Atmel AVR-based Constant Current Supply

By Jeffrey R. Skinner P.E., Chief Engineer, Controltek Inc.

I. Application Requirements:

In September of 2002, a new client approached Controltek requesting design services for a constant current power supply. This supply would receive a regulated 48-volt input and was required to generate a user adjustable constant current output in the range between 3 and 8 Amps. Current regulation including ripple was to be less then ±10% of selected output level. This circuit had to operate over a varying load range between 1_ and 16_ and drive four such independent loads switched in one at a time. In addition the constant current supply is be followed by an H-bridge that switches the direction of current flow through the load at fixed intervals. Regulatory agency approval to UL 508 and CSA 22.2 was also required. The supply had to operate over a temperature range between -20˚C to +70˚C in a NEMA 4 enclosure with no ventilation or external heat sinking.

II. Architectural Approach:

For flexibility, field upgradability, cost, minimum component count and to minimize PCB real-estate requirements, we decided to use a small footprint, low cost microcontroller, implementing current regulation in a software based algorithm rather then in analog hardware. Constant current control would be done in a buck regulator topology with current feedback. The more functions we needed in the circuit that were provided internal to the microcontroller, the fewer additional parts would be required. This would save both cost and PCB real estate. Criteria for the selection of a microcontroller were cost, packaging and an internal peripheral set matching our needs. Adding in the requirement to adjust the target output current over a range between 3 and 8 Amps drove the overall peripheral set requirements for the microcontroller. We would need a two-channel analog to digital converter and a single PWM output. Additionally, the microcontroller needed to have internal FLASH program memory, static RAM and the ability to be in circuit programmed.

Before a decision on the best-fit microcontroller for this application could be made, we needed to know the required resolution and accuracy of the analog to digital converter and the required resolution and frequency for the PWM output. PWM frequency is matched to the inductor and determines the amount of ripple current. Higher switching frequencies result in smaller inductor values and thus smaller magnetic components. Higher switching frequencies also result in greater switching losses in the pass transistor and thus greater power dissipation. Resolution and accuracy of the analog to digital converter is determined by the required current regulation and by the feedback control algorithms stability requirements.

III. Detailed Design Requirements:

A simulation model for the circuit was created in PSPICE as an aid in determining inductor size and PWM frequency. This same model was also used to verify the control algorithm. Initially a P-channel MOSFET was selected for the pass transistor to avoid the high side gate drive requirements of an N-channel device. With an input voltage of 48 volts, commercially available high side gate drive IC's are very limited. Due to the potential switching losses a push-pull gate drive was used to speed up the turn off time. To meet the requirement of total current error less then ±10% of selected output; the total root sum squared error from all contributing sources had to be less then 90 mvolts. This corresponds to 10% of the minimum output current level of 3.0 amps based on using a 3.0 volt reference for the analog to digital converter, a 0.05_ sense resistor and a differential op-amp with a gain of 6 in the feedback path. With this set of values, and assuming an 8 bit analog to digital converter, full scale input range to the analog to digital converter would be 10 Amps and scaling in the circuit would be as follows:

(10 Amps)(0.05_)(6) = 3.0 volts = reference voltage
1 volt = 3.33 Amps
1 bit = 3.0 volts/255 = 0.0118 volts per bit or 0.039 Amps per bit.

The Atmel ATtiny15L AVR microcontroller was selected for this project because of its small footprint, internal FLASH memory, in circuit programmability and high speed PWM. Additionally, the potential to make use of its internal voltage reference and differential amplifier was attractive. Unfortunately, the accuracy of the internal voltage reference is not very good. With ±160 mvolts of error its contribution alone would exceed the 90 mvolts allowed by the requirements. Also the differential amplifier gain setting is limited to values of only 1 and 20, neither of which fit well in this application. Using a gain of 1 would require a sense resistor of 0.3_ and at 8 Amps this would dissipate 19.2 watts, not a good choice for a sealed NEMA 4 enclosure with out any external heat sinking. A gain of 20 would require a sense resistor value of only 0.015_ which is equivalent a 1 in length of board trace using 1 oz. Copper 1/32 inch wide. At this low of an ohmic value, trace resistance would cause significant errors in the feedback signal. As a result, the decision was made to use an external voltage reference and operational amplifier. Because the Atmel ATtiny15L has a 10-bit analog to digital converter the scale factor per bit is:

1 bit = 3.0 volts/1023 = 2.9 mvolts or 0.0098 Amps per bit

The complete PSPICE simulation model is shown in figure #1 (see next page). Simulation of the control algorithm for current feedback control is done with the use of analog behavioral models which represent the microcontroller based algorithm. Analog behavioral models used include a gain block, integrator block, sum block, difference block and limiter blocks. The control algorithm is a very simple proportional plus integral loop. It's important to keep in mind that the ATtiny15L does not have a multiply or divide instruction and has very limited storage for variables, providing only the 32 registers in the general purpose register file. This makes crafting of the control algorithm interesting as selection of the coefficients in the constant coefficient difference equation must be carefully made so that when combined with the selected sample rate the result is a value equal to 2n or 1/2n which can be implemented as bit shifts. To keep the algorithm simple enough for implementation in the ATtiny15L AVR processor, a simple proportional plus integral gain was used. Examination of the frequency response of the circuit combined with a series of transient simulations determined the initial proportional and integral gain values. Proportional gain was set at Kp = 0.5 while the integral gain was set at Ki = 500. Performance of the constant current supply with these gain values can be seen in the simulation plots of figures #2 and #3 (see next page). As can be seen, the system is over damped. This is by choice. In this system, a single constant current supply is used to drive four independent loads one at a time with the output of the supply switched between the four independent loads as discussed in the application requirements. When the load is switched, there is a momentary period when there is no load on the output of the supply. To prevent the output of the constant current supply from rising up to the 48 volt rail during the momentary no load condition the circuit is over damped. There is still a small increase in voltage at the constant current supply output but it is very manageable. When the next

