REMOTE MONITORING AND CONTROL VIA ROBOT WEB SERVICES

Liliana STAN[1*], Cristina PUPĂZĂ[2], Adrian Florin NICOLESCU[3]

Close to virtualization, sustainability is another Industry 4.0 advancement that will have a substantial impact on the planning and management of future factories, as modern manufacturing processes become characterized also by social sustainability. Environmentally and economically, the energy consumption of machines and robots is a crucial part of a production line. Consequently, analyzing, monitoring, and minimizing the consumption of energy continues to be an active field of research. This paper presents a new approach to remotely monitor and control an industrial robot by employing robot web services. A web-based framework is proposed to monitor and assess the performance of a robotic application for assembly by tracking the robot energy consumption and production rate. Experimental results validate the reliability of the proposed approach.

Keywords: industrial robot, intelligent manufacturing, robot web services, remote monitoring.

1. Introduction

The next generation factory floor features collaborative robots, advanced Computerized Numerical Control (CNC) machinery, service robots and Human Machine Interface (HMI) technology. Availability and maintainability, which can be defined as the probability that a system will operate satisfactorily and effectively at a particular moment and its capability to be repaired, are critical for all machinery. The modern manufacturing architecture must consider the requirements of interconnected Cyber-Physical Systems (CPS), such as interoperability, scalability, concurrency, and synchronization. Consequently, concurrent CPS have to be fault and failure tolerant and diagnostic subsystems embedded into the CPS network must be designed such that fault/failure localization and repair can be done in real-time. In modern manufacturing, the joining of Information Technologies (IT) and Operational Technologies (OT), also known as the OT/IT Convergence [1], is having a revolutionary effect. Web

[1*] Corresponding author, PhD student, Dept. of Robots and Manufacturing Systems, University POLITEHNICA of Bucharest, Romania, e-mail: elena_liliana.stan@stud.fiir.upb.ro

[2] Prof., Dept. of Robots and Manufacturing Systems, University POLITEHNICA of Bucharest, Romania, e-mail: cristina.pupaza@upb.ro

[3] Prof., Dept. of Robots and Manufacturing Systems, University POLITEHNICA of Bucharest, Romania, e-mail: afnicolescu@yahoo.com

Services and APIs are prevalent methods in modern manufacturing processes for achieving interoperability, which is the capability of devices and systems to communicate with each other.

The ability to remotely manage and monitor machines is one of the primary applications of IoT. Important aspects in the field of industrial automation include remote monitoring and data visualization. The information may be applied for remote operation, maintenance planning, incident analysis, and optimization. Remote equipment control is essential when machines are inaccessible or pose serious risks to human safety. The manufacturing system's performance and condition can be assessed by employing sensors to collect and process data on each device and system. Implementing Digital Twins (DT) to enable a DevOps approach for Cyber-Physical Production Systems (CPPS) [2] creates a fully integrated and automated production process, enabling continuous improvement. The deployment of edge computing technologies is advancing concurrently with the advancement of IIoT. Edge computing moves probing closer to the source, while reducing latency and ensuring reliable collection of real-time data. Consequently, new IIoT systems are shifting from centralized cloud-computing which are ideal for non-real-time application that require vast amounts of storage and high-processing power, to distributed intelligent edge computing, suitable for real-time services as they exhibit low latency [3].

In the wake of COVID-19, the ability to remotely access and control critical processes has become rather essential. Consequently, the pandemic has accelerated the development of cyber security technologies, and services involved in safeguarding factory floor assets that use industrial Ethernet and have an Internet connection.

Production Analytics software is a platform that enables real-time visualization of machine, equipment, and production data, and provides analysis tools to examine past data and visualize production losses to devise solutions successful in modern manufacturing. However, data processing in traditional manufacturing poses many challenges in the process of data transmission, storage, and analysis, mainly due to old equipment with low or no computational power, and lack of standard interfaces and protocols between the machines. Bridging the gap between OT and IT is one of the many roles of a Supervisory Control and Data Acquisition (SCADA) system which, for modern manufacturing, should be based on open standards that promote interoperability, scalability, collaboration, facilitate integration, and be web-deployable. The key capabilities of a SCADA system are to provide user-friendly HMI on the factory floor, to make OT data instantly available to operators, and to enable remote monitoring and control. There are currently powerful and dedicated commercially available solutions to enable the shift to intelligent manufacturing, but they are also expensive for small and medium sized companies. This paper presents a cost-effective, web-based solution to remotely monitor and control industrial robot applications.

The paper is structured as follows: Section 2 presents a brief introduction to remote monitoring and control of industrial robotic applications; Section 3 introduces the proposed method applied in a case-study of robotic assembly and its key components; Section 4 describes the experimental results; and finally, Section 5 summarizes the conclusions and highlights the contributions.

2. Remote monitoring and control of industrial robotic applications

Industry 4.0 is the digitalization of production through embedded sensors in all product components and manufacturing equipment, ubiquitous CPS, data collection and analysis. CPS are embedded systems with both a cyber and a physical world, between which there is a constant and iterative exchange of information [4]. Sensors are employed to collect environmental conditions in the physical world, which are subsequently assessed using processing power from the cyber world. The collected information can be distributed to other entities via communication interfaces or employed for predefined rules of behavior to interact with the physical world via actuators. In the last decade, the DT paradigm has been studied as an approach to virtualize real-world entities by designing software counterparts that enable intelligent services, such as tracking the current state of the physical entity, monitoring to detect and predict potential critical issues, optimize the performance, and enhance the physical counterpart's functionality [5]. In industrial applications, the real-time capability and the computation power for production process, energy analysis, etc., are both essential.

