



REMOTE USER INTERFACE FOR ABB IRB120 ROBOT USING TCP/IP COMMUNICATION

Stelian-Emilian OLTEAN¹, Paul-Valentin VINKA²

^{1,2}George Emil Palade University of Medicine, Pharmacy, Science, and Technology of Targu Mures

Faculty of Engineering and Information Technology

N. Iorga, 1, Targu Mureș, 540139, Romania

¹stelian.oltean@umfst.ro

²vinkavali@gmail.com

Abstract

This paper presents the development and implementation of a remote user interface for the ABB robot using the TCP/IP protocol. The study focuses on building a user-friendly interface in Matlab environment that enables precise and secure control of the robot's operation with an educational support role in teaching the basics of robotics. The remote application offers a simple and intuitive controlling mode, providing users with a firsthand experience of the robot's capabilities through predetermined actions. Additionally, the advanced mode allows users to test the robot's functionality and perform simple tasks. The use of asynchronous communication contributes to the stability and performance of the remote control application.

Keywords: industrial robot, remote control application, TCP/IP communication, graphical user interface.

1. Introduction

The modern era of industrial automation and industrial robotics relies heavily on efficient communication between devices and systems to achieve optimal performance and precision. The transformation of factories into smart factories and digitized industry in the context of the new industrial revolution given by the transition to the concept of Industry 4.0 has required the development of several divisions and areas ([1], [2]). Industrial automation using robotic applications is one of the fastest growing divisions.

According to [3], by 2020, more than 2.7 million industrial robots are already operating in factories

worldwide. The successful application of industrial robots and the active implementation of the Industry 4.0 concept have stimulated an interest in researching new implementations, educational training, but also addressing new non-typical areas.

Typical well-known applications mostly found in the industrial field are: material handling, welding and soldering, painting and coating, assembling, machining, cutting, casting, packing and palletizing, logistics and storage, inspection and quality control. Other non-industrial areas where the use of robots is increasing are: military and defence, medical and health services, agriculture, underwater, outer space, security ([3]-[5]).

Nowadays due to the complexity, not only the design, construction, and commissioning of a robotic

system require various knowledge ([4],[6]), but even their simple use, operation, and programming. In addition, there are multiple approaches to robot programming languages, as manufacturers offer dedicated software with complex graphical user interface (GUI), hardly accessible to the uninitiated.

So different studies mention graphical user interfaces and remote communication either to simplify the programming or to test and control the industrial robot for required purposes.

In [7] a web application and GUI that acts as a front-end for programming robot motion tasks lowers the engineering complexity. As stated in [8] a more general solution (independent of the robot manufacturer) can be done by using GUI automation tools and generic programming language to generate intermediary programs which are translated into vendor-specific program or template. In both cases certain programming skills and robotics knowledge are still required.

The paper [9] presents the possibility of using a graphics tablet to control an industrial robot without knowledge of ABB robot programming language. The application communicates with the robot via Ethernet using the TCP/IP data transmission protocol. Similarly, in [10] a web application was created as graphical user interface (GUI) and using a stylus pen an operator can maneuver the industrial robot ABB IRB120 to draw a picture with less subjective difficulty. The communication is made on a Non-Standalone 5G Mobile Network. The authors of [11] used a KINECT v2 module to control the ABB IRB120 robot by means of gestures and voice commands, the GUI application communicates with the robot via Ethernet using the TCP/IP data transmission.

In any situation, programming or just using the robots, the early use of interactive and educational applications helps to accumulate simple concepts/knowledge in the field (e.g. robot representation by coordinates, trajectories and movements, performing tasks, operating status, data transfer). In this context, the present paper delves into the development and implementation of a remote graphical user interface tailored for the ABB IRB120 industrial robot, where an important part is also to establish seamless communication between the robot and the client application, leveraging the TCP/IP protocol as the means of data exchange. Even if in the client application, the graphical user interface serves to test the functionality of the robot for educational purposes, the communication application demonstrates also the potential and advantages of efficient and

scalable communication protocols in industrial settings, supporting increased production efficiency and resource optimization.

