# An Adaptive Digital Controller for a Unity Power Factor Converter

Ahmed H. Mitwalli, Steven B. Leeb, *Member, IEEE,* George C. Verghese, *Member, IEEE,* and V. Joseph Thottuvelil, *Member, IEEE*

*Abstract*—**This paper describes an adaptive digital controller for a unity power factor ac–dc converter. The controller is based on a linear large-signal model of the boost converter. A hardware design is presented and analyzed, followed by the software implementation of the control algorithm. Issues in digital control of power converters, such as quantization effects and fixed-point representation of system parameters, are examined in the context of this system. Experimental results are presented and compared with simulations.**

## I. Introduction

With the increasing need for high-power converters [1], unity power factor (UPF) supplies have rapidly gained popularity. Power supplies in computers and other equipment, for example, must be not only efficient but also responsive to disturbances and reliable in the presence of component drift or aging. Digital controllers are promising in such applications because they are able to implement flexible control algorithms. Relatively few digital controllers have been built for this class of power converters; some exceptions are described in [5]–[8]. This paper describes the implementation of an adaptive, large-signal, linear, digital controller for a UPF converter.

A popular scheme for achieving unity power factor uses the switching preregulator circuit in Fig. 1. The boost converter in the circuit receives as its input a rectified waveform. An inner current loop controls the input current $i_L(t)$ to the shape and phase of the (rectified sine) input voltage $v_{\rm in}(t)$ by providing a switching sequence for the controllable switch that forces the inductor current toward a desired current $i_p(t)$. The reference current $i_p(t)$ is made proportional to the input voltage, $i_p(t) = kv_{\rm in}(t)$. The outer voltage loop regulates the output voltage $v_o$ to the desired reference voltage $V_o$ by adjusting the proportionality constant $k$ used to generate $i_p$ every line cycle [10].

The next section of this paper describes a linear large-signal model of the boost converter with inner current control loop. This model is used to develop a sampled-data controller for the output voltage [9], [10]. Section III describes the hardware prototype. The software implementation of the voltage loop controller is presented in Section IV. Section V reviews the design and implementation of adaptive control for the power supply. Simulations and experimental results are evaluated in Section VI.

Manuscript received November 3, 1993; revised October 4, 1995. This work was supported by Digital Equipment Corp. and the MIT/Industry Power Electronics Collegium. Additional hardware was available through a grant from Intel Corp.

A. H. Mitwalli, S. B. Leeb, and G. C. Verghese are with the Laboratory for Electromagnetic and Electronic Systems, Massachusetts Institute of Technology, Cambridge MA 02139 USA.

V. J. Thottuvelil is with AT&T Bell Laboratories, Mesquite, TX 75149 USA.

Publisher Item Identifier S 0885-8993(96)01922-9.

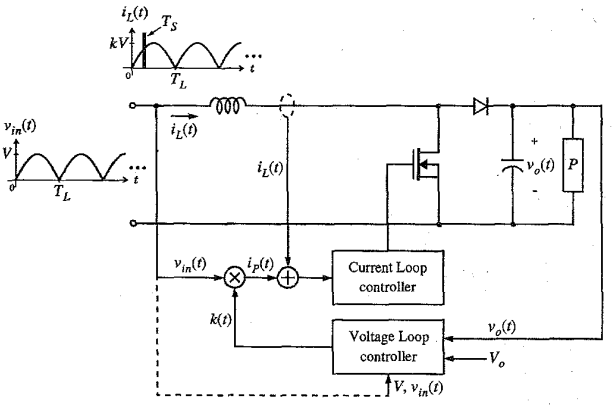



Fig. 1. High-power factor preregulator.

## II. Modeling and Control Design

Typical models of UPF power supplies are developed by linearizing nonlinear models of the converter circuits [2], [3], [11]. These models describe converter operation well only in the vicinity of a nominal operating point. A linear large-signal model for the boost converter with inner current loop is developed in [10], based on a dynamic power balance equation for the circuit. Squared output voltage is the state variable. A brief description of the model is presented below.

Let $T_L$ denote the period of the rectified input voltage $v_{\rm in}$. We assume that the current loop maintains $i_L(t) = i_p(t)$, and we ignore the switching ripple. The proportionality factor $k(t)$ is varied once every rectified line cycle to produce the desired reference current for the inner loop (faster variation of $k$ can also be considered, [17]). Modeling the load as a constant-power load that draws power $P$, the following power balance equation for the boost converter can be obtained with the help of Tellegen's theorem

$$\frac{1}{2}C\frac{dv_o^2}{dt} = kv_{\rm in}^2 - k^2\frac{1}{2}L\frac{dv_{\rm in}^2}{dt} - P. \tag{1}$$

The reference current $i_p$ has been substituted for $i_L$ and $i_{\rm in}$, and the relationship $i_p = kv_{\rm in}$ has been used to develop (1). A load comprising switching converters with regulated outputs is

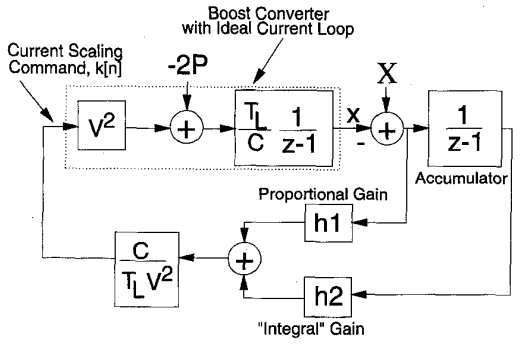



Fig. 2. DT controller for the UPF converter.

generally well represented as a constant power load. The model is easily augmented to incorporate a resistive component into the load. The experimental tests described later are actually carried out with a (nonlinear) resistive load, although the design is based on constant power. This makes evident the robustness of the design.

