IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 16, 2019, accepted December 31, 2019, date of publication January 10, 2020, date of current version January 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2965639

Easy and Secure Handling of Sensors and Actuators as Cloud-Based Service

REYES SÁNCHEZ-HERRERA, MARCO A. MÁRQUEZ, AND JOSÉ M. ANDÚJAR, (Senior Member, IEEE)
Huelva University, Escuela Técnica Superior de Ingeniería, 21007 Huelva, Spain
Corresponding author: Reyes Sánchez-Herrera (reyes.sanchez@die.uhu.es)

ABSTRACT The internet of things is rapidly becoming a fact of life. There remain, however, various challenges with respect to the concept of Industry 4.0 before this new reality is fully integrated into the industrial sector. One such challenge is ensuring security, an essential issue in terms of both management and operational technology involving the configuration and administration of physical equipment. Another is integrating equipment from different manufacturers which must be configured and/or administered with incompatible protocols. Likewise, remote access to systems must be carried out from a browser in the web frame. This paper presents a set of tools, available as a cloud-based service, which facilitates access to the physical equipment located in a business. The access enables the end user to see and/or modify the configuration and management of data in a secure, controlled, organized and collaborative manner. In order to unify different protocols, the service makes use of the Modbus TCP/IP (Transmission Control Protocol/Internet Protocol) protocol, one of the most commonly used by industry in local networks. The local Modbus communications are encapsulated in the Websocket protocol to make them accessible from the cloud. The tools proposed in this paper are based on open hardware platforms and free software, and they permit local and remote sensors and actuators to be fully and easily accessed. Hence, their chief advantage is that they allow access from any internet connection, via a browser, to the physical equipment located in an industry by means of user profiles in accordance with the security level set by each individual company.

INDEX TERMS Modbus, remote access, open source hardware/software, EJS/EJSS, data acquisition, Internet-of-Things, Industrial Internet of Things, cloud, Industry 4.0.

I. INTRODUCTION

Companies are now developing digital strategies which enable them to incorporate all the advantages and possibilities of the digital world into their production lines. The information generated by these advances will be digitally accessible from computers and other networked devices, the analysis and evaluation of which opens up a wide range of possibilities. This information can be considered a transformable raw material that can be given added value. The possibilities of digitalization are greatest in the industrial field, in which advanced digital techniques drive increases in productivity [1], such as the introduction of augmented reality in the assembly and maintenance of facilities, and the use of autonomous robots in production lines. If, in addition, all a company's systems are interconnected and accessible, then the company has achieved full digitalization and connectivity and can be described as an Industry 4.0 company.

The cornerstones of Industry 4.0 are the advances in digital and communications technologies. In the new industrial paradigm everything is interconnected. In addition, advances facilitated by the internet are tending to replace limited, local visions of industry with more global and distributed ones. This new vision enables total connectivity without the need for resources to be physically adjacent.

The degree of connectivity sought by the Internet of Things (IoT) or the Internet of Everything (IoE) [2, p.], [3], [4] already exists in the personal and social spheres. We live in a society permanently connected to the network, with an ever growing number of connected devices. The use of smart phones, for example, is increasingly widespread, and almost half the population have profiles on social networks. All these devices and applications generate data which we are largely unaware of. Nor do we control the applications installed on our devices for managing these data, as they exact our

The associate editor coordinating the review of this manuscript and approving it for publication was Theofanis P. Raptis.

mandatory consent which cannot be subsequently retracted. In addition, our online activity, like the devices, is constantly registered on an unknown server. In short, we have little control over our information and our devices. Moreover, they are vulnerable to those who know to obtain it and have the will to do so, without mentioning the consequences of disconnecting the server. In the industrial sphere, where information management and plant operation coexist, security and privacy are absolute essentials. In such a context, loss of information and control of devices is unthinkable, and consequently the adoption of IoT within this sector, referred to as the Industrial IoT (IIoT) [5], [6] represents a major challenge to industry.

Nevertheless, total connectivity has advantages for the industry that cannot be ignored. Among these is a much more open and immediate communications model [7] for all types of information exchange, whether between devices (M2M, Machine to Machine) [8], or between devices and people (M2P, Machine to People). The integration of management and production levels within industry means facilitating the exchange of information between the two. According to the current tendency to integrate the management information into the content management systems (CMS), cloud technologies have a very important role to play [9]. These technologies can be managed by the company itself or provided by third partly entities (IaaS, Infrastructure as a Service) that guarantee information security and privacy.

The adoption of Industry 4.0 by an industrial company involves the digitalisation and integration of all its communication processes. Moreover, to use CMS, that integration must be carried out into the cloud. Digitalisation makes it necessary to use very different communications technologies to those currently used in the industry. The limitations of local operation through local area network (LAN) can be overcome with SCADA (Supervisory Control and Data Acquisition) systems, which use different protocols, such as Fieldbus, Profinet [10] to carry out the communication between equipment. One of the most commonly used in heterogeneous scenarios (involving devices from different manufacturers) is Modbus TCP/IP [11], which has the following advantages in comparison with similar protocols:

- It is an open protocol, used locally (without internet access) across a wide range of industrial scenarios.
- It is interoperable with a large number of independent devices supporting the protocol, for example, frequency inverters, PLCs (programable logical control), I/O devices, etc.
- It can act as a link between devices and industrial networks running different protocols specific to different manufacturers (proprietary software).
- It is widely used on open hardware platforms, such as SBCs (Single Board Computers) and MCUs (MicroController Units).

These design features make Modbus ideal for situations involving a multiplicity of devices and software, linked together in LANs over the TCP/IP protocol.

It is possible for companies to access their data and devices remotely through the use of cloud-based services via a standard internet connection and a browser. However, several technical challenges need to be resolved. This paper presents solutions to the connectivity issues associated with the LAN and – a more complex challenge – the internet. In section 3.1, it also deals with connectivity via browser, the use of which introduces an additional layer of complications.

Against this background, this paper presents a set of tools for accessing all the devices on a company's industrial network by means of Modbus. The devices are accessed via the cloud and allow data queries to be carried out and devices to be configured and/or administrated in a secure, controlled, organized and collaborative environment. The specific nature of these design features is as follows: security is ensured by data encryption and the use of user profiles for access; control refers to the ability to apply different filters and user criteria, such as the time zone; the tools are organized in the sense that they allow access to be sequenced through, for example, a reservation system; and finally, they are collaborative in that they allow concurrent access to the same resource. In short, the tools permit access to an industrial network from any internet connection by means of a browser with a user profile enabled and the required level of security specified by each company, and as such they fulfil all system requirements facing companies for integration into Industry 4.0 and into CMS.

