Recibido 26 de octubre 2021. Aceptado 03 de diciembre 2021. Publicado 23 de diciembre 2021.

ISSN: 2448-7775

Reconfigurable Digital FPGA Based Architecture for 2-Dimensional Linear Convolution Applications

MARTIN A. ALARCÓN CARBAJAL[1*], RICARDO PÁEZ VILLA[2], DAVID E. CASTRO PALAZUELOS[1], GUILLERMO J. RUBIO ASTORGA[1].

[1] Postgraduate and Research Department of the National Technological Institute of México Campus Culiacan, Culiacán, Sinaloa, México.
[2] Telecommunications Department of the Center for Research and Advanced Studies of the National Polytechnic Institute Guadalajara Unit.
*Corresponding author: M11170966@itculiacan.edu.mx

ABSTRACT Discrete time linear convolution operation is a fundamental tool in digital signal processing with multiple applications such as audio, filter implementation, image processing, control and renewable energy just to mention a few. This article presents a digital architecture based on Field-Programmable Gate Array (FPGA) for the implementation of linear convolutions of any signal length and binary format in fixed-point arithmetic. The architecture was designed to be reconfigurable and easily programmable in multiple hardware description languages adding flexibility. The article culminates with the implementation of the architecture using Very High-Speed Integrated Circuit Hardware Description Language (VHDL) for a specific audio application.

KEYWORDS— Convolution, Discrete filters, Digital Architecture, FPGA.

I. INTRODUCTION

Due to the increase in the calculation capabilities and cost reduction of microcontrollers (MCUs), digital signals processors (DSP) and microprocessors (MPUs), linear discrete time convolution has played a fundamental role in digital signal processing [1]. Is one of the most frequently used operations in many engineering applications like digital signal filtering [2,3], image processing [4,5], spectral analysis [6], computer vision [7] and acoustics [8], just to mention a few.

The accuracy and precision of the linear convolution algorithm depends entirely on the capabilities of the digital processing platform. Therefore, it is difficult to design a flexible solution with the ability to perform linear convolution operations for different types of applications. Consequently, this article shows a Field Programmable Gate Arrays (FPGA) based digital fixed-point arithmetic architecture whose binary format is easily modifiable to meet the requirements demanded by each application. The architecture design allows the datapath and the control unit to be generated in any hardware description language. In addition, a methodology based on the Signal-to-Quantization-Noise Ratio (SQNR) analysis is presented to select the necessary binary Q-format based on required precision.

This paper is organized as follows: In section II the bases of discrete time convolution are presented as well as the proposed flowchart for the algorithm. Section III shows the design of the datapath and the control unit of the digital architecture. Section IV presents the analysis of the dynamic range and selection of the binary format. Section V shows the implementation of the architecture in FGPA and the methodology to perform the validation experiments. Finally, section VI presents the conclusions.

II. CONVOLUTION ALGORITHM

Convolution is a mathematic operator that establishes a relationship between a system response $y(n)$ to an input signal $x(n)$ in terms of the unit impulse response $h(n)$ [9]. For a discrete-time system, convolution is applicable, and it is expressed in Eq. (1).

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) \times h(n-k), \quad (1)$$

where: $x(n)$ has a length of m samples, $h(n)$ has a length of n samples and $y(n)$ has a length of $n + m - 1$ samples.

A. PROPOSED FLOWCHART OF THE CONVOLUTION ALGORITHM

Based on Eq. (1), a convolution algorithm flowchart is developed and shown in Fig. 1. The algorithm is designed so it can work on any number of samples, once the length of vectors x and h is given. The algorithm uses variable i to check if the length of y has reached its limit value. In

addition, it provisionally stores the product of current indexed samples in variable sum, and adds it to the previous product. As a result, the value of output y in index i is equal to the value of sum once u has reached a value greater or equal to the minimum value between i and m.

III. DIGITAL ARCHITECTURE

The architecture is organized in two sections. A datapath in charge of algebraic operations and a state machine to control the entire process.