Regarding robotic structures with 6 degrees of freedom (suitable for the ABB IRB120 industrial robot under discussion) there are numerous theoretical analyses of kinematics and dynamics, modeling and simulation, respectively validation studies using different software environments (e.g. Matlab, Octave, MathCad, Catia, RobotStudio [12]-[18]). For this reason, these aspects are no longer explained in this paper.

The subsequent sections of the paper will provide a comprehensive overview of the theoretical background, design considerations, and the methods employed in the development of the graphical user interface for the remote control. The implementation details, including the various modes of control and the communication structure between the robot and the client application, will be elucidated. In addition, it presents the main findings and contributions, while also highlighting the potential impact of this research in terms of industrial automation and robotics.

The paper aims to extend the knowledge of efficient communication technology in industrial robotics and demonstrates the potential and advantages of communication protocols for creating remote control applications for industrial and didactic purposes.

2. Materials and methods

A primary goal of this study was to develop a GUI for the remote control of an ABB robot using TCP/IP protocol in order to introduce basic robotics concepts. At the same time, the application has a flexible and modular architecture, in order to be an example that can be adopted in various industrial environments, not only according to the educational requirements.

By approaching the proposed solution in terms of efficient communication between the educational GUI and the industrial robot, the aim is to enhance the operational potential of the robot for novice users who want to experience the full range of its predetermined actions. Thus, the paper has a secondary contribution to the broader field of industrial automation and control by demonstrating the advantages of employing TCP/IP communication in robotic applications.

As already stated, to fully unlock the robot potential for the uninitiated, a user-friendly interface capable of precise and secure control becomes imperative. The proposed system's architecture is modular and consists of three major integral components:

- The Matlab Client application: developed with

user-friendliness in mind, the Matlab client serves as the primary interface for interaction. It allows users to issue commands to the robot, view real-time data, and monitor task status, without having the robotic programming skills.

- TCP/IP Communication Layer: this layer serves as the networking backbone, responsible for transmitting data between the Matlab client and the ABB industrial robot in real-time. It ensures that data packets are delivered accurately and promptly.
- ABB Industrial Robot Server Application: the ABB robot is controlled via custom-written modules that interpret the incoming commands from the Matlab client and execute them. The modules are written in the robot's specific programming language.

For the present paper, the Agile methodology underpins the project's development, allowing for quick iterations and the inclusion of feedback at various stages. Prototype models were used to validate each feature and design decision. The codebase is modular, making it easy to manage, maintain, and scale.

The proposed functionality of the system considers:

- Real-Time Control: the system provides real-time control over the robot's speed, position, and orientation. It allows dynamic adjustments to be made while operations are ongoing.
- Command Library: a pre-defined set of commands is available for common tasks, including waving, linear movement, circular movement, etc., simplifying the user experience.
- Custom Commands: the interface allows users to create and execute custom 3D movements, providing flexibility for advanced applications.
- Data Logging: the system features a robust data logging function. It automatically captures key performance metrics, logs errors, and stores operational data for analysis and reporting.
- experimental Setup: the system was initially developed and tested using a simulated ABB robot and a localhost setup for the TCP/IP communication layer. Rigorous testing was later performed on ABB IRB120 industrial robot under controlled conditions to validate the system's capabilities.

Although applicable to other structures, the application focused on an experimental setup with the ABB IRB120 industrial robot, which is one of ABB's smallest industrial robots. It weighs 25 kg and can handle loads up to 3 kg with a range of 580 mm ([19]-

[21]). The ABB IRB120 robot stands as a compact and versatile solution, widely utilized across various industrial sectors for tasks such as material handling, assembly, and adhesive application.

3. Implementation details

This chapter focuses on the nuts and bolts of the system implementation, with a particular emphasis on the different control modes facilitated by the Matlab based client interface ([22]). The objective is to give a detailed understanding of how the system operates.

The *Matlab based client application* offers two control modes for operating the ABB industrial robot:

Simple Mode

This mode is intended for users who do not have prior experience or knowledge of the robot. In this mode users can experiment with the robot through a series of pre-established demonstrations, each explained in detail.

