# A Fixed-Frequency Quasi-Sliding Control Algorithm: Application to Power Inverters Design by Means of FPGA Implementation

Rafael R. Ramos, *Student Member, IEEE*, Domingo Biel, *Member, IEEE*, Enric Fossas, and Francesc Guinjoan, *Member, IEEE*

***Abstract*—In this paper a fixed-frequency quasi-sliding control algorithm based on switching surface zero averaged dynamics (ZAD) is reported. This algorithm is applied to the design of a Buck-based inverter, and implemented in a laboratory prototype by means of a field programmable gate array (FPGA), taking into account processing speed versus computational complexity trade-off. Three control laws, namely sliding control (SC), fixed-frequency quasi-sliding ZAD and PWM-based control have been experimentally tested to highlight the features of the proposed algorithm. According to the experimental results presented in the paper, the ZAD algorithm fulfills the requirement of fixed switching frequency and exhibits similar robustness properties in the presence of perturbations to those of sliding control mode.**

***Index Terms*—Fixed-frequency, FPGA, power converters, quasi-sliding control.**

## I. Introduction

POWER conditioning systems are often designed to supply an ac load from a dc source. The uninterruptible power supplies (UPS), photovoltaic systems (PV) connected to the utility grid and ac power sources constitute the most classical applications. The design of such systems must achieve a behavior as close as possible to ideal ac voltage or current sources, in the sense of fast transient response to load variations, steady-state accuracy and low total harmonic distortion (THD).

Several control schemes have been suggested for dc to ac conversion, depending on how the signal error is processed. For instance, in order to preserve the benefits of a fixed-frequency design, many tracking control techniques based on high-frequency pulse width modulation (PWM) have been proposed for Buck-based inverters [1]–[5]. In all of these cases, the control design is based on a power stage model, leading to output waveforms sensitive to power stage parameter variations, such as output load. Considering this sensitivity, several advanced linear control algorithms, for example multiple-loop dead-beat and adaptive controllers, have been suggested to improve the robustness and the dynamic response of inverter design. Moreover, the corresponding inverter implementation includes a digital programmable platform in the control loop because of the computational complexity involved in such control algorithms. In this sense, several implementations based on microprocessors, DSP and FPGA can be found in the literature [6]–[12], in agreement with switching frequency versus control loop processing time trade-off.

Sliding-mode control techniques have also been proposed as an alternative to PWM control strategies in dc–dc switching regulators since they make these systems highly robust to perturbations, namely variations of the input voltage and/or in the load [13]–[17]. For this reason, tracking control schemes based on these techniques have also been applied to the design of high-efficiency Buck-based inverters. In this case, the converter output is forced to track an external sinusoidal reference by means of an appropriate sliding-mode control action [18]–[21]. However, because of the sliding mode control principles, the resulting designs operate at a variable switching frequency, this leading to an undesirable chattering phenomenon and hindering the design of the inverter filter elements. In order to partially overcome these drawbacks, analog implementations of sliding mode control have traditionally been carried out by means of a hysteresis comparator [22]–[24], which provides a variable bounded switching frequency. In addition, and regarding hysteresis comparator based implementations, several authors have suggested alternative designs for combining the robustness properties of sliding mode control with a fixed-switching frequency operation mode. One approach is based on fixing the switching frequency by means of a variable width hysteresis comparator [25]–[27], this leading to a cumbersome analog implementation where hysteresis width depends on the converter parameters. As an alternative, the addition of a fixed-frequency external signal to the switching surface has also been presented in [28]–[30]. Hence, in these works the switching instants do not only depend on the switching surface behavior.

At the same time, the increasing performances of digital platforms such as DSP or FPGA allow specific discrete-time sliding controllers as in [31]–[33]. This approach, known as quasi-sliding control, also seeks fixed switching frequency operation by obtaining control algorithms which force a switching surface null value at the end of the desired switching period. Several applications of such algorithms to the inverter design can be found in the works of [34] and [35].

Manuscript received January 30, 2002; revised October 7, 2002. This work was supported by the Ministerio de Ciencia y Tecnología, Spain, under Grant DPI2000-1509-CO3-02,03. Recommended by Associate Editor S. B. Leeb.

R. R. Ramos and D. Biel are with the UPC-EUPVG, Vilanova i la Geltrú 08800, Spain (e-mail: lara@eel.upc.es; biel@eel.upc.es).

E. Fossas is with UPC-ETSEIB, Institut d'Organització i Control de Sistemes Industrials, Barcelona 08028, Spain (e-mail: fossas@ioc.upc.es).

F. Guinjoan is with UPC-Mòdul C4, Barcelona 08034, Spain (e-mail: guinjoan@eel.upc.es).

Digital Object Identifier 10.1109/TPEL.2002.807164

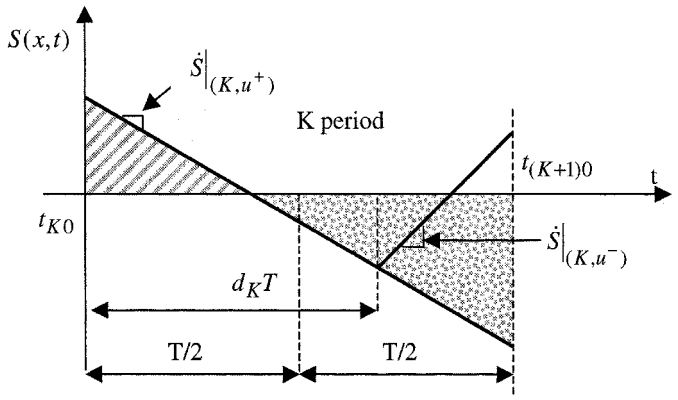


Fig. 1. ZAD principle.

Alternatively, a fixed-frequency quasi-sliding control algorithm based on switching surface zero averaged dynamics (ZAD) is reported in this paper. This algorithm is applied to the design of a Buck-based inverter, and implemented in a laboratory prototype by means of a field programmable gate array, taking into account the processing speed and computational complexity trade-off.

The paper is organized as follows: the ZAD control algorithm is introduced in section two; the main characteristics of a sliding mode controlled power inverter are presented in section three for the subsequent application of the ZAD control algorithm; section four is devoted to the description of the FPGA implementation of the proposed algorithm; for comparative purposes, simulation and experimental results of three control laws, namely sliding control (SC), fixed-frequency quasi-sliding ZAD and PWM-based control, are presented in section five to highlight the features of the proposed algorithm. Finally, the conclusions of this work are pointed out in section six.

## II. Zero Average Dynamics Control Algorithm

This section contains a summary of the ZAD algorithm [36] and deals with a quasi-sliding mode strategy based on the achievement of switching surface zero average dynamics in each cycle in steady-state. Starting from a general, SISO, autonomous, nonlinear system defined by

$$\hat{x} = f(x) + g(x) \cdot u \tag{1}$$

where $x \in \Re^n$ and $f$ and $g$ are vector fields defined on $\Re^n$, it is assumed that the system is governed by a switching surface $S(x, t)$ and a sliding control law

$$u(x, t) = \begin{cases} u^+ & \text{if } S(x, t) > 0 \\ u^- & \text{if } S(x, t) < 0 \end{cases} \tag{2}$$

which will be modified into a pulse width modulation as

$$u(x, t) = \begin{cases} u^+ & \text{if } KT \le t < (K + d_K)T \\ u^- & \text{if } (K + d_K)T \le t < (K + 1)T \end{cases} \tag{3}$$

where $T$ is the switching period and $d_K$ is the duty cycle in the $K$-period. The ZAD control algorithm imposes the control variable $u$ to force a switching surface zero average dynamics in steady-state, that is

$$\langle S(x, t) \rangle = \frac{1}{T} \cdot \int_{KT}^{(K+1)T} S(x, \tau) \cdot d\tau = 0. \tag{4}$$

Referring to Fig. 1, where $t_{K0}$ and $t_{(K+1)0}$ stand for the instants at the beginning and the end of the $K$ period, the control law is obtained under the hypothesis of linear approximation for the switching surface. As a consequence, the switching surface derivatives defined as $\dot{S}|_{(K, u^+)}$ and $\dot{S}|_{(K, u^-)}$ are considered constant during the period, and given by

$$\begin{aligned} \dot{S}\Big|_{(K, u^+)} &= \frac{\partial S}{\partial x} \cdot \left[ f[x(t_{k0})] + u^+ \cdot g[x(t_{k0})] \right] \\ \dot{S}\Big|_{(K, u^-)} &= \frac{\partial S}{\partial x} \cdot \left[ f[x(t_{k0})] + u^- \cdot g[x(t_{k0})] \right]. \end{aligned} \tag{5}$$

The ZAD control algorithm is deduced according to two different cases that can be distinguished from the switching surface behavior, namely the following.

A) If $S[x(t_{K0}), t_{K0}] \ge 0$ and $S[x(t_{K0}), t_{K0}] + (T/2)\dot{S}|_{(K, u^+)} \ge 0$ (condition verified during transient state), the dashed area will always be greater than the dotted one. Since in this case eq. (4) cannot be fulfilled for the $K$ period, the control action at $t_{K0}$, that is $u(t_{K0}) = u^+$, holds throughout the period, forcing the value of $S(x, t)$ to decrease. Thus, no switching action occurs and $d_K = 1$.