For digital control, it is convenient to develop a discrete time (DT), sampled-data model of the plant. Let the value of $k$ in the $n$th cycle be $k[n]$. Integrating (1) over a time interval $T_L$, and denoting $v_o^2$ at the beginning of the $n$th cycle by $x[n]$, we find

$$\frac{1}{2}C(x[n+1]-x[n]) = T_L\left(k[n]\frac{V^2}{2} - P\right) \tag{2}$$

or

$$x[n+1] = x[n] + \frac{T_L V^2}{C}k[n] - \frac{2T_L}{C}P \tag{3}$$

where $V$ is the amplitude of the input voltage waveform and is assumed constant in this model. This is a state-space representation of a first-order linear, time-invariant (LTI) discrete-time system, with control input $k[n]$ and disturbance input $P$.

The design steps for a DT version of a PI controller for the voltage loop are outlined in references [9] and [10]. The structure of the controller is shown in Fig. 2. This controller regulates $x = v_o^2$ to its reference value $X = V_o^2$, despite errors in the model and disturbances to the system. The accumulator serves to integrate the output error. The closed-loop system will reach a constant, stable steady state if the gains $h_1$ and $h_2$ are chosen carefully and if $P$ is constant.

In steady state, the accumulator state will, by definition, be constant. The input to the accumulator in steady state must be zero, and therefore $v_o^2[k] = V_o^2$. This controller regulates $x = v_o^2$ to its reference value $X = V_o^2$, despite errors in the model and constant disturbances to the system. Stability is obtained by choosing the gains $h_1$ and $h_2$ so that the roots of the characteristic polynomial

$$z^2 - (h_1 + 2)z + 1 + h_1 - h_2 \tag{4}$$

have magnitude less than unity.

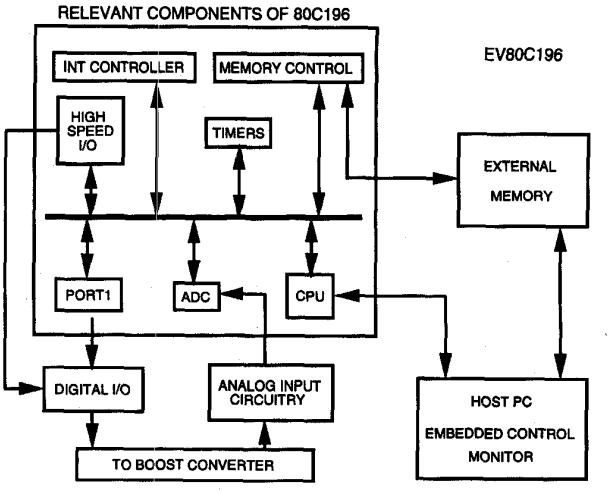



Fig. 3. Digital development system.

## III. Hardware Implementation

The following sections describe the prototype hardware implementation of the outer voltage loop, the inner current loop, and additional interface circuitry that connects the digital controller and the power electronics.

### A. Digital Voltage-Loop Controller

In the hardware prototype, the voltage-loop controller is implemented digitally. The output command computed by this loop is updated once every $T_L$ seconds. The digital implementation provides desired flexibility and robustness without excessive expense.

The 16-bit 80C196KB from Intel was chosen as the microprocessor for our prototype. Much less expensive than 32-bit microprocessors and DSP chips, this embedded microcontroller provides ample processing power for this application. In fact, a straightforward implementation of the PI controller discussed above does not even require a processor as sophisticated as the 80C196KB. However, the difficulties and complications, e.g., quantization, created by a very much simpler microprocessor, could substantially complicate a practical implementation. To achieve high levels of performance and to explore the various control schemes and features made possible by operating in discrete time with a digital controller, including adaptive control techniques, a reasonably capable microprocessor is required. Fig. 3 illustrates the components of the 80C196KB control board employed in the system.

*1) Interrupt Handling:* In a digital control system, tasks such as sampling and command generation need to be synchronized to specific points in time. Others are event generated, such as shutdown or other precautionary measures taken when a certain condition is detected. Still other tasks, such as system operation services or computations carried out during intervals free of other processing tasks, have no time restriction.

In the UPF prototype, all three types of tasks are present. Sampling and control need to be carried out at specific, periodic instants (8.33 ms apart). System shutdown is event-

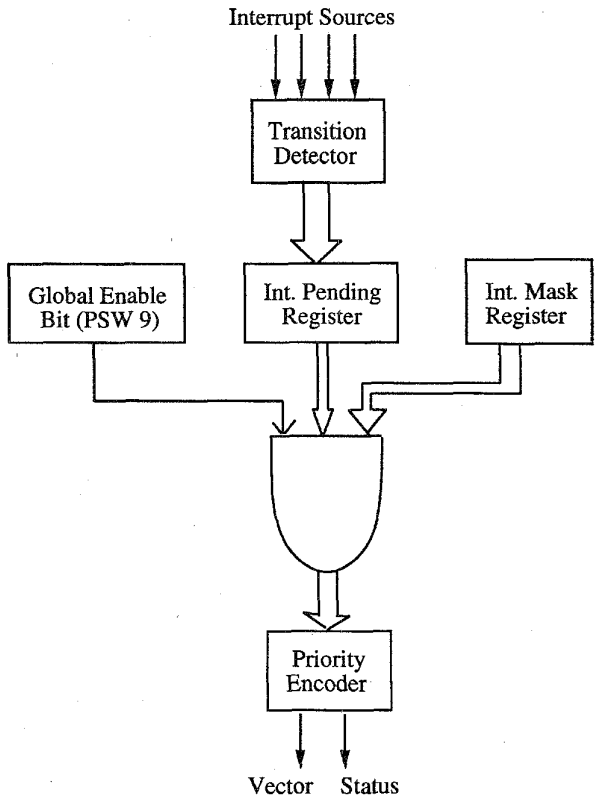



Fig. 4. Interrupt structure.

generated. The microcontroller can tune the system over several control cycles by estimating circuit parameters whenever the processor is free. An interrupt-driven software structure is convenient for implementing all three types of tasks. On the 80C196KB, special function registers (SFR's) and other devices handle interrupt generation, as shown in Fig. 4.