The outline of the remainder of the paper is as follows. Section 2 provides a review of the technical literature. Section 3 gives a detailed description of the set of tools proposed; it is divided into two subsections, the first dealing with the restrictions imposed by the browsers, as mentioned above, and the second, the technical aspects. Section 4 covers the aspects relating to integration in the cloud, while Section 5 presents a case study. It is divided in two subsections, the first to describe the case and the second to present the obtained results. Finally, in Section 6, some conclusions are drawn.

II. STATE OF THE ART

The communications scenario described in the introduction must integrate the traditional communications model currently used in industrial frameworks, denominated the computer integrated manufacturing (CIM) pyramid, with the new technologies emerging from cloud computing. The highest levels in CIM architecture correspond to information technology (IT), and are traditionally related to the commercial management, whilst the lowest levels correspond to the operational technology (OT) and are directly related to the physical devices involved in the manufacturing process. In the majority of companies, CIM remains embedded in local communications.

In addition, in many cases, unlike the IT applications, the OT devices are not interconnected via a LAN, and instead use proprietary software oriented towards closed systems. Integration of the IT and OT layers would allow the communication between them, making production more

efficient and improving profits [12]. IT/OT convergence enables direct control of the complete monitoring system in a company through automation and integration of communications systems and industrial networks. Without this convergence, a certain degree of operative inefficiency is inevitable and the work flow cannot be optimized.

However, this convergence is not easy to achieve. Commercial platforms exist, but they are rarely compatible with other applications within the same company because each manufacturer provides its own products and solutions, [13]–[17]. One proposal that is becoming increasingly popular is that of open solutions [18]. In this regard, Botta *et al.* [19] promote research efforts to define standard protocols, libraries, languages, and methodologies to develop the full potential of IoT. An alternative to commercial packages can be found in open hardware and free software platforms, which not only offer more control over the system and data, but also allow developers to work at lower levels, and, more importantly, do not require configuration data to be sent to external databases, [20].

As a result, low-cost platforms, such as Arduino [21], Phidget [22] or Raspberry Pi [23], are becoming increasingly popular. They display characteristics analogous to more expensive proprietary devices, and are highly suitable for monitoring and controlling systems via the internet, such as real-time monitoring of fuel cells using LabVIEW [24], and Arduino [25], [26], accessing applications in solar energy facilities [27], [28], developing the IoT [29], [30], remotely operating robotic applications [31], [32] and transmitting data [33], [34, p.], among others.

There are two procedural options for achieving the kind of convergence illustrated by the examples above of devices and industrial networks across different platforms:

1. The substitution of the protocols used up to now by others compatible with access from the cloud, such as HTTP (HyperText Transfer Protocol), Websocket and SSE (Server Sent Event). A major disadvantage of this option is that it discards the protocols currently used by the majority of devices in operation in manufacturing and other fields.
2. The design of an appropriate runways to make it possible to access the protocols currently used in the industry from the cloud.

This paper proposes a solution of the second type, using Modbus TCP/IP as the ideal protocol to unify heterogeneous industrial networks. However, the availability of the unified platform is not the only requirement for the development of remote monitoring and control systems. The integration of the relevant hardware/software into a communications network is also required. Remote systems can be classified as open or closed according to their degree of heterogeneity and accessibility, [35]. Hence, if the devices involved in the remote system are from a single manufacturer or compatible software, and are all connected to the same LAN [36], the remote system will be closed or proprietary. It becomes a little more open if – assuming the devices remain mutually compatible - it is accessible via the Internet [37], [38]. Finally, if the system involves mutually incompatible devices or devices from different manufacturers, and is also accessible via the internet, it can be considered fully open.

Some fully open systems can be found in the literature, but not many. Furthermore, those that are discussed tend to be local ad hoc solutions [39]–[41], with the result that the technology is non-transferable from one system to another. For example, in [42] a new flexible framework based on social instant messaging (IM) application architecture is proposed for integrating an offline remote experiment into a communication node via a unique identifier. However, the system is only valid for plants controlled with LabView.

In the approach outlined in this paper, the problem of heterogeneity is solved by the use of the unification protocol TCP/IP, while accessibility is achieved through integrating the systems involved with Modbus, as addressed in a paper by the same authors [43] in which a set of tools were presented for accessing a set of devices unified by Modbus over the internet. This solution is easy to implement over open platforms. In [43], easy java simulations (EJS)[44] is proposed as a suitable medium for implementing the application which accesses the physical devices (henceforth the user interface, UI). EJS is an open-source authoring tool created by Francisco Esquembre for building discrete simulations. An important feature of EJS is that it enables the development of the UI. However, it does not incorporate the tools necessary for connecting hardware. To overcome this problem, the authors [43] present a set of software tools, called elements, which are integrated within EJS, and provide a simple way of connecting physical devices across the network (through the use of controllers). Finally, they materialize the unification of communications through the corresponding EJS elements (software elements), which hide the details of the communications technology so as to make it easier to develop applications when the designer has little knowledge of network-based applications.

In short, the procedure proposed in [43] makes it possible to access accessible devices unified by Modbus TCP/IP over the internet. Establishing a connection between any internet-enabled PC (or SBC) and the physical system in question requires an application programming interface (API). However, APIs are not supported by the new generation of browsers, so in order to make the proposed software resident in the cloud, further software is needed. This new software has been developed by the authors and is the subject of this paper.

III. DESIGN OF THE USER INTERFACE IN THE FRAMEWORK OF THE NEW GENERATION OF BROWSERS

Cloud technology is becoming the most used to integrated IT in the internet. One of the causes is the fact that a cloud-based service does not require any specific application to be installed on the end user PC in order to carry out its function. However, access to the service is necessarily carried out from

a browser. This means that, for the integration of IT and OT layers of the CIM, they both must operate within the execution framework of the browser, a consequence which imposes limitations on the interaction with the local operating system and the possibilities of remote communications. Such limitations include, for example, the capacity of the local operating system to store data, execute an application or initiate a socket accessible from the operating system.

