Short informative title: Towards a new tendency in embedded systems in mechatronics for the engineering curricula

Miguel A. Martínez-Prado PhD¹ | Juvenal Rodríguez-Reséndiz PhD¹
Roberto A. Gómez-Loenzo PhD¹ | Karla A. Camarillo-Gómez PhD²
Gilberto Herrera-Ruiz PhD¹

¹ Universidad Autónoma de Querétaro, Facultad de Ingeniería, Querétaro, México
² Tecnológico Nacional de México en Celaya, Departamento de Ingeniería Mecánica, Querétaro, México

Correspondence
Juvenal Rodríguez-Reséndiz PhD,
Universidad Autónoma de Querétaro,
Facultad de Ingeniería, Querétaro, México.
Email: juvenal@uaq.edu.mx

Funding information
Consejo Nacional de Ciencia y Tecnología

Abstract
This paper presents a robot motion controller for an undergraduate laboratory study program. It is designed to help the students learn and to assess specific learning outcomes proposed by ABET by solving a real-life problem. The main objective of this project is to enable the engineering students to learn some core concepts about embedded systems and motion controllers for robotics by applying them in practice. Also, the proposal shows how to introduce the students to a new tendency in the embedded system market, namely, an All Programmable System on a Chip (SoC). This methodology incorporates interdisciplinary knowledge, technical and professional skills required for pursuing a successful career. In the present study, we surveyed the observations and interests of students towards the learning process, and the results indicate that the inclusion of the robot prototype has a significant impact on providing students with new learning outcomes.

KEYWORDS
learning outcomes, programmable logic, robotics, software development, student project

1 | INTRODUCTION

The Mechatronics is a discipline based on the synergy of control, electronics, computer science, and mechanics. Of course, each of these areas demands a lot of time by itself to master. Linking together this vast amount of knowledge and identifying transversal learning outcomes is one of the main problems educators around the world have been facing for over a couple of decades. Another critical issue is related to teaching digital systems in a quickly changing technological market.

Grimheden and Hanson[11] describe the process of inclusion of mechatronics in the educational offer of several institutions in northern Europe; they identify four questions related to the definition of mechatronics: Identity, Legitimacy, Selection, and Communication. Former research works address thoroughly the first three questions; however, the last question, concerning how to teach mechatronics is still under discussion in present days.

Craig[6] establishes that there must be a balance between two essential skills: first, modeling, analysis and control design for dynamic systems; and second, experimental validation of models and analysis and understanding the critical issues on the hardware implementation of such designs. Craig proposes a set of laboratory practices for a two-semester course. Although Craig covers well the fundamental
aspects of the theory, most of the presented systems are not of the kind of those found in industry.

Analogously, Parkin[19] describes a kit for the teaching of mechatronics which includes several sensors and actuators, a micro-controller training board, motors, and mechanical parts to build a mobile robot. There is no doubt that this hands-on approach has demonstrated to be very successful; however, most of the materials used in this kit are now obsolete or even though some of them are out of production.

On the other hand, Ceccarelli[3] remarks the importance of the use of low-cost robots for teaching and research purposes; he describes the experience at the Cassino University with this approach. Similar results are reported by Rojko et al.[20] who developed and tested a web-based course. Authors do agree on the idea of teaching mechatronics with a hands-on approach, and we also consider that the development of a (low-cost) robot reinforces the knowledge acquired in the classroom for subjects as CAD, mechanical design, control theory, power electronics, programming, robotics, and servo-mechanisms. Moreover, a project as the one developed in this document is related directly with the industry demands.