The robot's repetitive operations must be performed precisely over time, posing a challenge for robot manufacturers who must ensure accurate fault detection and diagnosis. Jaber [6] presented a comprehensive study on robot hardware, transmission faults and data acquisition. Soualhi et al. [7] presented a practical methodology for online diagnostics of multi-axis robots to identify small deviations by employing a direct monitoring approach that exploits the already installed encoders on each servomotor and an indirect monitoring approach that uses heterogeneous sensors placed at the robot tool level. Zhao and Li [8] developed an advanced method of robot working-state monitoring based on intelligent video surveillance, with real-time capability and high transmission frame rate. Wescoat et al. [9] presented a comparative study of various state of the art algorithms employed in detecting robot abnormalities.

In the manufacturing industry, a Production Key Performance Indicator (KPI) is a particular kind of measurement used to assess the performance of the production process. Such examples are the machine uptime rate, cycle time, throughput rate, or production attainment. Maintenance can be described as the aggregate of all technical and administrative procedures, including supervisory actions, intended to keep an item in a state where it can perform a specified function or restore it to that state, for the duration of its design life or longer. Conventional control charts are statistical tools typically employed for illustrating a quality feature plotted against the sample number (or time) with a center line and two control limits (upper and lower) to clearly distinguish between changes caused by unpredictability in the system and changes caused by a system malfunction. Predictive Analytics is the practice of extracting data to identify patterns and forecast future outcomes and trends. Predictive Maintenance (PdM) intends to use data collected from sensors and connected machines to identify when a failure is imminent and order preventive maintenance. The objective of a PdM program is to maximize equipment use by minimizing downtime.

The aim of the proposed web-based platform is to present the energy consumption in an interactive control chart that facilitates the monitoring of robot energy usage. The web Graphical User Interface (GUI) enables the monitoring of the robotic system based on sensor signals, as sensor changes are immediately signaled.

3. The web-based framework

Fig. 1 illustrates the proposed solution for remote monitoring robotic applications via robotic web services (RWS). Color coded in green is the current state of the lab equipment, whereas the digital replica is grey-colored. The blue elements represent the key components of the web-platform and violet represents possible applications of the web-platform.

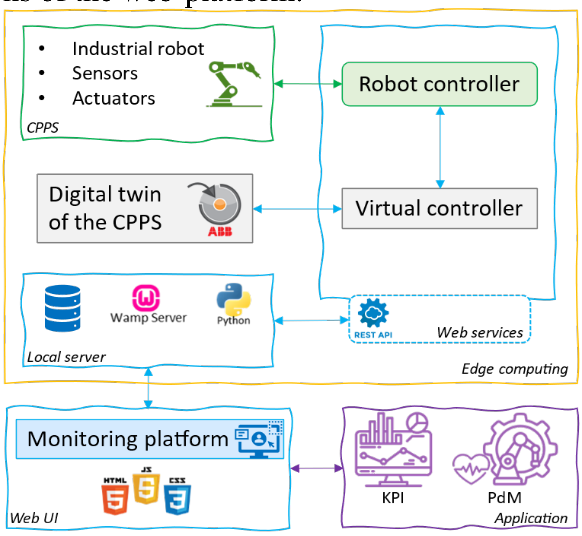

Fig. 1. RWS-based solution for remote monitoring robotic applications

It is possible to access information from the robot controller using robotic web services that adhere to the architectural form of Representational State Transfer (RESTful) Application Programming Interface (API), using the

Hypertext Transfer Protocol Secure (HTTPS) protocol, returning messages consisting of Extensible Markup Language (XML) and JavaScript Object Notation (JSON) data as further detailed in Section 3.3.

3.1 Case study: robotic assembly

This case study focuses on a production system in which a robot assembles components transported to its workspace by a conveyor. Robotic assembly is a manufacturing process in which a robot (embedded with specialized end-effectors) positions, matches, fits, and assembles interchangeable parts or sub-assemblies sequentially to produce a functional product (Fig. 2). This procedure requires a high level of repeatability, dependability, adaptability, and sequencing. Typical robot-assembly processes require compliance and force control to provide stable contact between the manipulator and the workpiece as well.

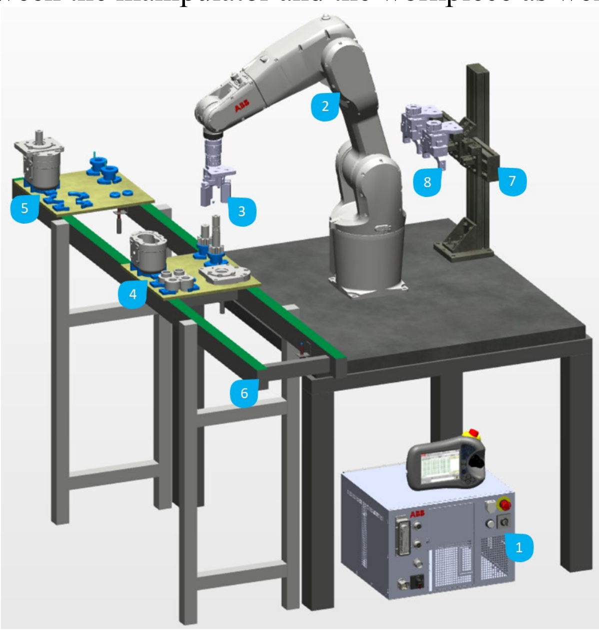

Fig. 2. Isometric view of the robot assembly application

As illustrated in Fig. 2, the robotic application comprises the robot controller (1); the ABB IRB 1200 robot arm (2) which is equipped with an automatic tool changer consisting of a compensation unit (Schunk AGE-F 040) together with a quick-change system (Schunk SWS 011), and an effector (3); pallets transporting the parts to be assembled (4) and the finished product (5) on a conveyor (6); a tool storage system (7) holding robot effectors (8). The articulated robot has a payload of 5 kg, a max vertical reach of 900 mm, and a repeatability of ±0.06mm (suitable for assembly applications, machine tending, or material handling). The robot is equipped with the IRC5 Compact controller and RobotWare software (RW v6.08) for robot control.