A. DATAPATH

The datapath is formed by four registers, one for each variable in the algorithm, as well as a multiplier and an adder. The datapath performs a full-length operation, giving as a result a word length of double the word length of the samples from x and h. The design of the datapath is shown in Fig. 2.

Signals sx, sh and sy correspond to the address of data in registers x, h and y respectively. This registers also admit an output enable (oe) and write enable (ld) signal, being oe the output enable signal for registers x and h, oey the output enable signal for register y, ldx the write enable signal for register x, ldh the write enable signal for register h and ldy the write enable signal for register y.

Register sum also has a write enable and output enable signal, *being ldsum* and *clrsum* respectively, which allows the algorithm to store or erase the accumulated product value of the last value of i.

It is important to note that the configuration of the multiplication block followed by the addition block, allows the synthesis of "multiply and accumulate" blocks (MAC) to accelerate the speed of the algorithm if the FPGA platform supports them.

B. CONTROL UNIT

The control unit consists of a Moore state machine with 5 states. The first is the $Start$ state. In it, all variables are set to an initial value, including the addresses of register x, h and y. If the algorithm has already performed a convolution operation, the architecture notifies with a flag, otherwise it only indicates that registers x and h are ready to receive data, giving access from the outside to signals ldx, ldh, sx and sh. Next, in $Test1$ state, the value of variable i is evaluated and, if it is less than the final length of vector y, the state machine goes to state $Test2$, otherwise it goes to state Done. In addition, in $Test1$, the current value of u is assigned. Control unit of the convolution algorithm is shown in Fig. 3.

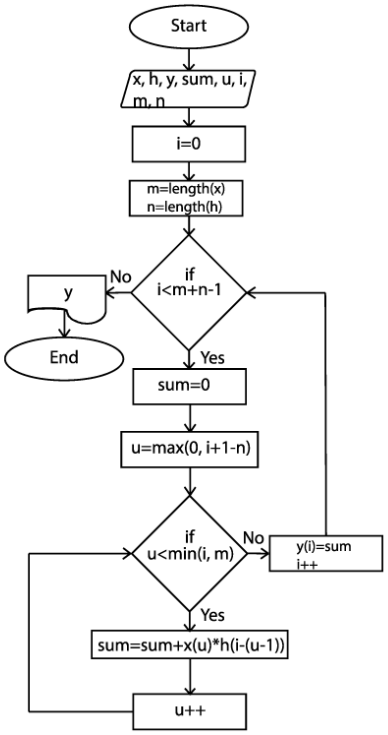


Fig. 1. Flowchart of the convolution algorithm.

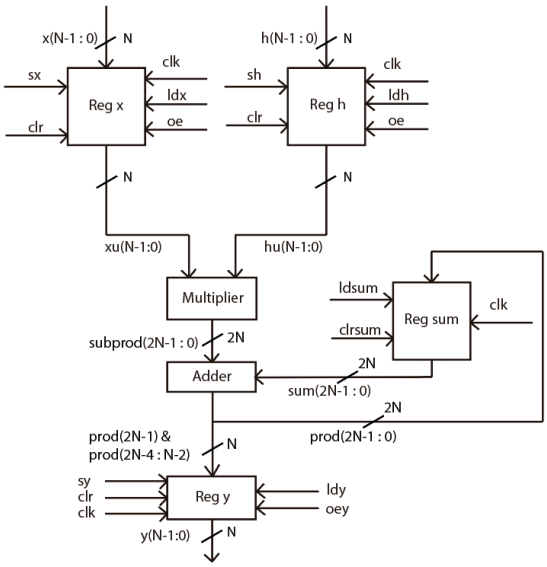


Fig. 2. Datapath of the of digital architecture to perform the linear convolution.