Chen et al.[5] detail an integral project consisting of the design and implementation of a numerical controller for an X-Y table, which is a mainline project, being two links the theoretical and practical knowledge that students require to become acquainted with the basic theory. There are three subtasks to be carried out as laboratory practices, namely, mechanical design, electronics design, and control and software development. In a similar way, the project proposed in this document relies on the same approach, i.e., one single main project which applies the knowledge that students acquire from other specific courses; however, in contrast with the project described by Chen et al.[5] this new proposal focuses the electronic design on the FPGA technology, because of its remarkable advantages.[15]

Taking into account all the tasks involved in the development of a low-cost robot, the design and implementation of a digital motion controller system is highlighted because the selection of the technology is crucial, i.e., the choice of a micro-controller or reconfigurable logic determines not only the system performance, flexibility, and cost, but also the implementation difficulty degree.[13]

Regarding reconfigurable logic, Aw[2] describes the design of a stepper motor controller and its implementation on a Field Programmable Gate Array (FPGA). This project results to be a good starting point to introduce to the reconfigurable logic for mechatronics courses; however, given its simplicity, this can be considered just as a single laboratory practice rather than a mainline project.

Robotics and the disciplines it involves are subjects that have been evolving for the last few decades. Since its beginnings in the 1960s, robotics has come a long way from being a field mastered by just a few[9] to the point it is a standard part of any mechatronics curriculum.[1]

The motion control approach used to implement robot control systems has moved from classical closed-loop analog controllers[24] to networked advanced digital controllers. [23] This evolution has occurred because of improvements based on the technologies available at a given point in time, especially the digital ones.

Since the 1980s, several approaches in the digital motion control field have arisen. Earlier implementations were based mainly on microprocessors and micro-controllers.[25] Both hardware platforms had severely constrained operation frequencies, a limited number of peripherals, and memory size.

A few years later, digital signal controllers (DSP) replaced these technologies because the DSP attain higher operation frequencies and incorporate dedicated buses and independent hardware modules which facilitate the implementation of algorithms for position and velocity control of servo-motors.[4,10,14,17] In the late 1990s, the reconfigurable logic industry released the first FPGA families which allowed the implementation of motion control algorithms in special-purpose streamlined hardware. Pioneer research works in this area describe the implementation of a quadrature counter for position control of a servo-motor[18] and the implementation of a controller for an inverter based on space vector pulse width modulation (SV-PWM).[26] Nowadays, the new FPGA families provide instances of an ARM Cortex processor into the same silicon device[7] to enable embedded applications to take advantage of both FPGA-based parallel processing and ARM Cortex serial processing. Many digital applications have taken advantage of this sort of technology. Recent research works relate to driver assistance,[12] brain-computer interfaces,[27] and implementation of Artificial Neural Networks (ANN).[8] to mention a few. There are motor control applications that also have migrated to this technology.[16,28]

2 LEARNING OUTCOMES

Of course, these technological changes require a new and different set of skills every time a novel technique raises; this involves updating the academic curricula every few years. At first glance, this may seem like there is a need for a different set of courses with different learning outcomes every time a novel technology becomes mainstream, but a careful instructor must reckon that skills and learning outcomes are not the same. Although courses might be updated every few years to teach the skills needed to adapt to new technologies, a proper learning outcome-based approach needs to recognize the general learning outcomes of an academic program and its
pertinence. Authors believe that learning outcomes, more than a mere formal nuisance, must be the basis of an ongoing modernization of academic programs, especially in the ever-changing engineering curricula.

In other words, while concrete skills are needed to deal with new technologies, general learning outcomes provide a pertinence context, and a direction for any academic program—pertinence may depend on social, economic and even political factors, while context relies more heavily on value creation chains. Without the direction provided by general learning outcomes, an academic engineering program may be trying to “jump on the wagon” of every new technology while paradoxically teaching no concrete and coherent set of practical skills.

In a mechatronics study program, there are different intertwined bodies of knowledge, such as project management,[21] mechanics, control theory, electricity, electronics, and computer science,[22] to name a few, each continually evolving at a different rate. Authors, as instructors in this field, have faced the need to update our course contents regularly—sometimes by trial and error, before adopting a general learning outcome approach for planning both the course and its assessment. Our institution, as most universities nowadays, provides global learning outcomes required for any exit profile, and our engineering faculty complements these with the general learning outcomes specific to each study program. Our mechatronics exit profile requires these general learning outcomes (GLO):