The user interface presented in Fig. 1 contains:

1. Status Light is an indicator light which displays the status of the connection. It turns green when communication is active and red when communication is stopped.
2. Start Communication Button initiates communication with the robot after properly configuring the network parameters and processing speed.
3. Stop Communication Button interrupts communication with the robot.
4. Tutorial Button opens a separate section that provides tutorials and information about each demonstration.
5. Robot IP Selection allows the user to enter the IP address of the robot to establish a connection.
6. Port Selection select the port through which communication is made.
7. Predefined Functions section provides to the user a series of buttons that, when pressed, trigger predefined movements of the robot. This is a convenient and intuitive mode for users who are not familiar with the robot's movements.

With the wave function, the robot moves its arm in a waving motion, similar to a human. Square function is a predefined program in which the robot moves its arm in a trajectory that forms a square. Absolute Joint function directs the robot to move to a predefined position, named "home position" or standard rest position. In the Linear function the robot performs a linear movement with its arm. In Circular function the robot's arm traces a circle in space. In Joint demonstration the robot performs a series of movements involving the flexing

and extending of its joints, highlighting its capability to execute complex and precise movements.

8. Robot Movement Speed Control allows user to adjust the speed at which the robot performs its movements, for quick and efficient adaptation to different tasks or working conditions. This is achieved through a control dial that enables the adjustment of the robot's movement speed.

9. Master Control switches the GUI to an advanced operating mode, thus offering more detailed control over the robot's operations.

10. Last Action Performed is a section which displays details about the last operation performed by the robot. This helps the user understand what actions the robot has recently carried out, thus facilitating the monitoring and control of its performance.

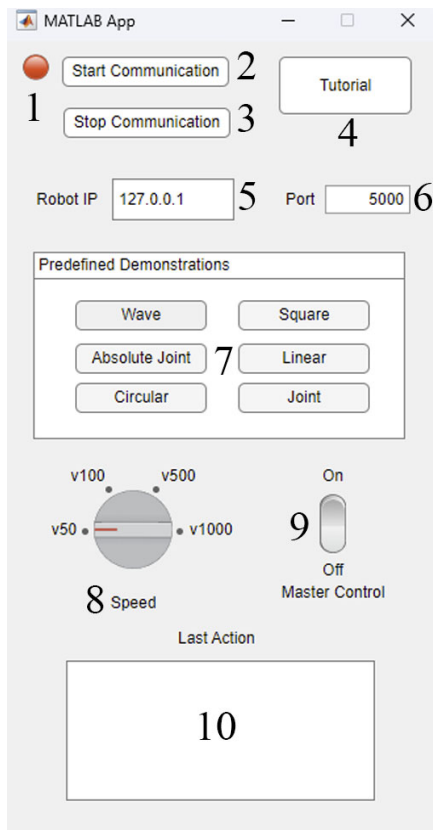


Fig. 1: GUI for the simple control mode
Advanced Mode

The advanced mode of the application is divided into other two distinct operating modes: *the use of world coordinates and the use of joint variables*.

The world coordinates operating mode is intended for situations where the user wants to control the robot using the world coordinates (x, y, z). In this case, the

user can send the exact world coordinates to which they want to move the tool center point (TCP), thus providing precise control over the position and movements in the workspace.

On the other hand, the *joint variable operating mode* refers to the direct control of the robot's individual axes. This mode allows the user to control the specific movements of each axes individually, thus offering greater flexibility in controlling the robot's movements.

Each operating mode has its own unique advantages and can be selected based on the user's specific needs. The user can opt for the world coordinates operating mode for a more generalized approach and for movements that require the TCP precise control. Alternatively, they can choose the joint variable operating mode for more detailed control over the robot's axes individual movements.

The GUI for the advanced operating mode with world coordinates is presented in Fig. 2.