With this in mind, subsection 3.1 details limitations imposed by the new generation of browsers, while subsection 3.2 presents the set of tools which is the main focus of the paper: a cloud-based, browser-enabled system for accessing heterogeneous industrial networks unified under Modbus TCP/IP.

A. CLOUD ACCESS BY THE USER

For security reasons, current browsers do not allow the execution of local applications, or connections between themselves and the operating system of the machine on which they are running. This places a limitation on remote access, so it is necessary to use HTTP or supported protocols over HTTP (such as Websocket or SSE) to access networked devices (both internet and local network). However, communication between the different devices, systems and plants in a company have to date generally been carried without taking into account the question of remote accessibility regarding OT (as this has only recently become viable). Consequently, the applications used in the design of such elements do not support integration with web technology and are often not easily adapted.

Until recently, the most common means of integrating web technology with functions such as the execution of applications on the operating system, access to files and remote access was through Java in the form of an API. Nevertheless, as mentioned above, this kind of integration is not longer allowed by the new browsers, as they do not allow the running of Java applets to reproduce the client-server architecture.

One means available to application programmers for retaining the capacity to connect the browser with the industrial network is a JavaScript engine which allows them to create connections based on HTTP. This is the procedure adopted in this paper for the integration of the application with the browser, in cases where the controller has been developed within open platforms.

The Modbus master and slave elements presented in [42] unify heterogeneous industrial networks and make them accessible from the internet by means of applets generated in EJS. However, those elements do not enable the system to be stored in, and accessed from, the cloud. The following subsection presents various easy Java and JavaScript simulations (EJSS) [44] elements designed to overcome this limitation, encapsulating the Modbus protocol on top of HTTP-based protocols.

B. PROPOSED EJJS ELEMENTS

EJSS is a new version of EJS to implement JavaScript interfaces. Four elements within the EJSS framework have been created to encapsulate the Modbus protocol on top of those based on HTTP. They are jointly denominated OverWebsocket (OW), and complement the elements presented in [43] and denominated Modbus (MB). Their operation is as follows: there are two sets of OW elements, the MasterOverWebsocket (MOW) and the SlaveOverWebsocket (SOW). MOW is composed of ClientMasterOverWebsocket (CMOW) and ServerMasterOverWebsocket (SMOW). They open communication links in HTTP protocols between the industrial network control application (INCA), developed in EJS, and the UI, developed in EJSS and executed on the end user PC. There are two SOW elements, the ClientSlaveOverWebsocket (CSOW) and ServerSlaveOverWebsocket (SSOW), whose function is the same as that of the MOW's. The use of the MOW or SOW elements is determined by the configuration of the MB elements in the industrial network.

Hence, on the one hand, if the element included in the INCA is the MasterModbus (MMB), the MOW elements will be used to encapsulate the Modbus communications on top of HTTP. Conversely, if the element in the INCA is the SlaveModbus (SMB), SOW will be used. On the other hand, the corresponding server element (SMOW or SSOW) must be located in the INCA, which has a valid and fixed IP. And the corresponding client element (CMOW or CSOW) must be located in the UI, which runs on the end user PC with any IP.

Among the HTTP protocols available, the proposed OW elements use Websocket to establish the communications links. In addition, the information is encapsulated in JSON (JavaScript Object Notation) as this is the most suitable for use in JavaScript. Like the MB elements, the OW also generate an API. However, the new one does allow write / read calls to be made from the browser to the Modbus registers that configure and manage the physical devices of the industrial network.

In the same way as the MB elements, OW only requires inclusion in the UI and the INCA. This can be done by selecting them with the mouse from the window "Elements for the model" and dropping them into the "List of elements" shown in figures 1 and 2. Figure 1 presents that operation in EJSS. As indicated above, CMOW and CSOW are available (both of which can be included in the UI). In the case shown in figure 1 it is CMOW which has been included, which also corresponds to the case presented in figure 2. Figure 2 presents the EJS framework and the INCA. Here, the MB element is necessarily MMB, and consequently SMOW must also be included. SMOW requires the following methods for its configuration:

```
SMOW.setModbus(MMB)
SMOW.port(8000)
```

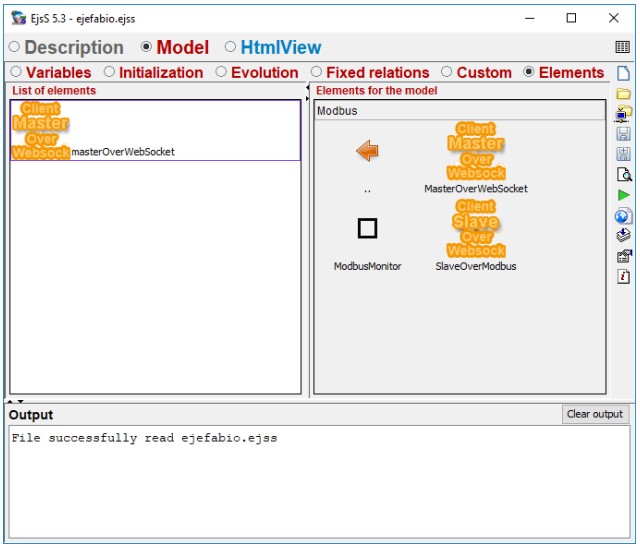


FIGURE 1. The EJSS application with available client elements. These appear on the right. On the left, CMOW has been selected and is included in the UI.

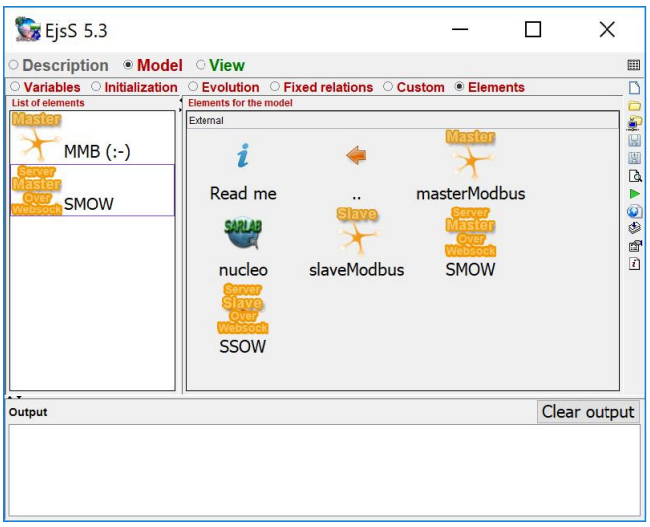


FIGURE 2. The EJS application with MB and OW elements. In this case, MMB has been required in the INCA and hence also SMOW. This case is the corresponds to the UI illustrated in figure 1.