1. Apply and use the knowledge of mathematics, basic science, and engineering to design and carry out research, application, technological, and social innovation projects using specialized methods and techniques.
2. Collaborate with disciplinary and multi-disciplinary teams to formulate and execute projects of automation solutions that are relevant to the context.
3. Design components, systems, and automated processes to meet specific needs and propose suitable solutions.
4. Formulate solutions to problems of automation, components, systems, and processes considering the impact and contributing to the improvement of the global, economic, environmental, and social context using current tools and techniques.
5. Assess and take care of the problems facing today's society, recognizing individual and cultural differences to live responsibly in the social and labor fields based on professional ethics and sticking to the criteria and quality standards to promote sustainable development.
6. Communicate ideas, concepts, and knowledge of engineering in a multicultural context.
7. Upgrade the knowledge on a continuous basis to improve their professional development, adapting to the changing needs of the environment.

Faculty members synthesized these learning outcomes by taking into consideration the ABET guidelines along with our pertinence context. Indeed, “robot motion control” is not listed per se in this list of general learning outcomes. But it is a topic that has been used in several practical applications in the mechatronics field and relates to most of the general learning outcomes above if taught correctly, that is when the instructor carries out a proper assessment; Table 1 outlines our approach.

Authors believe that educational activities are not just a means for learning the topics of an academic curriculum, but also to help the student to become acquainted with up to date technologies—even if those technologies have not become widespread yet. In this case, there are no application notes about the implementation of an entire motion control system implemented on the Zynq-7000 or other similar devices with a real-life application such as the control of a robot.

The aim of the student project outlined in this document is to help students attain a meaningful learning of several topics within the academic curricula by understanding the advantages of the All Programmable System on a Chip (SoC) technology for motion control applications in the context of the real-life application of controlling a three degree-of-freedom (DoF) robotic arm.

### 3 | STUDENT PROJECT DESCRIPTION

The application described in this work is related to the motion control of a three-degree of freedom robotic arm as

| Topic             | Assessment                                                                 | GLO |
|-------------------|---------------------------------------------------------------------------|-----|
| Vocabulary        | Student project report describing antecedents, basic concepts, methodology and including a glossary | 1, 2, 5, 6, 7 |
| System models     | A model of the robot to be controlled                                     | 1, 3, 4 |
| Applied control   | A complete description of the actuators and their drivers                | 1, 3, 4, 6 |
| Closed loop control | PID control design for each robot joint                                    | 1, 3, 4 |
| Velocity profiles | Proper computation of inverse velocity kinematics and the resulting velocity profile for each robot joint | 1, 3, 4 |
the one shown in Figure 1. The proposed motion control system to be assembled by the student comprises several tasks such as the hardware implementation of a digital position controller, the procedures to tune-up such a controller, to obtain the kinematic model and to implement its solution.

The complete system, from a structural point of view, involves dealing with several components, namely:

- **Mechanical system.** The tree DoF robotic arm has a popular architecture used in those manipulators that decouple the positioning and orientation of the end-effector.[1] Since the robotic arm only has three DoF, it only serves the purpose of positioning.

- **Actuators.** Each actuator for the small robot in our laboratory comprises a medium-power 12V brushed DC motor combined with a 171.79:1 metal spur gearbox, and it has an integrated 48 counts per revolution (CPR) quadrature encoder on the motor shaft, which provides 8245.92 counts per revolution.

- **Servo-drivers** (one per motor). A power amplifier stage has been included which is composed of three Texas Instruments LMD18245 power amplifiers required to drive and control the current of the servo-motors. These amplifiers can operate with an analog current reference, and for this reason, a digital to analog converter (DAC) is required, namely, an Analog Devices AD5668 which has eight analog outputs with a resolution of sixteen bits.

- **Motion controller.** The motion controller is the most elaborate component since it includes two subunits, the Programmable Logic (PL) unit, and the Processing System (PS) unit. To control the position of each robot joint the student is required to implement a Servo-Controller module, as the one illustrated in Figure 2, using programmable logic (e.g., VHDL), and then to instantiate it three times within the PL unit. Complementarily, the student also develops a software application to interpolate the points along the desired trajectory and to solve the inverse kinematics; this application runs on the Linux Ubuntu 12.04 operating system (OS) as part of the PS unit.

