

UNIVERSITY OF WESTMINSTER



WestminsterResearch

<http://www.wmin.ac.uk/westminsterresearch>

Adaptive high-performance velocity evaluation based on a high resolution time-to-digital converter.

John N. Lygouras¹
Theodore P. Pachidis¹
Kostas N. Tarchanidis²
Vassilis S. Kodogiannis³

¹ Department of Electrical and Computer Engineering, Democritus University of Thrace

² Department of Petroleum and Natural Gas Technology, Kavala Institute of Technology

³ Harrow School of Computer Science, University of Westminster

Copyright © [2008] IEEE. Reprinted from the IEEE Transactions on Instrumentation and Measurement, 57 (9), pp. 2035-2043.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Westminster's products or services. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of the University of Westminster Eprints (<http://www.wmin.ac.uk/westminsterresearch>).

In case of abuse or copyright appearing without permission e-mail wattsn@wmin.ac.uk.

Adaptive High-Performance Velocity Evaluation Based on a High-Resolution Time-to-Digital Converter

John N. Lygouras, Theodore P. Pachidis, *Member, IEEE*,
Kostas N. Tarchanidis, *Member, IEEE*, and Vassilis S. Kodogiannis, *Member, IEEE*

Abstract—In this paper, an improved method is presented to derive the velocity information in a pulse-number measurement/time-duration-type digital tachometer by processing its pulse train. The method incorporates encoder pulse counting and very accurate time measurement. The velocity sampling interval is not constant but is continuously modified. An adaptive algorithm provides a wide-range velocity evaluation with very good accuracy. The adaptation of the next sampling period, according to the instant velocity, results in better response times at low speeds and a very high accuracy at medium and high speeds. Compared to currently known methods, the time measurement resolution and, consequently, the velocity accuracy is improved by using the proposed method due to the inclusion of high-resolution time-to-digital converters in the design. The proposed configuration can be implemented in specific hardware by using field-programmable gate arrays (FPGAs), thus saving the computational power of the digital signal processor that supervises the system for higher level control tasks.

Index Terms—Adaptive sampling interval, digital encoder, field-programmable gate arrays (FPGAs), time-to-digital converter (TDC), velocity evaluation.

I. INTRODUCTION

MODERN position control techniques require increased processing speed and, consequently, computational power, particularly in high-speed systems. Recent trends in position control problems have adopted the use of microcontrollers or digital signal processors (DSPs) and digital techniques rather than analog circuits to implement system control functions. Direct digital control techniques have clear advantages in terms of improvement in performance, flexibility in implementation, and user adjustments of control strategies/configurations. In addition, microcontrollers or DSPs could utilize functions such as data acquisition and processing, monitoring, display, and diagnostics. Complex control algorithms are also easy to implement. The DSP's microcontroller

architecture contributes to the reduction in system hardware and its cost. Since simple position feedback provides poor performance and the controller gain increment results in an oscillatory system response, the digital positioning loop is usually supplemented by a velocity loop to initiate damping. Hence, actual velocity feedback becomes necessary.

The problem of optimum velocity evaluation, given a sequence of outputs from a digital position sensor, is important in many process applications and motion control systems. A large number of digital tachometers that have been mentioned in the literature [1]–[5] are based on the measurement/time duration (M/T)-type digital tachometer. Velocity measurement can be obtained by processing the frequency or the period of the encoder pulses. There are many methods on velocity measurement or estimation by an optical encoder. Among them there are two commonly used methods: 1) the M method and 2) the T method. In the M method, which is also called the fixed-time method, the number of pulses from the optical encoder is calculated during successive fixed-time intervals, and arithmetic differentiation is executed. In a variation of this method, the position history is accumulated over more than two successive time intervals. The phase difference of the codes between two succeeding sample periods corresponds to the velocity. To obtain an average velocity, considering that single differentiation results in poor accuracy due to spatial quantization, P_{dk} (i.e., the rotating angle or displacement in T seconds) is accumulated during N_c sampling periods. However, the required time for the velocity value calculation is increased by the nature of process itself. The method of arithmetic differentiation is used because of its simplicity in low-cost and low-requirement applications.

In the T method or the fixed-position method, the velocity is calculated as the interpulse angle divided by the time between sequential pulses. A small accuracy is achieved at low-speed ranges with the M method, whereas a high accuracy is achieved with the T method, which is applicable only to low-speed ranges. Several combined methods have also been proposed, e.g., the M/T method, which works in all speed ranges and has a high accuracy in the low-speed range. The method is extended to a system that is called constant sample-time digital tachometer (CSDT).

The constant elapsed time (CET) method is based on counting encoder pulses in a specific time interval, and the corresponding time is measured by counting the clock pulses of a high-frequency clock with period T_c . The measured time

Manuscript received August 6, 2007; revised January 16, 2008.

J. N. Lygouras and T. P. Pachidis are with the Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece.

K. N. Tarchanidis is with the Department of Petroleum and Natural Gas Technology, Kavala Institute of Technology, 65500 Kavala, Greece.

V. S. Kodogiannis is with the Harrow School of Computer Science, University of Westminster, W1B 2UW London, U.K.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2008.919039

interval should satisfy the conditions that it should be larger than or equal to the desired CET T_{el} and contain an integer number of encoder pulses. The double-buffered method that was presented in [1] is based on pulse counting and pulse period measurement during a fixed sampling time. A new architecture for processing pulses that are derived from an optical incremental encoder is described in [3], where information about absolute position and velocity is discussed. Encoder interface circuitry is implemented in a very high speed integrated circuit hardware description language. The elapsed time between two subsequent readouts is managed by the processor, according to previous estimates. In [4], a novel method of velocity measurement for motor drives with optical encoders is proposed. The method, which is called the “synchronous measurement method or the S method,” measures the velocity, which is synchronized with the alteration of pulse numbers. Accurate velocity measurement is reported in all speed ranges. A processing algorithm that employs a novel mixed time–frequency measurement technique of the encoder signals is presented in [5]. In all the relevant methods for speed evaluation, the accuracy, among other parameters, strongly depends on the resolution that the time intervals are measured.