*2) The A/D Converter:* In the prototype, an analog-to-digital (A/D) converter is used to sample the input voltage level, the output bus voltage, and the load current. The 80C196KB includes a 10-bit analog-to-digital (A/D) converter with an eight-channel analog multiplexer. On the 12-MHz version of the microprocessor, employed in our application, a conversion is completed in approximately 26 $\mu$s. As is the case with other on-board peripherals, the A/D converter is controlled by an SFR. In the prototype, a sample is acquired by writing a value to an SFR that specifies the channel to be sampled, initiating the acquisition. When the conversion is complete, an interrupt is generated. The result of the conversion is placed in other SFR's. The A/D operation is outlined in detail in [22].

*3) Timers:* A precision timer is used in the prototype to coordinate periodic sampling for the sampled data voltage loop controller. The 80C196KB contains a 16-bit timer that is incremented every 16 clock cycles, providing a timing resolution of 1.33 $\mu$s. As preprogrammed times are reached by the timer, a software timer interrupt can be generated. When the interrupt is generated, time-specific tasks, such as sampling

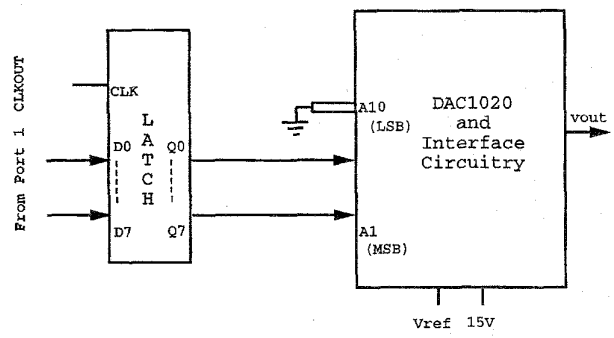



Fig. 5. Output port 1 and multiplying DAC configuration. Vout = −vref (A 1/2 + A 2/4 + ··· + a 10/1024). An = 1 if $n$th digital input is high.

analog inputs, are carried out. Details of the timing algorithm are described in detail in [22].

### B. D/A Conversion

The output of the digital voltage-loop controller computation provides the reference for the inner current loop controller. One approach for interfacing the digital and analog control loops would be to compute the commanded current $i_P$ in Fig. 1 inside the microprocessor and then convert it to an analog signal for the current-loop controller. This approach would require the microprocessor to deliver samples to the current loop at a sufficient rate to provide the desired current wave shape. The processor would face substantial sampling and processing chores.

A better method, employed, for example, in [5], utilizes a multiplying digital-to-analog converter (multiplying DAC or XDAC). A multiplying DAC computes a function of a digital and an analog input, usually a product. In our prototype, a multiplying DAC is used to implement the multiplier in Fig. 1, whose inputs are $k[n]$ and $v_{\text{in}}(t)$. Fig. 5 illustrates the multiplying DAC in the circuit configuration employed in the prototype. The latch at the input to the multiplying DAC is used to synchronize updates of the scale factor $k$. The inputs to the latch come from a parallel port of the 80C196KB.

### C. Analog Current-Loop Controller

For convenience, the current-loop controller in the prototype was implemented using the Unitrode UC3854. (The UC3854 may be used to implement both analog voltage and analog current control loops, [21].) Three inputs to the chip, $A, B$, and $C$, are fed to an analog function block in the chip, which computes the function $AB/C$. This output is used as the current reference. The input current is controlled to follow this reference value through an average current mode control scheme. The current loop and interface to the digital voltage loop are shown schematically in Fig. 6. For our prototype, the UC3854 inputs $A$ and $C$ are pinned to fixed voltage levels, and input $B$ is connected to the output of the multiplying DAC described in the previous section. This arrangement is attractive for testing and development because it permits the *a priori* limitation of the commanded power to a safe maximum

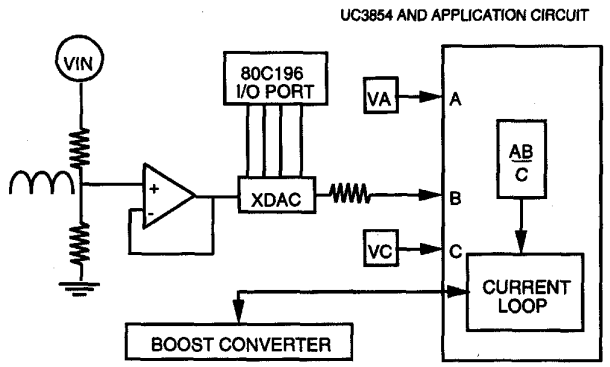



Fig. 6. Implementation of the current loop.

value defined by the voltages on pins $A$ and $C$ and the upper voltage limit on $B$ [22].

### D. A/D Resolution Enhancement

The effects of quantization associated with the 10-bit resolution of the A/D converter can be mitigated in some cases by a refinement described in this section. For some variables in a system, e.g., the output voltage in the UPF prototype, only a small portion of the possible variable range needs to be resolved. For output voltages above or below a comparatively narrow range, the control command will be saturated. It is possible, therefore, to increase the input resolution by "magnifying" the voltage in the linear region of the controller's operation with the analog subtractor shown in Fig. 7. The input voltage, $V_{\text{in}}$, is the output bus voltage of the UPF, $v_o$, divided by 100. In the linear region of the subtractor, the output voltage of the operational amplifier is

$$V_{\text{out}} = \frac{R_1}{R_2}(V_{\text{in}} - 2.7). \tag{5}$$

If $V_{\text{in}}$ goes below 2.7 V (in other words $v_o$ goes below 270 V, the point below which the output command $k$ will be saturated at its maximum value) the output of the subtractor saturates at 0 V. If the ratio of the resistors, $R_1/R_2$ is chosen carefully, the output of the circuit in Fig. 7 will saturate at its maximum value whenever the power supply voltage output goes over 430 V. Since the output command of the digital controller in the prototype saturates for $v_o$ below 270 V and shuts off for $v_o$ above 430 V, no relevant information is lost by not resolving voltages outside of the 270–430 V region. Moreover, by amplifying the voltages in the region of interest, between 270 V and 430 V, we are able to map this region to the full 5-V range resolved by the A/D converter. The improvement in resolution is equal to the slope of the linear region in the diagram, $R_1/R_2$. Careful implementation is required to prevent amplification of unwanted noise. (The popular operating point of 390 V is assumed in this discussion. In the actual implementation, this and other parameters in the system are varied for convenience and ease of prototype construction.)