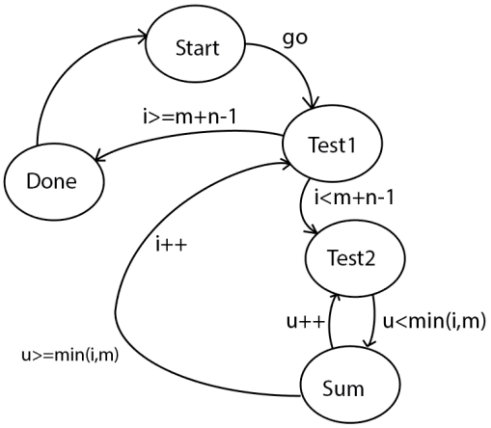


Fig. 3. Control unit of the of digital architecture to perform the linear convolution.

In *Test2* the current values of sh, sx, and sy are assigned, and the algorithm makes sure that u is less or equal to the minimum value between the variable i and the length of vector x. Also, in this state the signal $h(n)$ is reflected and shifted along signal $x(n)$. In the state *sum*, u is incremented in a unit, and if its new value exceeds the minimum between i and the length of vector x, register y is loaded with the actual value of signal *prod*, besides, the value of i is incremented in 1. If the latter condition is not met, register *sum* is loaded. Finally, in *Done* state a flag that indicates the end of the operation is raised.

In order to calculate the number of cycles in the process, Eq. (2) is used, as it is a function of the number of samples involved in the calculus.

$$Cycles = (\sum_{n=1}^{N}(2n+1)) \times 2 + (2N+1) \times (|Ix - Ih| - 1) + 3, \quad (2)$$

where $N = min(Ix, Ih)$, $Ix =$ length of vector x and $Ih =$ length of vector h.

IV. DYNAMIC RANGE AND PRECISION OF THE ARCHITECTURE

A. SPECIFIC AUDIO APPLICATION

A single channel audio filtering application is used to test the algorithm and digital architecture. Using a fixed-point format, the architecture's fixed-point arithmetic behavior is met with the one shown in Table I.

A 41-order low-pass FIR filter is made in MATLAB for it to be signal *h(n)*. For the FIR filter realization, a Kaiser window with a sample frequency of 44100 Hz (standard CD sampling ratio) and a cut frequency of 8000 Hz (voice-frequency transmission channel) is used.

Using the fixed-point toolbox from MATLAB, an estimated value of measured signals shown in Fig. 4 is obtained and it is shown in Table II.

TABLE I. ARCHITECTURE`S FIXED-POINT ARITHMETIC PARAMETERS.

Parameter	Value
Rounding method	Floor
Overflow action	Saturate
Product mode	Full Precision
Product word length	Sample Word Length (x2)
Product fraction length	Sample Fraction Length (x2)
Sum mode	Full Precision
Sum word length	Sample Word Length (x2)
Sum fraction length	Sample Fraction Length (x2)

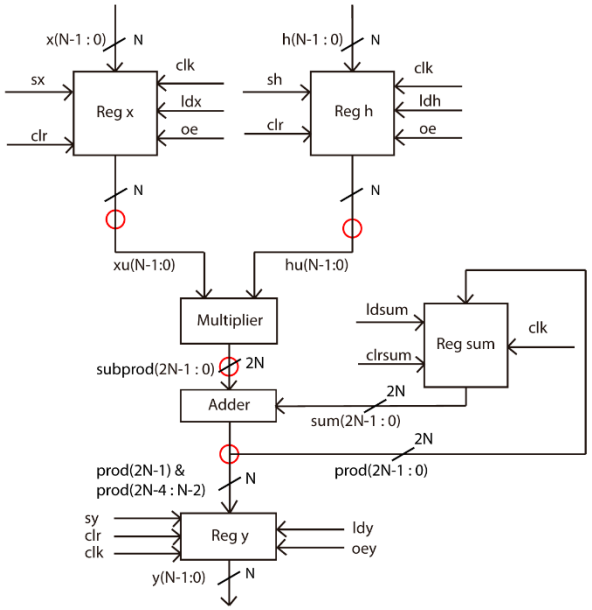


Fig. 4. Measure points (red circles) of the of digital architecture datapath to perform the linear convolution.