In [6], the proposed speed measurement system is based on an advanced single-buffered method that utilizes both the pulse counting and the period measurement. The authors report an extremely wide speed range of 5×10^{-5} –1830 r/min, which is independent of the sampling time and was chosen to be between 100 μ s and 22 ms. The maximum static relative error is limited to 6.67×10^{-4} for a 1-ms sampling time or to 6.67×10^{-3} for a 100- μ s sampling time. This method is suitable for a real-time control and an offline speed analysis. Accurate determination of the surface velocity profile of an almost uniformly rotating drum is considered in [7], where a measuring wheel is placed in direct contact with the drum and is rigidly coupled to an encoder of high line count. A new algorithm is described, which computes the nonlinear filtering of the encoder output in such a manner to yield a very close approximation to the actual drum velocity under most circumstances.

Kavanagh [8] examines sensor nonideality that can introduce very significant errors into the tachometer output. It is shown that performance can greatly be improved (i.e., the noise present in the velocity signal is significantly reduced) by oversampling the counter values used for velocity calculation. The counting and oversampling operations that are inherent in the oversampled CSDT are implemented using a field-programmable gate array (FPGA). The FPGA acts as a peripheral device to a DSP. To exactly measure the duration between two (e.g., rising) edges of tachometer pulses, an auxiliary counter is commonly used. The addition of this auxiliary counter provides very accurate measurement for the period T of the encoder pulses and, thus, more accurate calculation of the velocity. To obtain high-precision time measurement, many authors use the interpolation technique that was proposed by Nutt [9]. In [10], Lygouras *et al.* describe a new method for processing information that is derived from an optical encoder to estimate velocity. The adaptation of the sampling interval according to instant rotational speed allows the evaluation of the velocity with very good accuracy at either low or high speeds,

and small response times at high speeds can be achieved. The circuit is a single-chip design that was implemented in a static random access memory (SRAM) FPGA. Recent improvements in FPGAs have offered new possibilities for the development of fast, precise, and cost-effective time-to-digital converters (TDCs) with direct coding. Using the FPGA devices that were produced in the 0.65- μ m (or less) complementary metal–oxide–semiconductor (CMOS) technology with user-programmable amorphous antifuse structures [11], TDCs whose resolution is better than 200 ps can be designed. Since the time coding is directly realized by the tapped digital delay lines, those TDCs feature very low conversion time [12].

In this paper, the design of a high-performance circuit for very accurate velocity evaluation over a wide range of speeds is described. A new technique for processing the pulses from an optical encoder is proposed, combining counters as in the M/T method and high-resolution TDCs on a single CMOS FPGA chip to derive the velocity information. The adopted scheme is an adaptive sampling interval technique that allows velocity evaluation with very high accuracy, both at very low and at high velocities. The instant encoder’s waveform period measurement is used to accordingly adapt the next sampling time interval. This is an adaptive method, which results in a greater accuracy at a wide range of velocity measurements rather than the simple CET or double-buffered method. Our method is based on a combination of three previous works: 1) the immediate measurement of encoder pulses and the corresponding time interval that many authors have previously described in the literature [1]–[4]; 2) the adaptive sampling interval method [10], which allows the widening of the measurement range of velocities with a very high accuracy; and 3) the improved method for measuring time intervals with 200-ps resolution and 43-s range [12].

In the following sections, the circuit’s most significant parts are described and analyzed, and the overall performance of the chip is discussed. The proposed design is described, and experimental results from the constructed velocity evaluation chip are presented. The circuit is implemented as a single-chip design on an all-layer copper SRAM process FPGA chip.

II. ACCURATE VELOCITY EVALUATION USING TDCs

The design of an interpolating time counter on a single CMOS FPGA chip that contains two 6-bit TDCs, two multibit synchronous counters that operate at a 100-MHz clock, one control register, and input/output circuits is described in [12]. The measured time range of the counter spans 0–43 s at a resolution of 200 ps. A similar technique for high-resolution measurement of the sampling period T is used in this paper. The operation principle of the tachometer is shown in Fig. 1. An encoder pulse counter (EPC) counts the encoder pulse number N_p that was contained in the time interval T (which is equal to 1 for the example in Fig. 1). The second main time counter (MTC) measures the duration of T_s from a stable high-frequency clock. The MTC counts the integer number N_c of the reference clock periods T_0 . To compensate for the fact that encoder edges asynchronously occur with the MTC clock edges and to obtain high precision in period T duration measurement

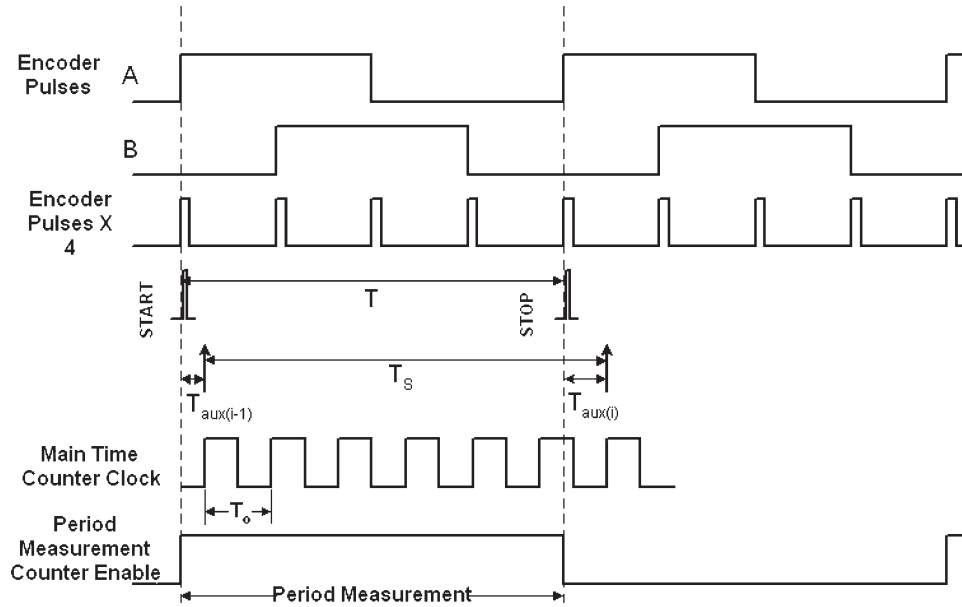


Fig. 1. Operation principle of the M/T -type tachometer that incorporates a TDC.