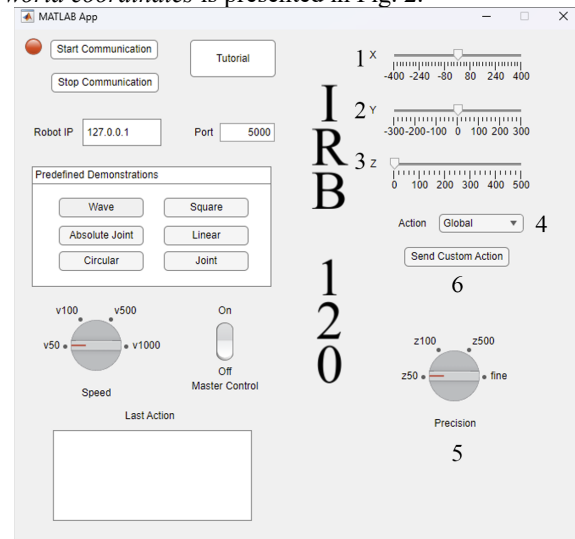


Fig. 2: GUI for the advanced operating mode with world coordinates

This mode employs multiple interface components to facilitate this interaction:

1. X Coordinate Slider feature allows the user to select the value of the X coordinate for the robot's movement. The user can slide to select the desired X coordinate value, providing granular control over the robot's position.

2. Y Coordinate Slider is similar to the X coordinate slider and allows the user to select the value of the Y coordinate for the robot's movement. It can be easily adjusted to achieve the desired accuracy in controlling

the robot's position.

3. Z Coordinate Slider allows the user to set the value of the Z coordinate for the robot's movement. By sliding it, the user can control the height at which the robot moves in the workspace.

4. Action Selection Dropdown (Global or Local) allows the user to switch between the advanced operating modes: world or joint. The user can select the desired mode from this dropdown menu, offering flexibility in the robot's operating mode.

5. Precision Selection Knob allows the user to adjust the accuracy with which the robot performs its movements. By rotating this knob, the user can increase or decrease the robot's movement precision, allowing fine control over its performance.

6. Send Action Button send the command to the robot after all settings have been properly configured. This triggers the robot's movement according to the settings specified by the user.

The advanced operating mode with local variables presented in Fig. 3 provides a higher degree of detailed control over the robot, allowing the user to individually control each motor of the robot. This is particularly useful in situations where fine-tuning of the robot's movements or manipulation of a specific arm or axis is desired.

1. Slider for Axis 1 controls the position of the robot's base, which forms Axis 1.
2. Slider for Axis 2 controls the robot's first arm.
3. Slider for Axis 3 is responsible for controlling the robot's second arm.
4. Slider for Axis 4 control the robot's third arm position.
5. Slider for Axis 5 controls the rotation of the robot's last arm.
6. Slider for Axis 6 is responsible for the rotation of the robot's "hand" meaning its tool or end effector.
7. Representative Image shows an image for better understanding of the actions of each slider. This helps users identify and better understand how each axis of the robot operates and moves.

The robot control system architecture is based on ABB Robot Studio software [23], an advanced platform that, through the RAPID language, allows us to design and implement the *server application* dedicated to robot management. This server plays a pivotal role in our system, being responsible for interpreting and processing the instructions received from the client graphical user interface (Matlab client application) and converting them into actual actions of the robot.

One important part of the robot controller infrastructure is the 616-6 PC Interface Module, which,

thanks to its dedicated function suite, facilitates the establishment and management of a TCP/IP connection between the client and server. The SocketCreate, SocketBind, and SocketAccept functions allow us to configure and initiate a robust and reliable connection for data transmission, thus providing us the opportunity to communicate effectively with the robot.