The components of the workcell were designed using modern Computer-Aided Design (CAD) tools in accordance with their physical configuration. The virtual model mirrors the manufacturing process in terms of design, collision avoidance, fault detection, testing, planning, process optimization, and monitoring. For designing the DT, ABB RobotStudio provided the necessary tools:

- To virtualize the robotic assembly workcell.
- To generate a collision-free toolpath.
- To program the robot considering sensor data and virtual models of the physical components.
- To simulate the process, analyze the motions and assess the robot energy consumption.

3.2 Robot controller

The IRC5 robot controller features the following communication technologies: robot web services, a programming interface based on Hyper Text Markup Language (HTML5) for communicating with robots from any device, regardless of operating system, and socket messaging, which enables machine-to-machine communication via the exchange of Transmission Control Protocol/ Internet Protocol (TCP/IP) messages over a network. *The virtual controller* can be described as a software that emulates the real robot controller to allow the same software that is controlling the robot, to run on a machine for offline programming, simulation, and analysis purposes. Whether virtual or real, the IP address of the controller can be discovered using the Bonjour discovery tool (Fig. 3).

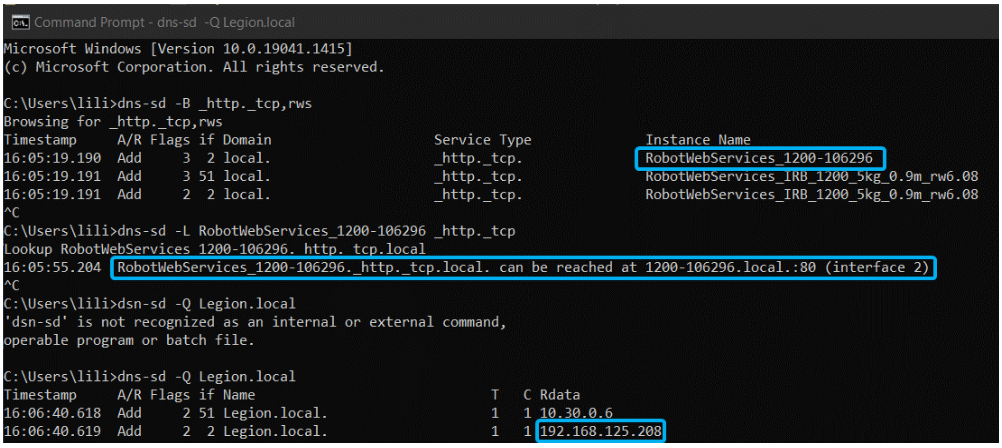

Fig. 3. Discovery of the robot controller employed in the experimental setup

3.3 Robotic Web Services

RESTful is a software architectural style that describes a uniform interface between physically separate components, often across the Internet in a Client-Server architecture. In RESTful communications there is a distinct request method for each of the operations to resources, namely GET, PUT, POST and DELETE. The web platform was designed to access information from the robot controller using web services that adhere to the architectural form of RESTful APIs [10] using the HTTPS protocol, returning messages consisting of XML and JSON data, the latter being more compact and returns less overhead. The resources are returned as Unicode, UTF-8, while the controller supports Latin1 (ISO-8859-1), so RWS converts characters between Latin1 and UTF-8. Operations that take a long time to carry out, such as backup actions, are handled asynchronously. Using cookies (ABCDX), a session can support several secure WebSocket connections.

The Robot Web Services (RWS) [11] consists of several services and each service may have additional services or one or more resources. Fig. 4 illustrates the RWS resource map and highlights the main services employed in this paper.

