Contents lists available at ScienceDirect

Heliyon

www.elsevier.com/locate/heliyon

Research article

Flexible robotic teleoperation architecture for intelligent oil fields

Gustavo Caiza [a], Carlos A. Garcia [b], Jose E. Naranjo [b], Marcelo V. Garcia [b,c,*]

[a] *Electronic Engineering, Universidad Politecnica Salesiana (UPS), 170146, Quito, Ecuador*
[b] *Department of Systems, Electronics and Industrial Engineering, Universidad Tecnica de Ambato (UTA), 180103, Ambato, Ecuador*
[c] *Department of Systems Engineering and Automation, Basque Country University (UPV/EHU), 48013, Bilbao, Spain*

ARTICLE INFO

Keywords:
Computer science
Electrical engineering
Systems engineering
Automation
Computer engineering
Robotics
Electromechanics
Petroleum industry
IEC-61499
MQTT protocol
Robot operating system (ROS)
Teleoperation
Oil &amp
Gas process

ABSTRACT

Teleoperation virtual platforms allow people to send their skills and capacities into machines located in either relative close (few meters away) or far (different continents) locations. With the use of lightweight protocols, people can remotely control the actions and movements of robots so they can avoid physical interaction with dangerous or risky places. Oil and gas well-pads stations are working zones considered hazardous due to the various chemical substances used in their daily processes. This characteristic makes these places the perfect candidates for the implementation of teleoperation solutions in order to reduce the direct interaction of humans with different chemicals and risky situations. The following investigation focuses on the development of a base teleoperation scheme to perform inspection and maintenance tasks in the inside one of these hydrocarbon facilities. The proposed system aims to generate an easily scalable teleoperation solution using distributed control schemes and a lightweight communication protocol to remotely manipulate a KUKA mobile manipulator. As the first stage of this investigation, the main result focuses on the development of the generic control and communication functions that allow the physical testing of the system using a KUKA YouBOT mobile manipulator and the help of a qualified operator of the station.

1. Introduction

Oil well-pad stations in Ecuador are complex facilities conformed by a numerous amount of infrastructural elements, machinery and control devices located in biodiversity hotspots. This demands to the maintenance workers to constantly perform inspection and maintenance tasks in order to maintain the production needs while reducing at the minimum the risk of failures which can severely damage the environment, the operators or the machinery and devices related to the station. To lessen the environmental impact caused by the building of this type of process oil station and their subsequent growth, its well-pads, buildings, and warehouses for machinery is constructed in reduced spaces. This results in greater difficulty for operators when they move through the facilities to perform inspection and maintenance tasks, that leads to a waste of time, effort and resources.

Daily oil production needs direct the industrial paradigm into a route where faster workflows and data exchange times are the main objectives. With this in mind, new industrial technologies and control schemes require a greater commitment of the station's workers for asset management [1]. One of the new concepts used to reach the objectives set the Industrial Internet of Things (IIoT) [2]. It is mainly referred to as an industrial framework where a large number of machines and intelligent devices are connected and synchronized through the use of software tools and specific hardware technologies [3]. Its implementation benefits the production scheme through the reduction of human errors, the increase in the overall efficiency and the cost of the final product in terms of time and money. The goal of the IIoT is not to fully replace human workers. Its final objective is to enhance and optimize it by reducing the amount of trivial and repetitive tasks performed by humans inside a manufacturing plant.

IEC-61499 takes advantage of the advanced features of intelligent devices to generate generic and modular automation elements meant to be easily implemented even in IEC-61131 control schemes. One of its main features is the generation of software elements using high-level programming languages. This allows a simple integration of complex algorithms or custom made libraries in the codification of the control functions.

* Corresponding author at: Department of Systems, Electronics and Industrial Engineering, Universidad Tecnica de Ambato (UTA), 180103, Ambato, Ecuador.
E-mail addresses: gcaiza@ups.edu.ec (G. Caiza), ca.garcia@uta.edu.ec (C.A. Garcia), jnaranjo0463@uta.edu.ec (J.E. Naranjo), mv.garcia@uta.edu.ec, mgarcia294@ehu.eus (M.V. Garcia).

https://doi.org/10.1016/j.heliyon.2020.e03833
Received 18 July 2019; Received in revised form 15 January 2020; Accepted 21 April 2020

The handling of large amounts of data generates an important concern surrounding the interoperability of the IIoT, the required speed of communications between devices and the number of resources they will take. The Message Queuing Telemetry Transport (MQTT) data transfer protocol is based on publications and subscriptions to the so-called "topics", whose low consumption of resources and speed of response make it suitable for the communication needs of the IIoT [4]. The methodologies of this protocol allow the data of an industrial system to be available to all the devices from the system and to the entire enterprise, without straining data bandwidth.

Intelligent Oil Fields (IOF) are the result of the integration of these new technological advances into the petroleum industry. They allow the monitoring, analysis and control procedures to be done in real time, achieving this way optimal management of deposits [5, 6]. In IOFs it is possible to visualize the totality of the operations that are being carried out, besides instantly accessing key data about assets, measures, and documents coherent and intuitive formats. This information provides a comprehensive and unified process vision, meant to be used in the decision making. Within this concept, the environment is proactive and oriented to action through alerts and events, allowing them to be modified as they occur [7]. The most representative advantages of IOFs are the effective management of resources, the improvement of production, the reduction of errors and the improved decision times.

The ease of IOFs to send and receive large amounts of information generates the possibility of integrating a wider range of technologies into oil companies. The following research work uses as base the guidelines depicted in the standard for distributed control applications IEC-61499 to generate a teleoperation system. It allows a human operator to interact with a virtual platform and remotely control in real-time a mobile robot to develop industrial assessment and maintenance duties in an oil extraction station. The main goal of the final product of this research project is to reduce the amount of time invested in the aforementioned operations caused by the difficulty which a human-sized being has while moving around the station facilities to reach the control and measurement elements. An indirect resulting benefit of this system is the reduction of labor risks to operators by carrying out their activities remotely.

The layout of the paper is as follows: Section 2 depicts some related previous academic literature, Section 3 describes some of the concepts used in this investigation, Section 4 illustrates the case study, Section 5 details the functions, applications, systems and virtual environments generated; Section 6 presents some results of the use of the proposed system, and finally in Section 7 some conclusions and ongoing work are discussed.

2. Related works

The aim of this section is to show how other research deal with the design of IEC-61499 applications in Cyber-physical Production Systems (CPPS) using Virtual Reality, Automation for Robots or Tele-operation for Oil & Gas process. In this sense, a set of related works, paradigms and development technologies, which provide agile and flexible architectures and CPPS are presented.

In the current industry, efficiency and high-performance turn robots into the best choice to do high accuracy tasks in industrial environments. The fact that they can be considered as "easily replaceable elements" makes them highly fit to perform tasks with hazards for workers' integrity. Robots in these times, have several applications in the industrial field. An example of the use of industrial robots in risky activities is addressed in [8], in that research, the automation system made by robots for steel beam manufacture in building constructions is developed. In contrast, an application where robots are used in high precision activities is presented in [9], where robots are used in the grinding and polishing process of convex and aspheric mirrors. The authors of the research depicted in [10] show a complex architecture with the aim of the develop a pick-and-sort application based on motion planning and real-time vision working together, allowing that a robotic arm picks square objects from a conveyor belt.

The growing industrial needs require more modular and flexible systems. This phenomenon results in a development of industrial solutions focused on the coordinated work of multiple devices. Evidence of this type of investigations is shown in [11], where the processes of: welding, transporting and polishing are done with robotic modules fully connected and capable of sharing information with any device of the system. This work focuses on the concepts of Smart Factories and Industry 4.0, characterized by the high efficiency and easy customization brought by the interconnection of the systems in one industrial network. The investigation presented in [12], illustrates how improved communications give to the system a general methodology for generating paths in mobile robot navigation applications. The upgrade in data exchange gives an advantage for control all elements that works in a industrial system. The authors in [13] presents an architecture scheme to interact between robots and humans. This proposal is based on the generation of a virtual environment capable of getting the data of the production system and accurately replicate the physical behavior of the robots in the virtual world.

Previous research works of this authors have contributed in recent years to the growth of the industrial scalability of the IEC-61499 standard. In [14] the researchers show a distributed architecture for control system integration used in the batch industrial process developing with the embedded low-cost board as controllers. Another contribution is given in [15], where an automation architecture for CPPS based on Unified Modeling Language (UML) for industrial environments is showed. One of the key factors involved in the rapid development and industrial reception of IEC-61499 standard is its potential of generating generic Function Blocks (FBs) using high-level languages for programming. For example in [16] and [17], where an Model Predictive Control (MPC) and an advanced Fuzzy control are integrated into the Function Blocks (FBs) codification.

This paper takes as reference the previous articles presented by the authors. In [18, 19] the researches show how the IEC-61499 standard is used in low-cost boards to develop a control scheme for the Kuka Robotic Arm. In these papers, the Robotic Arm is used for welding tasks and pick and place process. The results of using the FORTE as IEC-61499 Runtime states the high level of precision while developing the different industrial processes. This paper goes beyond because implementing an MQTT communication protocol transmisión between Kuka youBot and a human operator into a virtual environment with the aim to set predictive maintenance actions in the Oil & Gas Industry.

Throughout the years, the programming required in the robot's operation has been improved to the point of generating smoother and more precise movements. One of the alternatives preferred by many researchers is using the kinematic equations. The research showed in [20] presents a controller based on the SCARA robot kinematics equations, integrating this robot in a distributed winding process. Similarly, in [21] the investigation states a switching algorithm between kinematic control architectures for the robotic arm operation. These algorithms are generally developed in high-level programming, making them fully compatible with IEC-61499. As an example, the research showed in [22] introduces a robotic control compatible with the IEC-61499 standard. The authors in [23] show a robotic cell control using the IEC-61499 under the Robot Operating System (ROS) architecture is created. These works demonstrate the success of integrating the control and information of robotic elements under this standard.