## 4 | HARDWARE DESCRIPTION

The student implements the digital design of the motion controller within the bundled demo of Xilinx which includes the necessary peripheral drivers to use the Digilent Zybo platform as a standard desktop environment; such demo incorporates a memory interface which is used to interchange data between the PS and PL units. As programmable logic becomes widespread this kind of interfaces are expected to become ubiquitous and, thus, are an essential asset for students.

Each of the Servo-Controller modules instantiated in the PL unit comprises four components, as shown in Figure 2:

A **summing point** to compute the difference between the reference position and the feedback position. The position reference (REF) signal is a 32-bit register whose value can be written directly from the PS unit. Both the REF and POS signals are subtracted and trimmed to 16 bits to avoid saturation during computations. This difference is labeled as ERR and is the input of the digital filter.

A **digital filter** to implement a proportional-integral-derivative (PID) controller. The student must implement a...
standard PID controller in the form of a fourth order digital filter, that is, a filter of the form:

$$u(k) = \sum_{i=0}^{4} q_i e(k - i) + u(k - i).$$  \hspace{1cm} (1)$$

where $u$ and $e$ are the filter output and input error, respectively (see Figure 3). The filter stores the coefficients $q_i$ in a fixed-point format with nine bits for the integer part and the remaining sixteen bits for the fraction; this representation is meant to exploit the $18 \times 25$ embedded multipliers, and therefore the position error signal is extended to 18 bits internally as well. In a similar fashion to the position reference, the filter coefficients are available for writing from the PS unit. The controller output $UOUT$, illustrated in Figure 3, has sixteen bits of width and it is directly related to the needed torque to reach the desired position.

A digital to analog converter (DAC) driver to compute the absolute value of $UOUT$ and then shift this signal to the left to fulfill the whole conversion range of the DAC. The amplifiers require an additional signal for the motion direction labeled as DIR (see Figure 2) which is the most significant bit (MSB) of $UOUT$.

An encoder quadrature interface to measure the motor position. This component requires the quadrature signals CHA and CHB from the incremental encoder, and its output (POS) has 32 bits of width.

These Servo-Controller modules are a core part of the hands-on approach being proposed because it is quite common for textbooks to use non-synthesizable language constructs that the student can only discern once the real code has been written. The summing point helps the student understand the concept of data path while the digital filter enables the student to understand practical concepts such as sampling time, discretization and the $Z$ transform. The development of the PL unit helps the student reaffirm both pure combinational and sequential logic in a relatively simple context since the most complicated part of the control system is stored in the PS unit programmed in software.

5 CONTROLLER TUNING PROCEDURE

The frequency response of a constant-coefficient linear system is the basis for the crossover frequency tuning method. It is well-known in control theory that a causal time-invariant linear system fed with a sinusoidal input always responds with another sinusoidal wave of the same frequency, but with different amplitude and its phase shifted. By measuring these values, it is possible to determine the gain and phase of the plant for a given frequency.

The design requirements involve the crossover frequency $\omega_c$ and the margin phase $\theta_m$ of the system. Frequencies for most of the servo-mechanisms are in the range of 50 and 600 radians per second, while the margin phase is between 30° and 60° degrees, to obtain a well-damped response.

The identification process is carried out by the student by feeding the plant $H(s)$ with a sinusoidal waveform given by

$$u(k) = A \sin(\omega_c k T), \hspace{1cm} k = 0, 1, ..., n$$  \hspace{1cm} (2)$$

where $T$ is the sampling period for the encoder. Thus, the system is expected to respond with a signal wave of the form

$$\theta(k) = B \sin(\omega_c k T + \phi_1)$$  \hspace{1cm} (3)$$

From equations (2) and (3), the student determines the gain as

$$K = \frac{A}{B}$$  \hspace{1cm} (4)$$

while the phase $\phi_1$ can be estimated from the differential time $\Delta t$ between common points of the sinusoidal wave, that is,

$$\phi_1 = \omega_c \Delta t$$  \hspace{1cm} (5)$$

Both $K$ and $\phi_1$ are computed experimentally from equations (4) and (5). In this stage, the student feeds the system with a set of different frequencies like those of Table 3, and logs the gain $K$ and the phase $\phi_1$ of the output with respect with the input; if the student records enough data, an empirical Bode diagram can be attempted using a logarithmic scale, which is quite enlightening.