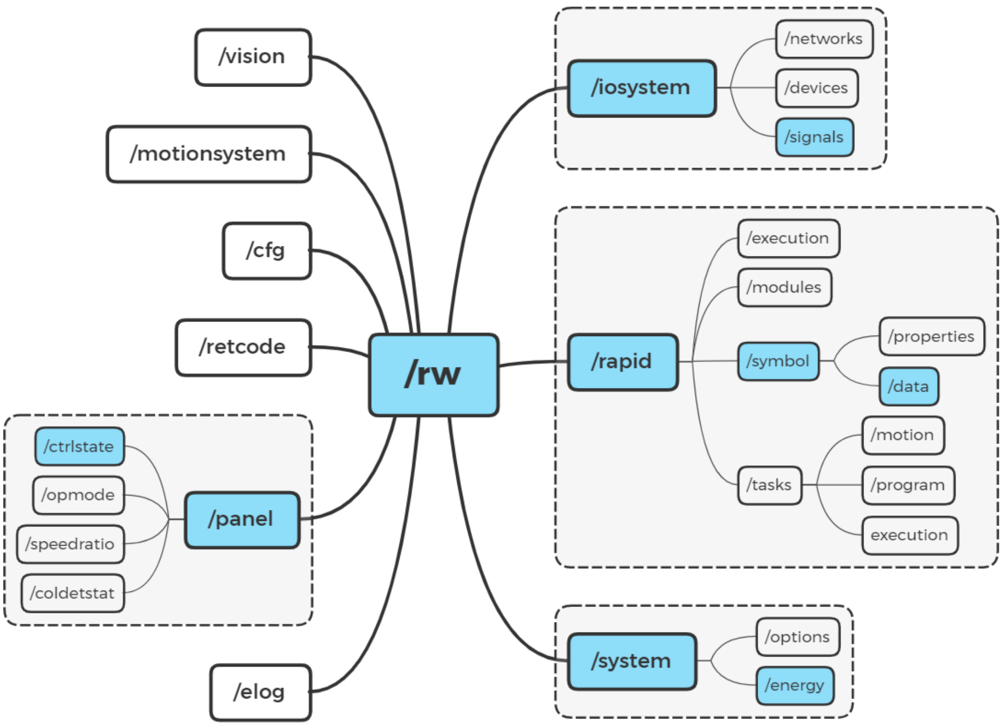

Fig. 4. RobotWare Services resource map

TCP and User Datagram Protocol (UDP) are typical Internet transport protocols deployed by ABB IRC5 devices. This means that IRC5 systems can be connected to a standard network (TCP/IP/Ethernet) just like any other computer or network equipment, reducing expenses and unifying network management. Moreover, the interconnection of control systems and "office" systems, such as Enterprise Resource Planning (ERP), enables a vast array of new applications that benefit from such vertical integration, from the shop floor to enterprise management. Nonetheless, the direct connection of control systems to the plant network introduces security concerns, such as malware infections (viruses, Trojans), denial of service, and leakage of confidential data. Consequently, for the proposed approach, it is important to identify the most severe security threats and mitigate actions against them. The User Authentication System (UAS), employed in this work, is a security feature that is implemented in every IRC5 controller. UAS restricts which personnel can operate which controller activities by specifying the users and groups with access to the controller and its functionality.

3.4 Development of the web platform

Table 1 summarizes the key characteristics of the proposed framework applied in the case study. The programming of the robot is based on sensor signals, and the assembly order is predetermined. Therefore, the web-platform has to communicate with every device in the production system. This criterion is accomplished, since the web-platform can communicate with every robot, sensor, and actuator that is controlled by the robot controller.

The two-way communication enables the web-based framework to acquire data from the robot controller and have control over it, such as stopping the robot in an emergency. The web-based framework must collect data on an hourly basis, while system control must be close-to-immediate, with no significant delays. The robotic process is entirely automated, and the DT includes robust simulation features, such as checking the toolpath against collisions and calculating the energy usage.

Table 1

Key characteristics of the proposed framework

Integration level	Robot	Production System	Factory environment
Connectivity mode	Uni-directional	Bi-directional	
Update frequency	Daily	Hourly	Immediate/real time
CPS intelligence	Human triggered	Automated	Autonomous (AI-driven)
DT simulation capabilities	Static	Dynamic	Predictive/prescriptive
Digital model	Geometry, kinematics	Control behavior	Multi-physical behavior
Computing level	Edge	Fog	Cloud

To compute the energy consumption, the DT must combine kinematic and mathematical models with an established behavior control module. Finally, since only real-time communication and no complex processing is necessary, it is sufficient for the web-based framework to rely on edge-computing capabilities.

The following key elements of the web-platform have been created (as described in Fig. 5):

1) A relation between the real and the virtual robot controller.

2) A communication link between the controller and the webserver.

3) A utility class to facilitate the controller-webserver communication.

4) A utility service to collect and store data (e.g., the latest measurement of the consumed energy).

5) Web GUI that enables a user to monitor a selection of sensor signals from the workcell (The Dashboard).

6) Visualization charts that enable a user to monitor the workcell productivity and the energy consumption of the robot (The Control Charts).