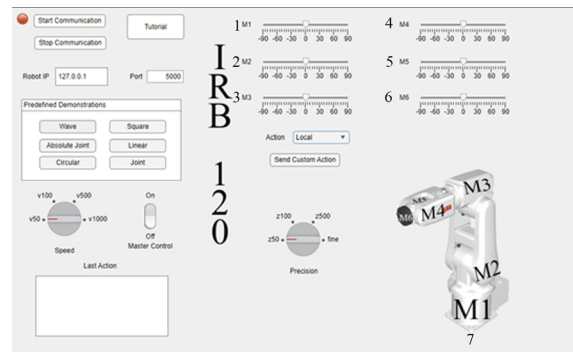


Fig. 3: GUI for the advanced operating mode with joint variables

The *server application* manages two main types of instructions:

- *Simple Instructions*: These consist of commands represented by character strings transmitted by the Matlab based application to the server. Each character string represents a distinct command that, once received and interpreted by the server, results in the execution of a specific action by the robot.
- *Composite Instructions*: These are more complex commands that, in addition to the character string indicating the desired action, also include motion parameters such as the x, y, z coordinates. The server processes this information and generates instructions for the robot, leading to the execution of a specific movement in space.

The server is created in ABB RobotStudio and provides a controlled and secure environment for simulating and testing the behavior of the robot. Through this, we can validate the performance and reliability of the robot control system in a virtual environment, thereby ensuring that each command is executed correctly and efficiently before moving on to real-world implementation. In conclusion, the server plays an essential role in ensuring the efficiency and safety of the operations performed by the robot.

The communication protocol has been implemented to manage the communication between the client and server via TCP/IP data communication, as well as the specific functions associated with each transmitted command. The below paragraphs presents each

character string sent, the additional parameters required, the corresponding RAPID procedure, and a description of the actions performed. Each of these commands initiates a specific RAPID procedure which, in turn, controls the behavior of the robot based on the transmitted parameters. The implementation of this TCP/IP protocol has allowed for precise controllability of the robot, adapting to the needs and requirements on a case-by-case basis.

Sent character strings and their functions implemented:

- *stop*
 - Additional Parameters: None
 - RAPID Procedure: stop
 - Description: Stops the connection between the client and server.
- *speed1*
 - Additional Parameters: None
 - RAPID Procedure: speed1
 - Description: Changes the robot's movement speed to 50 units.
- *speed2*
 - Additional Parameters: None
 - RAPID Procedure: speed2
 - Description: Changes the robot's movement speed to 100 units.
- *speed3*
 - Additional Parameters: None
 - RAPID Procedure: speed3
 - Description: Changes the robot's movement speed to 500 units.
- *speed4*
 - Additional Parameters: None
 - RAPID Procedure: speed4
 - Description: Changes the robot's movement speed to 1000 units.
- *wave*
 - Additional Parameters: None
 - RAPID Procedure: wave
 - Description: Initiates the waving procedure.
- *joint*
 - Additional Parameters: None
 - RAPID Procedure: joint
 - Description: Moves the robot in Joint mode.
- *circular*
 - Additional Parameters: None
 - RAPID Procedure: CIRCLE
 - Description: Moves the robot in

Circular mode.

- *JointAngles*
 - Additional Parameters: Local variables for the movement of all motors
 - RAPID Procedure: JointAngles
 - Description: Moves the robot's joints according to specified parameters.
- *LinearMovement*
 - Additional Parameters: Global variables for point
 - RAPID Procedure: LinearMovement
 - Description: Moves the robot's arm to the specified position.
- *Square*
 - Additional Parameters: None
 - RAPID Procedure: SQUARE
 - Description: Initiates the procedure for moving the robot in the shape of a square.

4. Discussion and results

The culmination of our work on the robotic control system offers insightful results and sheds light on the implications of our design choices. The system consists of a client-server typical configuration with a Matlab-based graphical user interface on the client side easy to use for educational purposes and a server application based on ABB RobotStudio and RAPID programming language. The following paragraphs summarize the outcomes and present some particularities based on the system objectives, successes, and limitations.

a) Efficiency in communication

The implementation of the TCP protocol significantly improved the efficiency of communication between the server and the client. Data packets were sent and received with minimal latency, thereby ensuring real-time control of the robot's movements.

b) Precision in movements

The sliders and controls developed for each axis of the robot proved to be highly precise, allowing the operator to make fine adjustments. Such precision is invaluable in applications requiring meticulous placement or movement, such as in medical surgeries or delicate material handling.