The student can then start the design procedure with the computation of the plant gain at the desired crossover frequency. In this case, the plant $H(s)$ is comprised by the servo-motor and the amplifier, whence the gain is given by

$$K = \|H(\omega_c)\| = \left\| \frac{K_a K_t}{j \omega_c (|j \omega_c| J_m + K_b)} \right\|$$  \hspace{1cm} (6)$$
where \( K_a \) is the gain of the servo-amplifier, and \( K_T, J_m, \) and \( K_b \) are the torque constant, inertia and viscous friction of the servo-motor, respectively. Besides, the student is also required to compute the phase at the same frequency as

\[
\phi_1 = \arg[H(j\omega_c)] = -\arctan\left(\frac{K_a\omega_c}{J_m\omega_c^2}\right) \tag{7}
\]

Thus, the proportional constant \( K_p \) is given by

\[
K_p = \frac{1}{K_aK} \tag{12}
\]

Finally, the Ziegler-Nichols criterion gives us the integral time constant

\[
T_i = 4T_d \tag{13}
\]

This tuning procedure allows the student to corroborate the principles of control theory in the frequency domain.

### 6 | KINEMATIC MODEL

The motion control performance is measured by following a sequence of points along a circular path in the space and comparing the programmed position with the actual position; therefore, it is necessary to obtain a kinematic model to convert coordinates from joint-space (generalized coordinates) to Cartesian coordinates and vice versa. Given that the robot, illustrated in Figure 4, has three degrees of freedom, only its position can be controlled but not its orientation. By following the Denavit-Hartenberg convention after assigning a \( Z_i \) axis along each robot joint as in Figure 4, the student obtains the structural parameters enlisted in Table 2, where \( a_i \) is the distance from \( Z_i \) to \( Z_{i+1} \), \( b_i \) is the coordinate along \( Z_i \) of the intersection of \( Z_i \) and \( X_{i+1} \), and \( \alpha_i \) is the angle between successive pairs axes, i.e., the angle between \( Z_i \) and \( Z_{i+1} \).

The tool-center-point (TCP) is located at the end \( O_4 \) of the last link, as shown in Figure 4, which is the point of interest.
for the experimental validation of the motion controller. Geometrically, the TCP is the sum of those vectors defined as the displacement $O_iO_{i+1}$ directed from the former to latter, which are given in the $i$-th frame by

$$[O_iO_{i+1}] = \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ b_i \end{bmatrix}$$

(14)

It is necessary to transform all those vectors to coordinates in the first (base) frame to describe their entries in the same coordinate frame. The change of basis matrix from the $(i + 1)$-st frame to the $i$-th frame is given by

$$[Q_i] = \begin{bmatrix} \cos \theta_i & -\cos a_i \sin \theta_i & \sin a_i \sin \theta_i \\ \sin \theta_i & \cos a_i \cos \theta_i & -\sin a_i \cos \theta_i \\ 0 & \sin a_i & \cos a_i \end{bmatrix}$$

(15)

and thus, given the joint angles, the Cartesian position of the TCP of the robot is

$$[O_4]_1 = [O_1O_2]_1 + [Q_1]_1[O_2O_3]_2 + [Q_1]_1[Q_2]_2[O_3O_4]_3$$

(16)

This practical use of linear algebra has proven to be very interesting to students.