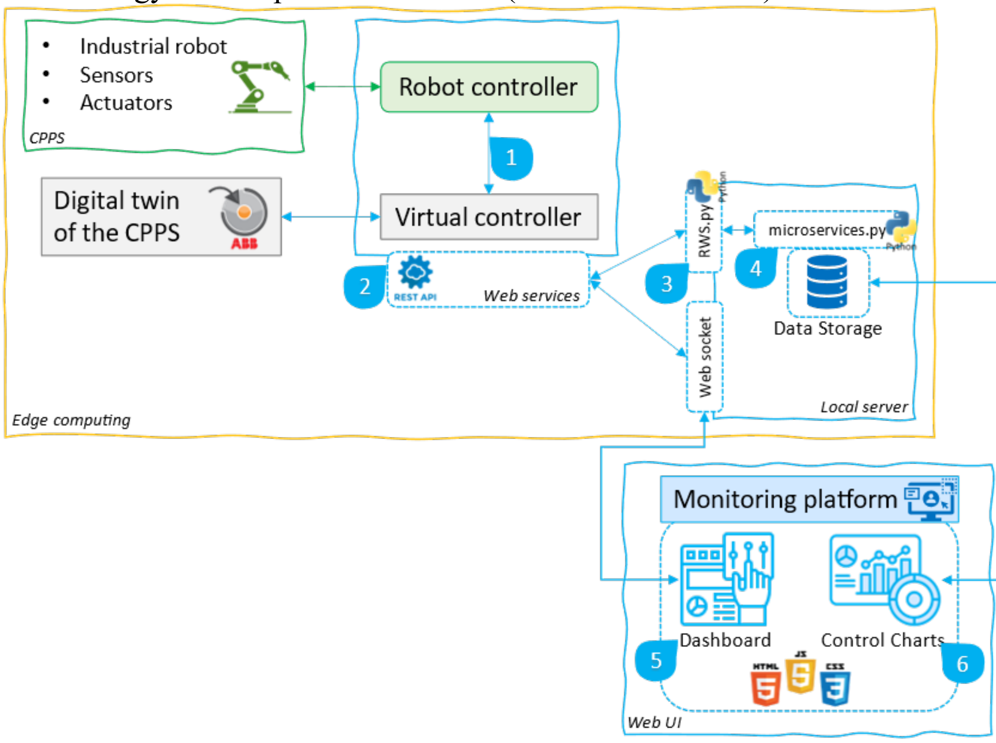

Fig. 5. The web-based monitoring framework applied in the case study

Fig 6 illustrates the Dashboard which enables a user to connect to a robot controller and to monitor a predefined selection of signals. The data (e.g., energy consumption) is collected and stored on a local server in JSON format, and the web-based GUI is created using HTML5, JavaScript, and CSS3 as well as the Bootstrap 4 [12] framework for the front-end development and the Highcharts [13] library to create the interactive charts that display the information stored on the server. The REST API paradigm is employed to allow communication between the front-end and the back-end, while the robot web services facilitate a communication link with the robot controller.

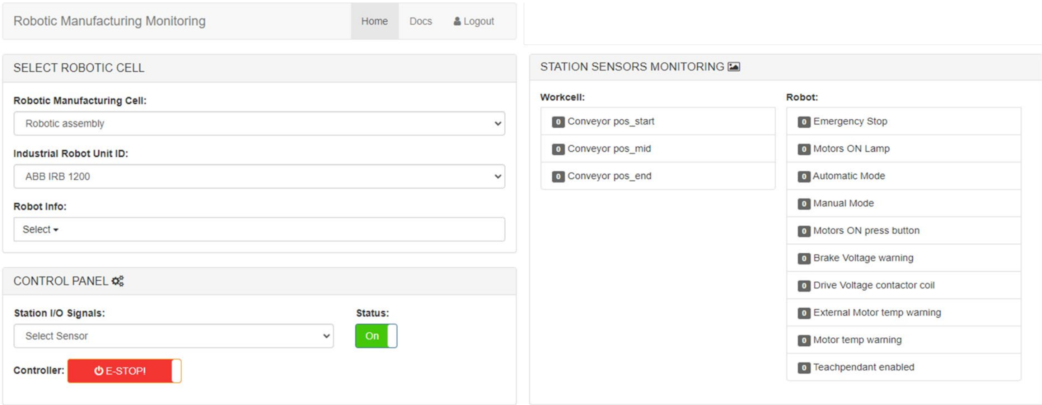

Fig. 6. The Dashboard: enables a user to connect to a robot controller (left) and to monitor a predefined selection of signals (right)

The automatic mode is a method of operation in which the robot performs in accordance with the task program, without manual operator control. The following tasks are typically performed in automatic mode: adjusting the robot speed; loading, starting, modifying, and terminating RAPID programs via remote clients. A simulation procedure to demonstrate the functionality of the platform was performed, as illustrated in Fig. 7: a signal (1) on the Dashboard indicates that the teach pendant is active (2), while other signals (3, 5), indicate that the robot control is in the Automatic Mode (4) and the motors are on (6), as stated on the teach pendant interface as well.

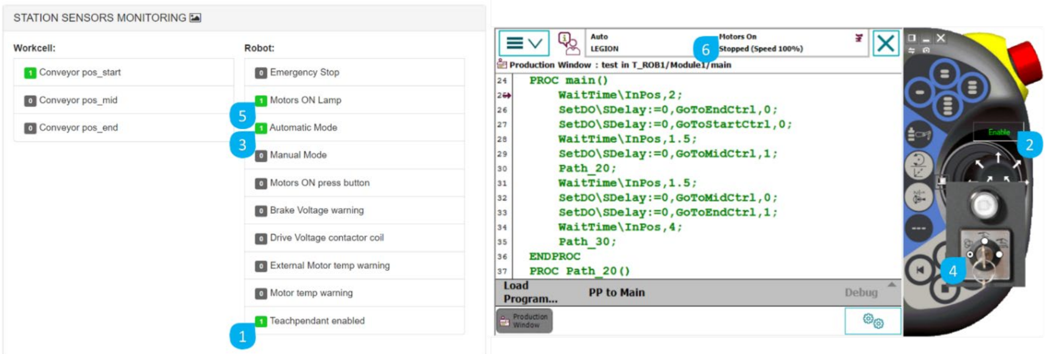

Fig. 7. The teach pendant is enabled, and the robot is in automatic mode

For testing purposes and to replicate the functionality of the control charts, RobotStudio was employed to continuously simulate the robotic process, while the microservices script ran in the background to collect and store data from the virtual controller every hour. Fig. 8 depicts the interactive chart that displays the energy consumption of the robot. An expected level and two control thresholds are defined for when the consumption exceeds or falls below the desired range. If the productivity of the workcell is constant but the energy consumption exceeds the thresholds, this may indicate a problem with the robot.

The records presented in Fig. 8 has only a demonstrative purpose. The experimental method starts with establishing a connection between the real and the virtual robot controllers to facilitate the transfer of the same RAPID program employed previously in the simulation, to the IRC5 controller and start its execution. As the controller executes the robot program step by step, the microservices script collects the required data, namely, the number of finished parts and the energy consumption of the robot. As soon as the data is collected in the corresponding JSON file, the visualization charts display it on the web-platform. A control script has been devised to periodically change the clock on the test machine so as not to waste resources while evaluating the capacity to gather data hourly. After data collection and storage, the control script would jump the time by 56 minutes. This procedure would be performed six times. The process ensures that the robot will not operate continuously for six hours without influencing the outcome of the test.

