
2. Kassakian, J. G., Schlecht, M. F., and Verghese, G. C., *Principles of Power Electronics,* Addison-Wesley, Reading, MA, 1991.
3. Krause, P. C., Wasynczuk, O., and Sudhoff, S. D., *Analysis of Electric Machinery,* IEEE Press, New York, 1995.
4. MathCAD® 2000 Professional, MathSoft Engineering & Education, Inc., 101 Main Street, Cambridge, MA 02142-1521, http://www.mathsoft.com.
5. MircoCap, SPECTRUM SOFTWARE, 1021 S. Wolfe Rd., Sunnyvale, CA 94086, http://www.spectrum-soft.com.
6. Mohan, N., Undeland, T., and Robbins, W., *Power Electronics: Converters, Applications, and Design,* 2nd ed., John Wiley & Sons, New York, 1995.
7. Mohan, N., *Electric Drives, An Integrative Approach,* MNPERE, Minneapolis, MN, 2000.
8. PSpice® Documentation, 555 River Oaks Parkway, San Jose, CA 95134; (408) 943 1234; http://pcb.cadence.com/.
9. Rashid, M. H., *Power Electronics: Circuits, Devices, and Applications,* Prentice-Hall, Englewood Cliffs, NJ, 1993.
10. Rashid, M. H., *SPICE for Power Electronics and Electric Power,* Prentice-Hall, Englewood Cliffs, NJ, 1993.
11. Rashid, M. H., S*PICE for Circuits and Electronics Using PSpice,* Prentice-Hall, Englewood Cliffs, NJ, 1990.
12. Saber® Mixed Circuit Simulator, Avant! Corporation, 46871 Bayside Parkway, Fremont, CA 94538; (510) 413 8000; info@avanticorp.com.
13. Simplorer®, Technical Documentation, SIMEC Corporation, 1223 Peoples Avenue, Troy, NY 12180; (518)-373-5838; Info@Simplorer.com.
14. Tuinenga, P. W., *SPICE, A Guide to Circuit Simulation & Analysis Using PSpice,* Prentice-Hall, Englewood Cliffs, NJ, 1988.