B) If $S[x(t_{K0}), t_{K0}] \ge 0$ and $S[x(t_{K0}), t_{K0}] + (T/2)\dot{S}|_{(K, u^+)} < 0$ (condition verified in steady-state), the dotted area will always be greater than the dashed one. In this case, condition (4) can be fulfilled during the $K$ period if the control action at $t_{K0}$, $u(t_{K0}) = u^+$, is switched to $u^-$ after a time interval $d_KT$, where

$$d_K = 1 - \sqrt{\frac{\left|\dot{S}\Big|_{(K, u^+)}\right| - 2 \cdot \frac{|S[x(t_{K0}), t_{K0}]|}{T}}{\left|\dot{S}\Big|_{(K, u^+)}\right| + \left|\dot{S}\Big|_{(K, u^-)}\right|}}. \tag{6}$$

A similar reasoning can be applied to $S[x(t_{K0})] \le 0$. All the cases and the corresponding control actions are summarized in Table I [36].

From these results, it can be concluded that the duty cycle $d_K$ corresponding to the ZAD control algorithm can be determined provided that $S[x(t_{K0}), t_{K0}]$, $\dot{S}|_{(K, u^+)}$ and $\dot{S}|_{(K, u^-)}$ as well as the switching period $T$ fixed by the user, are known.

## III. Sliding Mode Controlled Buck Inverter

The previous control algorithm has been applied to the design of a Buck-based high-frequency inverter. This work is focused on the sliding control loop design of a full-bridge Buck

TABLE I
ZAD CONTROL ALGORITHM

| | |
|---|---|
| $S[x(t_{K0}),t_{K0}]\geq 0$ and $S[x(t_{K0}),t_{K0}]+\frac{T}{2}\dot{S}\|_{(K,u^+)}\geq 0$ | $u(t_{K0})=u^+;d_K=1$ (no switching) |
| $S[x(t_{K0}),t_{K0}]\geq 0$ and $S[x(t_{K0}),t_{K0}]+\frac{T}{2}\dot{S}\|_{(K,u^+)}<0$ | $u(t_{K0})=u^+$, switching at $d_KT$: $d_K=1-\sqrt{\frac{\left\|\dot{S}\|_{(K,u^+)}\right\|-2\cdot\frac{\|S[x(t_{K0}),t_{K0}]\|}{T}}{\left\|\dot{S}\|_{(K,u^+)}\right\|+\left\|\dot{S}\|_{(K,u^-)}\right\|}}$ |
| $S[x(t_{K0}),t_{K0}]\leq 0$ and $S[x(t_{K0}),t_{K0}]+\frac{T}{2}\dot{S}\|_{(K,u^-)}\leq 0$ | $u(t_{K0})=u^-;d_K=1$ (no switching) |
| $S[x(t_{K0}),t_{K0}]\leq 0$ and $S[x(t_{K0}),t_{K0}]+\frac{T}{2}\dot{S}\|_{(K,u^-)}>0$ | $u(t_{K0})=u^-$, switching at $d_KT$: $d_K=1-\sqrt{\frac{\left\|\dot{S}\|_{(K,u^-)}\right\|-2\cdot\frac{\|S[x(t_{K0}),t_{K0}]\|}{T}}{\left\|\dot{S}\|_{(K,u^+)}\right\|+\left\|\dot{S}\|_{(K,u^-)}\right\|}}$ |

inverter, as that depicted in Fig. 2, where the Buck stage behavior can be represented by means of the following piecewise state equations:

$$\frac{d}{dt}\begin{bmatrix} i \\ vo \end{bmatrix}=\begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}\cdot\begin{bmatrix} i \\ vo \end{bmatrix}+\begin{bmatrix} \frac{E}{L} \\ 0 \end{bmatrix}\cdot u. \quad (7)$$

The control signal $u$ drives the power switch states and takes discrete values, namely $u\in\{-1,1\}$, this resulting in an LC filter input voltage of $-E$ or $+E$. The desired ac regulated output voltage is achieved by designing a sliding control loop based on the following switching surface proposed by Carpita *et al.* [23]

$$S(x,t)=\alpha\cdot(Vref(t)-vo)+\left(\frac{dVref(t)}{dt}-\frac{dvo}{dt}\right) \quad (8)$$

and the control law

$$u=\begin{cases}+1 & \text{if } S(x,t)>0\\ -1 & \text{if } S(x,t)<0\end{cases} \quad (9)$$

where $Vref(t)=A\cdot\sin(2\pi ft)$ is the reference signal. As the authors have shown, this design leads to the desired steady sliding motion, that is $vo=Vref(t)=A\cdot\sin(2\pi ft)$.

Starting from this previous work, the following section is devoted to the implementation of the quasi-sliding ZAD algorithm applied to the switching surface given by (8) for a Buck-based inverter design.

## IV. FPGA-BASED ZAD CONTROL LOOP IMPLEMENTATION

Control loop implementation assumes the design of a Buck-based inverter operating at a fixed switching frequency ranging from 20 to 40 kHz. In addition, the parameters of the Buck power stage are designed to fulfill the linear behavior approximation of the switching surface given in (8) for the desired switching period.

As has been previously shown in Table I, the control algorithm must take into account several cases, some of which involve nonlinear arithmetic. Because of the algorithm complexity, a digital implementation appears to be the best solution.

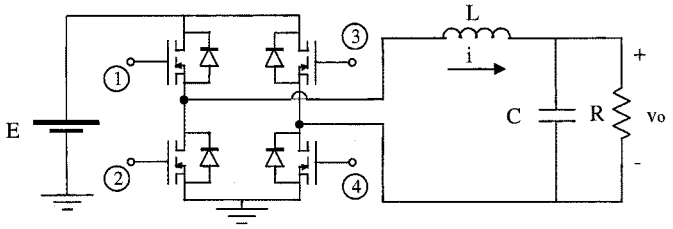


Fig. 2. DC/AC Buck power stage.

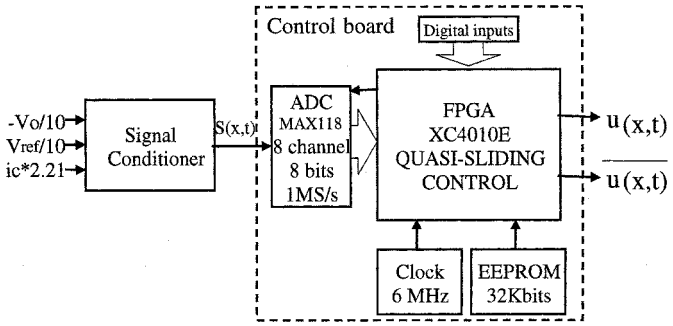


Fig. 3. ZAD quasi-sliding control block diagram.

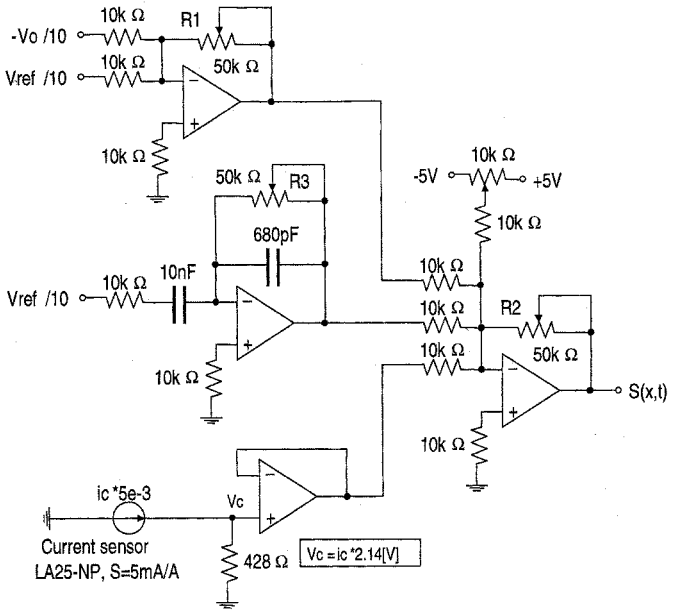


Fig. 4. Analog signal conditioner circuit.

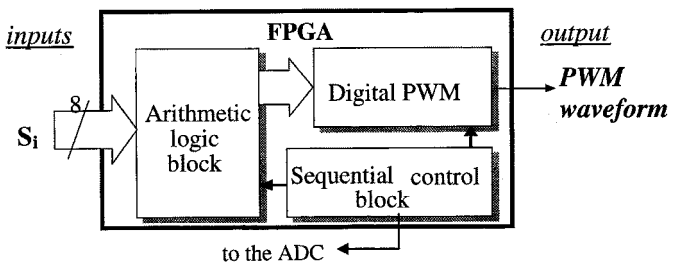


Fig. 5. Functional block diagram of the designed FPGA.