(with a quantization step of 400 ps instead of 10 ns), we have used a time interpolation technique, as in [12]. It involves splitting the measurement time interval T into three parts, as shown in Fig. 1. The first part is the integer number N_c of the reference clock periods T_0 ($T_s = N_c \times T_0$) and the two intervals (at the initial and final parts of the measurement interval) $T_{aux}(i-1)$ and $T_{aux}(i)$, each having a duration of less than one clock period, as shown in Fig. 1. T_0 is measured by a synchronous 23-bit binary counter to a 10-ns accuracy (at a 100-MHz clock), whereas the remaining two parts $T_{aux}(i-1) = N_A \tau_A$ and $T_{aux}(i) = N_B \tau_B$ are measured with the aid of two TDCs with quantization steps τ_A and τ_B , respectively. All these counters are simultaneously and synchronously reset with an encoder pulse edge. We have taken into consideration that, since the two TDCs are implemented in the same FPGA that uses identical delay cells, the average quantization step of each cell is the same for both TDCs (i.e., $\tau_A = \tau_B = \tau$).

The relation between the time intervals T and T_s is

$$\begin{aligned} T &= T_s + T_{aux}(i-1) - T_{aux}(i) \\ &= N_c T_0 + N_A \tau - N_B \tau \\ &= N_c T_0 + (N_A - N_B) \tau. \end{aligned} \quad (1)$$

The calculation of an accurate velocity estimate (in units of code changes per second) is then possible by using the following equation:

$$\bar{v} = \frac{N_p}{T} = \frac{N_p}{N_c T_0 + (N_A - N_B) \tau}. \quad (2)$$

The TDC process is directly realized by a time-coding line that consists of two tapped delay lines that work in differential mode, as shown in Fig. 2(a). The first line is a chain of 63 AND gates B_1, B_2, \dots, B_{63} , and the second line is a chain of 63 latch flip-flops L_1, L_2, \dots, L_{63} . If the latch delay propagation

time τ_1 is longer than the AND gate delay τ_2 ($\tau_1 > \tau_2$), the time quantization step (or the incremental resolution) τ_A (in seconds) of the TDC A is determined by their difference (i.e., $\tau_A = \tau_1 - \tau_2$). This difference can be made considerably shorter than the propagation delay time of the fastest gate in the FPGA logic block. The time intervals that we are interested in measuring are the time differences between rising edges of the pulses IN (the $START$ pulse in the rising edge of the encoder pulse at the start of the sampling interval for TDC A or the $STOP$ pulse at the end of the sampling interval for TDC B) and \overline{EN} (i.e., the rising edge of the next MTC clock pulse for both counters). When $\overline{EN} = L$, all latches in the chain are transparent, and their states follow the input, i.e., $IN = L$. Due to this autoreset feature, there is no need for a separate reset signal for the chain. When $IN = H$, this logic “ H ” propagates by sequentially changing the state of the register flip-flops, with a delay of τ_1 (in seconds) in each. After $n\tau_1$ (in seconds), the first n F/Fs in the chain will have to change their output Q to “ H ,” whereas the remaining F/Fs will still be at “ L .” The transition of \overline{EN} from low to high sets out the “hold” condition for each F/F until it is “Reset” by the corresponding AND gate output (i.e., L_1 by gate B_3 , L_2 by gate B_4 , L_3 by gate B_5 , and so on). Since $IN = H$ due to the $START$ signal and $\overline{EN} = L$ (and, thus, $\overline{ENOUT} = L$), the “ H ” that is produced in the Q output of the first F/F (with delay τ_1) “runs” through the F/F chain with a speed of $(1/\tau_1)$ F.Fs/s. Similarly, when $\overline{EN} = H$ (and, thus, $\overline{ENOUT} = H$), this “ H ” also runs through the AND gates chain with a speed of $(1/\tau_2)$ gates/s. Since $1/\tau_1 < 1/\tau_2$, the “ H ” state in the first chain will prevent the “ H ” state in the second chain after time $T_{aux}(i-1) = N_A(\tau_1 - \tau_2)$, where N_A is the number of F/Fs when the “ H ” state has already “run.” This state is coded in the second delay line by setting a high level at the output Q of the last cell whose CP input changes (i.e., from low to high). The “one-out-of-63” code that is produced in the output of the 63 F/F latch is coded in a 6-bit binary word through