TABLE II. ARCHITECTURE`S NUMERICAL DYNAMIC RANGE.

Measured signal	*Minimum*	*Maximum*
Input signal ($\boldsymbol{x}$)	*-1*	0.99996948
Impulse signal ($\boldsymbol{h}$)	-0.07708096	0.40838001
Adder input ($\boldsymbol{subprod}$)	-0.40838001	0.40836755
Adder output ($\boldsymbol{prod}$)	-1.20607794	1.20556727
Convolution output ($\boldsymbol{y}$)	-1.12475930	1.12796640

From the latter, a Q-Format with an integer part QI of 2-bits is needed, and it is calculated with Eq. (3).

$$QI = floor(\log_2(\max(abs[\alpha_{max}, \alpha_{min}]))) + 2, \quad (3)$$

where α_{max} and α_{min} are the maximum and minimum dynamic range value from floating-point variable to represent.

For the fractional part of the Q-Format QF, a number of signal to quantization noise ratio (SQNR) analysis are done in order to select a minimum SQNR value of 60 dB, following Eq. (4)

$$SQNR = 10\log_{10}\frac{\sum_0^{M-1} FP(n)^2}{\sum_0^{M-1}(Fix(n) - FP(n))^2}, \quad (4)$$

where FP is the floating-point 64-bits signal, Fix is the fixed-point signal and M is the number of elements of the vector or matrix. In Fig. 5, the SQNR values for each measured signal in different word lengths are shown, Fig. 6 shows the SQNR values for different number of samples of the output y.

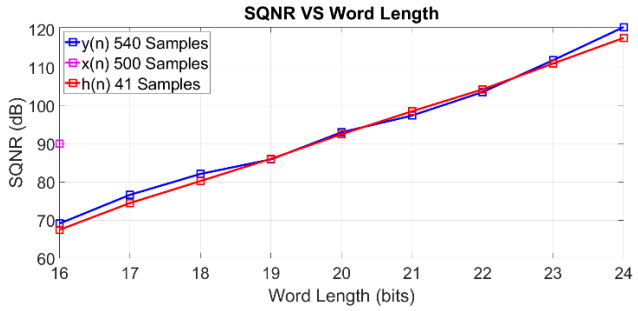


Fig. 5. Relationship between SQNR and word length of all signals involved.

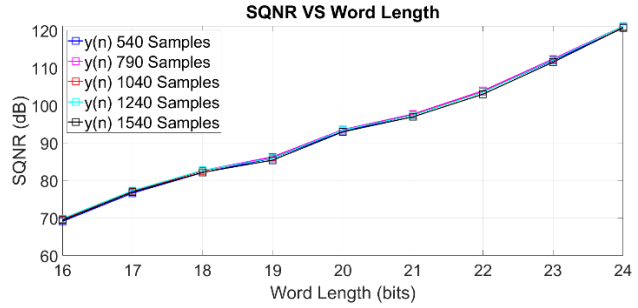


Fig. 6. Relationship between SQNR and number of samples of the output $y(n)$.

From the Fig. 5 and Fig. 6, a Q-Format of Q2.14 is chosen, as it represents the floating-point number with enough accuracy.

B. ERROR CALCULATION

Using 540 samples, a calculus of error is made. In order to estimate the tolerance of error ε_s, Eq. (5) must be met [10]

$$|\varepsilon_a| < \varepsilon_s, \tag{5}$$

where ε_a is the relative error, given in Eq. (7), and ε_s is given by Eq. (8). Additionally, the absolute error E is given by Eq. (6).

$$E = true\ value - aproximate\ value \tag{6}$$

$$\varepsilon_a = \frac{E}{true\ value} \tag{7}$$

$$\varepsilon_s = 5x10^{-t}, \tag{8}$$

where t is the number of significant figures. Using 3 significant figures, the tolerance of error is equal to 0.005. With Eq. (6) and Eq. (7), an average of error with 540 samples is obtained, as shown in Table III.