With teleoperation, robots quit performing following predefined actions and become physical intermediates that interact with the environment according the control in real time of a human operator. It allows humans to perform activities in extremely adverse environments without direct physical interaction. An example of this is the spatial application depicted in [24], where the authors detail the teleoperation architecture adopted by the investigators to remotely control the Korean Lunar Rover. Depending on how well the teleoperation system is struc-

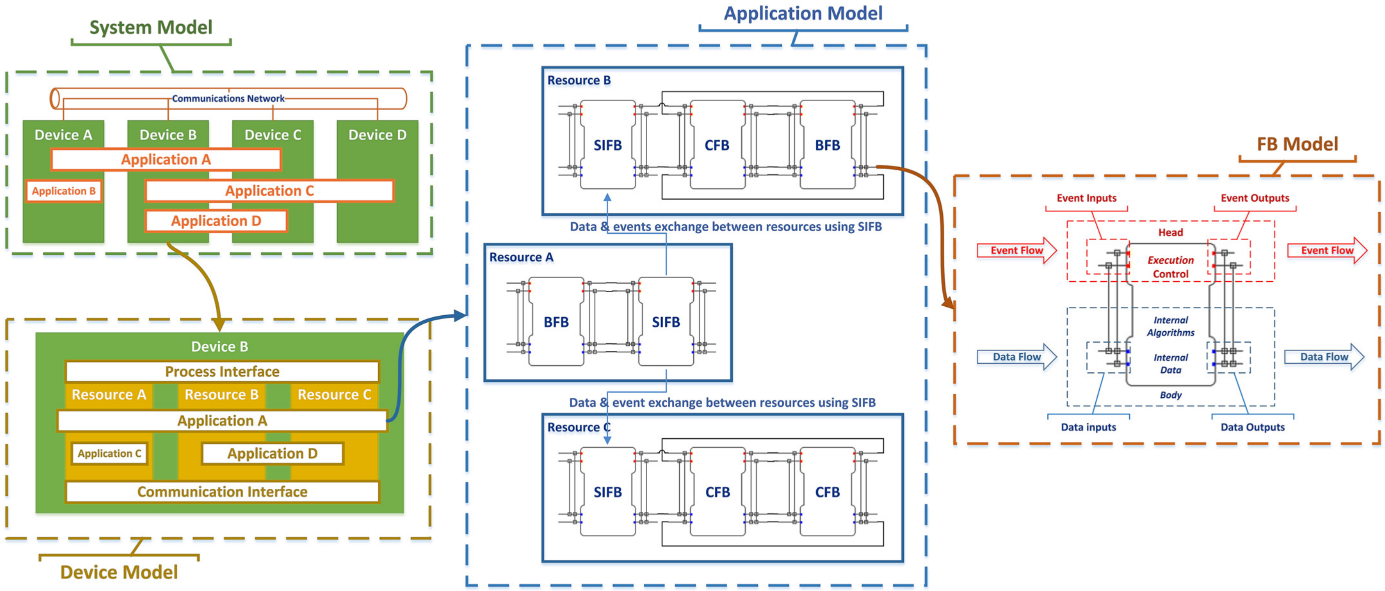

Fig. 1. IEC-61499 Architecture Models: System Model, Device Model, Application Model and Function Block Model.

tured, it can allow humans to do activities which require a high level of accuracy from far distances. The work presented in [25], presents a teleoperation system for needle steering aiming to be implemented in human surgeries. As time goes by, teleoperation can be implemented in more local applications such as the industrial field. In [26] the authors present a teleoperation system for the operation of an industrial robotic arm in manufacturing processes. With all these backgrounds as starting point, this work aims to develop even further the proposal depicted in [27], and generate a teleoperation system compatible with the new industrial standard IEC-61499 to perform maintenance and inspection activities in oil extraction stations.

In this paper the integration of a flexible architecture for Industrial Robot Tele-operation and IEC-61499 Service Interface Function Blocks (SIFBs), with MQTT protocol using inside ROS, is addressed enabling an easy and fast building of this kind of automation systems.

3. Technical background

3.1. *Intelligent oil fields*

To keep pace with the demanding technological world we live in, companies operating in the field of oil and gas must develop a complete set of hybrid skills where production processes can interact with information technologies [28]. Due to these needs, the IOFs are born, as an extension of the commercial capacities where the leadership can manage the value chain and not only the equipment. New possibilities are presented through large production environments with multiple assets. The intelligent oil field allows production optimization in all sites and consents the collaborative management of manufacturing performance. It allows assets to operate closer to their optimum point by improving business processes, including real-time visibility and process collaboration [29].

IOF is known by many names, including “digital oilfield”, “field of the future” and “real-time optimization”. IOF shows great promise for a future of greater productivity, greater recovery, lower costs and less exposure to health, safety and the environment [30].

3.2. *IEC-61499 standard*

IEC-61499 is an automation-oriented standard developed to overcome the limitations shown by its standard predecessor IEC-61131 while facing the challenges of the new industrial paradigm. They can be summarized in the need of manufacturing enterprises to generate and implement flexible, modular and custom control systems. The IEC-61499 standard depicts a solution based on a distributed scheme to provide its systems the characteristics of [31]:

- **Portability.-** The capacity of accepting and correctly execute systems developed using other software tools compatible with the standard.
- **Reusability.-** The capacity of developing generic functions and applications suitable for multiple systems.
- **Interoperability.-** The capacity of integrating multi-brand hardware solutions in the same production system.
- **Configurability.-** The capacity of easily performing software modifications and even on the run modifications.

As shown in Fig. 1, the IEC-61499 standard describes a series of generic models for developing distributed control systems. They specify communication networks and process networks as the environment for intelligent devices, resources, and applications. The upper level of the standard is the System Model. It states that a distributed application can operate in a single device or have its operation distributed in many devices, up to the point where it is difficult to define which is the main controller.

The Device Model depicted in the standard presents a process interface, and a communications interface. The first one provides the services that enable the exchange of data between resources, while the second one provides communications services that allow the resources of different devices to exchange data between them. In this model, a device can support one or more resources, providing independent execution and control of networks of function blocks [32].

Distributed applications are defined in the Application model as a set of interconnected Function Blocks (FBs) linked by event and data connectors, whose objective is to solve a particular automation control problem. All the application blocks can be mapped to the different devices defined in the system configuration. Therefore when the application is deployed, it will be downloaded as a whole in a single device or as individual functions in different devices [33].

The core of the IEC-61499 standard is the Function Block (FB) Model introduced for the first time in its predecessor. However, it improves the execution of functions and data exchange by adding event handling mechanisms to the FBs. IEC-61499 defines the FB as the core software element needed for the development of control applications.

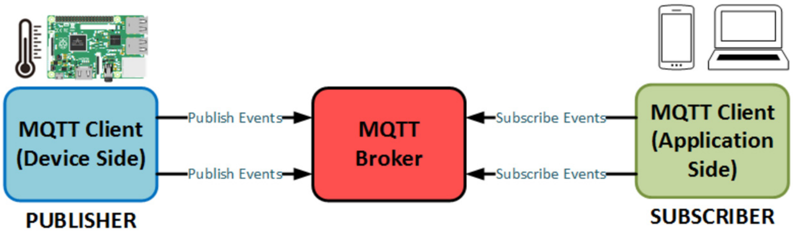

Fig. 2. MQTT Protocol scheme.

It has its own data structure and is used to encapsulate programming algorithms related to: basic operations (i.e. mathematical and Boolean operations, string manipulation), specific functions (i.e. configuration and manipulations of hardware resources), communications (i.e. channel configuration, object setup, message creation), etc [34].

As shown in Fig. 1, each FB is divided into two fundamental parts. Its upper part is called "Head" or "Execution Control Chart". It contains definitions (usually given in terms of a state machine) which map events with algorithms and define the order of execution in case of multiple event arrival. The lower part is called "Body". It manages the data flow and contains the algorithms and internal needed for the operation of the block.

In the IEC-61499 several FB types are defined, all of which can contain a behavior description in terms of service sequences:

- **Basic Function Blocks (BFBs)**.- Its functionality is described in terms of an Execution Control Chart (ECC), where every state can be related with several actions.
- **Composite Function Blocks (CFBs)**.- Its functionality is defined by a network of different types of FBs which can include other CFBs. Sub-applications are included in this type.
- **Service Interface Function Blocks (SIFBs)**.- This type of FBs provides an interface between the function block domain and external services (i.e. read of the current value of a hardware data input).
- **Adapter Interfaces**.- This type of block is not a real FB. However, it provides an interface concept to separate specification and implementation in the systems.

The development of the FBs is carried out in two stages. First is the "Generation of the External Structure" where an Integrated Development Environment (IDE) compatible with the standard is required. In this stage, the type name, the instance name and the set of data and event inputs and outputs are defined. The final product at the end of this procedure is an empty block structure which programming only includes the declaration and base configuration of its inputs and outputs [34].

The second stage is the "Programming Integration". In this stage, all the algorithms related to the FB operation are integrated into the block source files. The codification can be done using any of the programming languages specified in the standard guidelines which include the languages listed in the IEC-61131 standard and high-level languages for programming like Java, C++, Python, Java, etc.