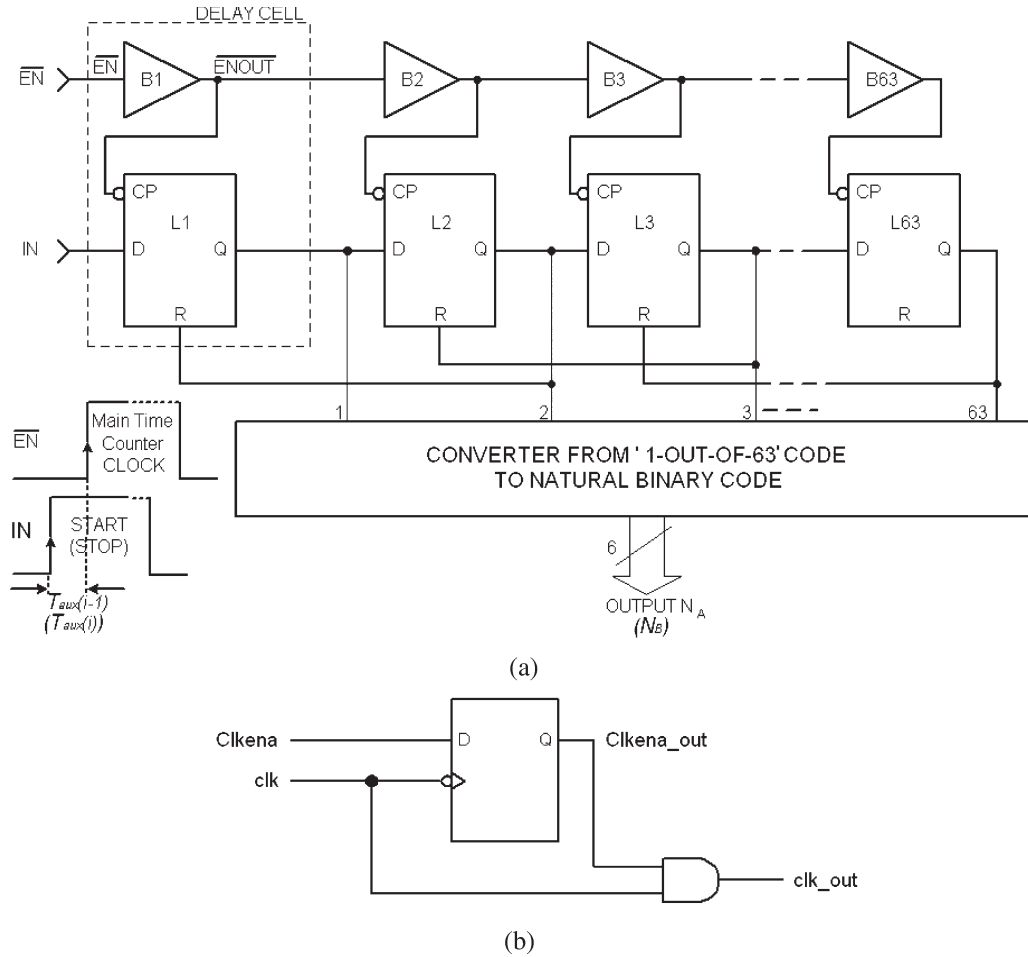


Fig. 2. Simplified logic diagram of (a) the TDC with direct coding and (b) detailed LE in the FPGA device that was used.

a combinational circuit. In the case where we assume a 100-MHz clock ($T_o = 10$ ns) and a resolution $\tau_A = 400$ ps for the TDC, 63 delay cells is sufficient, since only 25 cells are, in fact, used to achieve the total time span that is equal to the single period of the reference clock. The above configuration has been favored instead of others that are possible, since it can more easily be implemented with the logic elements (LEs) that are available in the used FPGA. An exactly similar configuration is also used to determine $T_{aux}(i)$ by using the second TDC B .

Currently available FPGA families are based on a 1.2-V, 90-nm, all-layer copper SRAM process and feature new logic structures that maximize performance and enable device densities that approach to 200 000 equivalent LEs. They support various I/O standards, along with support for 1-Gb/s source synchronous signaling with dynamic phase alignment circuitry and offer a complete clock management solution with an internal clock frequency of up to 550 MHz. The LE that has been used in our application is shown in Fig. 2(b). This LE can be used either as a delay cell as it stands (combining the L_1 F/F with the AND gate B_2 , the L_2 F/F with the AND gate B_3 , and so on) or as a single AND gate to implement the gate B_1 , provided that we give a permanent “H” to the “clkena” input and a low level in the “clk” input, which is also used here as the \overline{EN} input. Using such an LE, where the time delay is τ_1 (in

seconds) for the F/F and τ_2 (in seconds) for the AND gate, their difference will be the resolution of the TDC, which is uniform in consecutive cells and is quite stable in temperature.

III. VELOCITY EVALUATION CIRCUIT DESCRIPTION

The block diagram of the FPGA design for position detection and adaptive velocity evaluation circuit, according to the new proposed technique, is presented in Fig. 3. To simplify the operation, the circuit is divided in three subblocks (included in dashed lines). The system position measurement is achieved using the 20-bit up/down-position binary counter, which counts the encoder pulses (multiplied by four). This binary-encoded number determines the position of a moving platform in one direction. The direction detection circuit output determines the direction of counting (Up/*DOWN*) for the position counter and the velocity sign (VS).

As previously stated, the EPC counts the encoder pulses that are contained in the (adaptively changing) time interval T , whereas the second MTC measures the duration of T_s from a (100-MHz) stable high-frequency clock. The time interval T is measured between the rising edges of the input pulses, i.e., START and STOP. These pulses are produced at the rising edges of the first and the final measured encoder pulses, as defined by the variable sampling interval T (see Fig. 1).