From Table III, it is shown that a minimum word length of 16-bits with a Q-Format of Q2.14 is required for this application.

V. EXPERIMENTAL RESULTS

A Cyclone V GX 5CGXFC5C6F27C7N FPGA is used to test the digital architecture, which is shown in Fig. 7. Also, a half-duplex UART component is implemented with the architecture to send and receive external data via PC, as seen in the Fig. 8. In it, the *RX* Control reads a character sent from data terminal equipment (DTE) and, if the first data received are the character "*x*" or "*h*", the following data are loaded into register *X* or *H*, also, it changes the memory address of the corresponding register each time a single data are fully loaded. Data are received in two 8 bits parts, first the most significant byte (MSB) and then the least significant byte (LSB), and temporarily stored in dedicated registers before being concatenated and gotten into the architecture, this process is made in the *RX Demux*. In addition, once the execution of convolution is done, data are sent out with signal transfer, from the user, and *TX* Control changes the memory address of register *Y* from beginning to end, each time an element is completely sent, sending first the LSB and then the MSB, both temporarily stored in registers in part *TX Demux*.

TABLE III. NUMERICAL ERRORS DEPENDING ON THE WORD-LENGTH.

Case (540 samples)	Average absolute error (E)	Average relative error (ε_a)	Average percentual error (ε_a %)
Q2.13	3.44E-04	5.80E-03	0.5789
Q2.14	1.87E-04	3.69E-03	0.3699
Q2.15	7.79E-05	1.60E-03	0.1608
Q2.16	4.24E-05	7.93E-04	0.0793
Q2.17	2.75E-05	3.57E-04	0.0357
Q2.18	1.22E-05	1.50E-04	0.015
Q2.19	7.43E-06	6.18E-05	0.0062
Q2.20	3.61E-06	3.88E-05	0.0039
Q2.21	1.38E-06	1.82E-05	0.0018
Q2.22	5.03E-07	9.48E-06	0.000948

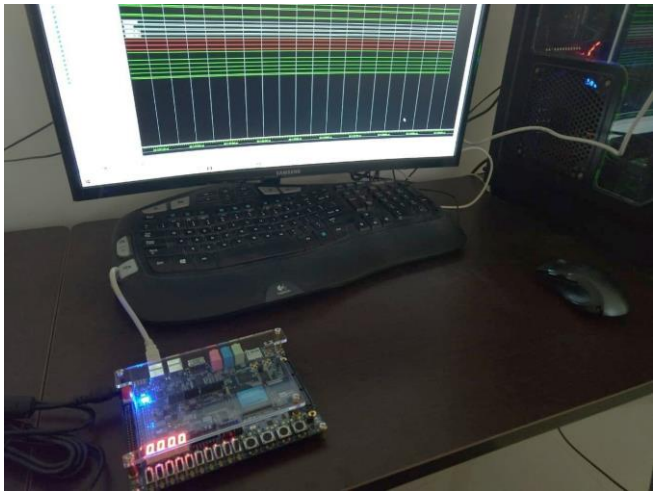


Fig. 7. Cyclone V (GX) FPGA and ModelSim

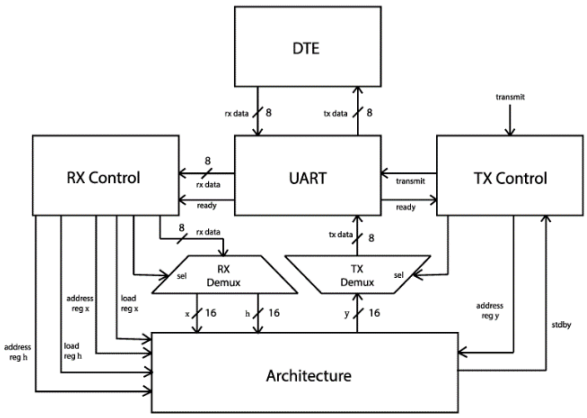


Fig. 8. Architecture with the convolution algorithm and a half-duplex UART.