Several digital platforms can be considered, such as general-purpose microprocessors, digital signal processors (DSP)

and high-density programmable logic devices like field programmable gate arrays (FPGA) or complex programmable logic devices (CPLD).

The final selection should take into account several features such as processing speed, device capability, design environment and device cost. As for fixed-frequency PWM waveforms, the main processing speed requirements to preserve the expected closed-loop dynamics are that the digital processor supplies the proper control action at the beginning of the period and that the computing time does not exceed in any case the time interval during which the control value holds before switching. If these two requirements are fulfilled, the control implementation can be considered as cycle-by-cycle control. In the case of the dc/ac power converters considered here, the switching frequency ranges from 20 to 40 kHz, which means that obtaining a minimum duty cycle of 10% will require a control law computing time less than 10% of the switching period, thus ranging from 2.5 $\mu$s to 5 $\mu$s for the present application. This computing-time requirement rules out the use of general-purpose microprocessors or DSP, which are based on software design. Alternatively, because of its high-speed processing capability and its embedded hardware design, an FPGA has been finally selected for the present case.

Fig. 3 shows a block diagram of the XC4010E FPGA-based implementation of the ZAD quasi-sliding control algorithm. This block diagram includes an analog signal conditioner, an analog-to-digital converter (ADC) and an FPGA programmable logic device with its corresponding external clock and EEPROM memory to store the FPGA configuration. Some aspects of the design of these blocks are described in the following paragraphs.

### A. Analog Signal Conditioner

The signal conditioner is in charge of supplying the value of the switching surface $S(x, t)$ given by (8) to the ADC, and is designed by means of conventional OpAmp's-based circuitry, as shown in Fig. 4. The Buck output voltage $v_o$ is sensed by means of an AD215BY wideband isolation amplifier, whereas the capacitor current is acquired with an LA25-NP current sensor.

### B. AD Converter

The switching surface value $S(x, t)$ is sampled and digitized by the ADC at a fixed rate for the subsequent FPGA processing. As far as ADC selection is concerned, the following related parameters must be taken into account.

1) Maximum allowable sampling frequency: the maximum sampling frequency is lower bounded by the effective sampling frequency used in the design, and upper bounded by the ADC cost.
2) Analog-to-digital time conversion. The ADC time conversion is added to the FPGA computing time, thus increasing the overall control-loop processing time. This parameter must be chosen in order to preserve the cycle-by-cycle control concept.
3) Number of bits of the digital conversion. This parameter affects both the desired output voltage waveform and the FPGA computing time, since the higher the number of bits, the smaller the quantization error affecting the output voltage waveform, but the longer the control algorithm computing time becomes. This trade-off can be solved from several simulations by including the effects of a quantizer of $n$ bits and by numerically evaluating the dependence of an error index such as Total Harmonic Distortion (THD) through the number of bits. Simulations show a good-enough closed-loop output voltage response for an 8-b quantizer.

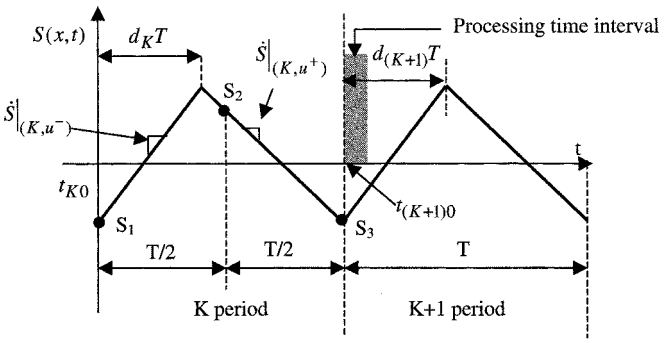


Fig. 6. Schematic algorithm procedure.

TABLE II
EXPRESSIONS FOR THE PARAMETER $D$

| | $d_K > 1/2$ | $d_K \leq 1/2$ |
|---|---|---|
| S1≥0 | $D = \frac{S_1 + S_3 - 2 \cdot S_2}{(1 - d_K)}$ | $D = \frac{S_1 + S_3 - 2 \cdot S_2}{d_K}$ |
| S1<0 | $D = \frac{2 \cdot S_2 - S_1 - S_3}{(1 - d_K)}$ | $D = \frac{2 \cdot S_2 - S_1 - S_3}{d_K}$ |

TABLE III
SWITCHING SURFACE DERIVATIVES IN TERMS OF $S_1$, $S_2$, $S_3$ AND $D$

| | $d_K > 1/2$ | $d_K \leq 1/2$ |
|---|---|---|
| S3≥0 and S1≥0 | $\left\|\dot{S}\right\|_{(K,u^+)} = 0 - 2 \cdot (S_2 - S_1)$ | $\left\|\dot{S}\right\|_{(K,u^+)} = D - 2 \cdot (S_3 - S_2)$ |
| S3≥0 and S1<0 | $\left\|\dot{S}\right\|_{(K,u^+)} = D - 2 \cdot (S_2 - S_1)$ | $\left\|\dot{S}\right\|_{(K,u^+)} = 0 - 2 \cdot (S_3 - S_2)$ |
| S3<0 and S1≥0 | $\left\|\dot{S}\right\|_{(K,u^-)} = D + 2 \cdot (S_2 - S_1)$ | $\left\|\dot{S}\right\|_{(K,u^-)} = 0 + 2 \cdot (S_3 - S_2)$ |
| S3<0 and S1<0 | $\left\|\dot{S}\right\|_{(K,u^-)} = 0 + 2 \cdot (S_2 - S_1)$ | $\left\|\dot{S}\right\|_{(K,u^-)} = D + 2 \cdot (S_3 - S_2)$ |

In accordance with the aforementioned trade-offs, a MAX118 A/D (eight channels, 8 b of resolution, 1 Msps of maximum sampling frequency and 660 ns of conversion time) has been adopted for the present design.

### C. FPGA Design

The functional block diagram of the designed FPGA is shown in Fig. 5, where three different blocks can be distinguished, namely.

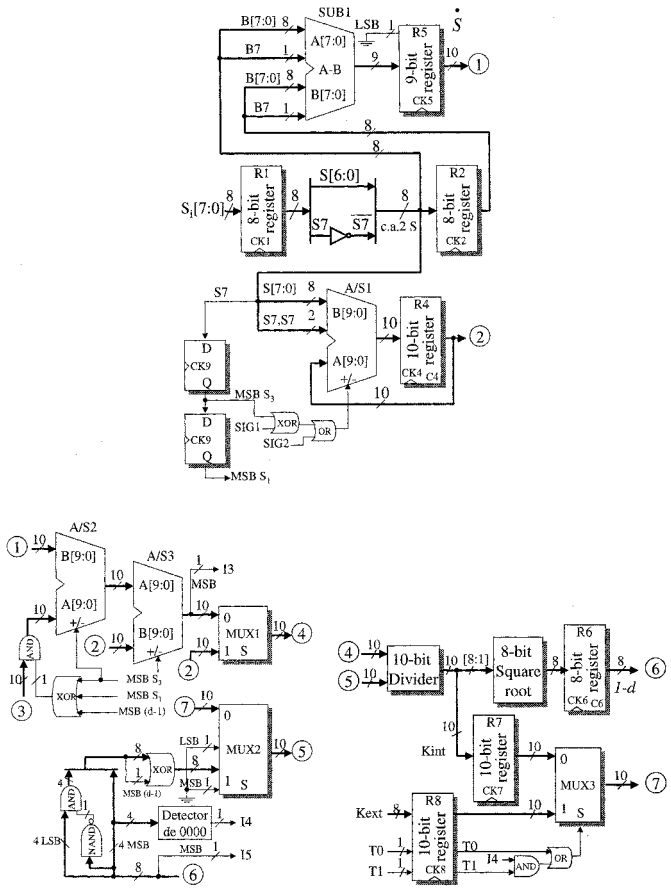


Fig. 7. Arithmetic block diagram.

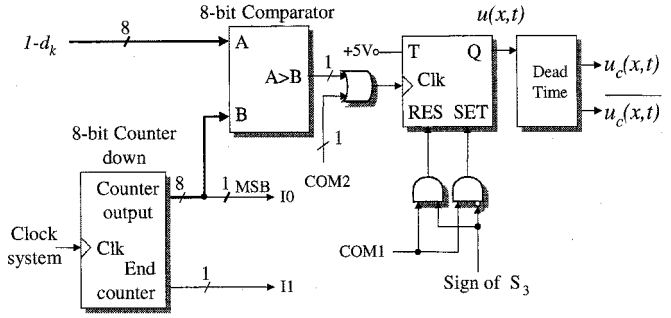


Fig. 8. Digital PWM block diagram.

1) The arithmetic block, which computes the ZAD algorithm from the digitized samples of the switching surface and supplies the corresponding duty cycle digital value, according to the expressions of Table I.
2) The digital PWM block, which is in charge of generating the PWM output waveform from the duty cycle value, and of fixing the desired switching frequency.
3) The sequential control block, which generates all the FPGA and ADC control signals.

Let us illustrate the main steps of the computational procedure embedded in the FPGA, by obtained the duty cycle determination of the $K+1$ period from the values sampled and computed during the $K$ period.