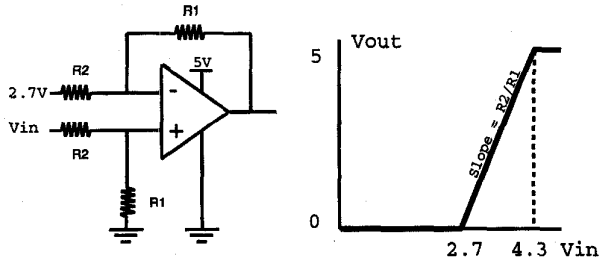



Fig. 7. Increasing input resolution.

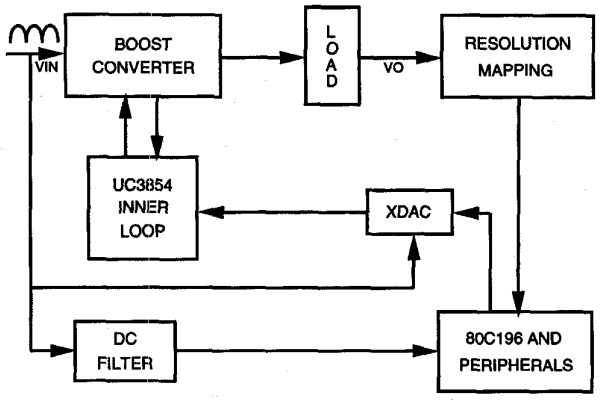



Fig. 8. Overall hardware system.

### E. The Complete System

Fig. 8 shows a schematic of the overall system. The boost converter uses a load capacitor $C = 100\ \mu$F and inductor $L =$ 1 mH. The capacitor is rated for 450 V. The load is composed of a positive temperature coefficient resistor with a nominal resistance of 3200 Ω at our output voltage. In our test stand, the load may be doubled by connecting an equal resistor in parallel with the initial resistive load, permitting observation of transient performance. The inputs to the UC3854 in Fig. 6 and the analog input to the multiplying DAC are chosen so that the maximum current command at an input voltage of 110 V RMS ac produces 50 W of output power, which at the lower load level of 3200 Ω produces 400 V across the load capacitor, sufficiently below its rated maximum.

## IV. Software Implementation

The software implementation of the controller uses fixed-point arithmetic to minimize cost and processing overhead. As in any fixed-point system, care needs to be taken in scaling variables to ensure that control calculations are made with necessary accuracy without over- or under-flowing the processor registers. To avoid changing the characteristics of the closed-loop system, care must also be taken to account for gains arising from the interface circuitry between the analog and digital subsystems, e.g., the A/D converter. Timing constraints are also of extreme importance when working with sampled-data models. Timing, parameter quantization, and scaling are the core challenges that must be met during the software design process.

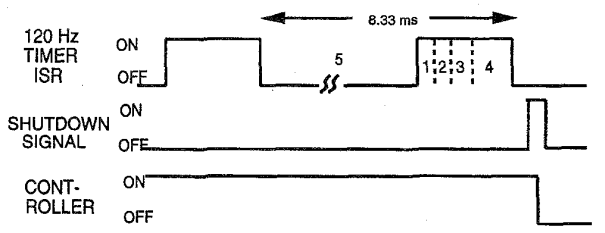



Fig. 9. System timing: 1) reload timer, 2) sample controller inputs, 3) compute PI output, 4) output controller command, and 5) estimation/tuning and operating system.

### A. Timing

Timing for the controller is implemented using the 80C196KB on-board timer described above. To implement the discrete-time PI controller, the sampling and control output frequencies need to be 120 Hz. A control cycle begins when software timer 0 on the microcontroller generates an interrupt. The processor executes an interrupt service routine (ISR) in response to the interrupt. The first instruction in the ISR reloads the timing register to initiate a new interrupt in 8.33 ms, maintaining the 120 Hz sampling frequency to within the resolution of the timer. The time-specific task of sampling the inputs is executed next. These inputs are used to compute the controller output, also inside the ISR. This computation consumes less than 1 ms of processing time. It is important that this computation delay is short compared to the sampling period. (Additional states due to delay between the input and output would otherwise need to be included in the model.) The controller output is then written to the output port. Additional time in the cycle is left for parameter estimation and adaptive (or more complex) control, or for use by the operating system. A shutdown signal generates an interrupt of higher priority that transfers control to its own ISR when detected. Fig. 9 illustrates the timing of operations in the prototype.

### B. Scaling and Quantization

For discrete-time PI control, the controller command is computed as illustrated in Fig. 2

$$k[n] = \frac{C}{T_L V^2}(h_1(v_o^2[n] - V_o^2)h_2\sigma[n]). \tag{6}$$

Several concerns arise in the digital implementation of this equation. Quantization of the inputs to the controller occurs as a result of A/D conversion, and output quantization is a function of the number of bits that are driving the input to the multiplying DAC. The equality above is therefore never met, and the desired zero-error steady state is not achieved. Instead, the output voltage and input current exhibit cyclic behavior around the infinite-precision (or analog) steady state. This behavior is shown in the simulations of a system with eight bits of A/D conversion resolution in Fig. 10. Both DT system poles have been placed at 0.5 for fast transient decay. Although the quantization in voltage is relatively insignificant, the current (the envelope of which is proportional to $k$, since $i_P = kv_{\text{in}}$) is oscillating far from its average value.