After solving for the generalized coordinates $\theta_1$, $\theta_2$, and $\theta_3$ in terms of the Cartesian coordinates of the TCP $[O_4]_1 = (x_p, y_p, z_p)$ there are four possible solutions for every single point in space. The first joint variable $\theta_1$ can be computed from (16) since

$$\sin(\theta_1 + \delta) = \frac{b_3}{\sqrt{x_p^2 + y_p^2}}$$

(17)

and

$$\cos(\theta_1 + \delta) = \pm \sqrt{1 - \sin^2(\theta_1 + \delta)}$$

(18)

where $\delta = \arg(x_p, -y_p)$. Therefore,

$$\theta_1 = \arg(\pm \cos(\theta_1 + \delta), \sin(\theta_1 + \delta)) - \delta$$

(19)
To compute $\theta_3$ both sides of (16) are squared, whence

$$\cos \theta_3 = \frac{x_p^2 + y_p^2 + (z_p - b_1)^2 - a_2^2 - a_3^2 - b_2^2}{2a_2a_3}$$  \hspace{1cm} (20)

and

$$\sin \theta_3 = \pm \sqrt{1 - \cos^2 \theta_3}$$  \hspace{1cm} (21)

Thus, as with $\theta_1$, there are two different solutions for $\theta_3$, namely,

$$\theta_3 = \arg(\cos \theta_3, \pm \sin \theta_3)$$  \hspace{1cm} (22)

Finally, once we have computed a solution for $\theta_1$ and $\theta_3$, from the first two rows of the system in the equation (16), we obtain a unique solution for $\theta_2$ since

$$\begin{bmatrix} \cos \theta_2 \\ \sin \theta_2 \end{bmatrix} = \begin{bmatrix} a_2 + a_3 \cos \theta_3 & a_3 \sin \theta_3 \\ -a_1 \sin \theta_1 & a_1 + a_3 \cos \theta_3 \end{bmatrix}^{-1} \begin{bmatrix} x_p \cos \theta_1 + y_p \sin \theta_1 \\ z_p - b_1 \end{bmatrix}$$  \hspace{1cm} (23)

implies

$$\theta_2 = \arg(\cos \theta_2, \pm \sin \theta_2)$$  \hspace{1cm} (24)

The above equations have been implemented by software within the PS unit of the Zybo Zynq platform. A series of points along a circular path in the Cartesian space is computed and transformed to the joints space in order to serve as set points for the three-axis motion controller. The current position of each servo-motor is recorded to apply the direct

![FIGURE 6](image-url) Joints response of the control system, (a) joint 1, (b) joint 2, and (c) joint 3
kinematics transformation in order to estimate the current Cartesian robot position.

7 | EXPERIMENTAL RESULTS

In summary, the student experimentation setup comprises the Digilent Zybo embedded platform, a couple of printed circuit boards (PCB) for data acquisition and signal conditioning, the power amplifiers, and a three DOF robotic arm manufactured with polyvinyl chloride (PVC) as illustrated in Figure 5.

A sample result of the identification process carried out by a student is shown in Table 3; frequency is given in rad/s and phase lag in degrees. The frequency spectra range only from 100 to 210 rad/s because higher frequencies result in gains

![Graph showing student perceptions of statements about the project]
near zero. It is clear from Table 3 that the gain and phase of the plant depend on the frequency as expected.

With the identification results, and given the crossover frequency and phase margin, the student tuned the PID parameters according to the procedure described above; a sample summary of results is enlisted in Table 4. The crossover frequency for each joint is chosen as the frequency where the gain is around 0.5. This tuning process helps many students realize the way control theory is used in practice.

The circular trajectory considered to evaluate the controller performance has a radius of 50 mm and its center is located at (110.0, 0.0, 70.0), while the Z-coordinate is kept constant during the whole movement, i.e., the entire motion is carried out only in the Z = 70.0 plane. The student interpolates a set of equally-spaced points along the circumference, each point in the set is used to solve the inverse kinematics, and the resulting angle set is converted to triads of encoder counts. Each triad of the encoder counts is written to the set point registers of each servo-controller with a sampling time for the generation of each point of $T_s = 0.001$ s.