The same IDE used in the creation of the blocks is used to generate the distributed control system by: defining the system model, specifying the interconnected devices in the system, interconnecting the FBs into distributed applications, mapping the applications to the necessary resources, and deploying all of the configurations into the physical system. In order for an intelligent device to execute the application downloaded in it, it is necessary to install a Runtime Environment in it. The operative system capable of executing the algorithms specified in the FBs of the system in accordance with the IEC-61499 guidelines.

3.3. *MQTT protocol*

Message Queuing Telemetry Transport (MQTT) is a Internet of Things (IoT) and Machine to Machine (M2M) lightweight protocol that works over the TCP/IP protocol. Its low consumption of resources makes it optimal for being implemented in networks with high latency and connections where a small code footprint is required. The MQTT architecture uses the publisher/subscriber model in combination with a client/server schema to accomplish the exchange of data between the interconnected devices. The elements of this architecture are: the publisher clients, the subscriber clients, and an intermediate message management server called "Broker"; all of them interconnected using a star topology [35]. (See Fig. 2.)

Inside the Broker, the transmitted information is stored in hierarchical topics with representations in the form of system file paths i.e. "block1 /extraction /sensors/PIT101". During the MQTT communication process, the publishers send messages to the available topics in the broker. Then, the broker assigns the information to all the clients subscribed to the topic with new data available. The use of an intermediate server eliminates the need for content negotiation mechanisms, turning MQTT into a protocol designed for one-to-many and many-to-many asynchronous communications that focuses on reliable messaging [36].

Despite the simplicity of its communication process, MQTT includes several security features in terms of authentication, data confidentiality, and authorization. There are two main methods of authentication in the protocol. The first is a simple user/password scheme implemented during the initial connection of the clients with the broker. The second method involves the use of a unique client identifier registered in the broker when establishing the first connection. This client identification can be formed up to a combination of 65535 characters and is given by the serial number of the device of by its MAC address [37].

The payload of MQTT messages can be encrypted to secure the data confidentiality. The encryption can be done either using an end-to-end or a client-to-broker scheme. In the end-to-end model, the publisher encrypts the message payload before sending the message to the topic. Then the message is distributed to all the subscribed clients, but only the client with the proper key is able to recover the original message. On the other hand, in the client-to-broker schema, the encryption is done between clients and the broker. To obtain this behavior, a custom developed broker plugin is used to decode the encrypted data before the message is distributed to the subscribers.

In MQTT the authorization management is done with the implementation of an Access Control Lists (ACLs) in the broker. Inside the ACL it's defined all the permissions for users and system processes to gain access to objects and their respective operations. The list also stores all the pairs of usernames and passwords available for a client and the information about the topics that a client can interact with. In the need for additional authorization management, enhanced features can be implemented in the broker in the form of plugins or web services [38].

3.4. *Kuka youBot™ mobile manipulator*

As automation grows in the current industry, companies see the need of integrating more robotic elements in their assembly lines to

Fig. 3. KUKA youBot™ mobile manipulator 3D model.

catch up with the market demand. Mobile manipulators are robots developed combining the manipulation skills of a robotic arm and the movement abilities of a robotic platform in an individual device. This design permits this kind of robots to be easily integrated with teleoperation implementation where a robot perform its movements in a large geographical area [39].

The KUKA youBot™, is a educational robotic platform which versatility makes it suitable for research/development, educational and industrial applications (see Fig. 3). It can be divided in three main parts: i) The KUKA youBot™ onmi-directional mobile platform, formed by the robot's structure, a set of four mecanum wheels with their servo-motors, a power driver, and an onboard computer into the omnidirectional mobile platform running a Linux distribution. ii) A five degrees of freedom KUKA™ robotic arm compatible with the EtherCAT industrial communication protocol. iii) Replaceable end effector which can be selected among a variety of options depending the availability [40].

A distribution of Robot Operating System (ROS) is installed in the youBot™ operative system. The operative system supplies the software resources like (i) a programming layer to interact with youBot hardware called hardware abstraction layer (HAL), (ii) drivers for all the components, (iii) a set of libraries, visualizers, message-passing, etc. needed to send messages with its physical components [41].

4. Case study

Oil extraction stations (usually referred to as "blocks") are physically distributed into several production areas where the tasks of extraction, transportation, and treatment of petroleum are performed. In these blocks, the measurement and control hardware is located in areas that naturally present several risks to the integrity of the operators. Despite the dangers, monitoring and maintenance tasks must be periodically performed in these blocks to secure an optimal operation and reduce at the minimum the occurrence of accidents.

The Ecuadorian enterprise Petroamazonas EP focuses its activities on petroleum exploration, extraction, and processing. It manages 21 blocks located in the Amazonian and coastal areas of Ecuador [42]. One of these blocks is called "Block 18" is located in Oriente Basin, the eastern part of Ecuador, which covers an area of 1,138 km^2. In this block currently 30 oil wells use bottom sensors to collect pressure, temperature and vibration measurements. As a result of the great amount of measurement and control elements, daily inspection and weekly maintenance tasks are programmed in the station to keep an optimal operation of the systems. In this type of scenario, the implementation of a teleoperation solution drastically decreases the possible risks to humans. This is done by transmitting in real time the abilities of an operator into a mobile robot manipulator.

5. Proposal implementation

An approach of this proposed implementation was presented previously in [43]. The tele-operated system proposed allows an experienced field worker to send his capabilities and skills to a robot mobile platform, as shown in Fig. 4. Following the human directions, the robot performs maintenance and inspection activities in adverse environments located far from the operator's location. This investigation proposes the use of the mobile manipulator KUKA youBot as the robotic actuator of the teleoperated system under IEC-61499.

The final product is divided into two main components. The first consists in a set of FBs designed under IEC-61499 to be in charge of: the control of the Kuka youBot hardware components, the manipulation of the motors of the joints, the reception of the audio and video feedback, and the data exchange through the system using the MQTT protocol. The second is a Human-Robot Interface (HRI) used to immerse the operator into a virtual environment (VE). The VE depicts closer to reality the distribution of the Oil and gas Block 18 and presents the audio and video feedback from the real location (obtained through MQTT) to the user in a friendly way.

5.1. *Robot manipulation FBs*

This group of FBs integrate ROS messages in their execution to configure and control the operation of the KUKA youBot. ROS is a component-based software framework for robotic applications thematically organized in stacks of packages. During the execution of the application, ROS nodes exchange information with each other using messages and services. The formats of these messages are defined in specific configuration text files and can be treated as standardized data types in a ROS environment. (See Fig. 5.)

Incorporating the youBot API wrapper into the source code of the developed FBs provides a mapping between ROS messages and the youBot driver method calls. This allows the interaction between the blocks and the logical resources of the robot. The youBot driver stack contains C ++ classes that grant low-level access to applications by means of: (1) the EtherCAT protocol driver management and (2) the API of the KUKA youBot. The EtherCAT driver used by ROS is built into the open source EtherCAT Master (SOEM) operating structure (located in the processing device). This allows the device to communicate at the lowest level and control all the actuators of youBot. Applying the operations given by the first layer of the stack, as shown in Fig. 5, the group of C++ classes of the youBot API implements a new range of operations that can be run to control the platform and arm kinematics and joint's manipulation.

The Function Blocks presented in Fig. 6 contain the algorithms used for manipulating the robotic arm and the mobile manipulator of the KUKA youBot. For their operations these FBs receive and generate events related to:

- The initialization of the block **INIT (input) & INITO (output)**
- The execution of a work cycle of the block **REQ (input) & CNF (output)**
- Errors encountered in the execution **EMR (input) & ERR (output)**

The four FBs of the set transmit a string value containing their "STATUS" information. This parameter comes useful while implementing additional system execution controls. The YouBotArm_READ and YouBotArm_WRITE functions (shown in Fig. 6a and Fig. 6b) are designed to operate with data related to each of the five joints and the gripper. The data inputs and outputs used by these two blocks are:

- **STATUS (STRING, output)**.- Give back a string showing the state of the FB. Possible values: initialized, operating, and stopped.
- **SOURCE (STRING, input)**.- String parameter used in the YouBotArm_READ function to specify the sensor from which the information is read. The available options are angle or current.
- **JR1… JR5 (REAL, outputs)**.- Data outputs from the YouBotArm_READ function that return the values of the joint sensors. These outputs are formatted as real type data.

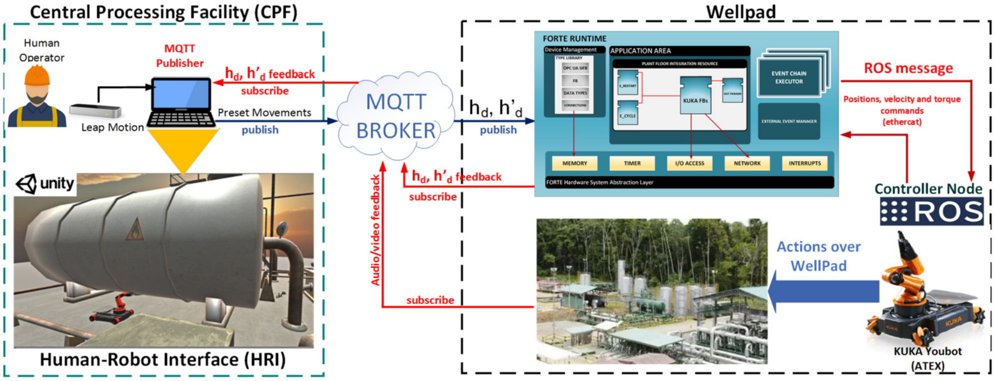

Fig. 4. Mobile Manipulator's Teleoperation Scheme implemented for Oil and Gas well-pads.