To improve the situation, $h1$ and $h2$ were chosen in the prototype to result in a slower system (poles closer to the unit

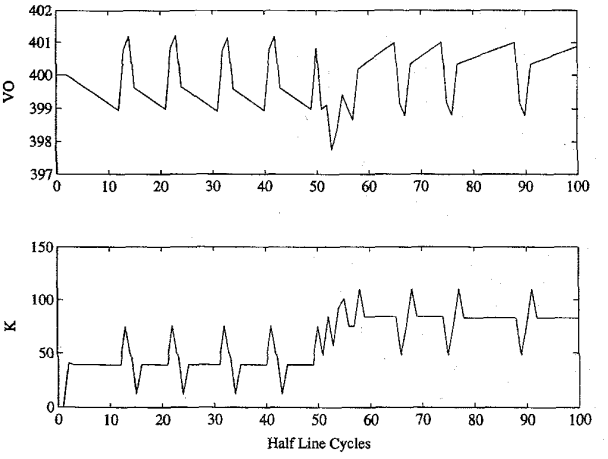



Fig. 10. Response to load power doubling with eight-bit sampling resolution.

circle). This system responds less aggressively to quantization error. Also, the sampling resolution was improved by using the full 10 bits of A/D converter resolution available in the 80C196KB and by using the analog subtractor circuit discussed above. These design choices resulted in the working system discussed in [22].

Fixed-point arithmetic requires that system parameters with values smaller than unity be scaled in order to be represented as integers. This scaling, along with other gains due to hardware interfacing, A/D conversion, or D/A conversion, enter the loop in the transfer relation between the inputs to the controller and its outputs. The software must be designed to unwrap these gains and scale factors to ensure that relationship (5) governs the output of the controller.

### C. Other Controller Features

Other features of the implementation serve to enhance its performance. Integrator windup is a problem that arises in PI controllers when the controller output is saturated. When saturation occurs, the integrator is effectively isolated from the closed loop system [16]. Its value may increase excessively when the controller is saturated. Substantial time might be required to unwind from this state, resulting in undesirable behavior, e.g., output voltage overshoot. As an antiwindup mechanism, accumulation is disabled when the output of the digital controller is saturated.

The voltage-loop controller includes soft-start during the turn-on transient and quantization error elimination in the steady state. Soft-start is implemented by initializing the reference voltage at a low value and stepping it slowly to a final value at which closed loop control will begin. Quantization error elimination is necessary for the implementation of adaptive control, which requires the estimation of circuit parameters based on the ripple in the output voltage. In the prototype, quantization error resulted in periodic deviations on the order of the ripple amplitude. It would be impossible to detect the ripple uniquely without additional measures. Elimination of quantization error in the steady state is outlined in Section V and discussed in more detail in [22].

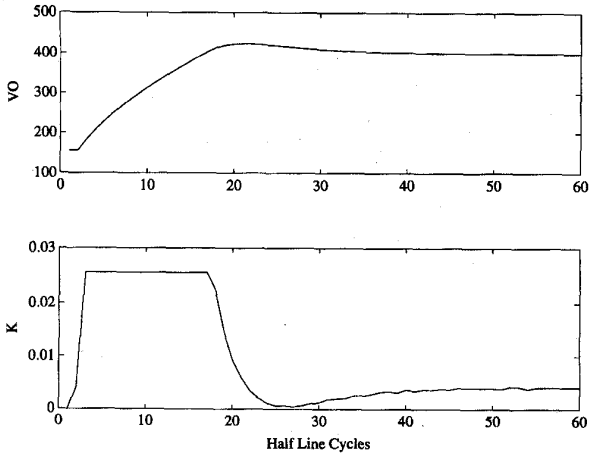



Fig. 11. Turn-on transient without antiwindup or soft-start.

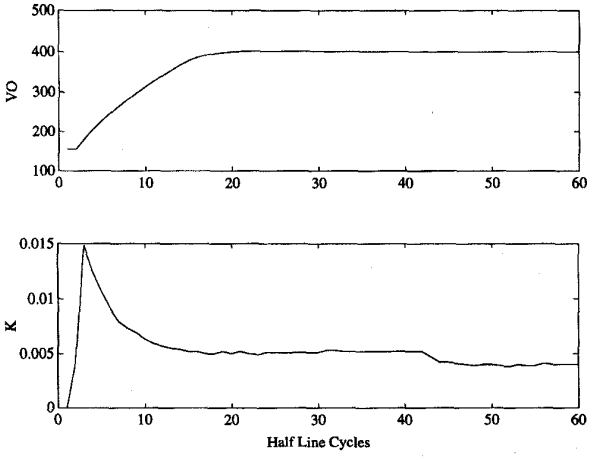



Fig. 12. Turn-on transient with antiwindup and soft-start.

The antiwindup and soft-start features of the controller were simulated. Fig. 11 shows the turn-on transient without either of these features. The output voltage overshoots to 420 V as a result of the accumulator windup during the transient. The input current (proportional to $k$ in the figure) saturates. Fig. 12 illustrates the behavior of the circuit on start-up with the antiwindup and soft-start schemes simulated. The voltage no longer overshoots.

### D. Overall Software Implementation

A simplified overall view of the software implementation is shown in Fig. 13. The software is interrupt driven. The diagram shows the general operation of the code as well as an example of how extraneous gains produced in the hardware are offset in the software (e.g., inverting the resolution mapping). Many of the details are described in depth in [22].

## V. Adaptive Control

The prototype provides a good test bed for parameter estimation and adaptive control techniques. The practical need for adaptive control motivates the use of a microprocessor-based controller. After the basic system was constructed and tested, estimation of the load capacitance and gain tuning based on the acquired estimate were included as additional features.

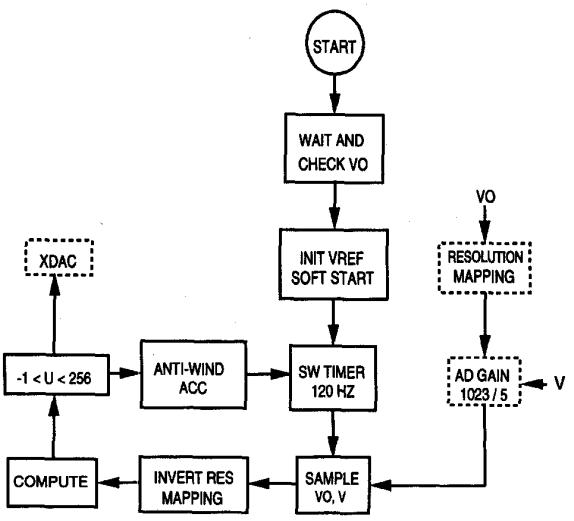



Fig. 13. Software structure.