The first of these connects the MB and OW elements. The second indicates the communications port between SMOW and CMOW in the UI, which in the case is the 8000. Hence the following method must be included in figure 1: masterOverWebsocket.port(8000).

In all the methods described above, the word occurring just before the period is precisely the name of the corresponding element, as it appears on the EJS / EJSS element screen, differentiating between upper and lower case letters. Thus, in the figure 2 methods, SMOW occurs just before the period, as does masterOverWebsocket in figure 1. Any name of the element can be used and is chosen by the programmer when are dropped in the ''List of elements'' panel.

If the required element in the INCA is SMB, then the OW for inclusion would be SSOW in the INCA and CSOW

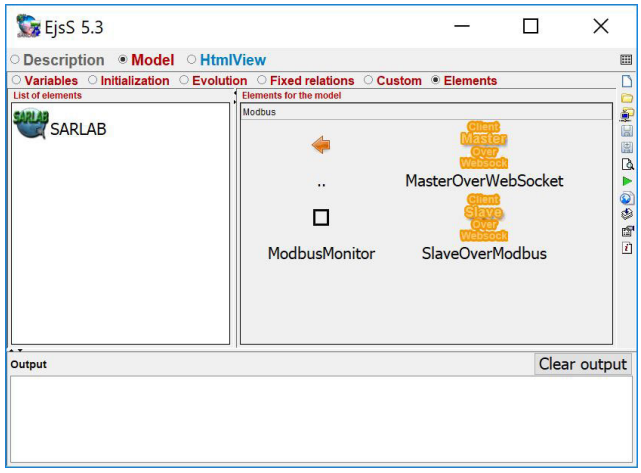


FIGURE 3. FIGURE 3. EJSS application with the OW elements available on the right. On the left, the SARLAB element has already been included in the UI.

in the UI. The methods for their configuration would be the same.

IV. INTEGRATION AS A CLOUD SERVICE

The high degree of data availability and expandability in the cloud make this service an excellent way to increase the efficiency of IoT systems. Data storage and analysis can be done in the cloud instead of on IoT devices, and hence this is where all the services described in this paper are located, making data and resources permanently available for all devices connected to the LAN.

To this end, the final step of the procedure proposed in this paper is the publication of the access to the controllers in the cloud in an organized, controlled and transparent way, irrespective of the communication protocol used.

In order to achieve this, all the physical devices in the industrial network are controlled by means of the Modbus protocol, whether the platform is open or commercial. In addition, as indicated above, the HTTP protocol used is Websocket.

The element designed to implement the communication links is SARLAB (an acronym derived from its Spanish name, Sistema de Acceso Remoto a LABoratorios, or remote laboratory access system in English), as illustrated in figure 3, and described in [43]. Regarding communication architecture, SARLAB uses a middleware layer to make the communication process between the user computer and the controller connected to the industrial network [45]–[47] transparent. Middleware works as a software distributed abstraction layer, and is located between the application and the lower layers (operating system and network layer). The software developed in the middleware layer provides an API which hides the complexity of the general communication problem.

In this way, SARLAB creates a service in the cloud allowing direct access from the UI, which it registers in the cloud IaaS. The communication layers in the industrial network are presented in figure 4. The upper is the cloud

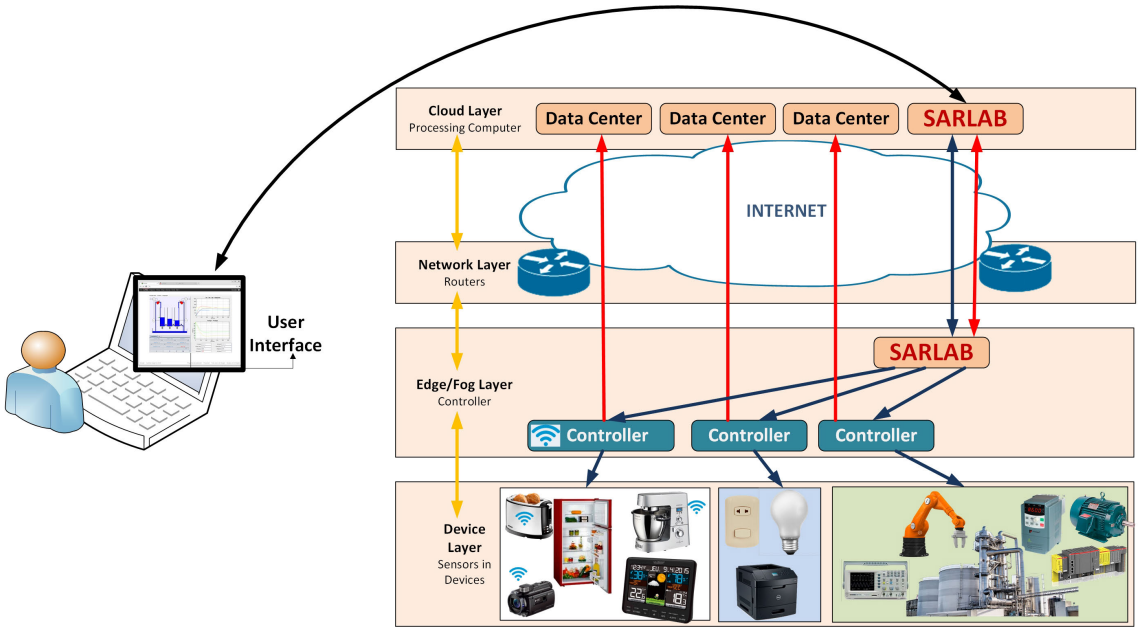


FIGURE 4. Communications arquitecture.