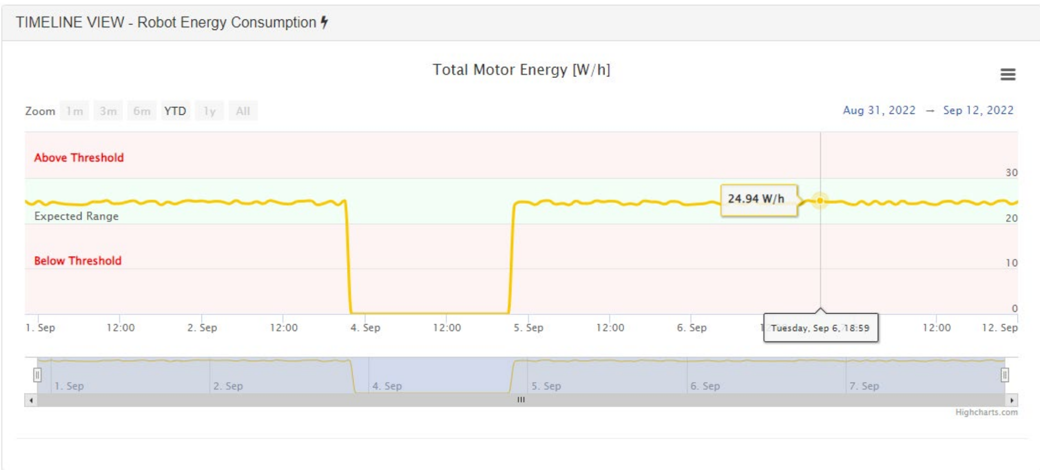

Fig. 8. The Robot Energy Consumption chart

The signals for the virtual robot's Total Motor Power and Total Motor Energy are derived from an ABB robot working under normal conditions. Nevertheless, as with any simulation model that employs approximations, the recorded energy data is slightly understated, as electromechanical losses in robot drives and operating condition impacts are likely to be neglected by the software model. Moreover, in a real setting, there are various components of the robotic system that consume energy, such as the controller, the cooling fans, HMI devices, or other auxiliary devices.

4. Experimental validation

To verify the effectiveness of the proposed approach, experiments have been carried out to validate the web-platform's signal monitoring capability: it must be determined that the changes of the sensors defined in the robot controller are signaled in timely fashion on the web-platform.

The experimental station (Fig. 9) consists of an ABB IRB 1200 articulated arm having a 5 kg handling capacity, 0.9 m reachability, and 0.06 mm position repeatability (suitable for assembly applications, machine tending, or material handling); an IRC5 Compact controller with the RobotWare software version: RW v6.08; sensors and actuators. RobotWare supports every feature of the robot system, including motion control, application program development and execution, and communication.

Fig. 9. The experimental setup consisting of an articulated robot IRB 1200, featuring the connection between the IRC5 controller (1) and the laptop running RobotStudio (2)

As part of the simulation procedure, Fig. 7 illustrated that when the Automatic mode is active, it is indicated both on the Dashboard and on the teach pendant interface. When the Manual mode is active, the motors are still on, but the motors lamp will flash intermittently. Consequently, Fig. 10 illustrates the motors lamp of the real controller alternating from 0 to 1. On the experimental setup, this signal is acquired employing RWS.

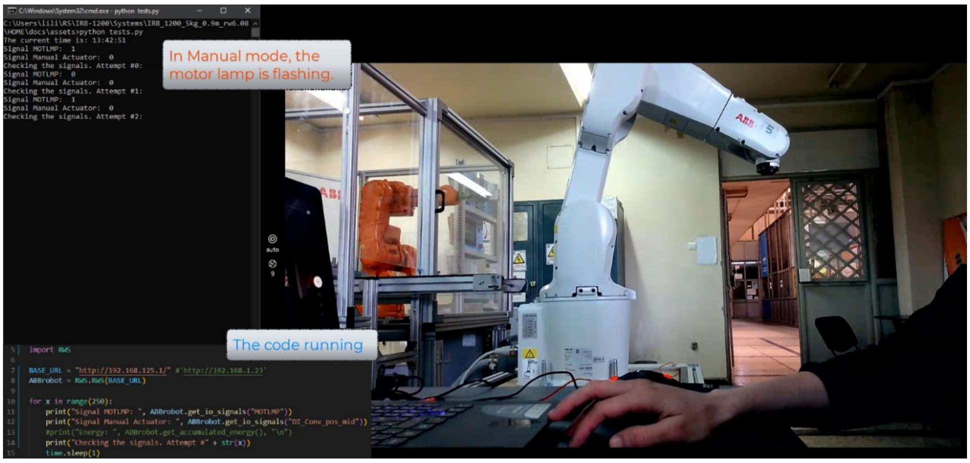

Fig. 10. The robot control is in Manual mode; therefore, the motor lamp is flashing. Consequently, the MOTLMP signal alternates from 0 to 1

A test button was employed to check the signal delay and thus the responsiveness of the web platform's functionalities. After physically connecting the test button to the controller, a digital signal was defined for it. Employing again the robot web services, Fig. 11 illustrates that when pushing the test button (1, 2), the signal collected from the robot controller will also change from 0 to 1 withing a timeframe of around 100ms, depending on the processing capabilities of the machine running the script (CPU: Intel I7-7700 @2.80GHz, SSD Samsung 970 EVO 1TB).