c) User-friendly interface

Our user interface was designed to be intuitive and accessible, mostly for unfamiliar users with robotic systems and programming languages, allowing it to be used for educational purposes. This was corroborated by user-testing sessions, where participants were able to successfully complete tasks using the robot within a

short learning curve, without the need for specialist knowledge and understanding.

d) Adaptability

The RAPID procedures, including commands for speed variations and specific movement types, provided users with adaptability. This means that the robotic system can be easily tailored to fit specific applications and environments.

e) Robustness and reliability

Our system showed robust performance during stress tests, where it executed complex tasks in succession without any downtime. This reliability is crucial also in the industrial settings, where the cost of mistakes or delays can be high.

f) Complexity in setup

One of the major drawbacks is the complexity involved in setting up the initial environment. The reliance on ABB RobotStudio as the back-end environment means that the experiments and the training with the real ABB IRB120 industrial robot under controlled conditions should be done with trained staff (at least one laboratory assistant or technician). He must have at least a basic understanding of physical equipment.

g) Scalability concerns

The TCP/IP data communication protocol used for communication, while efficient, has its limits in terms of scalability. If there is a requirement to control multiple robots, additional work will be needed to manage simultaneous connections.

h) Meeting objectives

The primary objectives of the project, which were to establish an efficient, reliable, remote control of the robotic system with user-friendly interface for learning activities in the field of robotics, have largely been met. The system is not only functional but also precise, adaptable, and robust. While we've achieved significant milestones, there is room for future enhancements. These include improving the scalability of the system and further refining the user interface according to the intended educational purpose. From this point of view, the use of the Matlab environment (more common in education) on the client side allows us to improve the application by integrating toolboxes for modeling and simulating the kinematics and dynamics of the robotic structure and getting into the depth of the robotics discipline.

i) Importance of testing

Rigorous unit tests, integration tests, and user acceptance tests ensured that each feature and component worked as intended. This provides protection against system failure, and perhaps more

importantly, gives the confidence to innovate and push the boundaries of what the robot could achieve.

h) Data security

The use of TCP/IP for communication opens up questions regarding data security. Future iterations could explore the integration of enhanced security measures, such as SSL/TLS encryption or secure authentication, to safeguard against unauthorized access to the robotic system.

5. Conclusion

In conclusion, this study has been a journey laden with both challenges and triumphs. It demonstrates that ABB RobotStudio, coupled with a comprehensive TCP/IP data communication protocol and a well-designed user interface, can create a robotic system that is efficient, robust, and adaptable. Although not without its limitations, the system provides an excellent foundation for future work in the area of robotics and automation.

We've achieved an effective proof-of-concept, and with continued innovation and rigorous evaluation, there is every reason to believe that such systems will only improve in terms of efficiency, reliability, and user-friendliness. With the fast advancements in robotics, our work is a timely and significant contribution to a field that is increasingly becoming a cornerstone of modern industry and daily life.

This is not the end, but rather, an exciting starting point for a multitude of possibilities in human-machine interaction and the broader field of robotics. The lessons learned and the results obtained pave the way for further research and development, adding another layer to the ever-evolving tapestry of robotics and automation.

Acknowledgement

This work was supported by the University of Medicine, Pharmacy, Science and Technology „George Emil Palade“ of Târgu Mureş Research Grant number 164/3/10.01.2023.

References

- [1] Vaidya, S., Ambad, P. and Bhosle, S. (2018), Industry 4.0 – A Glimpse, *Procedia Manufacturing*, vol. 20, pp. 233–238.
- [2] Xu, X., Lu, Y., Vogel-Heuser, B. and Wang, L. (2021), Industry 4.0 and Industry 5.0—Inception, conception and perception, *Journal of Manufacturing Systems*, vol. 61, pp. 530-535.