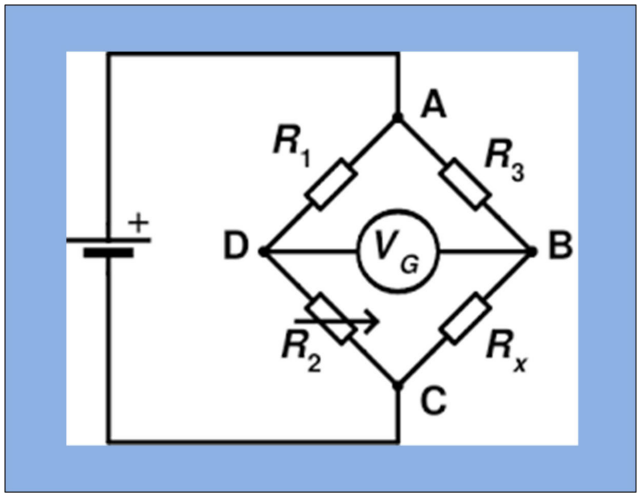


FIGURE 5. Wheastone brigde.

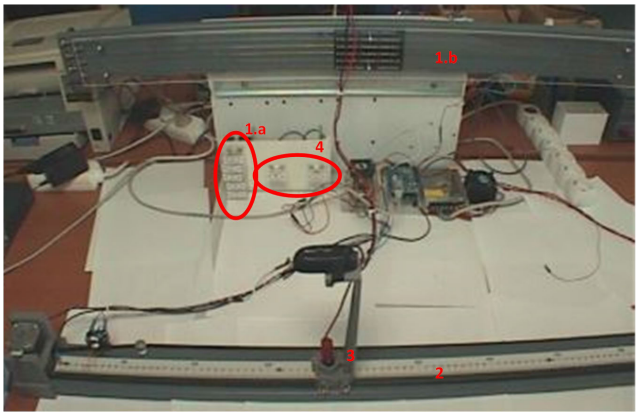


FIGURE 6. A general view of the use case plant.

layer, and going down, network layer with the routers, fog layer where controllers are connected and finally device layer with physical devices. The links between the end user computer and the fog layer are identified by the corresponding URL (uniform resource locator).

SARLAB controls access from outside the cloud according to the corresponding user profiles. However, SARLAB does not limit direct access to the internet from the controllers in the fog layer. Consequently, events occurring in the industrial network of a company can make use of the services in the cloud without being controlled by SARLAB. Figure 4 also represents the communication links created by SARLAB, based on [48]. As can be seen, SARLAB controls the communications and access to the fog layer from the cloud so that the end user can remotely access the industrial network by means of the cloud service.

V. USE CASE

A. DESCRIPTION

In this section, a use case is presented, consisting of a Wheatstone bridge, the corresponding circuit for which can be seen in figure 5. R2 in figure 5 represents a set of resistors, each of which is of unknown value and needs to be ascertained. The system also provides the possibility of connecting two or three resistors in series or parallel. Figure 6 represents a general view of the plant. The set of resistors indicated above is labelled 1.a. The element labelled 1.b is a set of wires, each of which is of unknown value and also needs to be found out. Figure 7 represents the UI, where the user can choose a set of resistors, after choosing between resistors and wires.

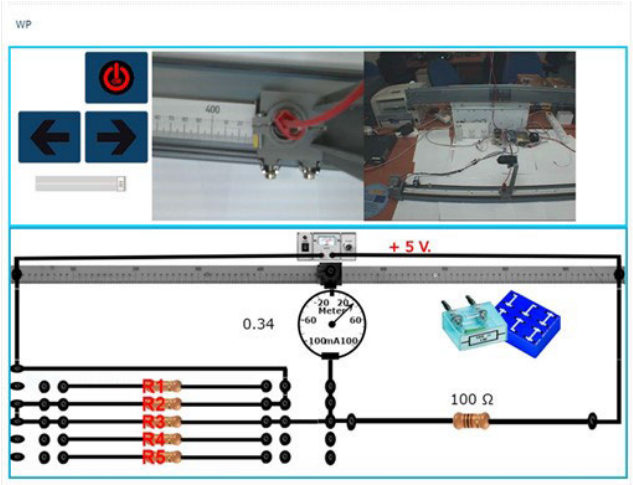


FIGURE 7. The use case user interface.

Elements R1 and R2 in figure 5 correspond to the rule at the bottom of figure 6 (labelled 2). A wire with a resistor of known resistance is divided into two parts by the isolated movable device (labelled 3 and henceforth denominated carriage). Finally, Rx in figure 5 is a resistor of known resistance with a value of 100 Ω if the user choses the resistors, and 1 Ω if the wires. This element is labelled 4 in figure 6.

Although this physical system is relatively simple, its characteristics allow different technical details regarding implementation to be illustrated. The intention is to show the potential in the functioning of the procedure rather than the complexity of the physical system itself. In fact, the procedure presented in this paper allows the inclusion of as many sensors and actuators as necessary, regardless of the complexity of the physical system.

In addition to the image shown in figure 6, the UI shows an amplified view of the isolated movable element on the right in figure 7, allowing the exact position on the rule to be clearly seen.

Regarding the architecture of the use case, the core of the system is a Raspberry Pi board [23] running the EJS where the INCA has been developed. On the one hand, INCA makes the communications with the UI, and on the other hand, it controls the set of relays and the carriage movement according to the orders received from the UI.

Figure 8 diagrams the structure of the complete system. The connections between the end user PC and the cloud can be seen, and likewise those between the cloud and the controller in the fog layer, which in this case is only the Raspberry Pi board. Likewise, the Raspberry Pi board is networked to the Arduino Mega with Shield Ethernet, which controls the carriage movement. These communications are carried out by Modbus TCP/IP, with the MB and OW elements in the INCA, as shown in figure 9.

In fact, the INCA involves an MMB element which is in communication with the Arduino board via the LAN. A Modbus TCP/IP slave, available in the Arduino libraries, runs on the Arduino board, which controls the carriage. The communications between those two elements are encapsulated in the Websocket protocol by means of a SMOW element, which corresponds to the CMOW element included in the UI (figure 10). In addition, the INCA controls the relays by GPIO (general purpose input/output) according to the end user selection in the UI. In order to do this, an SMB element has been included in the INCA, which receives the UI orders by means of the SSOW element. Both can be seen in figure 9. In figure 10, the CSOW element can be seen, which is in communication with the earlier SSOW. Finally, the Rasberry Pi also switches the physical devices on when an authorized

