

Article

# Industrial robot control by means of gestures and voice commands in off-line and on-line mode

Wojciech Kaczmarek <sup>1</sup>, Jarosław Panasiuk <sup>1</sup>, Szymon Borys <sup>1,\*</sup>, Patryk Banach <sup>1</sup>

<sup>1</sup> Faculty of Mechatronics, Armament and Aerospace, Military University of Technology, Kaliskiego 2 Street, 00-908 Warsaw; wojciech.kaczmarek@wat.edu.pl (W.K.), jaroslaw.panasiuk@wat.edu.pl (J.P.), patryk.banach95@gmail.com (P.B.)

\* Correspondence: szymon.borys@wat.edu.pl

**Abstract:** The paper presents the possibility of using KINECT v2 module to control an industrial robot by means of gestures and voice commands. It describes elements of creating software for off-line and on-line robot control. The application for KINECT module was developed in C# language in Visual Studio environment, while the industrial robot control program was developed in RAPID language in RobotStudio environment. The development of a two-threaded application in RAPID language allowed to separate two independent tasks for the IRB120 robot. The main task of the robot is performed in thread no. 1 (responsible for movement). Simultaneously working thread no. 2 ensures continuous communication with the KINECT system and provides information about the gesture and voice commands in real time without any interference in thread no. 1. The applied solution allows the robot to work in industrial conditions without negative impact of communication task on the time of robot's work cycles. Thanks to the development of a digital twin of the real robot station, tests of proper application functioning in off-line mode (without using a real robot) were conducted. Obtained results were verified online (on the real test station). Tests of correctness of gesture recognition were carried out, the robot recognized all programmed gestures. Another test carried out was the recognition and execution of voice commands. A difference in the time of task completion between the actual and virtual station was noticed - the average difference was 0.67 s. The last test carried out was to examine the impact of interference on the recognition of voice commands. With a 10dB difference between the command and noise, the recognition of voice commands was equal to 91.43%. The developed computer programs have a modular structure, which enables easy adaptation to process requirements.

**Keywords:** KINECT; industrial robot; vision system; RobotStudio; Visual Studio; gesture control; voice control

## 1. Introduction

The development of electronics, especially sensorics, results in a constant change of the way people interact with electronic devices. An example of using gestures and voice commands to operate devices can today be TVs, computers, building automation systems, mobile robots, etc. Development of electronics and vision systems contributes to the development of a new way of human interaction with devices, which creates completely new possibilities for designing and applying new computer applications. This is particularly evident in interfaces to video surveillance and game applications. A major advantage of using gestures and speech in interaction with the machine is the possibility of dynamic adaptation. The movements of human body can be considered as segments [1] that express specific meaning in specific time periods. Hence, during a conversation, people often gesture, emphasizing the meaning of spoken words. Therefore, sequences of gestures can indicate to the machine the next actions to be performed. By adding voice commands to this, it turns out that the method of communication with machines will become very close to the method used in everyday interpersonal relations. When considering this problem, taking into account robotics, it should be

noted that the use of gestures and voice commands will make it easier to program and operate very complex devices even for beginners.

Already today, industry 4.0 integrates people and digitally controlled machines with the Internet and information technologies [2]. On the one hand, the information flow is carried out at the level of production process between the machines, on the other hand, the process information obtained supports company management systems, which makes it easier to forecast production efficiency. During the implementation of industrial processes it is necessary to take into account many technological and hardware parameters [3,4]. A great challenge is to achieve high productivity and product quality in relation to the costs incurred for the construction of production lines and production itself. Therefore, it is necessary to develop, as far as possible, a flexible production system that would ensure that orders are met today, and in the future would allow to adapt to changing customer needs. Techniques and tools are sought to analyse the current technological process with the possibility of anticipating future use of the system [5,6].

Research efforts are being carried out worldwide to highlight the importance of human-machine integration in various aspects. An example is a study [7], which highlights the importance of human-machine integration in the context of obtaining data for reliability analysis.

In the near future, vision systems used for many years in industrial applications may be used for programming industrial robots or may give the possibility of easier cooperation between the operator and machines at the workstation [8]. Traditional robot programming with the use of dedicated teaching panels slowly becomes a rarity and is replaced by computer programming [9–11]. Thus, the question will arise as to what form the programming of robots will take in the future, and whether it is possible to create flexible programming that will allow the operator to interact with the robots using natural tools available to man, such as body movement (gestures) or voice. In research laboratories, applications are created that bring us closer to this goal. Developments on programming and controlling robots using gestures and voice are carried out in many fields, e.g.: industrial applications (assembly robots), transport (mobile robots), home use (cleaning robots) [12]. An example can be projects in which images are used to recognize hand gestures in order to control a mobile robot through a specially made control circuit [13]. Of course, the engineer's or operator's safety is important in this case, but modern robots today are equipped with safety systems that allow for full collaboration between robots and people [14].