A. DATA ANALYSIS

A testbench of the architecture is performed using ModelSim introducing 540 samples. The architecture is descripted by VHDL language to the FPGA and the data is sent by means of a PC and the serial protocol shown in Fig. 8. The samples filtered by the FPGA are sent back to the PC by the same protocol to be analyzed. Finally, a comparison is made between the algorithm implemented in MATLAB using 64-bits floating-point arithmetic, the 16-bits with fixed-point arithmetic and its corresponding test bench in ModelSim. This comparison is shown in Fig. 9.

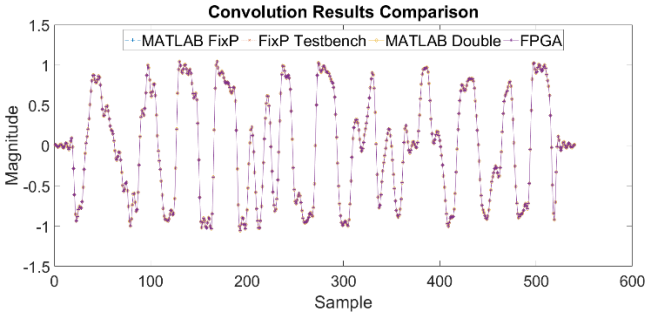


Fig. 9. Comparison between the algorithm implemented in MATLAB (fixed and floating point), testbench (ModelSim) and FPGA.

It is observed in Fig 9. That there are no significant differences between the system implemented in FPGA with 16-bit fixed-point arithmetic and the system implemented in MATLAB with 64-bit floating-point arithmetic. Likewise, it is observed that there is no difference between the result of the FPGA, the testbench and the system implemented in MATLAB with 16-bit fixed point arithmetic. With this the effectiveness of the architecture is verified.

VI. CONCLUSIONS

In this paper, a reconfigurable digital architecture to perform 2-dimensional convolution operations was presented. The architecture is reusable; therefore, it offers flexibility to extend to other algorithms and systems.

The complete system was designed using advanced analysis tools such as MATLAB and ModelSim. It was downloaded to an Intel Cyclone V FPGA using the VHDL hardware description language. The necessary precision analyzes were performed to determine the word length required for a specific audio filtering application. The parameterizable and reconfigurable hardware description allowed the architecture to be modified to meet the minimum error and SQNR requirements through minor changes to the original code. The system was tested through simulations and experiments, demonstrating good performance when is compared to the 64-bit implementation with floating point arithmetic. Because the architecture can resolve any type of two-dimensional convolution, it can be implemented in any type of system or project that requires performing this type of operation effectively like control systems, digital signals processing, image processing etc.

REFERENCES

[1] J. W. Pierre, "A novel method for calculating the convolution sum of two finite length sequences," in *IEEE Transactions on Education*, vol. 39, no. 1, pp. 77-80, Feb. 1996, doi: 10.1109/13.485235.

[2] , M.E. Nordberg, "A fast algorithm for FIR digital filtering with a sum-of-triangles weighting function," in *Circuits Systems and Signal Process,* vol. 15, pp. 145–164, March 1996. https://doi.org/10.1007/BF01183775.

[3] Chao Cheng and K. K. Parhi, "Hardware efficient fast parallel FIR filter structures based on iterated short convolution," in *IEEE Transactions on Circuits and Systems I*: *Regular Papers*, vol. 51, no. 8, pp. 1492-1500, Aug. 2004, doi: 10.1109/TCSI.2004.832784.

[4] Y. Jiao, C. Qian and S. Fei, "Mask Convolution for Filtering on Irregular-Shaped Image," in *17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pp. 115-118, December 2018, doi: 10.1109/DCABES.2018.00039.