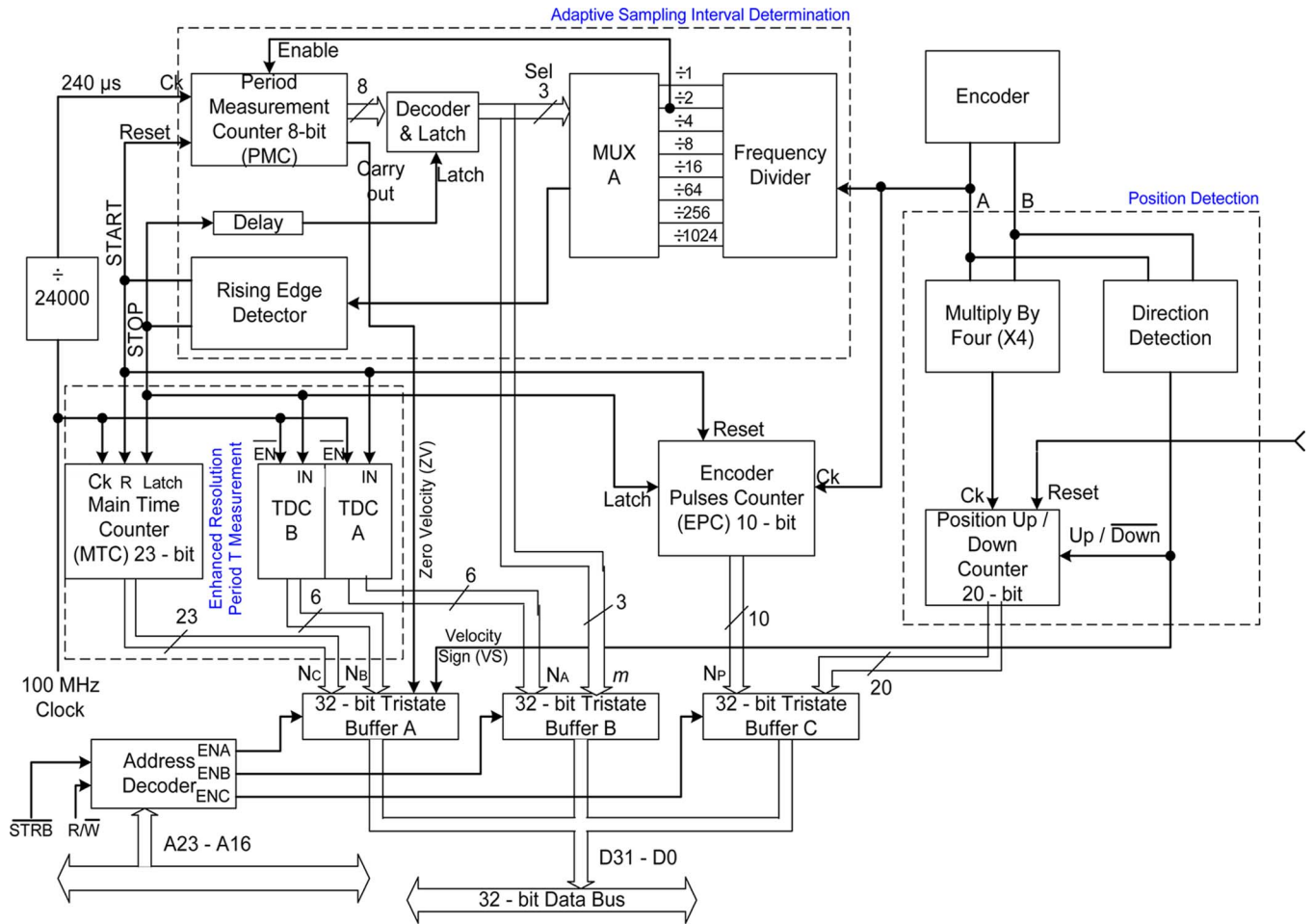


Fig. 3. Block diagram of the digital circuit for position detection and adaptive velocity evaluation, in combination with two TDCs, that is included in the FPGA chip.

The two TDCs, namely *A* and *B*, are used to measure $T_{aux}(i - 1)$ and $T_{aux}(i)$. The period measurement counter (PMC), MTC, EPC, and TDC *A* are reset at a rising edge of the encoder pulse waveform by the START pulse, whereas the TDC *B* is reset by the STOP pulse. The MTC and EPC are stopped by the next rising edge of the encoder pulse waveform, which has previously been divided by the frequency divider (e.g., divided by 1, 2, 4, 8, 16, 64, 256, or 1024) by using the STOP pulse. Our proposed method essentially deals with the adaptive encoder pulse period measurement but with a high accuracy, starting from measuring one encoder pulse period, when the previous measured period's maximum value is $T(i - 1) = T_{max}$. Then, the PMC counts to near its maximum value, and the ÷ 1 input is selected in MUX *A*, thus enabling the encoder pulses to pass undivided to the MUX *A* output. On the other hand, in the maximum measured velocity (minimum encoder pulses period), the PMC counts to near zero, thus selecting the ÷ 1024 input of MUX *A*. In this case, the time duration of 1024 encoder pulses is measured to accordingly increase accuracy. It is assumed here that the velocity is constant during one sampling interval, which can vary from 61.2 ms for the slower measurable rotational speed (1 r/min and supposing a 1000-p/rev encoder) to 7.44 ms for the higher velocity (6000 r/min). The current encoder's waveform period T_i measurement is used

here as a basis for adjusting the next sampling period T_{i+1} . This is accomplished by a decoder that decodes the output number of the PMC. We consider that the system is capable of measuring velocities from 1000 pulses per minute (ppm) = 16.667 pulses per second (pps) (= 1 r/min for rotational speed measurement) up to 6 million ppm = 1000 kpps (= 6000 r/min). In the case where the PMC reaches its maximum and an overflow occurs, it is considered to be a zero-velocity ($ZV = H$) condition. The relation between the PMC's output number and the decoder's output number m is shown in Table I. The relation between the encoder's pulse period T_i (measured by the PMC) and the decoder's output value is also shown, as well as the next sampling interval duration T_{i+1} for each of the eight different sampling intervals. In the last column, one missing pulse error for the TDC for each sampling interval is given.

The actual position and velocity-related information that is provided by the FPGA chip are directly fed to the DSP data bus by using three binary words. These data words are isolated from the DSP data bus by three tristate buffers, which are memory mapped to the processor in specific addresses and are selected through the address decoder. Table II indicates the format that was used in acquiring the actual position and the necessary information for the velocity evaluation. The memory-mapped I/O is used in the DSP, and the addresses F0FFFFh–FFFFFFh in the