Figure 6 shows a sample response for each servo-controller to the generated path logged by a student. It is possible to appreciate that the response of the first and third joints are very close to the reference path; however, the second joint differs significantly from the commanded set points. This could be caused by the influence of non-linear dynamics of the servo-motor and/or the effect of gravity since the control loop does not deal the velocities nor accelerations. It is remarkable that, despite the low-performance response of the second joint, the Cartesian position of the robot remains quite close to the commanded trajectory; this can be observed in Figure 7.

A highlight of the Zynq-7000 device family is its affordable cost for students when compared with solutions which integrate a standard PC, a DSP or a micro-controller with the programmable logic—the achieved cost saving is up to 70%.

Students were polled to gain insight on their opinion of the project; they were shown these six statements:

1. I consider this project to be useful and it boosts my interest in studying mechatronic systems.
2. This project enriches my knowledge about specifics such as programming, robotics, servo-mechanisms, control theory, digital systems, CAD, and power electronics.
3. I consider the hands-on approach of this project to be a significant contribution to the knowledge I acquired in the mechatronics field.
4. The student project has a clear and well-defined methodology.
5. I find this student project both challenging and compelling.
6. The knowledge I acquired during the development of this student project gave me a bigger picture of the kind of problems I will face during my professional career.

Two 10-student groups were asked, on a scale from 0 to 5, how much did they agree with these statements. The average results of the 20 students are shown in Figure 8. After completing the project, students expressed more motivation towards the study program and curiosity in deepening their knowledge of the related topics.

**ACKNOWLEDGMENTS**

Authors would like to thank the support of CONACyT for the funding provided for this research. Trials of this student project were carried out with the kind support of José Luis Avendaño Juárez and Suresh Thenozhi.