[5] C. Lin, C. Liaw and M. Sheu, "Image reconstruction by convolution with piecewise linear polynomal kernel," in *2009 7th International Conference on Information, Communications and Signal Processing (ICICS)*, pp. 1-4, January 2010, doi: 10.1109/ICICS.2009.5397474.

[6] J. Tan, B. Wen, Y. Tian, et al, "Frequency Convolution for Implementing Window Functions in Spectral Analysis," in *Circuits Syst Signal Process (2017)*. https://doi.org/10.1007/s00034-016-0403-7

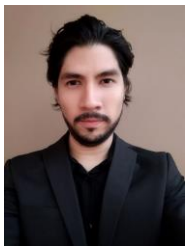
[7] A. Kumar, S. Srivastava, "Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector" in *Procedia Computer Science,* vol 171, pp. 2610-2617, 2020, https://doi.org/10.1016/j.procs.2020.04.283.

[8] V. Peddinti, Y. Wang, D. Povey and S. Khudanpur, "Low Latency Acoustic Modeling Using Temporal Convolution and LSTMs," in *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373-377, March 2018, doi: 10.1109/LSP.2017.2723507.

[9] J. G. Proakis, D. K. Manolakis, "Digital Signal Processing: Principles, Algorithms, and Applications," 4th Edition. Prentice-Hall, New York 2007.

[10] W. Padgett, D Anderson "Fixed-Point Signal Processing Synthesis Lectures on Signal Processing," 1th Edition, Morgan & Claypool, 2009, doi: 10.2200/S00220ED1V01Y200909SPR009.

BIOGRAPHIES



MARTIN A. ALARCÓN-CARBAJAL received the title of Mechatronic Engineer from the Instituto Tecnologico Nacional de México, Campus Culiacán in 2016 the master's degree in sciences in 2019. He is currently in the Ph.D. at the Instituto Tecnologico Nacional de México, Campus Culiacán. His current research lines include digital control, non-linear control algorithms, electronic power systems, renewable energy and digital systems.

RICARDO PAEZ VILLA was born in Culiacán, Sinaloa in 1995, is a graduate student of Mechatronics Engineering by Instituto Tecnológico Nacional de México Campus Culiacán on march 2019. Currently, he's a postgraduate student in CINVESTAV Guadalajara and his line of research is the study of numerical methods and algorithms for VLSI digital design and implementation.

DAVID E. CASTRO-PALAZUELOS was born in Culiacan, Sinaloa, Mexico, in 1977. He received his B.S. degree in Electronic Engineering from the Tecnologico Nacional de Mexico (TecNM)/Campus Culiacan (ITCuliacan), Culiacan, Mexico, in 2000, the M.S. degree in Telematics from Universidad de Colima (UdeC), Colima, Mexico, in 2004, and the Ph.D. degree at the Department of Electrical Engineering, Telecommunications Division at the Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional (CINVESTAV-IPN), Guadalajara, Jalisco, Mexico, in 2015.
From 2005 to 2010, he was a full professor with the Department of Electrical and Electronic Engineering at the TecNM/ITCuliacan. Since 2015, he has been with the TecNM/ITCuliacan, as a teacher and researcher. His research interests are in applications of signal processing to remote sensing with parallel data processing, high performance computing and hardware-software co-design for multimode sensing, communications systems and electronic instrumentation.

GUILLERMO J. RUBIO-ASTORGA was born in Culiacan, Mexico, in 1976. He received the Engineers degree in electrical engineering from Tecnologico Nacional de Mexico, Campus Culiacan, Mexico, in 1998, in 2001 obtained the master's degree in sciences in Electrical Engineering, from the Advanced Studies and Research Center, National Polytechnic Institute, Campus Guadalajara, Mexico, and the degree of Doctor of Science in Electrical Engineering from the Advanced Studies and Research Center, National Polytechnic Institute, Campus Guadalajara, Mexico, in 2013.
He is a Professor with the Electronic-Electric Engineering Department, Tecnologico Nacional de Mexico, Campus Culiacan. His current research interests include the nonlinear control of electric machines.