### A. Motivation

The load capacitance is usually imprecisely known and may change due to variations in the connected loads. Part of the output capacitor may also be moved to the input during brown- or blackouts to improve holdup time. As is clear from Figs. 1 and 2, the capacitance value is an important parameter in the controller design. If the assumption of perfect knowledge of the capacitance is removed, and the value of $C$ in the controller is replaced by $\hat{C}$, the roots of the new characteristic polynomial are

$$z = \frac{1}{2}\left[\left(\frac{\hat{C}}{C}h_1 + 2\right) \pm \frac{\hat{C}}{C}\sqrt{h_1^2 + 4\frac{C}{\hat{C}}h_2}\right]. \tag{7}$$

Clearly, given that the gains are negative for this system, a decrease in the true load capacitance moves the poles apart on the real axis and may move one of them outside the unit circle, while an increase in the capacitance increases the negative term under the square root sign, moving the poles off the real axis. In the implemented digital system, an error in $\hat{C}$ may have other effects as well.

### B. Design and Implementation

Reference [17] notes an approximate relationship between the output voltage ripple amplitude $\epsilon$ and the capacitance

$$\epsilon \approx \frac{P}{\omega_2 V_o C} \tag{8}$$

where $\omega_2$ is twice the line frequency. A bandpass filter in the prototype isolates the output voltage ripple. The output of the bandpass filter is digitized on a separate A/D channel. A current sensor provides a measurement of the load current.

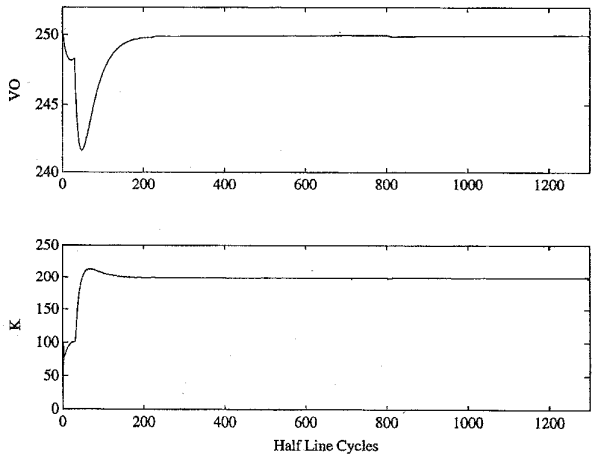



Fig. 14. Simulation of prototype system response to a doubling of load power.

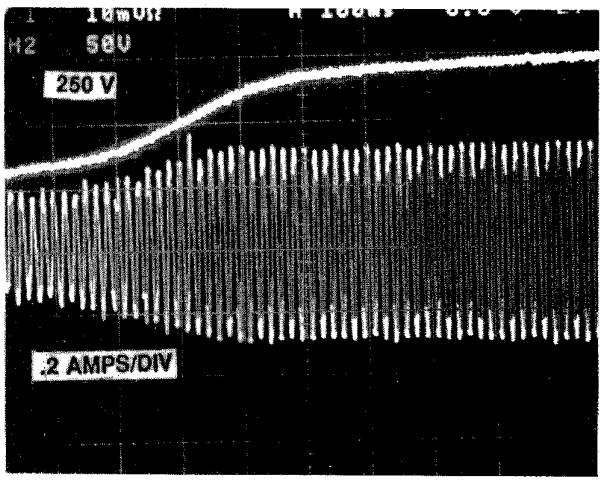



Fig. 15. Start-up of the prototype.

Once these measurements are made, the microcontoller software can estimate the output capacitance using (8) and tune the controller gains appropriately.

Since sine interpolation based on a few samples could necessitate the use of both the math and floating point libraries with the microprocessor (which would slow down the operation and result in higher cost in terms of memory), a simpler sampling scheme that sweeps the waveform in search for the ripple peaks is used in the prototype. In order to obtain accurate, relatively noise-free measurements, an average of the peaks obtained over a few cycles is used for estimating the load capacitance. Once an estimate is acquired, the gains are recomputed based on the old and the new estimates. Details of this scheme are given in [22].

It is also noteworthy that the quantization error mentioned earlier makes it very difficult to resolve the output voltage ripple necessary for implementing adaptive control. Even with the resolution mapping described in Section III, the quantization noise is on the order of the ripple amplitude in the prototype. It is therefore necessary to eliminate quantization in order to make this measurement. In order to prevent

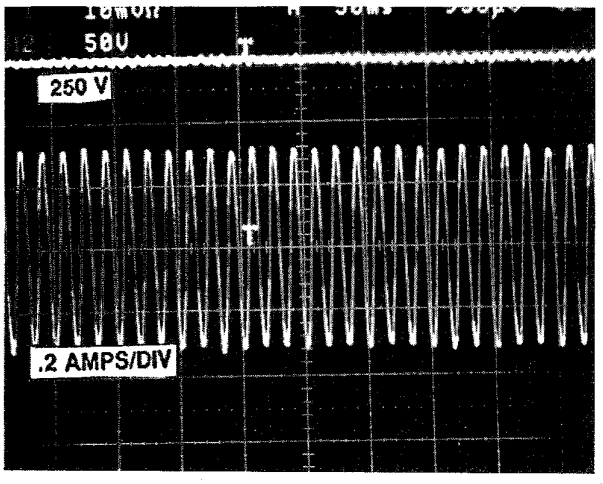



Fig. 16. Steady-state operation of the prototype.

the controller command from fluctuating during the ripple measurement, the PI controller is disabled in steady state and a fixed command is used. This fixed command is computed based on an average of the control commands during a period when the output voltage is within a certain distance from the target voltage. Any deviations from steady-state conditions result in an immediate return to active control. The complete algorithm and further discussion are found in [22].