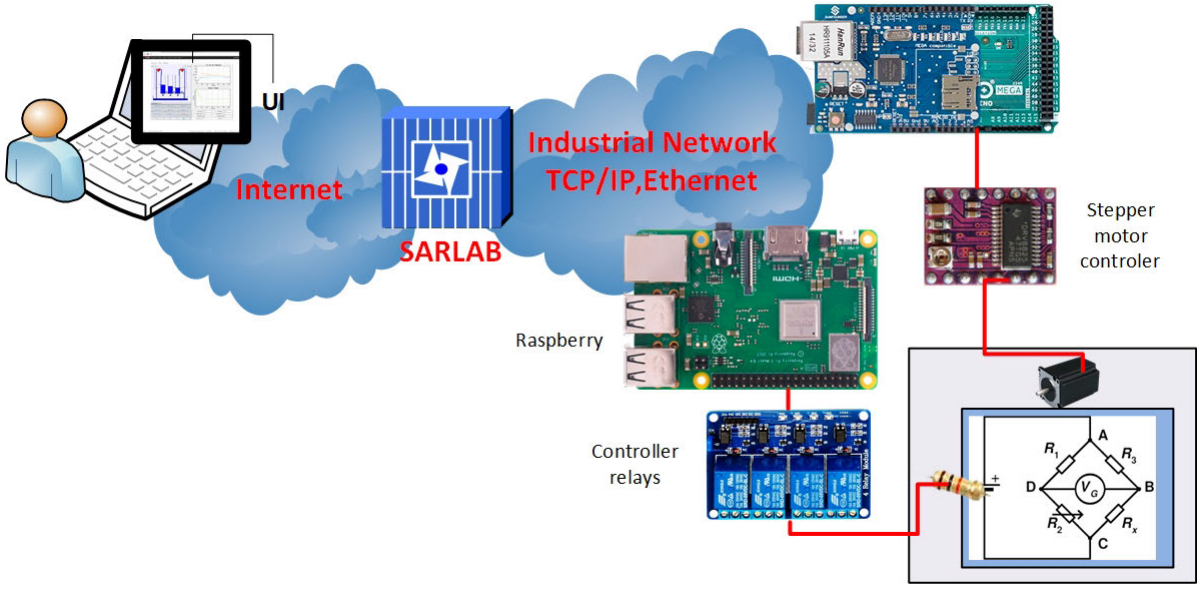


FIGURE 8. General scheme of the use case with the communications between the user and the physical system.

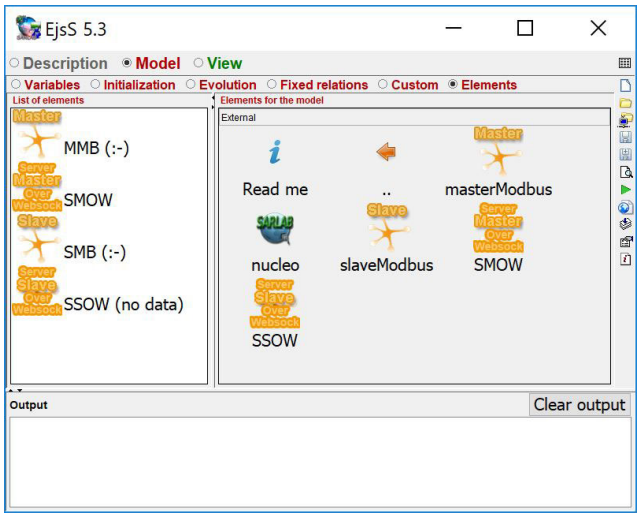


FIGURE 9. Elements in the use case INCA.

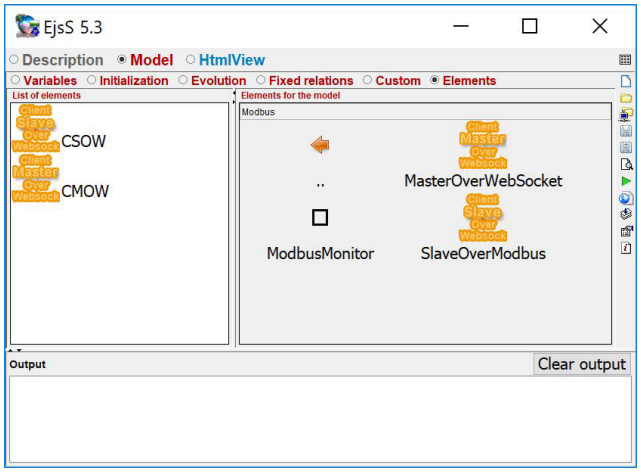


FIGURE 10. Elements in the use case UI.

user accesses them and switches them off when he/she leaves. SARLAB verifies the permissions of users at point of access to the UI.

Therefore, this use case represents a physical system connected to an industrial network, whose INCA has been implemented in EJS using the MB and OW elements, and the UI in EJSS using OW. The physical system consists of different devices, including relays controlled by digital signals and DC-DC motors controlled by PWM to establish the position of a mobile physical element. The nature of the physical system could have been any other whose control could be implemented digitally and analogically.

B. ACCESS AND RESULTS

The use case described in the previous subsection is in operation and use at the University of Huelva. Access can be made from anywhere with an Internet connection worldwide through a CMS, which in this case is moodle (one of the most widespread learning management systems in the world). The CMS, whose URL is http://sarlab3.uhu.es/sarlab/, includes a reservation system to regulate the access; although it can be also concurrent between remote users or remote and local users.

In addition, any authorized user (identified in the CMS) can access and manage the plant through any electronic device (PC, tablet or smartphone).

The result of applying the proposed procedure and set of tools to the described physical system in the previous subsection makes its access secure, controlled, organized and collaborative. Secure because it is done in an encrypted way and with the corresponding user profiles. Controlled because it may be subject to filters with different operating criteria, such as time zone. Organized because access can be sequenced by, for example, a reservation manager. Finally, collaborative because concurrent access to the physical plant is allowed.

In addition, the physical plant is available through the internet 24 hours a day, 7 days a week and the UI can combine real content (usually observed through some electronic device such as cameras and HMD displays) and virtual computer-generated content (augmented reality), adequately superimposed to improve the usability.

All these characteristics obtained in the use case can be extrapolated to other kind of systems as for example an industrial plant or smartdevices installed in a building. Although the use case is a plant in the university field; actually, the scalability of the developed solution is total, there is no technical barrier to incorporate more and more sensors and actuators. In fact, the main problem in industrial fields can be the security of communications and this is solved with this proposal.