The whole computational procedure is based on a switching surface synchronous sampling at twice the desired switching

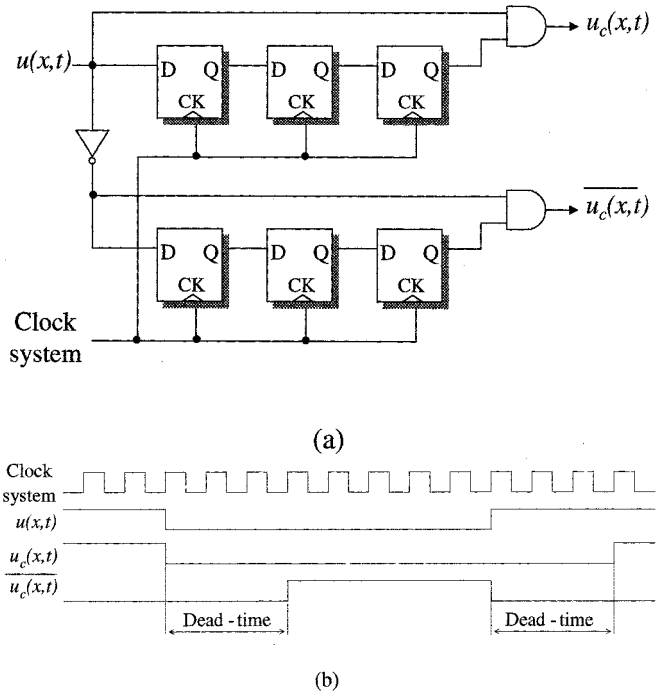


Fig. 9. Dead-time generator: (a) block diagram and (b) time diagram.

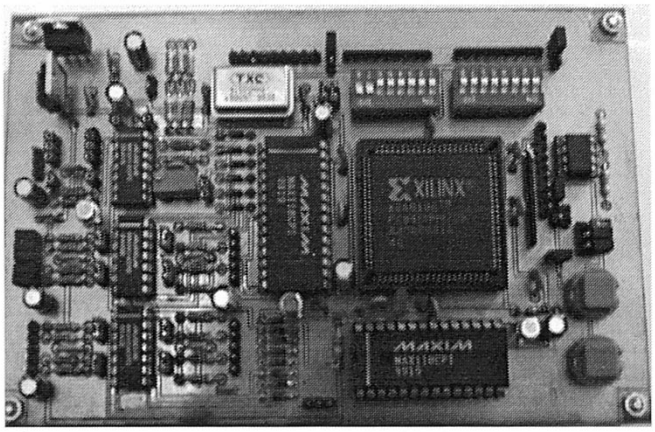


Fig. 10. Board of the ZAD control loop implementation.

frequency. Then, as can be seen in Fig. 6, during the $K$ period the following samples are known:

$$S_1 = S[x(t_{K0}),\, t_{K0}]; \quad S_2 = S[x(t_{K0}+T/2),\, t_{K0}+T/2];$$
$$S_3 = S\left[x(t_{(K+1)0}),\, t_{(K+1)0}\right]. \tag{10}$$

It should be pointed out that the values of $S_1$ and $S_3$ are obtained by sampling 826 ns prior to the end of the period to avoid the switching noise.

Assuming that the duty cycle of the $K$ period, $d_K$, is known, the first step is the computation of the parameter $D$, defined as

$$D = \left|\dot{S}\right|_{(K,\,u^+)}\left| + \right|\dot{S}\left|_{(K,\,u^-)}\right| \tag{11}$$

which corresponds to the denominator of the control laws given in Table I. This parameter can be easily calculated from $S_1$, $S_2$,

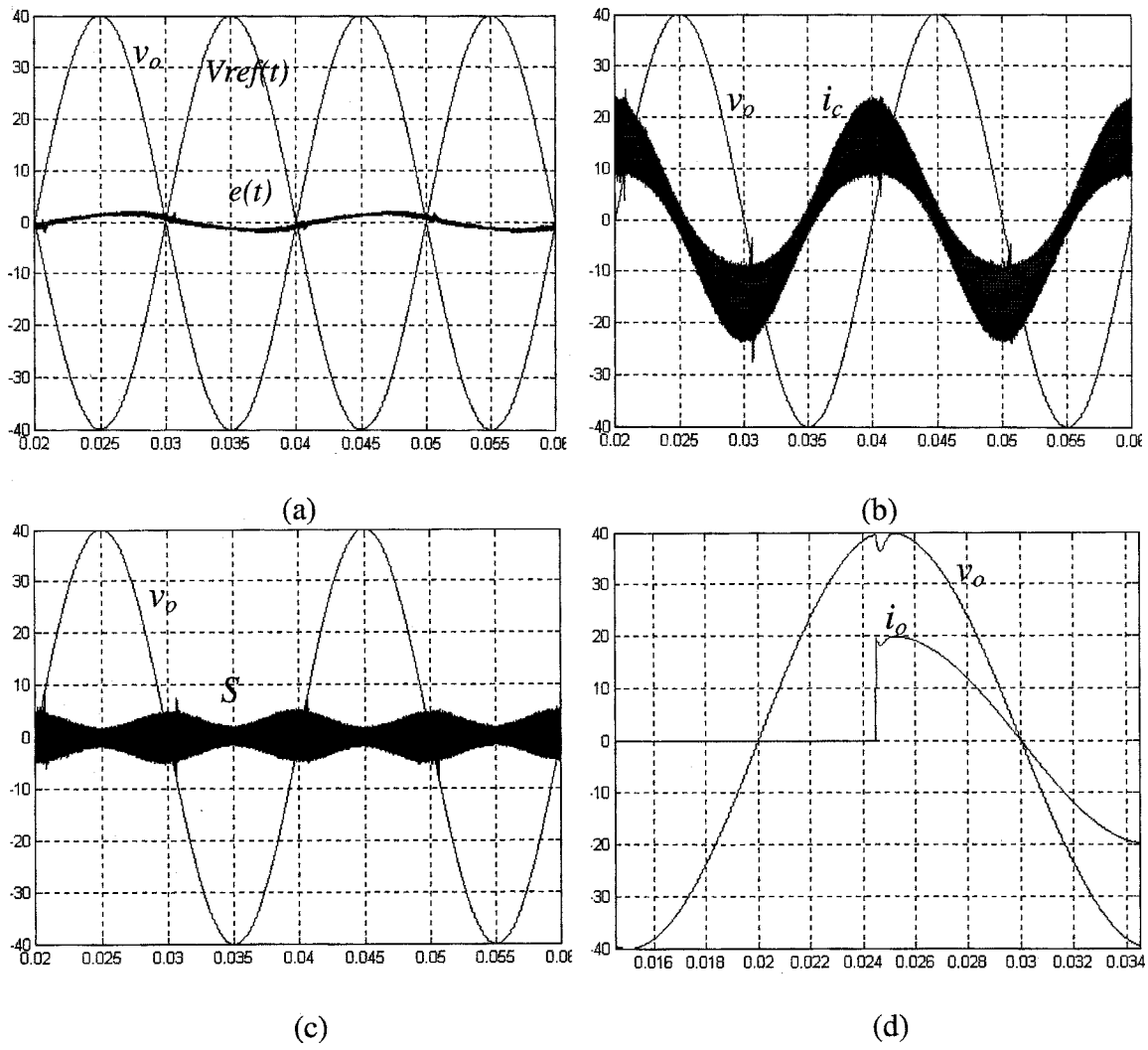


Fig. 11. ZAD simulation results. (a) Steady-state output voltage, $v_o$ [10 V/div], 180° shifted reference, $V_{ref}(t)$ [10 V/div], and voltage error, $e(t)$ [1 V/div]. (b) Steady-state output voltage, $v_o$ [10 V/div], and capacitor current, $i_c$ [0.5 A/div]. (c) Steady-state output voltage, $v_o$ [10 V/div], and switching surface value, $S$ [1 V/div]. (d) Output voltage, $v_o$ [10 V/div], and load current, $i_o$ [1 A/div], transient response for a load step change from open circuit to 20 Ω.

$S_3$ and $d_K$. That is, if $S_1 < 0$ and $d_K \leq 1/2$ (as depicted in Fig. 6), the following relations hold:

$$\begin{cases} \dot{S}\Big|_{(K,\,u^-)} = \dfrac{2(S_3 - S_2)}{T}. \\ S_3 = S_1 + \dot{S}\Big|_{(K,\,u^+)}.d_K T + \dot{S}\Big|_{(K,\,u^-)}.(1 - d_K)T \end{cases} \tag{12}$$

hence

$$D = \frac{2S_2 - S_1 - S_3}{d_K T}. \tag{13}$$

Similarly, the expressions of the parameter $D$ depending on the sign of $S_1$ and the value of $d_K$ can be easily derived. These expressions, normalized with respect to the switching period $T$, are summarized in Table II.