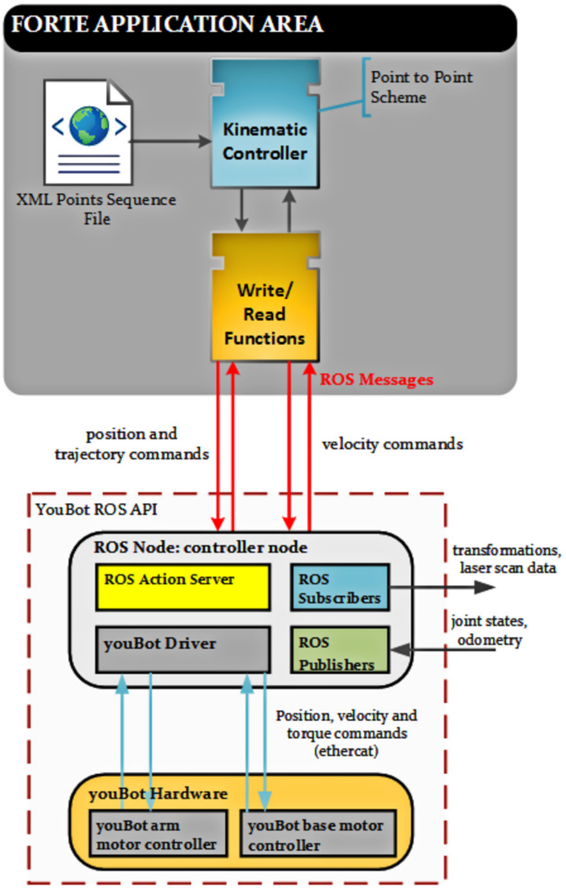

Fig. 5. IEC-61499 FB message communication scheme to ROS.

- **GR (REAL, output)**.- Data output from the YouBotArm_READ function which returns the gripper opening length value. This output is formatted as real type data.
- **JOINT_1... JOINT_5 (REAL, inputs)**.- Data inputs used by the YouBotArm_WRITE function. In them, it is specified the real type values of the desired joint angles
- **GRIPPER (REAL, input)**.- Data input used by the YouBotArm_WRITE function. In it, it is specified the real type value of the desired gripper opening length (in millimeters).

The YouBotBase_READ and YouBotBase_WRITE functions (shown in Fig. 6c and Fig. 6d) are designed to operate with data related to the three velocities which intervene in the movements of the platform. The data inputs and outputs used by these two blocks are:

- **LVR (REAL, output)**.- Data output used by the YouBotBase_READ function. Returns the actual value of the longitudinal velocity of the mobile platform.
- **TVR (REAL, output)**.- Data output used by the YouBotBase_READ function. Returns the actual value of the transversal velocity of the mobile platform.
- **AVR (REAL, output)**.- Data output used by the YouBotBase_READ function. Returns the actual value of the angular velocity of the mobile platform.
- **LONG_V (REAL, input)**.- Data output used by the YouBotBase_WRITE function. Used to set the new longitudinal velocity of the straight moments of the robot.
- **TRANS_V (REAL, input)**.- Data output used by the YouBotBase_WRITE function. Used to set the new transverse speed of the lateral movements of the robot.
- **ANG_V (REAL, input)**.- Data output used by the YouBotBase_WRITE function. Used to set the new angular velocity of the rotational movements of the robot.

5.2. Control FB

The control strategy was presented previously in [18, 19]. This strategy is supported on the kinematic equations of the Kuka™ robotic arm. The model representation of the KUKA youBot Arm configuration returns the position of the effector represented as h. With its operational coordinates represented as function of the general coordinates of the arm, it can also be described as presented in the Eq. (1):

$$f : \mathcal{N}_b \to \mu; (q_b) \to h = f(q_b) \tag{1}$$

In (1) $\mathcal{N}_b$ depicts the spatial pattern configuration.

Hence, the position's derivate of the gripper is calculated as the function's derivate of the KUKA youBot arm configuration is the output of the instantaneous kinematic model, as it can be seen in the Eq. (2):

$$\dot{\boldsymbol{h}}(t) = \boldsymbol{J}(q)\,\boldsymbol{v}(t) \tag{2}$$

Where the vector $\dot{\boldsymbol{h}} = \begin{bmatrix} \dot{h}_1 & \dot{h}_2 & \dots & \dot{h}_n \end{bmatrix}^T$ presents the speed of the end effector, and the vector $\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix}^T$ stores the velocities of the arm joints. The dimension m of the vector $\boldsymbol{v}$ is equal to the number of links of the robotic arm.

The Jacobian matrix presented in the Eq. (3):

$$\boldsymbol{J}(q) = \frac{\partial f(q)}{\partial q}\boldsymbol{T}(q) \tag{3}$$

defines the linear mapping between the gripper speed of the arm $\dot{\boldsymbol{h}}(t)$. And the velocity vectors of the robotic arm $v(t)$.$\boldsymbol{T}(q)$ is the transformation matrix that relates the joint velocities $\dot{q}(t)$ with the robotic arm velocities $v(t)$, resulting in the Eq. (4) which:

$$\dot{q}(t) = \boldsymbol{T}(q)\,\boldsymbol{v}(t). \tag{4}$$

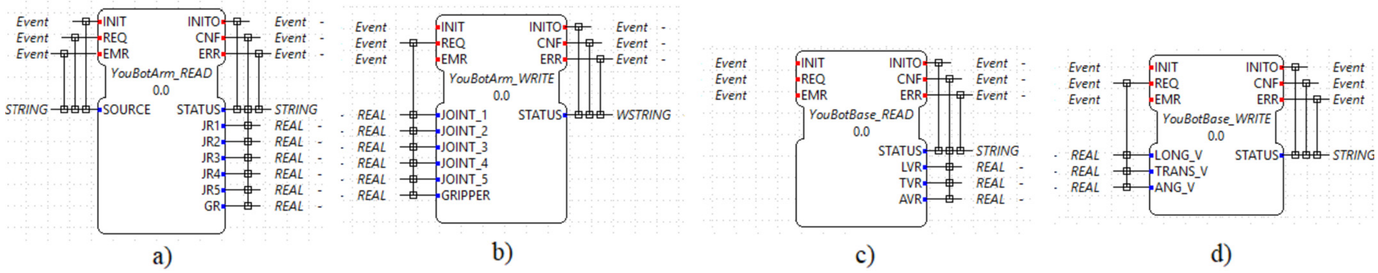

Fig. 6. FBs to send Robot Operating System (ROS) message; a) YouBotArm_READ, b) YouBotArm_WRITE, c) YouBotBase_READ, d) YouBotBase_WRITE.

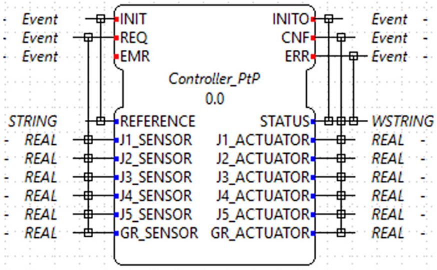

Fig. 7. Controller_PtP FB.

The HRI sends to the controller references containing the desired position and orientation of the end effector of the arm. Taking into account the time difference between references, a Point to Point Controller is implemented to operate the robot. The control algorithm is developed to be stable using the given orientation to the end effector of the youBot to a specified point, where $\boldsymbol{h}_d = \left[\boldsymbol{h}_{dpos}^T \quad \boldsymbol{h}_{dor}^T\right]^T$. Consequently, the speed and the position for arm stability can be calculated using the Equations (5) and (6).

$$\boldsymbol{h}_d(t, s, \boldsymbol{h}) = \boldsymbol{h}_d \tag{5}$$

$$v_{\boldsymbol{h}_d}(t, s, \boldsymbol{h}) = \dot{\boldsymbol{h}}_d \equiv 0 \tag{6}$$

To reach the specific endpoints on the control tasks of the youBot arm is defined as shown in the Eq. (7):

$$v_{ref}(t) = f\left(\tilde{\boldsymbol{h}}(t)\right) \tag{7}$$

Where $\tilde{\boldsymbol{h}}(t) = \boldsymbol{h}_d - \boldsymbol{h}(t)$, outcoming in $\lim_{t\to\infty} \tilde{\boldsymbol{h}}(t) = \mathbf{0}$.

All the discussed algorithms are encapsulated in the CFB presented in Fig. 7. This FB is designed to use as data inputs the information related to the actual values of the joint angles. Then, with these values, it extrapolates the end effector current position and orientation. The resulting data is compared in the control algorithm with the reference data. Finally, the FB returns as outputs, the actions that the joints need to perform in order to reach the desired reference.

5.3. MQTT FBs

These SIFBs incorporate the C++ library corresponding to Eclipse Mosquitto which implements the MQTT protocol versions 3.1 and 3.1.1. Mosquitto is a lightweight alternative and is suitable for use on devices from low power single board computers to full servers. The C++ library incorporates the methods to: initialize and clean up the library, create and eliminate clients, specify authentication and encryption methods, configure wills, connect the created clients to the broker, publish messages, subscribe to a topic, among others.

The final SIFBs created to exchange data using the MQTT protocol are shown in Fig. 8. To avoid redundant initialization of the mosquitto library, the first executed block performs this action and activates a flag that specifies that the "library initialization process" can be omitted during the execution of the rest of MQTT blocks. Like the Robot manipulation FBs, these functions operate with the events: REQ, INIT, INITO, EMR, CNF, and ERR. Following the MQTT operation, the six SIFBs have the HOST_IP and TOPIC inputs. In these inputs, the IP direction of the device where the broker operates and the topic where the data is uploaded and downloaded are specified as string parameters.