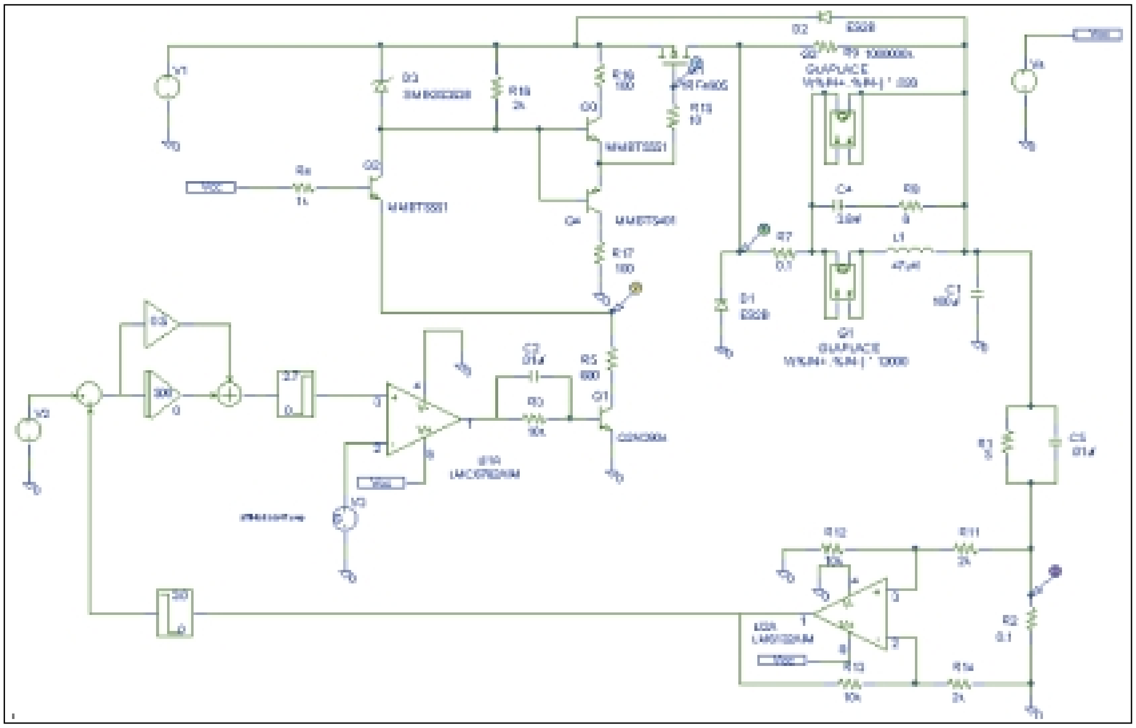


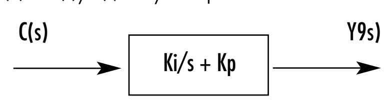
Figure 1 Simulation Model

load is engaged, the circuit adjusts the PWM duty cycle to produce the required current. Had the circuit been allowed to rise to the 48 volt rail there would be a large surge of current through the new load when it was engaged. A PWM switching frequency of 50 kHz was initially chosen to balance inductor size against heat dissipation in the P-channel MOSFET due to switching losses. Higher switching frequencies result in smaller magnetic component sizes thus allowing us to use a smaller less expensive inductor. Unfortunately, higher switching frequencies also result in greater switching losses in the MOSFET and the associated higher heat dissipation. From the PSPICE simulation results this initial set of gains and switching frequency seemed a good starting place for the design of the actual circuit.

IV. Algorithm Implementation in the Atmel ATtiny15L:

Implementation of the control algorithm in a digital processor requires translation of the system differential equations from their S domain Laplace form into the Z domain where they can be coded as constant coefficient difference equations. Constant coefficient difference equations are based on past and present values of the inputs and outputs of the system and the system sample rate. In this case load current error and applied PWM duty cycle to the P channel MOSFET transistor. Coefficients are constants based on the selected gain values and the rate at which the feedback parameter is sensed also called the sample rate of the system. First we start with the block diagram of the control loop shown in figure #4 (see nexy page). Only the feed forward proportional plus integral gain block H(s) is mapped to the Z domain. One of the most popular methods for converting from the S domain into the Z domain is by use of the bilinear transformation. The bilinear transformation successfully maps the S domain transfer function into the Z domain with the right half S plane mapping into the unit circuit in the Z domain. In fact, the bilinear transformation maps the entire imaginary axis of the S plane onto the unit circle in the Z plane thus avoiding the problem of aliasing found with the use of impulse invariance.

H(S) = Y(s)/C(s) = Ki/S + Kp



H(S) is converted to H(Z) by the following substitution:

S = 2/T[(1-Z-1)/(1+Z-1)]

Where T = sample rate of the system.

The price paid for this is distortion in the frequency axis. Proper smoothing of the current into the load is dependent on the frequency response of the circuit thus we are better off to avoid use of the bilinear transformation and instead base the transformation on the numerical solution to the differential equation. Mapping to the Z domain is again done by a substitution of variables but the substitutionary value for S is a little different:

S = (1 – Z-1)/T

H(S) = Y(S)/C(S) = [0.5S + 500]/ S

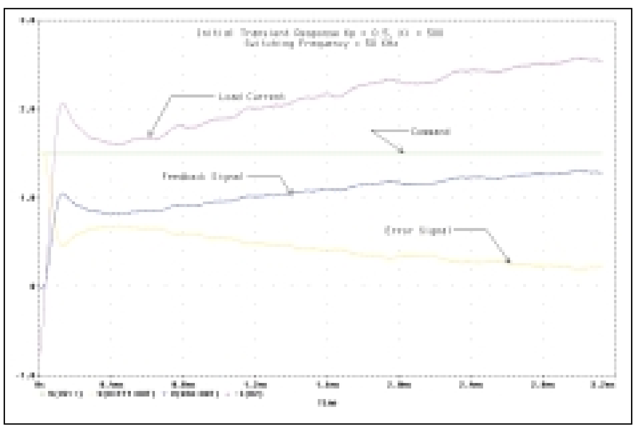


Figure 2

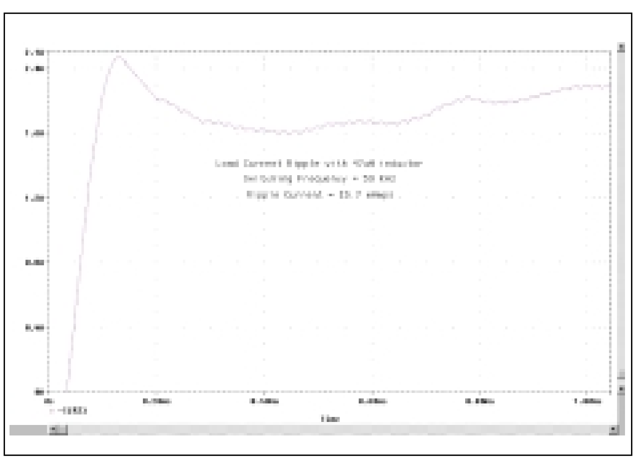


Figure 3

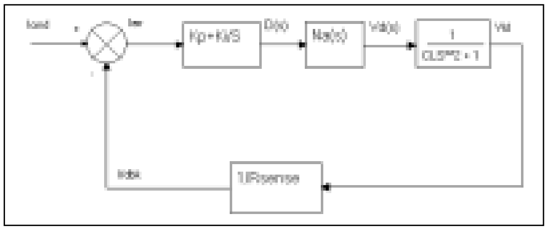


Figure 4

$$SY(S) = 0.5SC(S) + 500\ C(S)$$

$$(1 - Z\text{-}1)Y(Z)/T = 500\ C(Z) + 0.5\ C(Z)\ (1 - Z\text{-}1)/T$$

$$Y(Z)\ \text{-}\ Z\text{-}1Y(Z) = 500T\ C(Z) + 0.5\ C(Z) - 0.5\ Z\text{-}1\ C(Z)$$

The term Z-1 represents the value of the associated parameter at the previous sample point. We can denote this by the notation Y(k) and Y(k-1) where Y(k) is current value of the output and Y(k-1) is the value of the output generated by the previous sample. Replacing the above equation with this new notation yields:

$$Y(k) = 500\ T\ C(k) + 0.5\ C(k) - 0.5\ C(k\text{-}1) + Y(k\text{-}1)$$

$$Y(k) = A * C(k) - B * C(k\text{-}1) + D * Y(k\text{-}1)$$

Where $A = 500 * T + 0.5$
$B = 0.5$
$D = 1$

Let T = .001 seconds, the rate at which the charge current will be sensed and the algorithm recalculated and adjustment to the duty cycle of the PWM made.

⇨ $A = 1.0$
⇨ $B = 0.5$
⇨ $D = 1$

$$Y(k) = C(k) - 0.5 * C(k\text{-}1) + Y(k\text{-}1)$$

This is now the starting point for implementation into the ATtiny15L microcontroller. In terms of how the ATtiny15L AVR processor will implement this algorithm, the following definitions apply:
Y(k) = New duty cycle to be applied to the PWM output.
Y(k-1) = Duty cycle applied at last sample.
C(k) = Current error (Ierr) between required current and actual current.
C(k-1) = Current error (Ierr)at last sample.

Code for the ATtiny15L was written in C so the compiler was left to assign the data variables to registers in the general purpose register file. Other then simple addition and subtraction there is one divide by 2 associated with the variable C(k-1). This can be accomplished by a simple bit shift one place to the right since 0.5 is equivalent to 1/2n where n = 1.

V. Performance of Circuit and Algorithm:

Initial testing of the circuit determined that power dissipation in the P-channel MOSFET was too great to reach the required 8 Amp limit. Reducing the switching frequency to 25 kHz did not provide sufficient improvement. As a result the circuit had to be redesigned to use an N-channel MOSFET with a high side gate drive circuit.

Final circuit schematic is shown in figure #5 (see next page). Even with the N-channel MOSFET, switching frequency had to be reduced to 25 kHz to keep down the power dissipation. This was primarily due to the requirement to operate in a sealed enclosure with no external heat sinking or air flow. Additional problems were encountered with the control algorithm at load resistances down near the required 1_ limit with the minimum current level of 3 Amps. This requires an output voltage across the load of only 3 volts. Switching of loads at this low resistance caused instability in the control loop. As a result the forward loop gain had to be reduced to stabilize the system. This posed an interesting challenge, to find a set of gains that no only resulted in a stable overdamped system over the full range of loads from 1_ to 16_ but also could be implemented in the ATtiney15L requiring only bit shifts. Fortunately a simple reduction in the proportional gain by a factor of 0.5 combined with a change in the sample rate to 0.5 msec was sufficient to stabilize the system.