It can be noticed that these expressions may be applied provided that the two derivatives of (11) are defined during the $K$ period, this implying that the control value switches during the period. However, in transient state the ZAD algorithm holds the control action and the switching surface may remain positive (or negative) throughout the period. In this case, one of the two derivatives of (13) is not defined. Nevertheless, the parameter $D$ can be deduced from (5) and (11) as

$$D = \frac{\partial S}{\partial x} \cdot g(x) \cdot [u^+ - u^-] \tag{14}$$

or equivalently, by replacing (7) and (8) in (14)

$$D = \frac{2E}{LC}. \tag{15}$$

The FPGA implementation algorithm can identify this fact and then assign $D$ the value given by (15), which may be previously introduced and stored in the FPGA by means of the digital input shown in Fig. 3.

Once the value of $D$ is known, the next step is the computation of the switching surface derivatives, $\dot{S}|_{(K,\,u^+)}$ and $\dot{S}|_{(K,\,u^-)}$. As shown in Table III these derivatives may also be easily computed from the values of $S_1$, $S_2$, $S_3$ and $D$.

The value of $d_{(K+1)}$ is computed by assuming that the switching surface derivatives vary slowly with respect to the switching period (this assumption is reasonable due to

the output voltage low ripple), which enables the following approximation:

$$\dot{S}\Big|_{(K,u^+)} \approx \dot{S}\Big|_{(K+1,u^+)}; \quad \dot{S}\Big|_{(K,u^-)} \approx \dot{S}\Big|_{(K+1,u^-)} \tag{16}$$

and for instance, in the case of the second row of Table I, the duty cycle is finally computed as

$$d_{K+1} \cong 1 - \sqrt{\frac{\left|\dot{S}\right|_{(K,u^+)} - 2 \cdot \frac{|S[x(t_{(K+1)0}),t_{(K+1)0}]|}{T}}{\left|\dot{S}\right|_{(K,u^+)} + \left|\dot{S}\right|_{(K,u^-)}}} \tag{17}$$

which can be rewritten in terms of $S_1, S_2, S_3$ and $d_K$, according to Tables II and III, as

$$d_{K+1} \cong 1 - \sqrt{\frac{\frac{S_1+S_3-2S_2}{d_K} + 2S_2 - 4S_3}{\frac{S_1+S_3-2S_2}{d_K}}} \qquad \text{for } d_K < 1/2. \tag{18}$$

Regarding the FPGA algorithm implementation, the Arithmetic block is in charge of both identifying the different cases of Tables I–III and of computing the corresponding expressions, such as eq. (18). These tasks are carried out on an XC4010E-3-PC84 FPGA from Xilinx by means of the proper connection of registers, adders, multipliers and a digital square root circuit extractor [37]–[40], as shown in Fig. 7.

In addition, although a processing time is needed to evaluate the whole algorithm (see Fig. 6), according to Table I, the control value at $t_{(K+1)0}$ can be easily known by considering the sign of the switching surface at the beginning of the $K+1$ period (i.e., the sign of $S_3$). The proposed FPGA implementation provides this control value and holds it during the processing time.

The digital PWM block, implemented as shown in Fig. 8, is composed of an 8-b comparator and an 8-b step-down counter. The outputs of the Arithmetic block and the counter are compared to carry out the voltage to time conversion. An additional T-flip-flop is included to achieve the desired fixed-frequency synchronism. A dead-time block of three clock signal periods, shown in Fig. 9, is also included to prevent an input power stage short-circuit.

Finally, the "sequential control" block generates 24 control signals in order to manage the operation of both the inner blocks of the FPGA and the external A/D converter. The XC4010E-1-PC84 FPGA includes 10 000 logic gates and 800 flip-flops embedded in 400 configurable logic block (CLB) and 61 input/output block (IOB). The current design has consumed 245 CLB's (61% of the available CLB resources), 30 IOB's (49% of the available IOB resources) and 84 flip-flops (10% of the available FF resources).

The duty cycle computation at the beginning of each switching period consumes approximately between 10 and 16 clock periods, which leads to a total processing time ranging from 1.6 $\mu$s to 2.7 $\mu$s for a clock frequency of 6 MHz. As a consequence, for a switching period of 42.5 $\mu$s (23 kHz), a minimum duty cycle of 6.3% may be achieved in the worst case, this allowing a cycle-by-cycle control design.

## V. Simulation and Experimental Results

This section is devoted to the verifying of the proper operation of the ZAD algorithm by means of simulation and experimental results, and to experimental comparing the features of the proposed control algorithm with its sliding counterpart. A full-bridge Buck inverter has been built for this purpose with the following parameters.

1) Buck converter: $E = 50$ V, $C = 60$ $\mu$F, $L = 1.5$ mH, $R = 20$ $\Omega$.
2) Switching surface: $S = 0.5 \cdot (Vref - vo) + 0.8 \cdot 10^{-4} \cdot d(Vref - vo)/dt$.
3) User-defined parameters: switching frequency = 23 kHz, desired output voltage (reference signal) of $Vref(t) = 40 \cdot \sin(2\pi 50t)$.

The ZAD control loop Board including the FPGA, the analog conditioner circuitry and the ADC converter are shown in Fig. 10.

### A. ZAD Simulation Results

A MATLAB-SIMULINK simulation of the ZAD-controlled inverter, including an accurate model of the FPGA, has been carried out prior to the experimental verification. The steady-state behavior of the output voltage, the voltage error, the capacitor current and the switching surface are shown in Fig. 11(a)–(c), respectively.

In addition, Fig. 11(d) shows the output voltage and the load current for a load step change from open circuit to $R = 20$ $\Omega$. As can be seen, a fast recovery (less than the twentieth of the output voltage period) of the steady-state is obtained.

### B. ZAD Experimental Results

The same simulation conditions have been experimentally tested in the laboratory. The corresponding steady-state and load step change experimental results are shown from Fig. 12(a) to (d). As can be observed, they are in close agreement with those obtained in the simulations despite a steady-state error of 3%. Fig. 12(e) shows both the output voltage and the output current when the inverter is loaded with a full-wave rectifier; the THD measured in this case is approximately 0.3%, which can be considered a good inverter performance. Finally, the switching control signal spectrum, which evidences the fixed-frequency operation, is presented in Fig. 12(f).

### C. ZAD Versus Sliding Control

In order to explore the differences between the fixed-frequency quasi-sliding ZAD control and its sliding counterpart, Fig. 13(a)–(d) shows the experimental steady-state and load transient responses under the same laboratory conditions using the same switching surface and the sliding control law given in (9). In this case, the steady-state error is slightly lower (2% instead of 3%), and the steady-state recovery slightly faster than that corresponding to the ZAD algorithm. Fig. 13(e) shows the switching control signal spectrum, evidencing the expected variable-frequency operation of the sliding-mode control.

Other comparisons of the ZAD controlled inverter are reasonable, for example with PWM-based inverters due to their