Three pairs of MQTT SIFBs allow the transmission and reception of: information of the whole robot using the MQTT_RobotRead and MQTT_RobotWrite SIFBs (shown in Fig. 8a and Fig. 8b), data only from the robotic arm by means of the MQTT_ArmRead and MQTT_ArmWrite (shown in Fig. 8c and Fig. 8d), or data only from the base by using the MQTT_BaseRead and MQTT_BaseWrite SIFBs (shown in Fig. 8e and Fig. 8f). The parameters HOST_IP and TOPIC are mandatory inputs in the six aforementioned blocks. However, the functions can operate without one or more inputs or outputs connected.

5.4. Control reference and feedback CFBs

The final FBs created for the system are the Control Reference and the Feedback CFBs. The first CFB includes in its programming the use of the MQTT_BaseRead and the MQTT_ArmRead functions to exchange information with the HRI. Using these MQTT functions and additional algorithms this FB returns the control reference for the robotic arm controller and the base's displacement. The programming of the second Composite FB includes the use of OpenCV libraries for obtaining real-time video and audio data from the camera mounted in the mobile manipulator. The internal algorithm of the function transmits the audio-visuals of the robot location to the HRI. In the interface, the feedback allows the operator to be aware of if the actions taken by the control had the desired physical result. In this system, the operator is responsible for "calculating the error" between the set paths and the achieved results. Thus, no additional real-time image processing is done by the platform in the recording and transmission processes.

The ControlReference CFB and the Feedback CFB are shown in Fig. 9a and Fig. 9b. Following the development structure used in the generation of the previous FBs, these FBs also operate with the events: INTI, REQ, EMR, INITO, CNF, and ERR. Additionally, they also return the STATUS data output.

6. Discussion of results

The final teleoperation application under IEC-61499 standard implemented using the created FBs is presented in Fig. 10. It is networked and configured so that in a work cycle the actions corresponding to: data exchange with the HRI, read of the robot sensors values, control calculations, and execution of the control actions are performed.

First step is to select at menu screen the functions assigned for each device that includes the following options: check the process in real time, see the device warnings, inspect the risks presented by each element, navigate through a configuration panel, obtain information on the element, and perform maintenance actions (preventive, predictive or corrective). The operator can select the option that best suits his needs through the digitalization of the movement of his hands.

Once the operators begin to interact with the proposed system through the HRI, as shown in Fig. 11, the operators are automatically

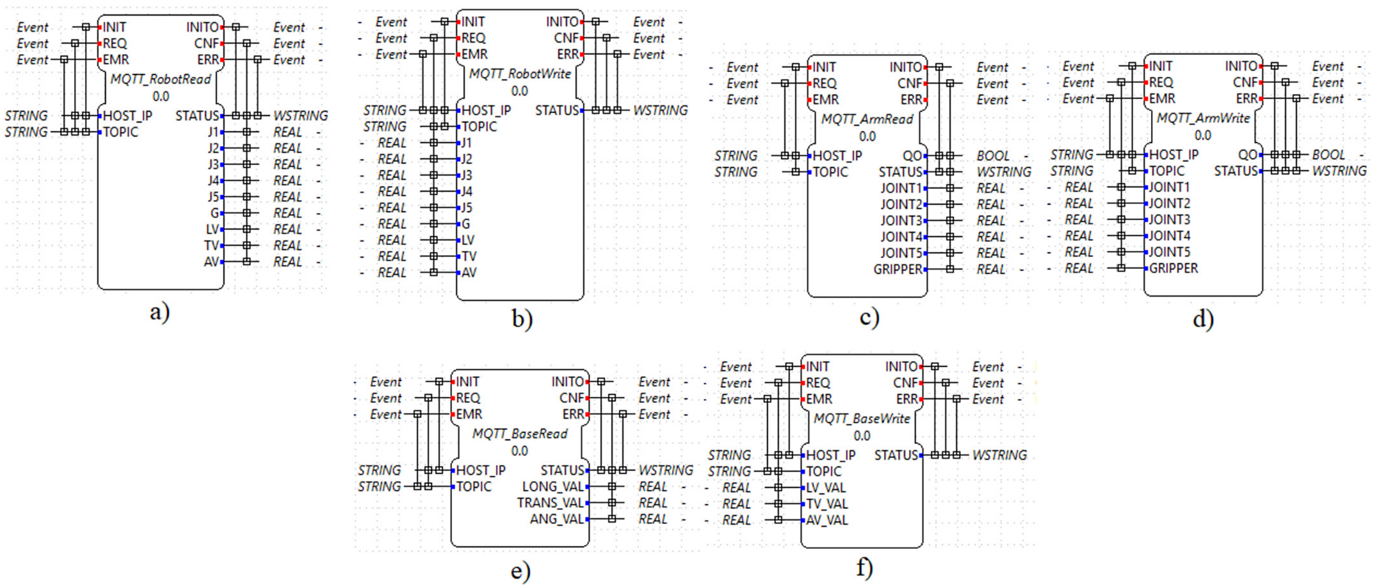

Fig. 8. SIFB for MQTT communication: a) MQTT_RobotRead, b) MQTT_RobotWrite, c) MQTT_ArmRead, d) MQTT_ArmWrite, e) MQTT_BaseRead, f) MQTT_BaseWrite.

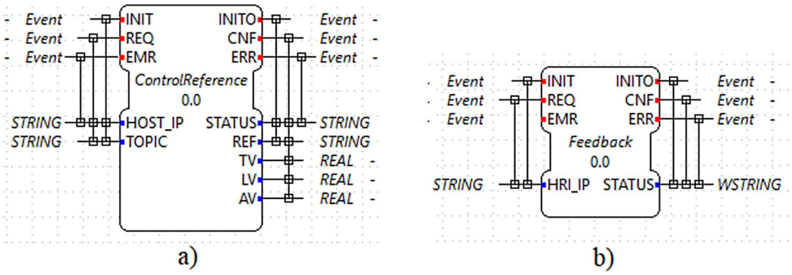

Fig. 9. FBs for Reference and Feedback; a) ControlReference CFB; b) Feedback CFB.

immersed in the real environment of the Block 18 station. Through the Leap Motion sensory device and its sense of kinesthesis, the users can control the movement and handling of the KUKA youBOT robotic arm. As the mobile manipulator moves in the real environment the operators can visualize the information of each measurement and control device. This is done with the menu of options developed in Unity 3D and programmed in VR, in combined work with the USB camera incorporated in the robot.

The combined use of real location feedback and the created VE gives a wider perspective of the robot location to the operators. As seen in Fig. 11, the VE behaves as a side map from which the operators can estimate the best route of operation or detect the possible obstacles in the robot motion. The KUKA youBot™ robot can be seen scouring the surroundings of the virtual oil storage tanks programmed into the Unity 3D graphics engine. In order to obtain better control of the activities to be carried out within the wellpads, a system of location and monitoring of the slave robot has been implemented. In the upper right corner, the wellpad map's is displayed where, with a red dot, the movement of the robot is represented in real time through the facilities. On the other hand, visual feedback of the real environment is presented in the upper left corner. In order for the tasks performed by the slave to be precise, in addition to position control, visual feedback is also needed. This in order to obtain effective control and infinite transparency of the system. In this figure, the slave robot can be observed verifying the pressure contained in the pipes and the state of the pressure valves thus fulfilling its control and maintenance function of oil equipment.

In a teleoperation system, the communication scheme must be able to guarantee an optimal data exchange in periodic and small periods of time. For the first tests of operation, the KUKA YouBot was configured to execute only the communication module of the final application and was placed in the center of the station. Test data was exchanged with the remote desktop that executed the HRI developed during a time period of 30 minutes. The results of the evaluation of the communication scheme of the system are shown in Fig. 12. The data traffic in bits per second that is generated by the publisher/subscriber MQTT communication. A bandwidth that reaches a maximum of 85000 bits/s and a minimum of 3000 bits/s with an average of 44000 bits/s for the whole group of packets captured as a function of time can be seen, i.e., it is the amount of widthband in use when the communication between our HRI and the KUKA youBot™ robot is generated.

In Fig. 13 the number of packages processed in a publication/subscription request is detailed, with a maximum of 74 Packets/s and a minimum of 5 Packets/s, which generate an average of 40 Packets/s for each mentioned time interval. In addition, it can be seen that, despite working in a dynamic oil environment, there is no loss of communication packages. This is because the MQTT protocol uses an openSource broker (Mosquitto) that, due to its lightness, allows us to easily use it in a large number of environments, even if they have few resources. On the other hand, it can also be seen that starting from the second 130 the packets sent per second show a uniform distribution, this occurs since from this time the manipulation with the robotic arm is finished and an automatic homing process begins.