This new set of gains and sample rate yields the following constant coefficient difference equation. This final equation is easily implemented with only bit shits, 0.5 being a single bit shift to the right for divide by 2 and 0.25 requiring two bit shifts to the right for a divide by 4.

$$Y(k) = 0.5 * C(k) - 0.25 * C(k\text{-}1) + Y(k\text{-}1)$$

The C code listing for the algorithm is as follows:

```
for (;;)/* loop forever */
  {
  if(TIFR & 0x02)   // if timer overflow (every .500
                       millisecond )
  {
  TIFR |= 0x02;     // clear overflow flag
  TCNT0=56;         // reset counter for 200 counts to
                      overflow (1 ms period)
  PORTB|=0x10;
  Ck=0;
  ADMUX=0x61;       // select channel 2, current setting
  ADCSR=0xC2;       // start single conversion, with
           CK/4,

  while(ADCSR & 0x40);
                    // wait for conversion complete
  Ck+=ADCH;
  ADMUX=0x62;       // select channel 1, current feedback
  ADCSR=0xC2;       // start single conversion, with CK/4,

  while(ADCSR & 0x40);
                    // wait for conversion complete
  Ck-=ADCH;         // Ck=setting - feedback (error)
  if(Ck1==-1) Ck1=0;
  Ck1=Ck1/2;                    // divide previous error by 2
  Yk=(Ck/2-Ck1/2)+Yk1;          // +/- 7FFF
  if(Yk<0) Yk=0;                // no negitive output
  if(Yk>0x00FE) Yk=0x00FE ;     // Maximum output is 0xFF

  Ck1=Ck;           // save error for next iteration
  Yk1=Yk;           // save result for next iteration
  OCR1A=(char)Yk;
  PORTB&=0xEF;
  }
}
```

Actual performance of the supply at the worst case condition with a 1_ load is shown in figures 6 and 7. In figure #6 we look at the system at initial power up with a 1.0_ load. Here we see that we do not have the classical overdamped transient response characteristics. With a 1.0_ load a 3 Amp current requires only 3.0 volts across the load. A very small duty cycle from the 48 volt source is required to achieve this 3.0 volts. Small changes in such a low duty cycle result in significant swings in the output voltage and current. This is what makes the 1.0_ load case the most difficult. Figure #7 shows not only the transient case when changing direction of current flow but also the steady state current ripple. Although we are below the 10% limit at 3 Amps of 300 mAmps, we have significantly more ripple in the 1.0_ case then we have in the 15.0_ case shown in figure #8. At 1.0_ we see about 250 mAmps of ripple current while we have no measurable ripple in the same scale for the 15.0_ case under identical switching conditions. We also see much more of the desired overdamped characteristic in the 15.0_ load case.