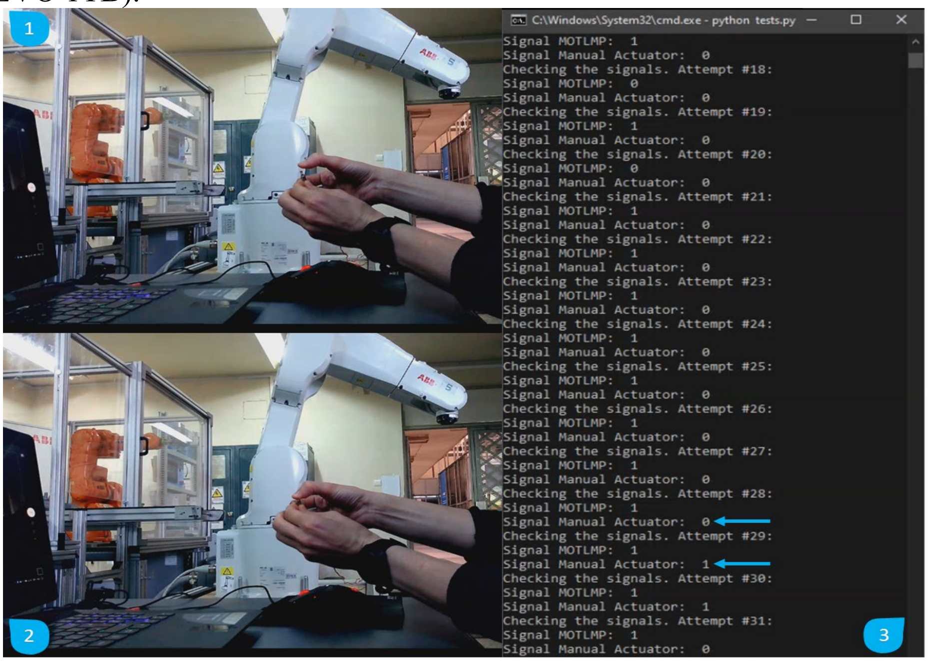

Fig. 11. The test button is connected to the robot controller and thus assigned to a digital signal

As a software employed heavily in industry, RobotStudio offers two methods to analyze the energy consumption of the robot: online and simulation mode. As illustrated in Fig. 12, in the simulation mode, the graphical robot moves as the robot program on the virtual controller commands (2) and the total motor power, and the total motor energy (4) are derived from an ABB robot working under normal conditions. In contrast to this, in the online mode, the graphical robot moves as the robot program is executed on the real controller, however, with a slight, expected delay (1). In online mode, the signals for total motor power and total motor energy (3) are read directly from the robot motor drives. As with any simulation model that employs approximations, the energy data recorded from the simulation is slightly understated, as electromechanical losses in robot drives and operating condition impacts are likely to be neglected by the software model. In future work, the errors of the simulation mode could be further evaluated. As illustrated, the simulated total energy consumption is approx. 180 J, while the total energy measured on the robot is approx. 140 J, therefore the absolute error is approx. 40 J and the relative error, 16.6%.

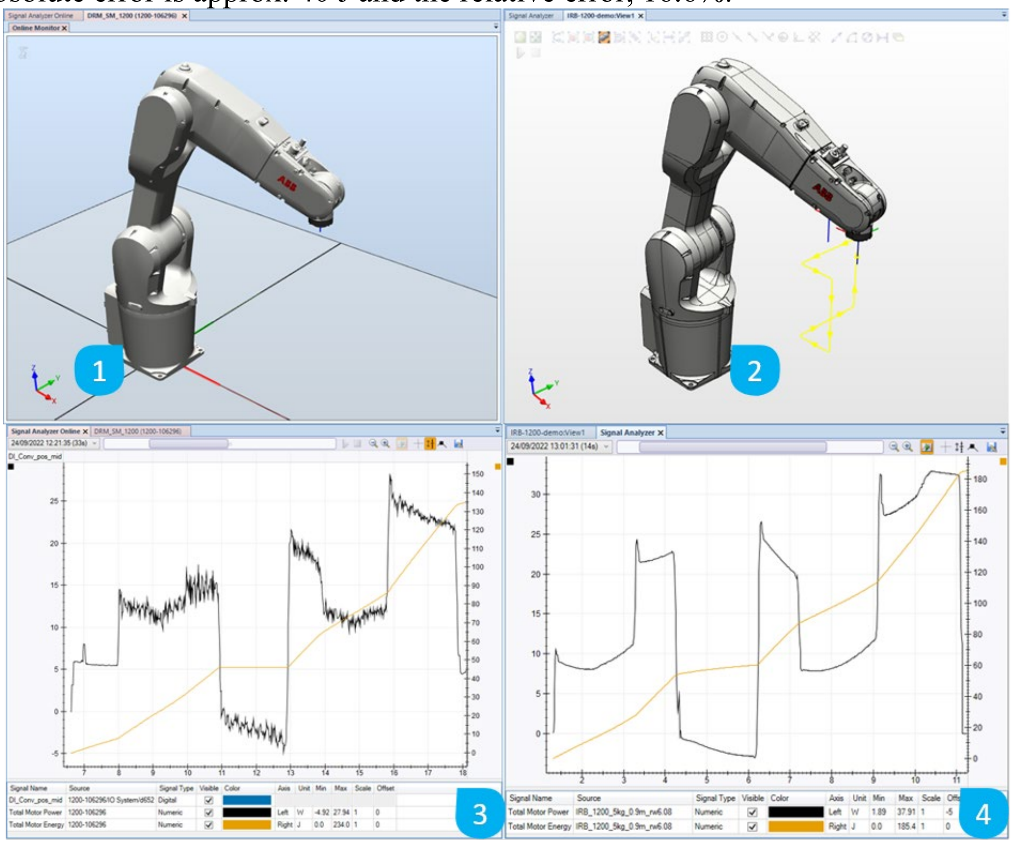

Fig. 12. RobotStudio Online monitoring: robot movement (1) and energy consumption analysis (3). RobotStudio Simulation monitoring: robot trajectory (2) and energy consumption (4)