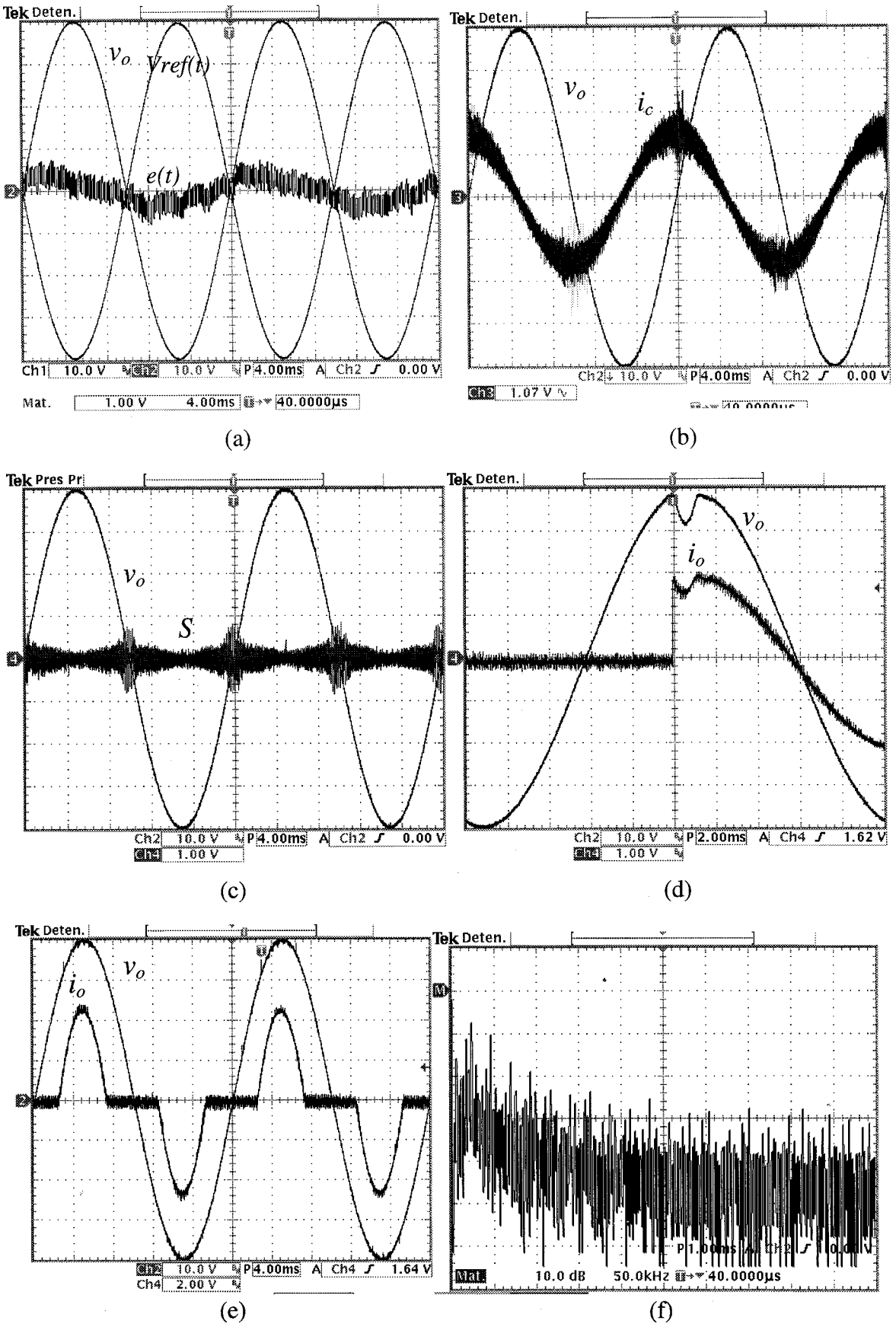


Fig. 12. ZAD experimental results. (a) Steady-state output voltage, $v_o$ [10 V/div], 180° shifted reference, $V_{ref}(t)$ [10 V/div], and voltage error, $e(t)$ [1 V/div]. (b) Steady-state output voltage, $v_o$ [10 V/div], and capacitor current, $i_c$ [0.5 A/div]. (c) Steady-state output voltage, $v_o$ [10 V/div], and switching surface value, $S$ [1 V/div]. (d) Output voltage, $v_o$ [10 V/div], and load current, $i_o$ [1 A/div], transient response for a load step change from open circuit to 20 Ω. (e) Steady-state output voltage, $v_o$ [10 V/div], and output current, $i_o$ [2 A/div], for a full-wave rectifier load. (f) Switching control signal spectrum [10 dB/div].

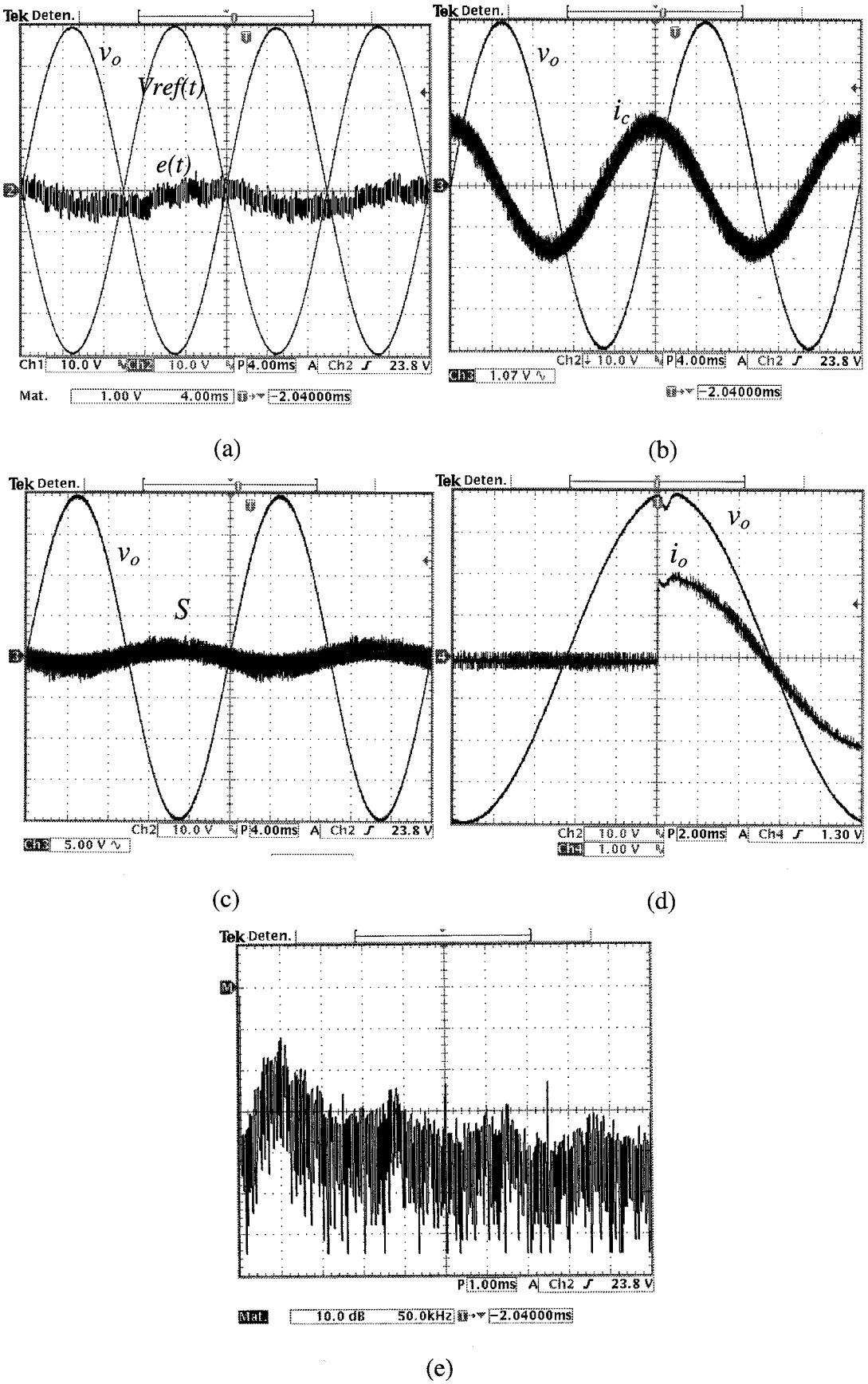


Fig. 13. Sliding control experimental results. (a) Steady-state output voltage, $v_o$ [10 V/div], 180° shifted reference, $V_{ref}(t)$ [10 V/div], and voltage error, e(t) [1 V/div]. (b) Steady-state output voltage, $v_o$ [10 V/div], and capacitor current, $i_c$ [0.5 A/div]. (c) Steady-state output voltage, $v_o$ [10 V/div], and switching surface value, $S$ [5 V/div]. (d) Output voltage, $v_o$ [10 V/div], and load current, $i_o$ [1 A/div], transient response for a load step change from open circuit to 20 $\Omega$. (e) Switching control signal spectrum [10 dB/div].