**ORCID**

Juvenal Rodríguez-Reséndiz http://orcid.org/0000-0001-8598-5600

**REFERENCES**

1. J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Springer-Verlag, New York, Inc., 2002, 2.
2. K. C. Aw, S. Q. Xie, and E. Haemmerle, A FPGA-based rapid prototyping approach for teaching of Mechatronics Engineering, Mechatronics. 17 (2007), 457–461.
3. M. Ceccarelli, *Robotic teachers’ assistants*, IEEE. Robot. Autom. Mag. 10 (2003), 37–45.
4. T. N. Chang, B. Cheng, and P. Sriwilaiaajaroen, *Motion control firmware for high-speed robotic systems*, IEEE. Trans. Ind. Electron. 53 (2006), 1713–1722.
5. D. Chen et al., A mechatronics control engineering class at Beihang University, China: Practicing and exploring, Mechatronics. 22 (2012), 881–889.
6. K. Craig, Is anything really new in mechatronics education? IEEE Robot. Autom. Mag. 8 (2001), 12–19.
7. L. H. Crockett et al., The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc. Strathclyde Academic Media, 2014.
8. P. Dondon et al., Implementation of a feed-forward Artificial Neural Network in VHDL on FPGA. In: Neural Network Applications in Electrical Engineering (NEUREL), 2014 12th Symposium On. IEEE, 2014, pp. 37–40.
9. J. F. Engelberger, Robotics in Practice: Management and Applications of Industrial Robots. Springer Science & Business Media, 2012.
10. G. Ferretti et al., The structured design of an industrial robot controller, Control. Eng. Pract. 4 (1996), 239–249.
11. M. Grimheden and M. Hanson, Mechatronics—the evolution of an academic discipline in engineering education, Mechatronics. 15 (2005), 179–192.
12. Y. Han and E. Oruklu, Real-time traffic sign recognition based on Zynq FPGA and ARM SoCs. In: 2014 IEEE International Conference on Electro/Information Technology (EIT). 2014, pp. 373–376.
13. A. Malinowski and H. Yu, Comparison of embedded system design for industrial applications, IEEE. Trans. Ind. informatics. 7 (2011), 244–254.
14. M. K. Masten and I. Panahi, Digital signal processors for modern control systems, Control Eng. Pract. 5 (1997), 449–458.
15. E. Monmasson et al., FPGAs in industrial control applications, IEEE. Trans. Ind. informatics. 7 (2011), 224–243.
16. Y. S. Moon et al., Hardware Implementation of Motor Controller Based on Zynq EPP (Extensible Processing Platform), J. Korea Inst. Electron. Commun. Sci. 8 (2013), 1707–1712. http://www., readcube., com/articles/10.13067%2F2FIKIECS. 2013.8.11.1707.
17. K. Nilsson and R. Johansson, Integrated architecture for industrial robot programming and control, Rob. Auton. Syst. 29 (1999), 205–226.
18. A. K. Oudjida et al., A reconfigurable counter controller for digital motion control applications, Microelectronics J. 28 (1997), 683–690 http://www.sciencedirect.com/science/article/pii/S0026269297000098.
19. R. M. Parkin, The mechatronics workbench, Eng. Sci. Educ. J. 11 (2002), 36–40.
20. A. Rojko, D. Hercog, and K. Jezernik, E-training in mechatronics using innovative remote laboratory, Math. Comput. Simul. 82 (2011), 508–516.
21. K. H. Rose, A guide to the project management body of knowledge (PMBOK® guide)-Fifth edition, Proj. Manag. J. (2013), https://doi.org/10.1002/pmj.21345.
22. M. Sahami (IEEE) and S. Roach (ACM), Computer Science Curricula 2013. ACM, New York, NY, USA, 2013, https://doi.org/10.1145/2534860.
23. M. Schwager, J. McLurkin, and D. Rus, Distributed Coverage Control with Sensory Feedback for Networked Robots. In: Robotics: Science and Systems. Massachusetts Institute of Technology Computer Science and Artificial Intelligence Lab, Cambridge, MA, USA, 2006, pp. 49–56.
24. B. Siciliano, A closed-loop inverse kinematic scheme for on-line joint-based robot control, Robotica. 8 (1990), 231–243.
25. T. M. Sobh et al., Concurrent design of a three-link manipulator prototype, Comput. Electr. Eng. 27 (2001), 445–458.
26. Y. Y. Tzou and H. J. Hsu, FPGA realization of space-vector PWM control IC for three-phase PWM inverters, IEEE. Trans. Power. Electron. 12 (1997), 953–963.
27. M. Won et al., Low-power hardware implementation of movement decoding for brain computer interface with reduced-resolution discrete cosine transform. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2014, pp. 1626–1629.
28. H. Yun et al., Implementation of motor controller using Zynq EPP. In: 15th International Symposium on Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on an Advanced Intelligent Systems. 2014, pp. 1224–1228.
control theory, digital systems, robotics and computation theory.

K. A. CAMARILLO-GÓMEZ is a professor and head on Research Projects of the Department of Mechanical Engineering of the Tecnologico Nacional de Mexico in Celaya. After finishing his PhD studies at Tecnologico Nacional de Mexico en la Laguna, she joined at the same institution in Celaya as professor in 2009. She has received several fellowships and awards on robotics by the Mexican Association on Robotics and Industry (AMRob) and the Federation International of Robo-sports Association (FIRA). She is a member of the AMRob and Co-chair of HuroCup of FIRA. Her current research field is focuses on modeling and control of robots, control of nonlinear systems, stability analysis of nonlinear systems, development of rehabilitation systems, humanoid robots, as well as assembly automatization and applications of vision control on industrial processes.

G. HERRERA-RUIZ received his BSc degree in electronics and master in electrical engineering both a ITESM Mexico, and PhD degree in mechanical engineering from the Technical University of Budapest Hungary since 1997. He is currently the Rector of the Autonomous University of Queretaro and he is National Researcher level three with the Mexican Council of Science and Technology, CONACYT.

How to cite this article: Martínez-Prado MA, Rodríguez-Reséndiz J, Gómez-Loenzo RA, Camarillo-Gómez KA, Herrera-Ruiz G. Short informative title: Towards a new tendency in embedded systems in mechatronics for the engineering curricula. Comput Appl Eng Educ. 2019;27:603–614. https://doi.org/10.1002/cae.22101