In Fig. 14, the round trip time (RTT) is shown, i.e., it shows the time it takes for a packet of data sent from the publisher to return to this same sender having passed through the destination subscriber. As can be seen, the maximum delay generated in the publication/subscription MQTT process does not exceed the 196 milliseconds, concluding that the response of the system in real time is efficient and that the deci-

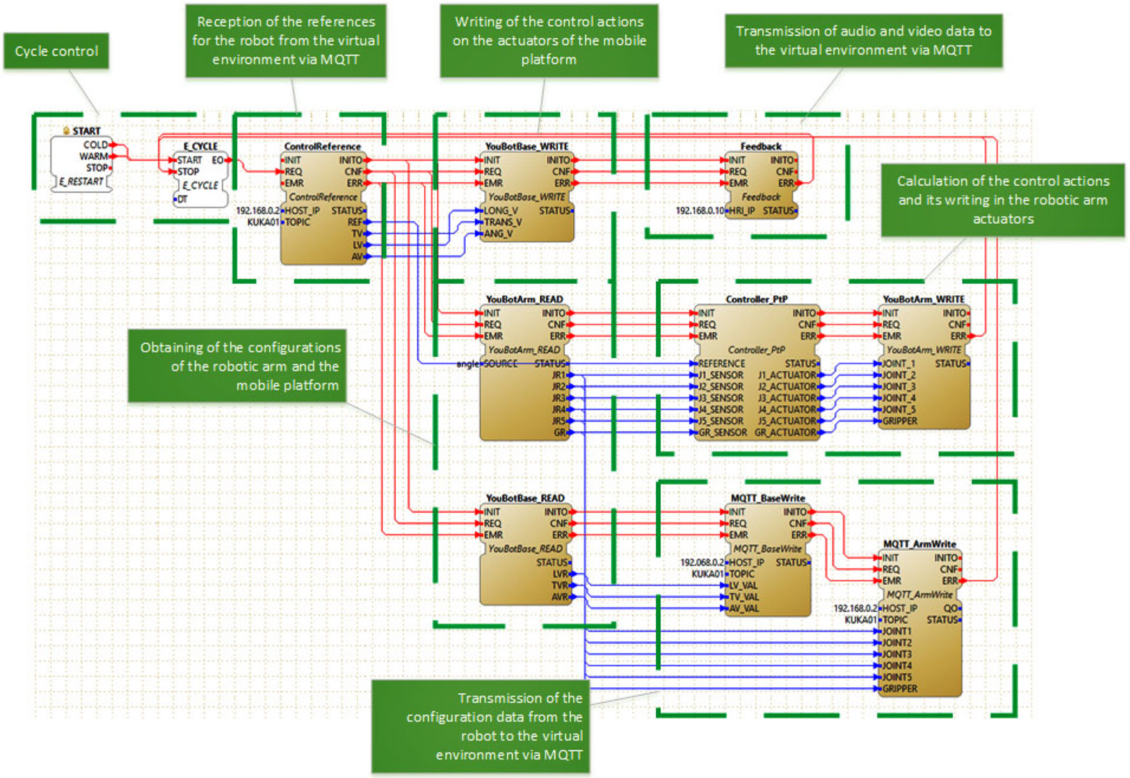

Fig. 10. IEC-61499 Application for Kuka youBot™ teleoperation for Oil& Gas process.

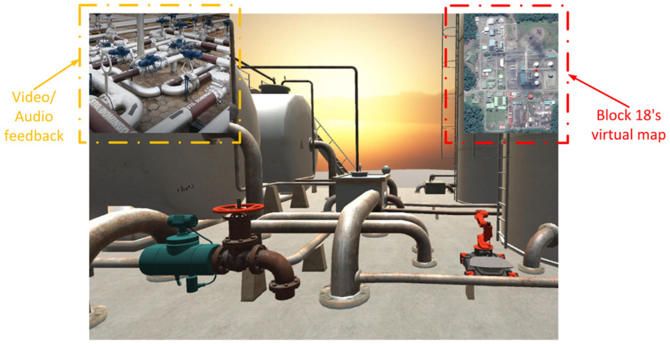

Fig. 11. KUKA youBot™ in Oil&Gas equipment maintenance activities.

sion of using MQTT as a data transmission protocol was accurate. In addition, the generated RTT corroborates the information obtained in Fig. 13, where there is no loss of packages or network congestion.

The simultaneous use of open-loop and closed-loop controllers allowed the resulting system to operate in an easy and simplified manner. The open-loop control scheme is in charge of the moving tasks of the mobile platform. It works in two stages: 1) reception of the three speed references (longitudinal, angular and transversal), and 2) writing of the new values in the main configuration file of the platform. The open loop controller performs these steps every time there is a new reference value available in the Broker. As a safety mechanism, if a disconnection occurs during the operation, the robot application is configured to automatically inhibit the movement of the platform. Correspondingly to the controller of the platform, the VE is programmed to sample the speeds of the virtual representation of the robot and upload its values to the Broker every 150 milliseconds. During the displacement tests, this value was demonstrated enough to allow the operator to move the YouBot through the station with a good degree of accuracy. As shown in Fig. 15, the open loop control scheme can execute a work cycle in an average time of 150.25 milliseconds.

The closed-loop kinematic controller is responsible for the manipulation tasks of the robotic arm. As a result of the implementation of a position error correction based on the location and orientation of the end effector, this control scheme requires a longer period of action. When collecting the new position and orientation references, the VE is programmed to: 1) allow the operator to change the references in relatively small and progressive steps, and 2) sample and load the new references every 400 milliseconds. As in the previous case, tests were carried out to measure the time of the work cycle in the closed-circuit controller. Fig. 16 illustrates how this control scheme can execute a duty cycle in an average time of 400.30 milliseconds. The efficiency of the controller can be verified with the results presented in Fig. 17. In this image, you can see how the algorithm is able to reduce the configuration error of the robotic arm to the minimum in the specified time.

7. Conclusions and ongoing works

The proposed teleoperation system has been validated through several tests conducted at Petroamazonas EP. The process of adaptation of the operators to the proposed architecture has been favorable, demonstrating a great development of their sense of kinetics. The KUKA youBot™ robot presents a great development in industrial environments, being ideal for manipulation and transporting tasks. The precision of the tasks of both end effector positioning and locomotion are high enough to fulfill the orders entrusted in a satisfactory manner.

This automation architecture provides to the field operator the capacity to send their abilities in real-time to the mobile robotic platform. This provides the robot with the knowledge to perform predictive maintenance tasks of all machines that are in the Well-Pad of Petroamazonas's oil extraction company. This virtual platform permits the operator field the execution of the chores in a hazardous environment with limited access, safeguarding the integrity of the personal.

When evaluating the robustness of the communication network of the system, it was taken as a reference to the number of orders sent and the number of actions performed. The efficiency of the system in this

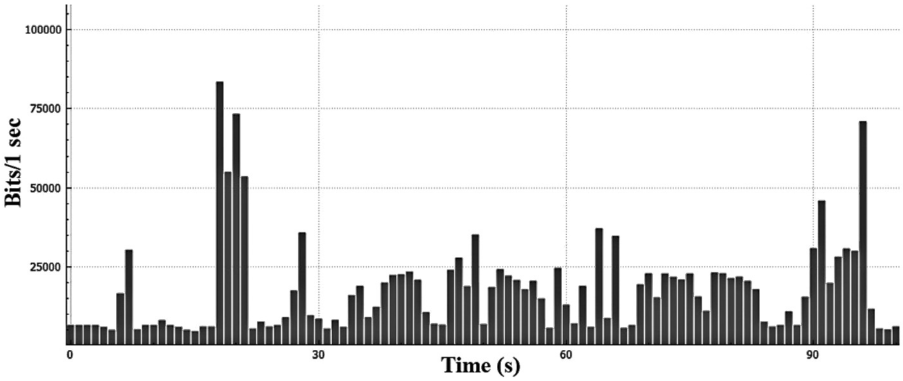

Fig. 12. Bandwidth used in a publisher/subscriber MQTT communication.

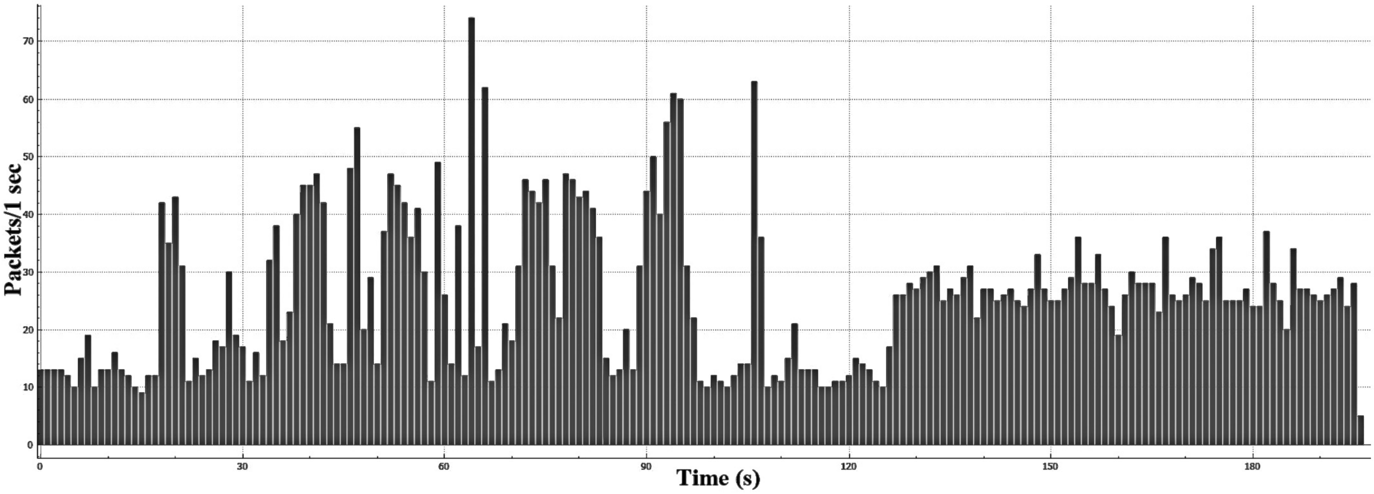

Fig. 13. MQTT packet processing.