## VI. System Simulation and Experimental Results

Using the large-signal model and PI controller developed above, Matlab[1] simulations were constructed for the closed-loop system. With the general parameters of the development system and techniques presented earlier, simulations of the digital system could be conducted to predict the effects of disturbances and parameter variations for comparison with results from the prototype. Simulations of different variations of the system were shown earlier in Figs. 10–12.

Fig. 14 shows a simulation of the working digital system responding to a doubling of load power. The system poles are relatively near to the unit circle (approximately .91) and input resolution has been maximized using the resolution mapping circuit discussed earlier. Note that in this implementation, the operating point is 250 V instead of 400 V. The effects of quantization are negligible in this design, which was used for the final implementation.

Experimental results on the start-up and steady-state operation of the hardware prototype are shown in Figs. 15 and 16. They agree with the simulations and demonstrate good performance for the UPF regulator. Quantization error can be seen in the current waveform and slightly in the voltage waveform, due to scale. The sinusoidal current yields a near unity power factor. Figs. 17 and 18 show the response of the system to a load disturbance with an initial error in the assumed capacitance, with and without adaptive control. Note that in both cases the initial value of the capacitance given to the controller was lower than the correct value. From (6), a controller that does not tune its gains will then respond

[1] Matlab is a trademark of The Mathworks Inc.

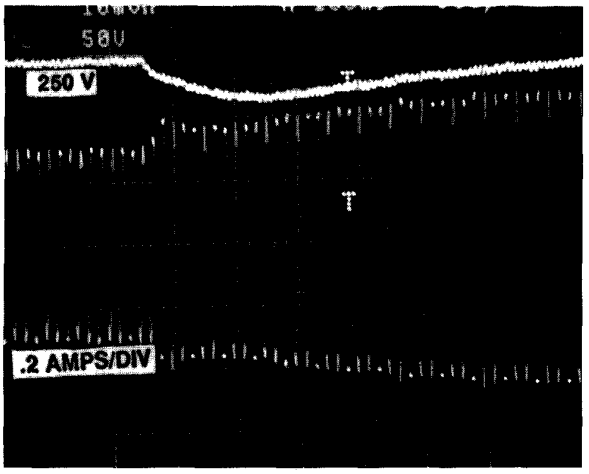



Fig. 17. Prototype response to load power doubling without adaptation.

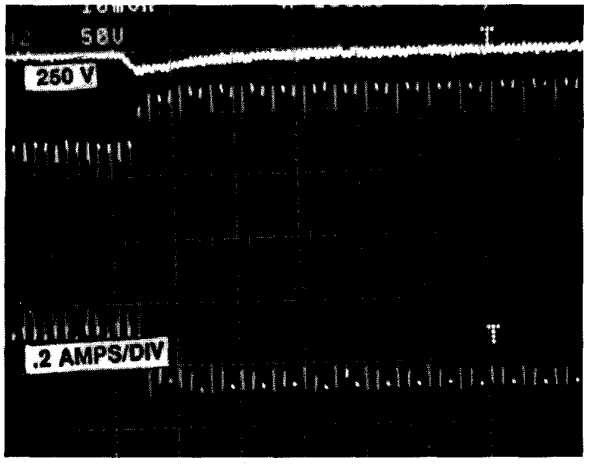



Fig. 18. Prototype response to load power doubling with adaptation.

more slowly to a disturbance. As expected, the response of the controller without adaptive control (Fig. 18) is slower than that which incorporates gain tuning (Fig. 17).

## VII. Conclusion and Future Work

The experimental results match the simulations well, verifying that the large-signal model is accurate. No load regulation and no steady-state error are evident, thanks to an integral component in the controller. An analog controller with no integral component implemented using the UC3854 exhibits substantial load regulation (nonzero steady-state error). Although an analog implementation of a PI controller should in theory have zero steady-state error, in reality this may be difficult to achieve. True PI control, with infinite gain at dc, is not possible due to parasitic resistances. The digital controller suffers from no such parasitics, and other effects that contribute to load regulation can be minimized.

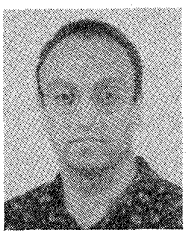
The basic closed-loop controller was successfully implemented, with a few additional features that were included with relative ease. These features illustrate the potential for complex control. For example, adaptive control was implemented. The microprocessor employed in the prototype is relatively inexpensive yet powerful, leaving room to implement more complex algorithms or to decrease the cost of the implementation. With the decreasing cost of microprocessors and their increasing functionality, the possibilities are clear. Digitally controlled power converters appear well suited to meet increasingly stringent power quality and performance requirements.

## Acknowledgment

The authors would like to thank A. Stankovic for his discussion and help.