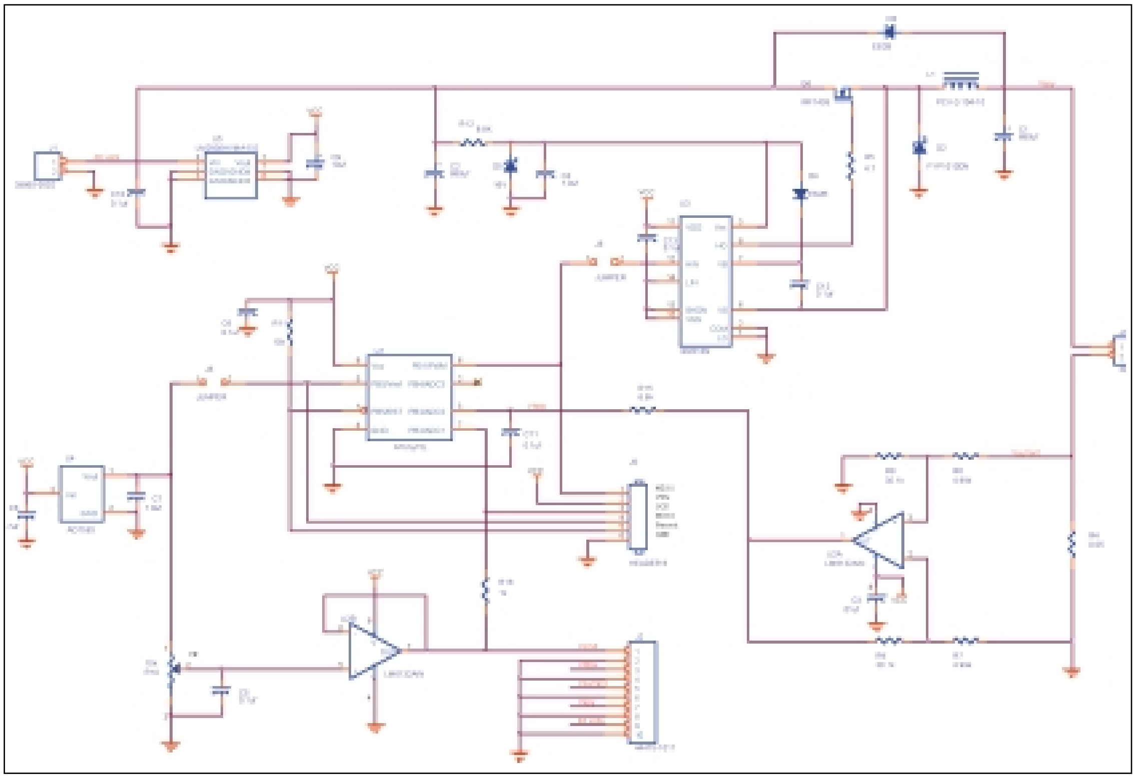


Figure 5

VI. Conclusions:

Although initial PSPICE simulation provided a good starting point for algorithm development, as well as inductor sizing and PWM frequency, there is still nothing quite like prototype testing. Computer simulation with PSPICE as with all simulation programs is only as good as the models and assumptions used. Care must always be taken with computer simulation to do a sanity check on the results and circuit breadboards and prototypes are still required to flush out all the issues. Minor tweaking of the algorithm was required to achieve stable operation over the required load range. The minimum 1.0_ load requirement proved to be the biggest hurdle as achieving stability at very low PWM duty cycles required a significant reduction in the proportional gain and increase in the sample rate. Heat dissipation on the main pass transistor was also a major challenge in this design. A P-channel MOSFET based design would have been simpler and less expensive but the larger die area and correspondingly higher parasitic capacitance of the P-channel devices results in slower switching times and greater heat dissipation. Higher switching frequencies have more turn on and turn off transient states generating greater heat dissipation in the pass transistor. We were able to overcome these obstacles in the final design and meet all the requirements in large part due to flexibility of the system architecture. Today's low cost microcontrollers such as the Atmel's AVR family make it feasible to do digital sampled data controllers where purely analog designs would have previously been used.

Selection of the Atmel ATtiny15L AVR microcontroller proved to be a good choice. It's small footprint and peripheral set made it a good fit for this design. It seems that the ATtiny15L processor would also be a good choice for battery charging circuits which could make use of the same buck regulator topology. A current sense resistor between battery negative and ground allows feedback for constant current charging independent of any time varying load currents beyond the battery which would be supplied by the charger supply. There are sufficient A/D inputs to support battery voltage and temperature monitoring for –_v or _T rapid charge termination allowing multi-chemistry support for NiCd, NiMH or Li-ion type batteries. It is unfortunate that the ATtiny15L is not offered in a grade that provides a laser trimmed voltage reference. At present, the on board voltage reference has insufficient accuracy to be useful in either the constant current supply or battery charging applications. Even though laser trimming of the voltage reference would add some cost to the part, the increase would be less then the addition of an external voltage reference with the required accuracy. PCB real estate would be saved and total parts count would be reduced with its associated parts stocking fees and carrying costs. Because of the lower current levels used in portable battery operated systems many battery charging applications could make use of the internal differential amplifier even with its limited gain choices. Lower currents allow for a broader selection of potential load sense resistor values without being driven to a high wattage part. Perhaps Atmel will consider offering a laser trimmed version of the ATtiny15L in the future targeted at battery charging applications. ❑

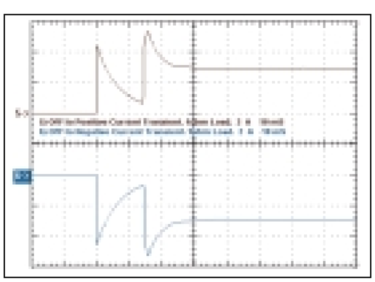


Figure 6

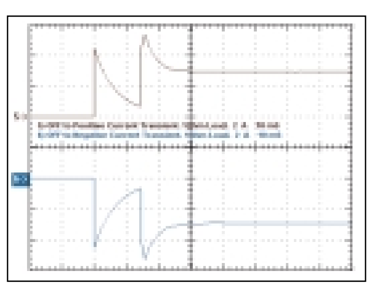


Figure 7

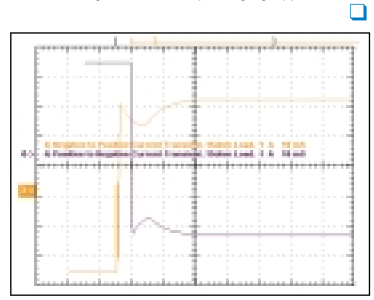


Figure 8