- [3] Dzedzickis, A., Subaciute-Zemaitiene, J., Sutinyas, E., Samukaite-Bubniene and U., Bucinskas, V. (2022), Advanced Applications of Industrial Robotics: New Trends and Possibilities. *Appl. Sci.*, vol. 12, pp.1-15.
- [4] Miller, M.R. and Miller R. (2017), *Robots and Robotics. Principles, Systems, and Industrial Applications*, McGraw-Hill Education.
- [5] Haider, A. F.A., Haidar F.Al-Q. and Nandha K. (2011), A Review of Application Industrial Robotic Design, Proc. Ninth International Conference on ICT and Knowledge Engineering, pp.105-112.
- [6] Oltean, S., Șoaita, D., Dulau, M. and Jovrea, T. (2008), Aspects of interdisciplinarity in the MRIP 02 system design, Proc. Optirob '07, pp. 1-5.
- [7] Astudillo, A., Sathya, A., Dirckx, D., Decre, W., Gillis, J, Pipeleers, G. and Swevers, J. (2022), A web-based graphical user interface for programming optimal control-based robot motion tasks, *IFAC Papers On Line* 55-27, pp. 120-125.
- [8] Ionescu, T.B. (2021), Leveraging graphical user interface automation for generic robot programming, *Robotics*, vol. 10, 3, pp. 1-23.
- [9] Kaczmarek, W., Lotys, B., Borys, S., Laskowski, D. and Lubkowski, P. (2021), Controlling an industrial robot using a graphic tablet in offline and online mode, vol. 21, 2439, pp. 1-20.
- [10] Yaovaja, K. and Klunngien, J. (2019), Teleoperation of an industrial robot using a non-standalone 5G mobile network, Proc. ECICE '19, pp. 1-6.
- [11] Kaczmarek, W., Panasiuk, J., Borys, S. and Banach, P. (2020), Industrial robot control by means of gestures and voice commands in off-line and on-line mode, *Sensors*, vol. 20, 6358, pp. 1-13.
- [12] IRB 120 ABB's 6 axis robot – for flexible and compact production (ROBO149EN_D), ABB Robotics, 2019.
- [13] Wang, C., Liu, D., Sun, Q. and Wang., T. (2021), Analysis of open architecture 6R robot forward and inverse kinematics adaptive to structural variations, *Mathematical Problems in Eng.*, vol 2021, pp.1-11.
- [14] Seven, O.F. and Ankarali, A. (2019), Inverse kinematics analysis of IRB120 robot arm, *SETSCI Proc. ISAS '19*, vol. 4, pp.383-390.
- [15] Nicolescu, A.F., Ilie, F.M. and Alexandru, T.G. (2015), Forward and inverse kinematics study of industrial robots taking into account constructive and functional parameters modeling, Proc. In manufacturing Systems, vol. 10, pp. 157-164.
- [16] Cristoiu, C. and Nicolescu, A. (2017), Validation of forward geometric models for ABB Robots using virtual models and the software applications CATIA and ABB Robot Studio, Proc. In manufacturing Systems, vol. 12, pp. 145-153.
- [17] Cristoiu, C. and Nicolescu A. (2017), New approach for forward kinematic modeling of industrial robots, *Research and Science Today Supplement*, vol. 2, pp. 136-144.
- [18] Hadi, M.B., Meigoli, V., Golbahar, M.R., Nayeri, S.A. and Ebrahimi, A. (2018), *J. Cent. South univ.*, vol. 25, pp. 2219-2244.
- [19] Sengaphone, P. (2019), Programming ABB industrial robot for writing a human language, *International Journal of Eng. Research and Technology*, vol. 8, issue 3, pp. 442-445.
- [20] Rosca, O.A., Nebunu-Barac, E.E., Covrig, T., Duka, A.V. and Oltean S.E. (2023), *A sorting solution based on an industrial robotic arm with image processing*, In: Moldovan, L., Gligor, A. (eds), ser. *Lecture Notes in Networks and Systems*, Germany: Springer, vol. 605.
- [21] The ABB Products and Services website (2023). [Online] <https://new.abb.com/offerings>.
- [22] Mathworks website (2023). [Online] <https://ch.mathworks.com/>.
- [23] Operating manual RobotStudio (3HAC032104-001), ABB, 2023.