VI. CONCLUSION

This paper presents a set of tools to facilitate the adoption of Industry 4.0 across all sectors of the business world. Although many industries have already migrated their IT processes to the cloud, the OT component remains accessible only at local level. The tools presented in this paper, based on open hardware and free software platforms, and implemented in EJS/EJSS, enable IT and OT to be integrated and accessed remotely. The integration of systems and devices running different protocols is achieved through Modbus TCP/IP within a single of level of the LAN. At the same time, access to devices is made available as a cloud-based service (via a browser) by means of communications encapsulated in Websocket. The result is a secure, controlled, organized and collaborative access to physical systems, which are in this way available 24 hours a day, 7 days a week through the internet by means of a UI that can combine real content and virtual computer-generated content (augmented reality), adequately superimposed. The solution and results addressed in this research are extrapolated to other sectors as the industry, service or household. Really, the application scope is not a problem, because the solution is completely general.

REFERENCES

[1] M. Savastano, C. Amendola, F. Bellini, and F. D'Ascenzo, "Contextual impacts on industrial processes brought by the digital transformation of manufacturing: A systematic review," *Sustainability*, vol. 11, no. 3, p. 891, Feb. 2019.

[2] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1443–1451, May 2014.

[3] G. Roussos, S. S. Duri, and C. W. Thompson, "RFID meets the Internet," *IEEE Internet Comput.*, vol. 13, no. 1, pp. 11–13, Jan. 2009.

[4] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID application development with the accada middleware platform," *IEEE Syst. J.*, vol. 1, no. 2, pp. 82–94, Dec. 2007.

[5] S. Kwon, J. Jeong, and T. Shon, "Toward security enhanced provisioning in industrial IoT systems," *Sensors*, vol. 18, no. 12, p. 4372, Dec. 2018.

[6] J. Park, "Advances in future Internet and the industrial Internet of Things," *Symmetry*, vol. 11, no. 2, p. 244, Feb. 2019.

[7] L. Yang, S. Yang, and L. Plotnick, "How the Internet of Things technology enhances emergency response operations," *Technol. Forecasting Social Change*, vol. 80, no. 9, pp. 1854–1867, Nov. 2013.

[8] B. W. Khoueiry and M. R. Soleymani, "A novel machine-to-machine communication strategy using rateless coding for the Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 937–950, Dec. 2016.

[9] T. Usländer and T. Batz, "Agile service engineering in the industrial Internet of Things," *Future Internet*, vol. 10, no. 10, p. 100, Oct. 2018.

[10] S. Jaloudi, "Communication protocols of an industrial Internet of Things environment: A comparative study," *Future Internet*, vol. 11, no. 3, p. 66, Mar. 2019.

[11] *The Modbus Organization*. Accessed: Apr. 17, 2019. [Online]. Available: http://www.modbus.org/

[12] K. L. S. Sharma, "20—Information technology-operation technology convergence," in *Overview of Industrial Process Automation*, K. L. S. Sharma, Ed., 2nd ed. Amsterdam, The Netherlands: Elsevier, 2017, pp. 359–375.

[13] L. De La Torre, M. Guinaldo, R. Heradio, and S. Dormido, "The ball and beam system: A case study of virtual and remote Lab enhancement with Moodle," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 934–945, Aug. 2015.

[14] J. Chacón, G. Farias, H. Vargas, A. Visioli, and S. Dormido, "Remote interoperability protocol: A bridge between interactive interfaces and engineering systems," *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 247–252, 2015.

[15] M. A. Prada, J. J. Fuertes, S. Alonso, S. García, and M. Domínguez, "Challenges and solutions in remote laboratories. Application to a remote laboratory of an electro-pneumatic classification cell," *Comput. Educ.*, vol. 85, pp. 180–190, Jul. 2015.

[16] U. Hernandez-Jayo and J. Garcia-Zubia, "Remote measurement and instrumentation laboratory for training in real analog electronic experiments," *Measurement*, vol. 82, pp. 123–134, Mar. 2016.

[17] I. González, A. Calderón, A. Mejías, and J. Andújar, "Novel networked remote laboratory architecture for open connectivity based on PLC-OPC-LabVIEW-EJS integration. Application in remote fuzzy control and sensors data acquisition," *Sensors*, vol. 16, no. 11, p. 1822, Oct. 2016.

[18] S. Weyer, M. Schmitt, M. Ohmer, and D. Gorecky, "Towards industry 4.0—Standardization as the crucial challenge for highly modular, multi-vendor production systems," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 579–584, 2015.

[19] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[20] B. P. Wong and B. Kerkez, "Real-time environmental sensor data: An application to water quality using Web services," *Environ. Model. Softw.*, vol. 84, pp. 505–517, Oct. 2016.

[21] *Arduino—Home*. Accessed: Feb. 4, 2019. [Online]. Available: https://www.arduino.cc/

[22] Phidgets Inc. *Products for USB Sensing and Control*. Accessed: Apr. 17, 2019. [Online]. Available: https://www.phidgets.com/

[23] Raspberry Pi Foundation. *Raspberry Pi—Teach, Learn, and Make with Raspberry Pi*. Accessed: Feb. 4, 2019. [Online]. Available: https://www.raspberrypi.org

[24] *LabVIEW 2019—National Instruments*. Accessed: May 20, 2019. [Online]. Available: http://www.ni.com/es-es/shop/labview/labview-details.html

[25] A. Calderón, I. González, M. Calderón, F. Segura, and J. Andújar, "A new, scalable and low cost multi-channel monitoring system for polymer electrolyte fuel cells," *Sensors*, vol. 16, no. 3, p. 349, Mar. 2016.

[26] T. Ewing, P. T. Ha, J. T. Babauta, N. T. Tang, D. Heo, and H. Beyenal, "Scale-up of sediment microbial fuel cells," *J. Power Sources*, vol. 272, pp. 311–319, Dec. 2014.

[27] H. Gad and H. E. Gad, "Development of a new temperature data acquisition system for solar energy applications," *Renew. Energy*, vol. 74, pp. 337–343, Feb. 2015.

[28] F. Salamone, L. Belussi, L. Danza, M. Ghellere, and I. Meroni, "An open source low-cost wireless control system for a forced circulation solar plant," *Sensors*, vol. 15, no. 11, pp. 27990–28004, Nov. 2015.

[29] G. Barbon, M. Margolis, F. Palumbo, F. Raimondi, and N. Weldin, "Taking Arduino to the Internet of Things: The ASIP programming model," *Comput. Commun.*, vols. 89–90, pp. 128–140, Sep. 2016.