## References

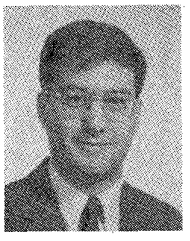
[1] B. Wilkenson, "Power factor correction and IEC 555-2," *PowerTechnics Mag.*, Feb. 1991, pp. 20–24.
[2] J. B. Williams, "Design of feedback loop in unity power factor ac to dc converter", in *Power Electron. Spec. Conf. Rec.*, 1989, pp. 959–967.
[3] R. B. Ridley, "Average small-signal analysis of the boost power factor correction circuit," in *Proc. 10th Ann. Virginia Power Electronics Center Sem. Power Electron.*, 1989, pp. 108–120.
[4] B. Sharifipour, P. Cacciola, and J. Maddox, "Designing a 1200 watt multiple output modular power system with high-power utilization for the workstation environment," in *Proc. IEEE Appl. Power Electron. Conf.*, 1989, pp. 439–444.
[5] C. P. Henze, "Digitally controlled ac to dc power conditioner," U.S. Patent 4 761 725, Aug. 2, 1988.
[6] B. K. Bose, *Microcomputer Control of Power Electronics and Drives*. Piscataway, NJ: IEEE Press, 1987.
[7] A. Mitwalli, S. B. Leeb, G. C. Verghese, and V. J. Thottuvelil, "A digital controller for a unity power factor converter," in *Proc. IEEE PELS 3rd Workshop Comput. Power Electron.*, Berkeley, CA, Aug. 1992.
[8] S. Sivakumar, K. Natarajan, and R. Gudelewicz, "Control of power factor correcting boost converter without instantaneous measurement of input current," *IEEE Trans. Power Electron.*, vol. 10, no. 4, July 1995.
[9] J. G. Kassakian, M. F. Schlecht, and G. C. Verghese, *Principles of Power Electronics*. Reading, MA: Addison-Wesley, 1991, pp. 395–399.
[10] K. Mahabir, G. Verghese, V.J. Thottuvelil, and A. Heyman, "Linear averaged and sampled data models for large signal control of high-power factor ac–dc converters," in *Power Electron. Special. Conf. Rec.*, 1990, pp. 291–299.
[11] C. P. Henze and N. Mohan, "A digitally controlled ac to dc power conditioner that draws sinusoidal input current," *Power Electron. Spec. Conf. Rec.*, 1986, pp. 531–540.
[12] K. Ogata, *Discrete-Time Control Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1987, pp. 1–36.
[13] B. Orlik and H. Weh, "Microprocessor-controlled three-phase motors with high resolution digital pulse width modulator for high pulse frequencies," in *Proc. Euro. Power Electron. Conf.*, 1985, pp. 3.39–3.44.
[14] C. Bergman and P. Goureau, "Direct digital control of a self-controlled synchronous motor with permanent magnet," in *Proc. Euro. Power Electron. Conf.*, 1985, pp. 3.269–3.273.
[15] J. Siebert, "Freely programmable digital open-loop/closed-loop control system for converters and converter drives," in *Proc. Euro. Power Electron. Conf.*, 1985, pp. 5.7–5.10.
[16] K. Astrom and B. Wittenmark, *Computer-Controlled Systems, Theory and Design*. Englewood Cliffs: Prentice-Hall, 1990, pp. 224–226.
[17] A. Stankovic, G. Verghese, X. Liu, and J. Thottuvelil, "Fast controllers for high-power-factor ac–dc converters," *Proc. Euro. Power Electron. Conf.*, 1990.
[18] *EV80C196KB Microcontroller Evaluation Board User's Manual, Release 001*, Intel Corp., Feb. 20, 1989.
[19] *80C196KB User's Guide*, Intel Corp., Oct. 1990.
[20] Y. Guijun and L. Norum, "Low cost digital controller for PWM converter," *Int. Fed. Automat. Control Low Cost Automat.*, 1990.
[21] *Power Factor Correction with the UC3854 Application Note*, Unitrode Integrated Circuits, 1991.
[22] A. Mitwalli, "A digital controller for a unity power factor power converter," Master's thesis, EECS Dept., M.I.T., Cambridge, MA, Dec. 1992.



**Ahmed H. Mitwalli** received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1991 and 1993, respectively. He is working toward the doctoral degree at the M.I.T. Laboratory for Electromagnetic and Electronic Systems.

His current research is in the development of novel actuator technologies (specifically, polymer gel actuators), with emphasis on system identification and control.

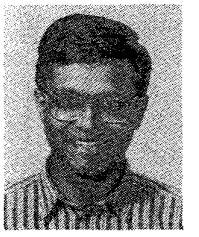
Mr. Mitwalli is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.



**Steven B. Leeb** (S'89–M'93) received the B.S., M.S., electrical engineering, and Ph.D. degrees from the Massachusetts Institute of Technology in 1987, 1989, 1990, and 1993, respectively.

He has been a Member of the M.I.T. faculty in the Department of Electrical Engineering and Computer Science since 1993. He currently serves as the Carl Richard Soderberg Assistant Professor of Power Engineering in the Laboratory for Electromagnetic and Electronic Systems. He is concerned with the design, analysis, development, and maintenance processes for all kinds of machinery with electrical actuators, sensors, or power electronic drives.

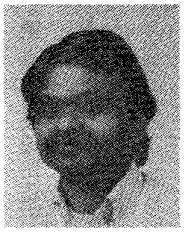
Dr. Leeb is a member of the IEEE Power Electronics, Control Systems, Power Engineering, and Signal Processing Societies. He is a Fellow of M.I.T.'s Leader's for Manufacturing Program and a member of Tau Beta Pi and Eta Kappa Nu.



**George C. Verghese** (S'74–M'78) received the B.Tech. degree from the Indian Institute of Technology, Madras, in 1974, the M.S. degree from the State University of New York, Stony Brook, in 1975, and the Ph.D. degree from Stanford University, Stanford, CA, in 1979, all in electrical engineering.

He is Professor of Electrical Engineering and a Member of the Laboratory for Electromagnetic and Electronic Systems at the Massachusetts Institute of Technology, Cambridge, which he joined in 1979. His research interests and publications are in the areas of systems, control, and estimation, especially as applied to power electronics, electrical machines, large power systems, and signal processing. He is co-author (with J. G. Kassakian and M. F. Schlecht) of *Principles of Power Electronics* (Reading, MA: Addison-Wesley, 1991).

Dr. Verghese has served as Associate Editor of *Automatica* and IEEE TRANSACTIONS ON AUTOMATIC CONTROL and is now Associate Editor IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY. He is on the AdCom of the IEEE Power Electronics Society, which he has served in various other capacities as well, including as Founding Chair of the Technical Committee and Workshop on Computers in Power Electronics.



**V. Joseph Thottuvelil** (S'77–M'78) received the B.Tech. degree from the Indian Institute of Technology, Madras, and the M.S. and Ph.D. degrees from Duke University, Durham, NC, all in electrical engineering, in 1978, 1980, and 1984, respectively.

He was with Digital Equipment Corp. from 1984 to 1993. Since 1993 he has been a Member of the Technical Staff in the Power Technology Group at AT&T Bell Laboratories, Mesquite, TX.