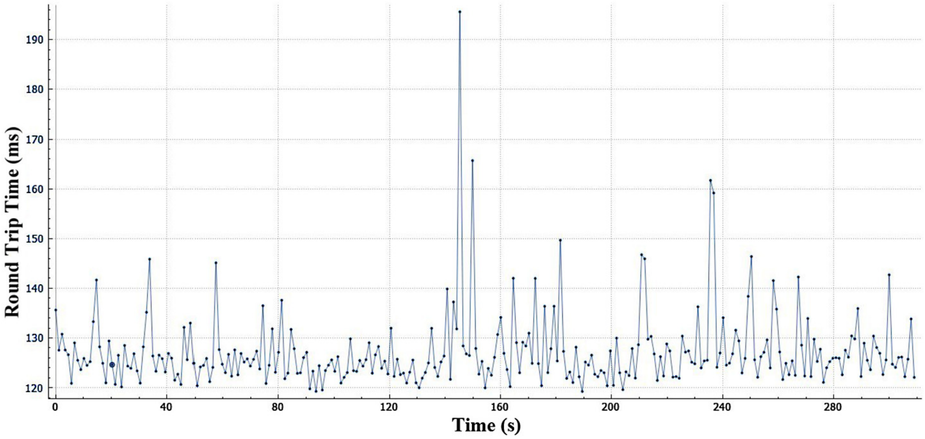

Fig. 14. MQTT round trip performance.

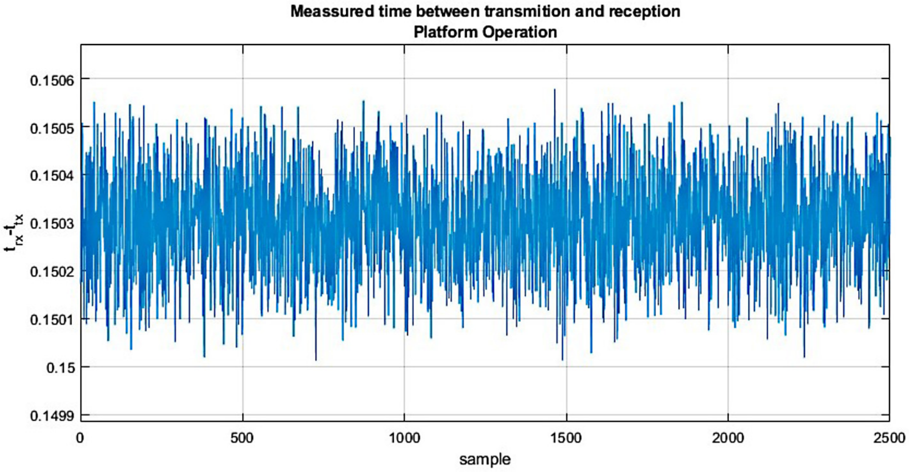

Fig. 15. Measurements of the work cycles of the platform control scheme.

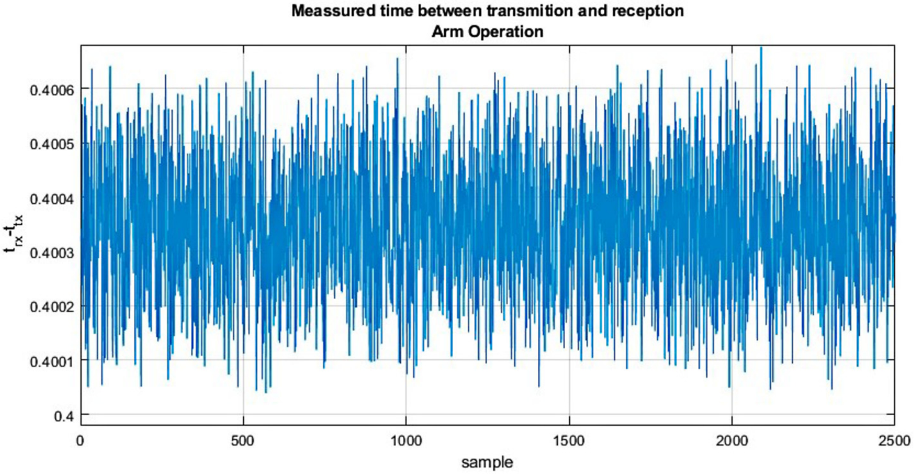

Fig. 16. Measurements of the work cycles of the arm control scheme.

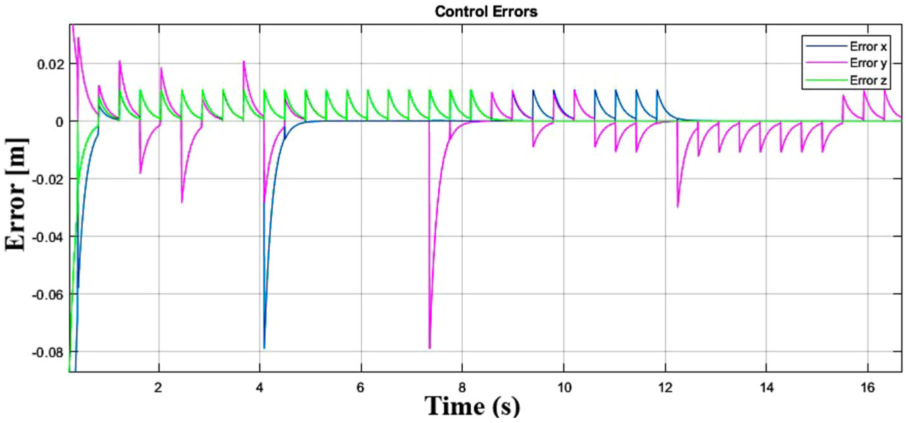

Fig. 17. Progressive correction of the robotic arm configuration errors.

aspect corresponds to 97%. This guarantees a communication link good enough to secure the minimum package lose. MQTT protocol is an efficient and fast communication protocol. The protocol security features make it suitable for industrial communications in hard environments.

Ongoing researches are focus on developing new software components like security mechanisms into all layers involved in this platform to be used in IIoT's applications/scenarios.

Declarations

Author contribution statement

Gustavo Caiza: Analyzed and interpreted the data.

Carlos A. Garcia: Conceived and designed the experiments; Wrote the paper.

Jose E. Naranjo: Performed the experiments.

Marcelo V. Garcia: Analyzed and interpreted the data; Wrote the paper.

Funding statement

This work was supported by Universidad Tecnica de Ambato (UTA) and their Research and Development Department (DIDE) under project CONIN-P-256-2019, and SENESCYT by grants "Convocatoria Abierta 2011", "Convocatoria Abierta 2013", and "Convocatoria Abierta 2018".

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] G. Roshani, E. Nazemi, M. Roshani, Intelligent recognition of gas-oil-water three-phase flow regime and determination of volume fraction using radial basis function, Flow Meas. Instrum. 54 (2017) 39–45, https://linkinghub.elsevier.com/retrieve/pii/S0955598616301820.

[2] X. Wu, S. Tian, L. Zhang, The Internet of things enabled shop floor scheduling and process control method based on Petri Nets, IEEE Access 7 (2019) 27432–27442, https://ieeexplore.ieee.org/document/8644004/.

[3] F. Tao, J. Cheng, Q. Qi, IIHub: an industrial Internet-of-things hub toward smart manufacturing based on cyber-physical system, IEEE Trans. Ind. Inform. 14 (5) (2018) 2271–2280, https://ieeexplore.ieee.org/document/8055624/.

[4] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K.F. Tsang, J. Rodriguez, Massive Internet of things for industrial applications: addressing wireless IIoT connectivity challenges and ecosystem fragmentation, IEEE Ind. Electron. Mag. 11 (1) (2017) 28–33, http://ieeexplore.ieee.org/document/7883984/.

[5] M.S. Hossain, G. Muhammad, Cloud-assisted industrial Internet of things (IIoT) – enabled framework for health monitoring, Comput. Netw. 101 (2016) 192–202, https://linkinghub.elsevier.com/retrieve/pii/S1389128616300019.

[6] M.Y. Aalsalem, W.Z. Khan, W. Gharibi, N. Armi, An intelligent oil and gas well monitoring system based on Internet of things, in: 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), IEEE, Jakarta, 2017, pp. 124–127, http://ieeexplore.ieee.org/document/8253159/.

[7] H. Regenbrecht, G. Baratoff, W. Wilke, Augmented reality projects in the automotive and aerospace industries, IEEE Comput. Graph. Appl. 25 (6) (2005) 48–56, http://ieeexplore.ieee.org/document/1528433/.

[8] B. Chu, K. Jung, Y. Chu, D. Hong, M.-T. Lim, S. Park, Y. Lee, S.-U. Lee, M.C. Kim, Kang Ho Ko, Robotic automation system for steel beam assembly in building construction, in: 2009 4th International Conference on Autonomous Robots and Agents, IEEE, Wellington, New Zealand, 2000, pp. 38–43, http://ieeexplore.ieee.org/document/4803937/.

[9] J. Kim, W. Lee, H. Yang, Y. Lee, Real-time monitoring and control system of an industrial robot with 6 degrees of freedom for grinding and polishing of aspherical mirror, in: 2018 International Conference on Electronics, Information, and Communication (ICEIC), IEEE, Honolulu, HI, 2018, pp. 1–4, http://ieeexplore.ieee.org/document/8330691/.

[10] Y. Zhang, L. Li, M. Ripperger, J. Nicho, M. Veeraraghavan, A. Fumagalli, Gilbreth: a conveyor-belt based pick-and-sort industrial robotics application, in: 2018 Second IEEE International Conference on Robotic Computing (IRC), IEEE, Laguna Hills, CA, 2018, pp. 17–24, http://ieeexplore.ieee.org/document/8329876/.

[11] O.I. Borisov, V.S. Gromov, S.A. Kolyubin, A.A. Pyrkin, A.A. Bobtsov, V.I. Salikhov, A.O. Klyunin, I.V. Petranevsky, Human-free robotic automation of industrial operations, in: IECON 2016 - 42nd Annual Conference of the IEEE, Industrial Electronics Society, IEEE, Florence, Italy, 2016, pp. 6867–6872, http://ieeexplore.ieee.org/document/7793922/.