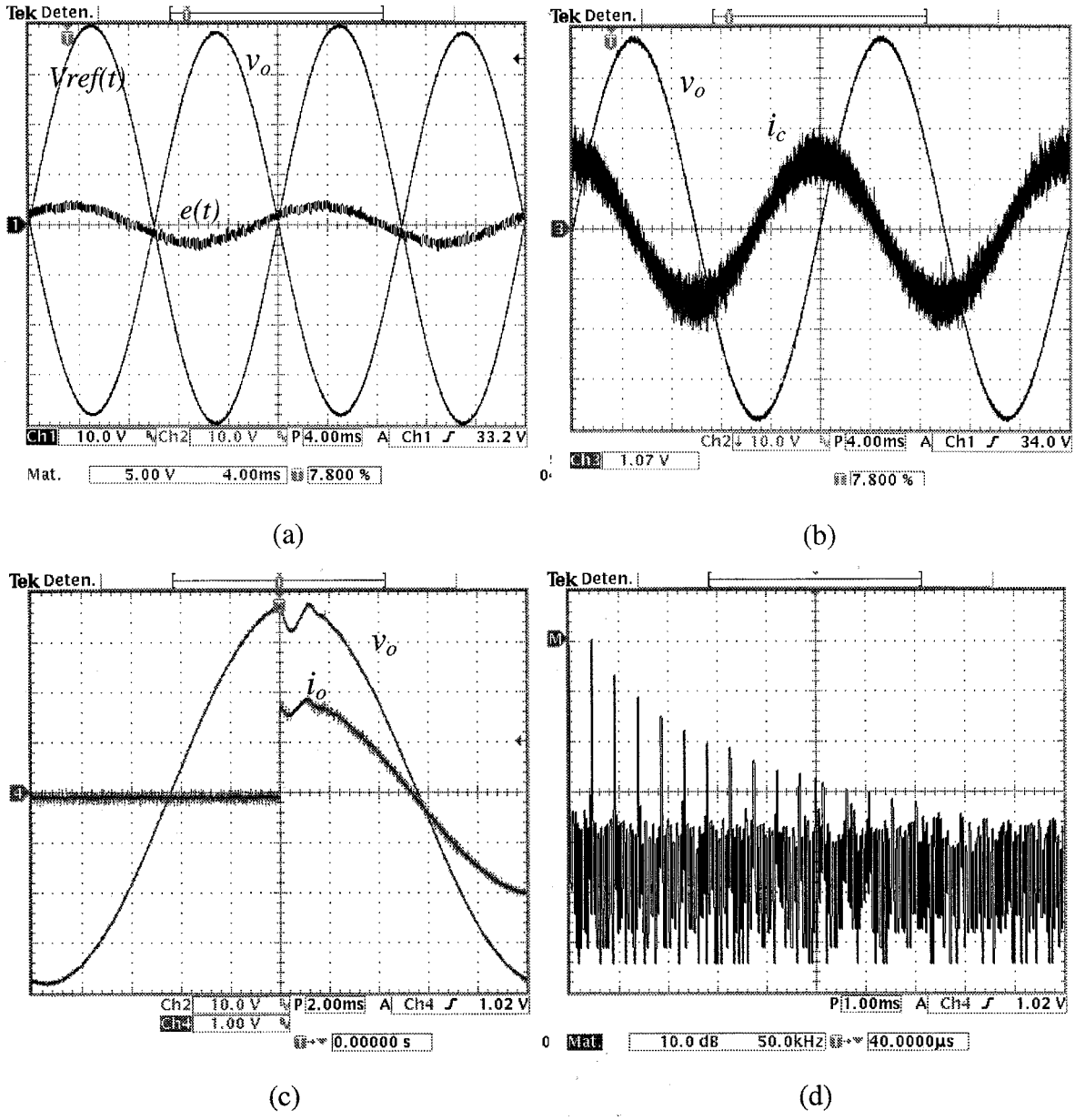


Fig. 14. PWM experimental results (controller adjusted for optimum transient response). (a) Steady-state output voltage, $v_o$ [10 V/div], 180° shifted reference, $V_{ref}(t)$ [10 V/div], and voltage error, $e(t)$ [5 V/div]. (b) Steady-state output voltage, $v_o$ [10 V/div], and capacitor current, $i_c$ [0.5 A/div]. (c) Output voltage, $v_o$ [10 V/div], and load current, $i_o$ [1 A/div], transient response for a load step change from open circuit to 20 Ω. (d) Switching control signal spectrum [10 dB/div].

inherent fixed-frequency operation. However, as many comparisons as existing PWM control algorithms can be considered. A first attempt is presented in this paper by comparing the proposed control algorithm with a PWM-based controller presented by Boudjema *et al.* [21], in terms of steady-state error and transient dynamics. The corresponding PWM control law of this work is deduced from the average dynamical model of the Buck converter by means of a pole placement technique, and is expressed in terms of the duty cycle as [21]

$$d = -5.58 \cdot 10^{-3} \cdot Vref(t) + k_P(Vref - vo) + k_D \frac{d(Vref - vo)}{dt}. \quad (19)$$

The same inverter has been experimentally tested under the above conditions with the PWM control law given in (19). A first set of measurements shown in Fig. 14(a) and (b) (steady-state) and 14(c) (transient response) is carried out adjusting the controller parameters $k_P$ and $k_D$ for an optimum transient response $(k_P = 1.39; k_D = 0.97{\cdot}10^{-4})$. The transient behavior of Fig. 14(c) is very similar to that obtained with the ZAD algorithm, but the steady-state error of 5 Vpp. detected in Fig. 14(a) is significantly greater (note that the scale for the middle trace is of 5 V/div. in this case). Finally, a second set of measurements now adjusting the PWM controller $(k_P = 1.95;\ k_D = 0.97 \cdot 10^{-4})$ to reduce the steady-state error to the minimum is shown in Fig. 15. As can be expected, the steady-state error reduction entails a transient response degradation.

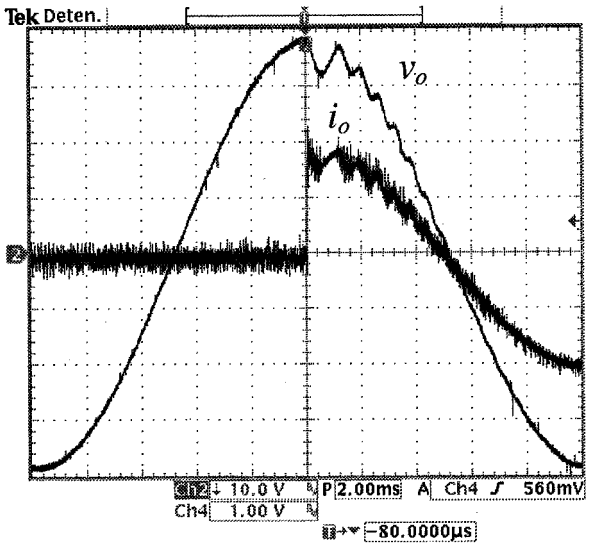


Fig. 15. PWM experimental results (controller adjusted for optimum steady-state error). Output voltage, $v_o$ [10 V/div], and load current, $i_o$ [1 A/div], transient response for a load step change from open circuit to 20 Ω.

## VI. Conclusion

This paper presents a fixed-frequency quasi-sliding control algorithm based on a switching surface zero average dynamics (ZAD) and its application to the design of dc–ac inverters. The algorithm is established under the hypothesis of linear dynamics of the switching surface during the switching period. A set of different control laws depending on the switching surface behavior has been obtained by means of straightforward geometrical considerations. It is worth noting that the resulting control laws do not depend on power stage parameters. Moreover, the ZAD algorithm can be applied irrespective of the switching surface, provided that the linear dynamics approximation is verified. In order to apply the ZAD algorithm to the design of a high frequency Buck inverter, a digital FPGA-based control loop implementation has been proposed taking into account the requirements of the control law nonlinear arithmetic and the overall control loop processing speed. A complete description of the ancillary circuitry, as well as the computational procedure embedded in an XC4010E-3-PC84 FPGA from Xilinx enabling a cycle-by-cycle control, have also been presented. Experimental results on a Buck inverter laboratory prototype are in close agreement with those obtained through numerical simulations and show good performances in terms of steady-state and transient responses, as well as good THD for nonlinear loads such as full-wave rectifiers. Additionally, the fixed-frequency operation requirement has been also evidenced. In addition, the same experiments carried out on the corresponding sliding-controlled inverter point out that the ZAD algorithm exhibits similar properties to its sliding control counterpart. As far as frequency operation is concerned, an additional comparison with a PWM inverter including a linear controller has shown that the trade-off between steady-state error and transient recovery time is better fulfilled by the ZAD control algorithm than by the classical PWM-based one. Other comparisons can be made, for instance with other PWM control algorithms in terms of steady-state error, dynamics and implementation complexity. However, since up to now the different PWM control techniques applied to the inverter design are mainly linear techniques based on a power stage model, it is believed that the ZAD fixed-frequency quasi-sliding algorithm would exhibit better robustness properties, considering its strong relation with the sliding control principles, this last statement being the subject of further research.

## References

[1] A. Capel, J. C. Marpinard, J. Jalade, and M. Valentin, "Large-signal dynamic stability analysis of synchronized current-controlled modulators. Application to sine-wave power inverters," *ESA J.*, vol. 7, pp. 63–74, 1983.

[2] A. Kawamura and R. G. Hoft, "Instantaneous feedback controlled PWM inverter with adaptative hysteresis," *IEEE Trans. Ind. Applicat.*, vol. IA-20, pp. 706–712, 1984.

[3] K. P. Gokale, A. Kawamura, and R. G. Hoft, "Dead-beat microprocessor control of PWM inverter for sinusoidal output waveform synthesis," *IEEE Trans. Ind. Applicat.*, vol. IA-23, pp. 901–910, Sept./Oct. 1985.

[4] P. Maussion, M. Grandpierre, J. Faucher, and J. C. Hapiot, "Instantaneous feedback control of a single-phase PWM inverter with nonlinear loads by sine wave tracking," in *Proc. IEEE Int. Conf. Ind. Electron. Contr. Instrum. (IECON)*, 1989, pp. 130–135.

[5] K. Jezernik, M. Milanovic, and D. Zadravec, "Microprocessor control of PWM inverter for sinusoidal output," in *Proc. Eur. Power Electron. Conf. (EPE)*, 1989, pp. 47–51.

[6] S. Jung, H. Huang, M. Chang, and Y. Tzou, "DSP-based multiple-loop strategy for single-phase inverters used in ac power sources," in *Proc. Power Electron. Spec. Conf. (PESC)*, 1997, pp. 706–712.

[7] H. Cha, S. Kim, M. Kang, and Y. Chung, "Real-time digital control of PWM inverter with PI compensator for uninterruptible power supply," in *Proc. Int. Conf. Ind. Electron. Contr. Instrum. (IECON)*, 1990, pp. 1124–1128.

[8] F. Botteron, H. Pinheiro, H. A. Grundling, J. R. Pinheiro, and H. L. Hey, "Digital voltage and current controllers for three-phase PWM inverter for UPS applications," in *Proc. Conf. Rec. IEEE Ind. Applicat. Conf.*, vol. 4, 2001, pp. 2667–2674.

[9] J. R. Espinoza and G. Joos, "DSP implementation of output voltage reconstruction in CSI-based converters," *IEEE Trans. Ind. Electron.*, vol. 45, no. 6, pp. 895–904, 1998.