5. Conclusions

This paper proposed a new approach to remotely monitor and control industrial robots via robotic web services. Moreover, an overview of the development of a web-based monitoring platform for robotic assembly workcell has been presented. The simulation results have demonstrated the web-platform's functionality and its smooth integration with the robotic system while the experimental results have validated its capabilities.

Summary of the main accomplishments:

- Access to the web-platform must be authorized and is only possible with a username and password. The digest authentication method upholds the security of the system when logging in by using the HTTP protocol and cryptographic hashing to avoid replay attacks.
- The interactive control charts facilitate the monitoring of the workcell output of the robot's energy usage.
- The Dashboard GUI enables the monitoring of the robotic system based on sensor signals. The simulation demonstrated that the sensor changes are immediately signaled.
- The simulation also revealed that it is possible to stop the robot from executing the RAPID program in the event of an emergency.

The main contributions: the development of the web-based monitoring platform for robotic systems employing robotic web services; the application of edge computing for data acquisition and processing in a robotic system; employing both a virtual and real robot controller for testing purposes and for replicating the functionality of the web-platform.

Future work may be directed on numerous challenging areas:

- The web-based platform may be extended to include more than one robotic system, granting it the ability to monitor factory floor production of various robotic applications. By employing web services, the web-platform can function as a communication core for numerous robotic systems, therefore, enhancing the operational performance and cost-efficiency.
- When the demand for processing capacity increases, the framework can be expanded to fog or cloud computing, which will also offer further analysis capabilities such as AI-based approaches for condition monitoring.
- The investigation of security weaknesses in networked machines and robotic systems, as well as the development of security methods and architectures, are equally promising directions.
- Expand the web-platform's capabilities to reduce the robot energy consumption through power management techniques while monitoring more data from the work environment to adjust the robot's power consumption based on workload, or through hybrid control methods, which combine

traditional control methods, such as energy-efficient motion planning with new techniques such as machine learning.

5. Acknowledgements

This work has been funded by the European Social Fund from the Sectorial Operational Programme Human Capital 2014-2020, through the Financial Agreement with the title "Scholarships for entrepreneurial education among doctoral students and postdoctoral researchers (Be Entrepreneur!)", Contract no. 51680/09.07.2019 - SMIS code: 124539.

REFERENCES

[1]. *L. Patera, A. Garbugli, A. Bujari, D. Scotece, and A. Corradi*, A Layered Middleware for OT/IT Convergence to Empower Industry 5.0 Applications, Sensors, **vol. 22**, no. 1, Dec. 2021, p. 190.

[2]. *M. Ugarte Querejeta, L. Etxeberria, and G. Sagardui,* Towards a DevOps Approach in Cyber Physical Production Systems Using Digital Twins, in Lecture Notes in Computer Science, **vol. 12235**, LNCS, 2020, pp. 205–216.

[3]. *D. Hastbacka*, Dynamic Edge and Cloud Service Integration for Industrial IoT and Production Monitoring Applications of Industrial Cyber-Physical Systems, IEEE Transactions on Industrial Informatics, **vol. 18**, no. 1, Jan. 2022, pp. 498–508.

[4]. *L. Monostori*, Cyber-physical production systems: Roots from manufacturing science and technology, At-Automatisierungstechnik, **vol. 63**, no. 10, Oct. 2015, pp. 766–776.

[5]. *F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee*, Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison, Engineering, **vol. 5**, no. 4, Aug. 2019, pp. 653–661.

[6]. *A. Jaber*, Robot Hardware, Transmission Faults and Data Acquisition, in Design of an Intelligent Embedded System for Condition Monitoring of an Industrial Robot, 2017, pp. 93–124.

[7]. *M. Soualhi, K. T. P. Nguyen, K. Medjahe, D. Lebel, and D. Cazaban*, Intelligent monitoring of multi-axis robots for online diagnostics of unknown arm deviations, Journal of Intelligent Manufacturing, 2022.

[8]. *C. Zhao and J. Li*, Working Condition Monitoring System of Substation Robot Based on Video Monitoring, Wireless Communications and Mobile Computing, **vol. 2022**, 2022.

[9]. *E. Wescoat, S. Kerner, and L. Mears*, A comparative study of different algorithms using contrived failure data to detect robot anomalies, Procedia Computer Science, **vol. 200**, 2022, pp. 669–678, 2022.

[10] “RESTful API”, https://restfulapi.net/ (accessed Sep. 20, 2022).

[11] “ABB Robot Web Services”, https://developercenter.robotstudio.com/api/rwsApi/ (accessed Sep. 18, 2022).

[12] “Bootstrap v4.6”, https://getbootstrap.com/docs/4.6/getting-started/introduction/ (accessed Sep. 20, 2022).

[13] “Highcharts”, https://www.highcharts.com/docs/index (accessed Sep. 18, 2022).