[30] A. Solano, R. Dormido, N. Duro, and J. Sánchez, "A self-provisioning mechanism in openstack for IoT devices," *Sensors*, vol. 16, no. 8, p. 1306, Aug. 2016.

[31] A. Cela, J. Yebes, R. Arroyo, L. Bergasa, R. Barea, and E. López, "Complete low-cost implementation of a teleoperated control system for a humanoid robot," *Sensors*, vol. 13, no. 2, pp. 1385–1401, Jan. 2013.

[32] C.-T. Chao, M.-H. Chung, J.-S. Chiou, and C.-J. Wang, "A simple interface for 3D position estimation of a mobile robot with single camera," *Sensors*, vol. 16, no. 4, p. 435, Mar. 2016.

[33] D. Piromalis and K. Arvanitis, "SensoTube: A scalable hardware design architecture for wireless sensors and actuators networks nodes in the agricultural domain," *Sensors*, vol. 16, no. 8, p. 1227, Aug. 2016.

[34] M. R. Senouci, A. Mellouk, N. Aitsaadi, and L. Oukhellou, "Fusion-based surveillance WSN deployment using Dempster–Shafer theory," *J. Netw. Comput. Appl.*, vol. 64, pp. 154–166, Apr. 2016.

[35] R. Sanchez-Herrera, A. Mejias, M. A. Marquez, and J. M. Andujar, "A fully integrated open solution for the remote operation of pilot plants," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 3943–3951, Jul. 2019.

[36] A. M. Borrero and J. M. A. Márquez, "A pilot study of the effectiveness of augmented reality to enhance the use of remote labs in electrical engineering education," *J. Sci. Educ. Technol.*, vol. 21, no. 5, pp. 540–557, Oct. 2012.

[37] V. L. Lasky, D. K. Liu, S. J. Murray, and Y. K. L. Choy, "A remote PLC system for e-Learning," in *Proc. 4th ASEE/AaeE Global Colloq. Eng. Educ.*, 2005, p. 1655.

[38] M. Domínguez, J. J. Fuertes, M. A. Prada, S. Alonso, and A. Morán, "Remote laboratory of a quadruple tank process for learning in control engineering using different industrial controllers," *Comput. Appl. Eng. Educ.*, vol. 22, no. 3, pp. 375–386, Sep. 2014.

[39] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control telelab: A user-friendly interface for distance learning," *IEEE Trans. Educ.*, vol. 46, no. 2, pp. 252–257, May 2003.

[40] N. Faltin, A. Böhne, J. Tuttas, and B. Wagner, "Distributed team learning in an Internet-assisted laboratory," in *Proc. Int. Conf. Eng. Educ.*, 2002, pp. 18–22.

[41] T. A. Fjeldly, J. O. Strandman, and R. Berntzen, "LAB-on-WEB—A comprehensive electronic device laboratory on a chip accessible via Internet," in *Proc. Int. Conf. Eng. Educ.*, 2002, pp. 1–5.

[42] N. Wang, G. Song, and X. Chen, "Framework for rapid integration of offline experiments into remote laboratory," *Int. J. Online Biomed. Eng.*, vol. 13, no. 12, p. 192, Dec. 2017.

[43] A. Mejías, R. Herrera, M. Márquez, A. Calderón, I. González, and J. Andújar, "Easy handling of sensors and actuators over TCP/IP networks by open source hardware/software," *Sensors*, vol. 17, no. 1, p. 94, Jan. 2017.

[44] *Welcome to Easy Java Simulations Home Page*. Accessed: Apr. 17, 2019 [Online]. Available: http://fem.um.es/Ejs/

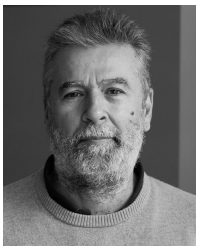
[45] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[46] D. F. Sadok, L. L. Gomes, M. Eisenhauer, and J. Kelner, "A middleware for industry," *Comput. Ind.*, vol. 71, pp. 58–76, Aug. 2015.

[47] N. Cai, M. Gholami, L. Yang, and R. W. Brennan, "Application-oriented intelligent middleware for distributed sensing and control," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 947–956, Nov. 2012.

[48] M. A. Marquez, R. S. Herrera, A. Mejias, F. Esquembre, and J. M. Andujar, "Controlled and secure access to promote the industrial Internet of Things," *IEEE Access*, vol. 6, pp. 48289–48299, 2018.

REYES SÁNCHEZ-HERRERA was born in Huelva, Southwest of Spain. She received the Industrial Engineering degree and the Ph.D. degree (Hons.) in electrical engineering, in 1995 and 2007, respectively. She is currently a Professor with the Department of Electrical Engineering, University of Huelva. She worked in electrical engineering within the first research group to which she belonged. In electronics and communications in the second. She is the main research of the applied multidisciplinary research group with the University of Huelva. Her main interests include electrical power quality, renewable energy systems, and engineering education.



MARCO A. MÁRQUEZ received the Industrial Engineering degree from the University of Seville, Seville, Spain, in 1979, and the master's degree in industrial engineering from the University of Huelva, Huelva, Spain, in 2009. He was an Associate Professor with the Department of Electronic Engineering, Computer Systems and Automatic Control, University of Huelva, from 1994 to 2002. He is currently a Researcher with Huelva University. He is also a regional Instructor of Cisco Systems with the CIT Business University Foundation, University of Cádiz, Cádiz, Spain. His research interests include new e-learning technologies and the communications aspects in remote labs. His developments are the basis UNILABS communications systems, a network of Spanish universities that share its laboratories through the Internet.

JOSÉ M. ANDÚJAR (Senior Member, IEEE) was born in Huelva, Spain. He received the Ph.D. degree, in 2000. He is currently a Full Professor of systems engineering and automatic control with the University of Huelva, Spain. Throughout his professional life he has received 23 awards and academic honors. He has conducted ten Doctoral Theses with eight prizes. He has 12 international patents. He has more than 300 publications, among them more than 100 articles published in indexed journals in the ISI Journal Citation Reports. Specifically, he has 43 quartile Q1 publications in 19 different journals; most of these journals are among the top ten in their categories, and several are number one. He has an H-index (SCOPUS) = 22. He has led or co-led 50 research projects funded by public institutions and companies. His main research interests are control engineering, renewable energy systems, and engineering education.