TABLE I
ADAPTIVE HIGH-PERFORMANCE VELOCITY EVALUATION SYSTEM DETAILS

PMC Count (Decimal)	PMC Count (Binary)	Ovfl	Condition	Decoder & Latch Output	Selected Freq. Div. Output (number m)	Next Sampling Interval $T(i+1)$	One TDC Missing Pulse Error (%)
255	11111111	1	Zero Speed	000	$\div 1$	61.2 ms	$0.0654 \cdot 10^{-5}$
		0	Maximum Measured Period / Minimum Speed $\cong 1$ r.p.m.				
223	11011111	0		001	$\div 2$	53.52 ms	$0.0373 \cdot 10^{-5}$
192	11000000						
191	10111111	0		010	$\div 4$	45.84 ms	$0.0218 \cdot 10^{-5}$
160	10100000						
159	10011111	0		011	$\div 8$	38.16 ms	$0.0131 \cdot 10^{-5}$
128	10000000						
127	•	•	•	•	$\div 16$		
•	•	•	•	•			
64	•	•	•	•	$\div 64$		
63	00111111	0		110	$\div 256$	15.12 ms	$0.01 \cdot 10^{-6}$
32	00100000						
31	00011111	0	Minimum Measured Period / Maximum Speed $\cong 6000$ r.p.m.	111	$\div 1024$	7.44 ms	$0.005 \cdot 10^{-6}$
0	00000000						

TABLE II
FORMAT FOR ACQUIRING THE ACTUAL POSITION AND THE NECESSARY INFORMATION FOR THE VELOCITY EVALUATION

Address	Data Accessed	Buffer														
F0FFFFh	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; text-align: center;">31</td> <td style="width: 10%; text-align: center;">30</td> <td style="width: 10%; text-align: center;">29</td> <td style="width: 10%; text-align: center;">20</td> <td style="width: 10%; text-align: center;">19</td> <td style="width: 10%; text-align: center;">0</td> </tr> <tr> <td colspan="3" style="text-align: center;">Encoder Pulses</td> <td colspan="3" style="text-align: center;">Position</td> </tr> </table>	31	30	29	20	19	0	Encoder Pulses			Position			C		
31	30	29	20	19	0											
Encoder Pulses			Position													
F1FFFFh	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; text-align: center;">31</td> <td style="width: 10%; text-align: center;">8</td> <td style="width: 10%; text-align: center;">6</td> <td style="width: 10%; text-align: center;">5</td> <td style="width: 10%; text-align: center;">0</td> </tr> <tr> <td colspan="3" style="text-align: center;">m</td> <td colspan="2" style="text-align: center;">TDC A</td> </tr> </table>	31	8	6	5	0	m			TDC A		B				
31	8	6	5	0												
m			TDC A													
F2FFFFh	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; text-align: center;">31</td> <td style="width: 10%; text-align: center;">30</td> <td style="width: 10%; text-align: center;">29</td> <td style="width: 10%; text-align: center;">28</td> <td style="width: 10%; text-align: center;">23</td> <td style="width: 10%; text-align: center;">22</td> <td style="width: 10%; text-align: center;">0</td> </tr> <tr> <td colspan="2" style="text-align: center;">ZV</td> <td colspan="2" style="text-align: center;">VS</td> <td colspan="2" style="text-align: center;">TDC B</td> <td colspan="1" style="text-align: center;">Main Time Counter</td> </tr> </table>	31	30	29	28	23	22	0	ZV		VS		TDC B		Main Time Counter	A
31	30	29	28	23	22	0										
ZV		VS		TDC B		Main Time Counter										
Others	Disabled															

memory map block are reserved for addressing I/O peripherals. The DSP is used in the microprocessor mode ($MC/\overline{MP} = L$). The eight most significant bits out of the DSP's 24-bit address bus (A23–A16), plus the R/\overline{W} and \overline{STRB} signals, are fed to a decoder to decode the specific address of the peripheral device. This imposes a limit of $2^4 = 16$ peripheral devices in the particular memory map block, which is enough for this application. Two of the decoder's outputs are used to select input data from MUX B according to Table II. For example, the 20-bit position word and the 10-bit EPC content are selected and fed through the tristate buffer C to the DSP's data bus (D0–D19 for position and D20–D29 for EPC content) when the address that was accessed is F0FFFFh. The 6-bit content of TDC A and 3-bit number m are accessed from the F1FFFFh address. The bits, namely ZV, VS, the 6-bit content of TDC B, and the 23-bit MTC, are accessed from the F2FFFFh address. Finally, if any other address is accessed, the tristate buffers are driven in the high- Z state. The 32-bit tristate buffers are enabled at the rising edge of the DSP R/\overline{W} signal, and the read cycle is synchronized by the \overline{STRB} signal.

The values of N_p , N_A , N_B , N_C , and m are used by the DSP to calculate the velocity by using (5) and, consequently, the linear velocity of the moving platform in our application. Since N_p is the angular position difference and T is the accurate sampling time of each measurement, the velocity is expressed as

$$n = \frac{\Delta\phi}{\Delta T} \tag{3}$$

where

$$\Delta\phi = \frac{2\pi N_p}{p} \tag{4}$$

and p is the number of encoder marks per rotation (or per length unit for linear motion).

The average rotational speed for the adaptive time interval is then calculated by

$$\overline{v} = \frac{2\pi N_p}{mpT} = \frac{2\pi N_p}{mp [N_C T_0 + (N_A - N_B)\tau]} \text{ (in radians per second).} \tag{5}$$

The following equations are used to evaluate the performance:

$$Q_v = \frac{2\pi}{mp [N_c T_0 + (N_A - N_B)(N_A - N_B - 1)\tau]} \quad (6)$$

$$\frac{Q_v}{v} = \frac{2\pi}{mp [N_c T_0 + (N_A - N_B)(N_A - N_B - 1)\tau] v} \quad (7)$$

where v denotes the actual angular (or linear) velocity, Q_v denotes the absolute velocity resolution, and (Q_v/v) denotes the relative velocity resolution.

It is clear, so far, that the measurement or sampling time T changes in relation to the actual measured velocity. With the encoder's very low rotational speed, e.g., less than 1 r/min, the PMC reaches its maximum, and a carry-out signal is considered as a "ZV" condition. This information is stored in a flip-flop (not shown in Fig. 3 for simplicity), which is also reset by the START pulse. This velocity is referred as the "minimum measurable velocity" n_{min} and is defined as

$$n_{min} = \frac{2\pi}{pT_{max}} \quad (\text{in radians per second}). \quad (8)$$

The maximum measurement range can be extended by making several modifications, e.g., using an EPC with more capacity (e.g., 12 bits).