[12] A.G.C. Gonzalez, M.V.S. Alves, G.S. Viana, L.K. Carvalho, J.C. Basilio, Supervisory control-based navigation architecture: a new framework for autonomous robots in industry 4.0 environments, IEEE Trans. Ind. Inform. 14 (4) (2018) 1732–1743, https://ieeexplore.ieee.org/document/8241718/.

[13] A. Richert, M. Shehadeh, L. Plumanns, K. Gros, K. Schuster, S. Jeschke, Educating engineers for industry 4.0: virtual worlds and human-robot-teams: empirical studies towards a new educational age, in: 2016 IEEE Global Engineering Education Conference (EDUCON), IEEE, Abu Dhabi, United Arab Emirates, 2016, pp. 142–149, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber = 7474545.

[14] C.A. Garcia, E.X. Castellanos, C. Rosero, C. Sanchez, M.V. Garcia, F. Perez, M. Marcos, CPPS on low cost devices for batch process under IEC-61499 and ISA-88, in: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), IEEE, Emden, 2017, pp. 855–860, http://ieeexplore.ieee.org/document/8104884/.

[15] E.X. Castellanos, C.A. Garcia, C. Rosero, C. Sanchez, M.V. Garcia, Enabling an automation architecture of CPPs based on UML combined with IEC-61499, in: 2017 17th International Conference on Control, Automation and Systems (ICCAS), IEEE, Jeju, 2017, pp. 471–476, http://ieeexplore.ieee.org/document/8204485/.

[16] C.A. Garcia, E.X. Castellanos, J. Buele, J. Espinoza, D. Lanas, C. Beltran, M.V. Garcia, MPC under IEC-61499 using low-cost devices for oil pipeline system, in: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), IEEE, Porto, 2018, pp. 659–664, https://ieeexplore.ieee.org/document/8472094/.

[17] C.A. Garcia, E.X. Castellanos, J. Buele, J. Espinoza, C. Beltran, M. Pilatasig, E. Galarza, M.V. Garcia, Fuzzy control implementation in low cost CPPS devices, in: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), IEEE, Daegu, 2017, pp. 162–167, http://ieeexplore.ieee.org/document/8170423/.

[18] C.A. Garcia, G. Salinas, V.M. Perez, F. Salazar L, M.V. Garcia, Robotic arm manipulation under iec 61499 and ros-based compatible control scheme, in: M. Botto-Tobar, L. Barba-Maggi, J. Gonzalez-Huerta, P. Villacres-Cevallos, O.S. Gomez, M.I. Uvidia-Fassler (Eds.), Information and Communication Technologies of Ecuador (TIC.EC), Springer International Publishing, Cham, 2019, pp. 358–371.

[19] C.A. Garcia, F. Salazar L, C. Marino, W.R. Villalba, M.V. Garcia, Design of flexible cyber-physical production systems architecture for industrial robot control, in: 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM), 2018, pp. 1–6.

[20] M. Gerngros, P. Herrmann, C. Westermaier, C. Endisch, Highly flexible needle winding kinematics for traction stators based on a standard industrial robot, in: 2017 7th International Electric Drives Production Conference (EDPC), IEEE, Würzburg, 2017, pp. 1–7, http://ieeexplore.ieee.org/document/8328147/.

[21] L.T. De Paolis, P. Bourdot (Eds.), Augmented Reality, Virtual Reality, and Computer Graphics: 5th International Conference, AVR 2018, Otranto, Italy, June 24–27, 2018, Proceedings, Part I, Lecture Notes in Computer Science, vol. 10850, Springer International Publishing, Cham, 2018, http://link.springer.com/10.1007/978-3-319-95270-3.

[22] A. Zoitl, T. Strasser, A. Valentini, Open source initiatives as basis for the establishment of new technologies in industrial automation: 4diac a case study, in: 2010 IEEE International Symposium on Industrial Electronics, IEEE, Bari, Italy, 2010, pp. 3817–3819, http://ieeexplore.ieee.org/document/5637502/.

[23] N. Iannacci, M. Giussani, F. Vicentini, L.M. Tosatti, Robotic cell work-flow management through an IEC 61499-ROS architecture, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Berlin, Germany, 2016, pp. 1–7, http://ieeexplore.ieee.org/document/7733528/.

[24] K. Choi, J. Cho, J. Han, H. Choi, G.-H. Yang, Space teleoperation scheme for Korean Lunar Rover, in: 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), IEEE, Jeju, 2017, pp. 706–709, http://ieeexplore.ieee.org/document/7992805/.

[25] Z. Wang, I. Reed, A.M. Fey, Toward intuitive teleoperation in surgery: human-centric evaluation of teleoperation algorithms for robotic needle steering, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Brisbane, QLD, 2018, pp. 1–8, https://ieeexplore.ieee.org/document/8460729/.

[26] T. Abut, S. Soyguder, Haptic industrial robot control and bilateral teleoperation by using a virtual visual interface, in: 2018 26th Signal Processing and Communications Applications Conference (SIU), IEEE, Izmir, 2018, pp. 1–4, https://ieeexplore.ieee.org/document/8404833/.

[27] J.E. Naranjo, E.C. Lozada, H.I. Espin, C. Beltran, C.A. Garcia, M.V. Garcia, Flexible architecture for transparency of a bilateral tele-operation system implemented in mobile anthropomorphic robots for the oil and gas industry, IFAC-PapersOnLine 51 (8) (2018) 239–244, https://linkinghub.elsevier.com/retrieve/pii/S2405896318307134.

[28] J. Krome, J. Matson, G. Mitchell, Integrating the intelligent oilfield, Hart's E and P (OCT.), https://www.scopus.com/inward/record.uri?eid = 2-s2.0-33750981559%26partnerID = 40%26md5 = bd414e30630b7022901829bd3c7a0dc5, 2006.

[29] J. Krome, J. Matson, G. Mitchell, The intelligent oil field: people and process, Hart's E and P (NOV.), https://www.scopus.com/inward/record.uri?eid = 2-s2.0-47749130871%26partnerID = 40%26md5 = eb20bca8c7ba2898f0c1891062aaecec, 2006.

[30] A. Al-Sulaiti, H. Al-Baker, M. Beg, Moving towards intelligent oilfields, in: SPE Production and Operations Symposium, Proceedings, vol. 1, 2012, pp. 490–497, https://www.scopus.com/inward/record.uri?eid=2-s2.0-84865755998%26partnerID=4%26md5=8146d318635a1465c84b0069e1989935.

[31] 4DIAC, IEC 61499 Implementation for Distributed, https://eclipse.org/4diac/, 2017.

[32] P. Gsellmann, M. Melik-Merkumians, G. Schitter, Comparison of code measures of IEC 61131–3 and 61499 standards for typical automation applications, in: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Turin, 2018, pp. 1047–1050, https://ieeexplore.ieee.org/document/8502464/.

[33] S. Patil, D. Drozdov, V. Vyatkin, Adapting software design patterns to develop reusable IEC 61499 function block applications, in: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), IEEE, Porto, 2018, pp. 725–732, https://ieeexplore.ieee.org/document/8472071/.

[34] L. Prenzel, J. Provost, Implementation and evaluation of IEC 61499 basic function blocks in Erlang, in: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Turin, 2018, pp. 123–130, https://ieeexplore.ieee.org/document/8502470/.

[35] MQTT Org, MQTT Standard, http://mqtt.org/, 2018.

[36] M. Houimli, L. Kahloul, S. Benaoune, Performance analysis of Internet of things application layer protocol, in: J. Mizera-Pietraszko, P. Pichappan, L. Mohamed (Eds.), Lecture Notes in Real-Time Intelligent Systems, vol. 756, Springer International Publishing, Cham, 2019, pp. 225–234, http://link.springer.com/10.1007/978-3-319-91337-7_22.

[37] Manickam Bahashwan, A.A.O. Selvakumar, A brief review of messaging protocol standards for Internet of things (IoT), J. Cyber Secur. Mobil. 8 (1) (2018) 1–14, http://www.riverpublishers.com/journal_read_html_article.php?j=JCSM/8/1/1.

[38] D.-H. Kim, H.-Y. Lee, D.-S. Kim, Enhanced industrial message protocol for real-time IoT platforms, IEIE Trans. Smart Process. Comput. 7 (5) (2018) 378–384, http://www.dbpia.co.kr/Journal/ArticleDetail/NODE07552995.

[39] T. Mercy, R. Van Parys, G. Pipeleers, Spline-based motion planning for autonomous guided vehicles in a dynamic environment, IEEE Trans. Control Syst. Technol. 26 (6) (2018) 2182–2189, https://ieeexplore.ieee.org/document/8022960/.

[40] B.I. Adamov, Influence of mecanum wheels construction on accuracy of the omnidirectional platform navigation (on example of KUKA youBot robot), in: 2018 25th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), IEEE, St. Petersburg, 2018, pp. 1–4, https://ieeexplore.ieee.org/document/8405889/.

[41] S. Blumenthal, A. Shakhimardanov, W. Nowak, KUKA youBot User Manual (2016) 48.

[42] M.V. Garcia, E. Irisarri, F. Perez, M. Marcos, E. Estevez, Engineering tool to develop CPPS based on IEC-61499 and OPC UA for oil&gas process, in: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), IEEE, Trondheim, Norway, 2017, pp. 1–9, http://ieeexplore.ieee.org/document/7991969/.

[43] C.A. Garcia, J.E. Naranjo, L.A. Campana, M. Castro, C. Beltran, M.V. Garcia, Flexible Robotic Teleoperation Architecture Under IEC 61499 Standard for Oil & Gas Process, in: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Torino, Italy, 2018, pp. 1269–1272, https://ieeexplore.ieee.org/document/8502520/.