[10] D. M. Baker, V. G. Agelidis, and C. V. Nayar, "Implementation of a zero-average current error control algorithm for inverters using a digital signal processor," in *Proc. IEEE Int. Power Electron. Congr. (CIEP)*, 1998, pp. 450–455.

[11] S. Jung, M. Chang, J. Jyang, L. Yeh, and Y. Tzou, "Design and implementation of an FPGA-based control IC for ac-voltage regulation," *IEEE Trans. Power Electron.*, vol. 14, no. 3, pp. 522–532, 1999.

[12] D. Bester, J. du Toit, and J. Enslin, "High performance DSP/FPGA controller for implementation of computationally intensive algorithms," in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, 1998, pp. 240–244.

[13] V. I. Utkin, *Sliding Mode and Their Applications in Variable Structure Systems*: Mir. Moscow, 1978.

[14] F. Bilanovic, O. Music, and A. Sabanovic, "Buck converter regulator operating in the sliding mode," in *Proc. 7th Int. PCI Conf. Power Conv.*, 1983, pp. 251–258.

[15] H. Sira-Ramirez, "Sliding motions in bilinear switched networks," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 8, pp. 919–933, 1987.

[16] R. Venkataramanan, A. Sabanovic, and S. Cuk, "Sliding mode control of DC-to-DC converters," in *Proc. IEEE Int. Conf. Ind. Electron. Contr. Instrum. (IECON)*, 1985, pp. 251–258.

[17] L. Martínez, J. Calvente, R. Griñón, A. Poveda, and E. Fossas, "Análisis of a bidirectional coupled-inductor Cuk converter operating in sliding mode," *CAS-1. IEEE Trans. Circuits Syst. I*, vol. 45, no. 4, pp. 355–363, 1998.

[18] K. Jezernik and D. Zadravec, "Sliding mode controller for a single phase inverter," in *Proc. Appl. Power Electron. Conf. Expo. (APEC)*, 1990, pp. 185–190.

[19] M. Carpita, M. Marchesioni, M. Oberti, and L. Puguisi, "Power conditioning system using sliding mode control," in *Proc. Power Electron. Spec. Conf. (PESC)*, 1988, pp. 623–633.

[20] D. Biel, E. Fossas, F. Guinjoan, E. Alarcón, and A. Poveda, "Application of sliding-mode control to the design of a Buck-based sinusoidal generator," *IEEE Trans. Ind. Electron.*, vol. 48, pp. 563–571, June 2001.

[21] F. Boudjema, M. Boscardin, P. Bidan, J. C. Marpinard, M. Valentin, and J. L. Abatut, "VSS approach to a full bridge Buck converter used for AC sine voltage generation," in *Proc. Int. Conf. Ind. Electron. Contr. Instrum. (IECON)*, 1989, pp. 82–89.
[22] H. Bühler, *Réglage par Mode de Glissement*: Presses Polytechniques Romandes, 1986.
[23] M. Carpita and M. Marchesoni, "Experimental study of a power conditioning using sliding mode control," *IEEE Trans. Power Electron.*, vol. 11, no. 5, pp. 731–742, 1996.
[24] B. Nicolas, M. Fadel, and Y. Chéron, "Sliding mode control of DC-to-DC converters with input filter based on the Lyapunov-function approach," in *Proc. Eur. Power Electron. Conf. (EPE)*, 1995, pp. 1338–1343.
[25] J. M. Ruiz, S. Lorenzo, I. Lobo, and J. Amigo, "Minimal UPS structure with sliding mode control and adaptative hysteresis band," in *Proc. Int. Conf. Ind. Electron. Contr. Instrum. (IECON)*, vol. 2, 1990, pp. 1063–1067.
[26] C. Chiarelli, L. Malesani, S. Pirondini, and P. Tomasin, "Single-phase, three level, constant frequency current reference hysteresis control for UPS applications," in *Proc. Eur. Power Electron. Conf. (EPE)*, 1993, pp. 180–185.
[27] L. Malesani, L. Rossetto, G. Spiazzi, and A. Zuccato, "An AC power supply with sliding-mode control," *IEEE Ind. Appl. Mag.*, pp. 32–38, 1996.
[28] J. F. Silva and S. S. Paulo, "Fixed frequency sliding modulator for current mode PWM inverters," in *Proc. Power Electron. Spec. Conf. (PESC)*, 1993, pp. 623–629.
[29] B. Nicolas, M. Fadel, and Y. Chéron, "Fixed-frequency sliding mode control of a single-phase voltage source with input filter," in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, 1996, pp. 470–475.
[30] H. Pinheiro, A. S. Martins, and J. R. Pinheiro, "A sliding mode controller in single phase voltage source inverters," in *Proc. Int. Conf. Ind. Electron. Contr. Instrum. (IECON)*, 1994, pp. 394–398.
[31] K. Furuta, "Sliding mode control of a discrete system," *Syst. Contr. Lett.*, vol. 14, pp. 145–152, 1990.
[32] W. Gao, Y. Wang, and A. Homaifa, "Discrete-time variable structure control systems," *IEEE Trans. Ind. Electron.*, vol. 42, pp. 117–122, Apr. 1995.
[33] S. Muthu and J. M. S. Kim, "Discrete-time robust sliding mode control for switching power converters," in *Proc. Power Electron. Spec. Conf. (PESC)*, 1997, pp. 210–216.
[34] S. Jung and Y. Tzou, "Discrete sliding-mode control of a PWM inverter for sinusoidal output waveform synthesis with optimal sliding curve," *IEEE Trans. Power Electron.*, vol. 11, no. 4, pp. 567–577, 1996.
[35] S. Muthu and J. Kim, "Discrete-time sliding mode control for output voltage regulation of three-phase voltage source inverters," in *Proc. Appl. Power Electron. Conf. Expo. (APEC)*, 1998, pp. 129–135.
[36] E. Fossas, R. Griñó, and D. Biel, "Quasi-sliding control based on pulse width modulation, zero averaged dynamics and the L2 norm," in *Advances in Variable Structure Systems, Analysis, Integration and Applications* , X. Yu and J.-X. Xu, Eds. New York: World Scientific, 2001, pp. 335–344.
[37] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Sofware Interface*. New York: Morgan Kaufmann, 1994.
[38] K. K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," *IEEE Trans. Comput.*, vol. 38, pp. 358–375, 1991.
[39] B. E. Saglam, G. Gosgül, and G. Dündar, "Comments on 'A systematic approach for design of digit-serial signal processing architectures'," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 369–370, 2000.
[40] J. C. Majithia, "A cellular array for the nonrestoring extraction of square roots," *IEEE Trans. Comput.*, vol. C-20, pp. 1671–1618, 1971.

**Rafael R. Ramos** (S'97) was born in Barcelona, Spain, in 1967. He received the B.S, and M.S. degrees in telecomunications engineering from the Universidad Politècnica de Catalufia, Barcelona, Spain, in 1990 and 1996, respectively.

Since 1990, he has been an Assistant Professor in the Departamento de Ingenierfa Electrónica, Universidad Politecnica de Cataluñia, Spain, where he teaches microcontroller. His research fields are related to nonlinear controller, sliding mode control, power electronics, adaptive signal processing and digital implementation of signal processing systems and nonlinear control systems.

**Domingo Biel** (S'97–M'99) received the B.S, M.S., and Ph.D. degrees in telecomunications engineering from the Universidad Politècnica de Cataluñia, Barcelona, Spain, in 1990, 1994, and 1999, respectively. His Ph.D. Thesis dissertation research was in the application of sliding-mode control to the signal generation in dc-to-dc switching converters.

Since 1994, he has been an Assistant Professor in the Electronic Engineering Department, Technical University of Catalonia, Spain, where he teaches power electronics and control theory. He is the author/coauthor of several communications in international congresses and workshops. His research fields are related to nonlinear control, sliding mode control, and power electronics.

**Enric Fossas** was born in Aiguafreda, Catalunya, Spain, in 1959. He received the M.S. and Ph.D. degrees in mathematics from the Universitat de Barcelona, Spain, in 1981 and 1986, respectively.

In 1986, he joined the Department of Applied Mathematics, Universitat Politecnica de Catalunya. In 1999, he moved to the Institute of Organization and Control, Universitat Politecnica de Catalunya and to the Department of Automatic Control and Computer Engineering where he is presently an Associate Professor. His research interests includes nonlinear control (theory and applications), particularly variable structure systems, with applications to switching converters.

**Francesc Guinjoan** (M'92) received the M.S. and Ph.D. degrees in telecommunications from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1984 and 1990, respectively, and the Doc.Sci. degree from the Universite Paul Sabatier. Toulouse, France, in 1992.

From 1984 to 1995, he taught control theory and analog electronics at the Escuela Universitaria Politècnica de Vilanova i la Geltrú. He is currently Associate Professor in the Departamento de Ingeniería Electrónica, Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona, Universitat Politècnica de Catalunya, Barcelona, Spain, where the teaches power electronics. His research interest include power electronics modeling, nonlinear circuit analysis and control, and analog circuit design.