IV. ERROR CONSIDERATIONS

Among the several methods that were used to evaluate speed, a very high potential accuracy of the velocity measurement can be obtained using the M/T-type method. Several methodologies for mathematical analysis have been applied to digital sensors and to other more general class of systems, which involve the digital differentiation of quantized noise-affected signals. These analyses have concentrated on the error that results from the digital nature of the auxiliary counter. The relative error due to this effect is usually approximated at most rates by $1/(Tf_{aux})$, where f_{aux} is the auxiliary counter frequency (e.g., an uncertainty on the order of $1 \times 10^{-4}\%$ is expected if $T = 10$ ms and $f_{aux} = 100$ MHz). The error that results according to the above approximation for each different sampling interval is shown as "one TDC missing pulse error" in the last column in Table I. However, in practice, the actual performance is substantially poorer than indicated in such analyses due to the encoder nonidealities and other sources of error, thus introducing a bias in the measured velocity output. The nonequal distances between pulses results in differential- and integral-type errors that appear as high- and low-frequency errors when viewed over the circumference of the encoder disk. Other sources of error are the nonconstant velocity inside one sampling interval, the bearing nonideality, the influence of torque transmission compliance, etc.

To test the performance of the designed TDC, a similar technique to that used in [12] is followed: A large number of measurements of random time intervals with uniform time distribution were taken under computer control to evaluate the differential and integral nonlinearity. A sample size that is equal to 500 000 was considered to reduce the statistical error. In

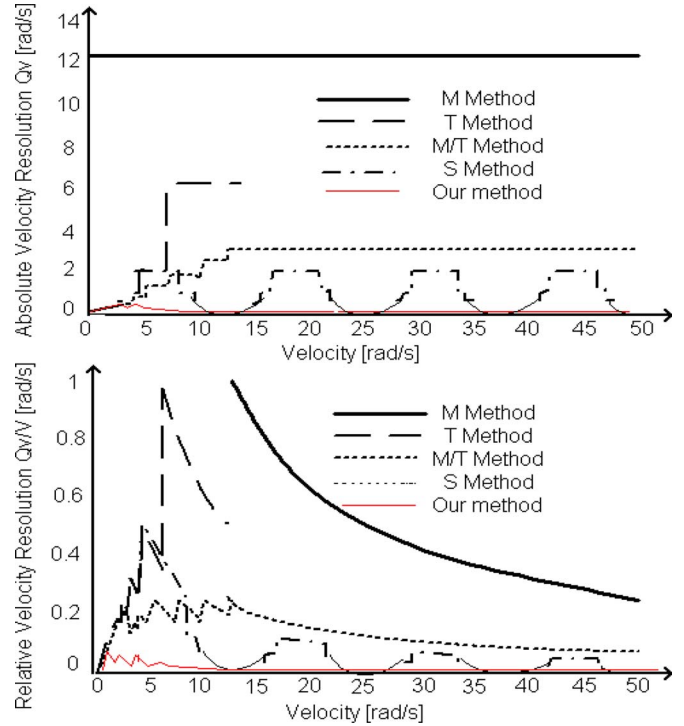


Fig. 4. Comparison of the theoretical resolution of our method with four other methods: 1) the M method (simple encoder pulses measurement in a fixed sampling period); 2) the T method (simple period measurement -division of interpulse angle by the pulse interval time); 3) the M/T method; and 4) the S method [4].

the 10-ns range and with 400 ps per channel resolution, the number of usable channels is 25. The mean channel count is 20 000, and the standard deviation of the count number in the channel is 141.42. The measured nonlinearity was expressed in terms of the system's least significant bit (LSB) or 10 ns per 25 channels = 400 ps. The differential obtained nonlinearity was within -0.5 and $+0.48$ LSB, and the integral nonlinearity range was between -0.2 and $+1.1$ LSB. The temperature also affects the differential and integral nonlinearities of the TDCs, which tend to be extended with the temperature increment.

The performance of the method can be estimated by comparing its resolution to other relevant methods. Fig. 4 gives the measurement resolution for the following five methods:

- 1) the M method (simple encoder pulses measurement in a fixed sampling period);
- 2) the T method (simple period measurement, i.e., division of interpulse angle by the pulse interval time);
- 3) the M/T method;
- 4) the S method described in [4];
- 5) our method.

V. CONCLUSION

In this paper, an improved method for evaluating the velocity of a moving system (rotating or linearly moving) that is based on the information that is provided by a digital encoder has been presented and compared with other similar techniques. A high-performance velocity evaluation algorithm that is based on the use of TDCs and a new adaptive sampling interval method

has been implemented and experimentally tested. The inclusion of the enhanced resolution sampling interval T measurement subsystem allows compensation for the fact that encoder edges asynchronously occur with the MTC pulses or instrumentation system that operates with a fixed sampling interval.

The circuit is a one-chip design and was implemented in an FPGA chip. Comparing the method with other related techniques that were presented shows that a very good "one missing pulse error" can be achieved by using the accurate measurement of the exact sampling interval T . The accuracy that was achieved with this method is far beyond the limits of the inaccuracies/imperfections of rotary encoders that originate in other sources. For example, the maximum relative error in Δp that was caused by nonequidistant encoder pulses for the encoder that was used in the implementation was specified, according to the information from the manufacturer, to be 0.07%.

The system can be used in high-accuracy position control applications for closed-loop motor control that require position and velocity feedback. The characteristics of the adaptive sampling interval method, together with a high-accuracy TDC, has been described, as summarized in Table I, which shows that a very high accuracy in velocity measurement can be achieved using the proposed method from very low rotational velocities (e.g., 1 r/min) to very high speeds (e.g., 6000 r/min). The sampling interval in high speeds is increased, as compared to other methods.

The instability of the clock frequency that was used as well as the truncation error are considered negligible. The performance standards that were used are the relative error approximated by $1/(Tf_{aux})$ and the response time at low and high velocities. An encoder with 1000 pulses per revolution has been used for testing purposes. The method that was presented here has the following advantages as compared to other methods.

- No input pulse synchronization is needed.
- The sampling period is not fixed but variable, resulting in a very good accuracy at medium and high speeds and in a faster response at low speeds.
- Due to its implementation in specific hardware, there is no need to utilize software timers and generate interrupts, which require significant processing time. This can elevate system performance by offloading computationally intensive functions from the DSP processor.
- The acceleration of the moving system that uses the digital encoder can be calculated using the simple differentiation method without accumulated quantization error.

REFERENCES

- [1] M. Prokin, "Extremely wide-range speed measurement using a double buffered method," *IEEE Trans. Ind. Electron.*, vol. 41, no. 5, pp. 550–559, Oct. 1994.
- [2] G. Bucci and C. Landi, "Metrological characterization of a contactless smart thrust and speed sensor for linear induction motor testing," *IEEE Trans. Instrum. Meas.*, vol. 45, no. 2, pp. 493–498, Apr. 1996.
- [3] E. Sisinni, A. Flammini, D. Marioli, and A. Taroni, "A PLD based encoder interface with accurate position and velocity estimation," in *Proc. ISIE*, vol. 2, pp. 606–611.
- [4] T. Tsuji, M. Mizuochi, H. Nishi, and K. Ohnishi, "A velocity measurement method for acceleration control," in *Proc. 32nd Annu. IECON*, Nov. 2005, pp. 1943–1948.

- [5] M. Faccio, P. Grande, F. Parasiliti, R. Petrella, and M. Tursini, "An embedded system for position and speed measurement adopting incremental encoders," in *Conf. Rec. 39th IEEE IAS Annu. Meeting*, Oct. 2004, vol. 2, pp. 1192–1199.
- [6] M. Prokin, "Extremely wide speed range measurement system," *Rev. Sci. Instrum.*, vol. 73, no. 4, pp. 1962–1964, Apr. 2002.
- [7] R. C. Kavanagh and A. Jones, "New low-cost technique for accurate surface velocity measurement of a rotating drum," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 2, pp. 416–422, Apr. 2004.
- [8] R. C. Kavanagh, "An enhanced constant sample-time digital tachometer through oversampling," *Trans. Inst. Meas. Control*, vol. 26, no. 2, pp. 83–98, 2004.
- [9] R. Nutt, "Digital time intervals meter," *Rev. Sci. Instrum.*, vol. 39, pp. 1342–1345, 1968.
- [10] J. Lygouras, K. Lalakos, and P. Tsalides, "High-performance position detection and velocity adaptive measurement for closed-loop position control," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 4, pp. 978–985, Aug. 1998.
- [11] *Very High Speed FPGA's*, QuickLogic Data book, Santa Clara, CA, 1995.
- [12] J. Kalisz, R. Szplet, J. Pasierbinski, and A. Poniecki, "Field-programmable-gate-array-based time-to-digital converter with 200-ps resolution," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 1, pp. 51–55, Feb. 1997.



John N. Lygouras was born in Kozani, Greece, in May 1955. He received the Diploma and Ph.D. degrees (with honors) in electrical engineering from the Democritus University of Thrace, Xanthi, Greece, in 1982 and 1990, respectively.

In 1982, he was a Research Assistant with the Department of Electrical and Computer Engineering, Democritus University of Thrace, where he has been an Associate Professor since 2000. For six months in 1997, he was with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, U.K., as an Honorary Senior Research Fellow. His research interests include analog and digital electronic systems design and implementation, as well as robotic manipulator trajectory planning and the execution and position control of underwater remotely operated vehicles.



Theodore P. Pachidis (S'99–M'04) was born in Drama, Greece, in August 1962. He received the B.S. degree in physics and the M.S. degree in electronics from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1985 and 1989, respectively, and the Ph.D. degree in robotics and machine vision systems from the Democritus University of Thrace, Xanthi, Greece, in 2005.

Since 1989, he has been a Teacher of physics and electronics in Kavala, Greece. From 1996 to 1998, he was a Schoolmaster with the Public Professional Training Institute, Kavala. Since 2005, he has been with the Kavala Institute of Technology. He is currently with the Department of Electrical and Computer Engineering, Democritus University of Thrace. His research interests include electronics, robotics, machine vision systems, and visual C++ and microcontroller programming.



Kostas N. Tarchanidis (S'86–M'93) received the Diploma in electrical engineering from the Democritus University of Thrace, Xanthi, Greece, in 1989 and the Ph.D. degree in robotics from the University of Liverpool, Liverpool, U.K., in 1992.

From 1996 to 1998, he was with Lamda Inc., Xanthi. He is currently a Full Professor with the Department of Petroleum and Natural Gas Technology, Kavala Institute of Technology, Kavala, Greece, where he is also the Director of the Laboratory of Instrumentation and Control and the Dean of the

School of Engineering.

Dr. Tarchanidis is a member of the Technical Chamber of Greece.



Vassilis S. Kodogiannis (M'01) was born in Heraklion, Crete, Greece, in 1966. He received the Diploma degree in electrical engineering from the Democritus University of Thrace, Xanthi, Greece, in 1990, the M.Sc. degree in VLSI systems engineering from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1991, and the Ph.D. degree in electrical engineering from Liverpool University, Liverpool, U.K., in 1994.

From 1996 to 1999, he was with the Technical University of Crete, Chania, Greece, and the University of Ioannina, Ioannina, Greece, as a Visiting Assistant Professor. He is currently a Principal Lecturer with the Harrow School of Computer Science, University of Westminster, London, U.K. His research interests include intelligent systems, image processing, and control.

Dr. Kodogiannis is a member of the International Neural Network Society, the European Society for Fuzzy Logic and Technology, and the Technical Chamber of Greece.