The development and testing of industrial robot control applications can take place in virtual reality in off-line mode. Robot manufacturers offer environments for programming and modeling robots (e.g. RobotStudio by ABB, ROBOGUIDE by FANUC, RT ToolBox by Mitsubishi Electric, KUKA SimPro by KUKA). Modern approach allows to create a virtual controller, which is a digital twin of its real counterpart. This means that it has exactly the same functionality, and that the way and principle of its operation almost 100% corresponds to the operation of real device [15]. This gives great possibilities especially because of the functional analysis of software under development. Many researchers use the KINECT sensor for gesture control of robots [16–22]. Important part of robot programming is preparing an optimal trajectory and gripping points [23–28]. An interesting use of the KINECT sensor is its use as a 3D scanning sensor mounted on a FANUC R2000 industrial robot [29]. This approach allows the scanning of large objects on the one hand, but on the other it is flexible and easily reprogrammable. 3D scanners can also be used to scan larger areas, such as robotic production cells or even entire buildings [30,31]. The limitations of the sensor's design must be taken into account, as the use of the IR detector requires appropriate environmental conditions to eliminate interference. This solution solves many important problems, but most of it is limited to control by gestures only, without considering the possibility to control voice commands as well. Moreover, it seems to be advisable to implement solutions allowing to create digital stations and use virtual environments.

This paper presents a method of controlling an industrial robot with the use of gestures and voice commands. The solution of the problem required the development of a computer program in C# language in Visual Studio environment, which allows to obtain data from KINECT V2 sensor and is

responsible for communication with the application developed for the industrial robot [32]. During the development of a program in Visual Studio, the ABB.Robotics.Controllers library was used to connect directly to the robot controller, set the robot's operating parameters and run the robot control program. The Microsoft.Kinect library was used to detect, connect to the application, receive and interpret data from KINECT V2. This library has the types of variables and mechanisms to operate this device. Voice control of the robot is provided by the System.Speech.Recognition library. The TCP/IP protocol and the System.Net library are responsible for proper communication and smooth robot control. The user interface is based on a WPF (Windows Presentation Foundation) application.

The application for IRB 120 robot from ABB was developed in RAPID language. It was decided that the application will be two-threaded. The main thread is responsible for execution of robot's movements and logic related to the production task. The second thread, working in a background of the main thread, is responsible for communication via Ethernet with the application supporting the KINECT sensor.

The prepared software enables testing of the application (e.g. communication of devices and realization of production tasks) and simulation of robot's work in off-line mode with the use of digital twin and on-line control of the real robot. Tests of the real robot allowed to validate the correctness of digital twin's operation.

Section 2 contains a project of robotic test station with its digital twin. Section 3 presents a description and principle of operation of the control programs and the results of tests carried out. Section 4 summarizes the results of our research.

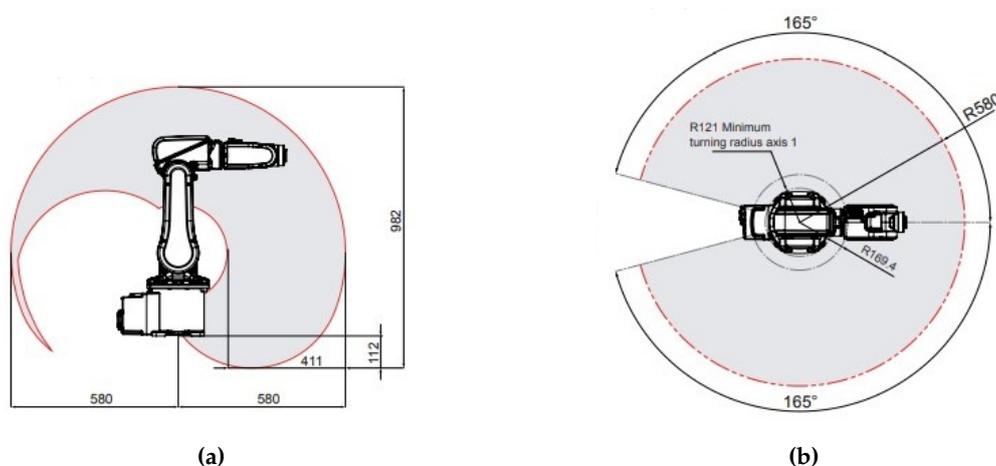
## 2. Station design

Tests were carried out on a research station in the Robotics Laboratory of the Military University of Technology. The station consists of:

- ABB six-axis IRB 120 robot with IRC5 Compact controller (Figure 1),
- KINECT V2 sensor (Figure 2),
- PC.

### 2.1. ABB IRB 120 robot with IRC5 Compact controller

The IRB 120 is one of ABB's smallest industrial robots (Table 1 and Table 2). It weighs 25 kg and can handle loads up to 3 kg with a range of 580 mm [33].



**Figure 1.** ABB IRB 120 motion range: (a) side view; (b) top view.

**Table 1.** IRB 120 robot parameters.

| Data                      | Parameter           |
|---------------------------|---------------------|
| Number of axes            | 6                   |
| Supply voltage            | 200-600 V, 50/60 Hz |
| Power consumption         | 0.24 kW             |
| Acceleration time 0-1 m/s | 0.07 s              |
| Position repeatability    | 0.01 mm             |
| Payload                   | 3 kg                |

**Table 2.** IRB 120 axis parameters.

| Axis movement   | Working range | Velocity |
|-----------------|---------------|----------|
| Axis 1 rotation | +165 to -165  | 250/s    |
| Axis 2 arm      | +110 to -110  | 250/s    |
| Axis 3 arm      | +70 to -110   | 250/s    |
| Axis 4 wrist    | +160 to -160  | 320/s    |
| Axis 5 bend     | +120 to -120  | 320/s    |
| Axis 6 turn     | +400 to -400  | 420/s    |

## 2.2. KINECT V2 sensor

The KINECT V2 sensor consists of three components, an RGB camera, a depth sensor (emitter and infrared camera) and a set of 4 microphones [34], as shown in Figure 2, which give it the functionality to control with gestures and speech.

A depth sensor is used to track a person's movement. The infrared emitter sends an infrared beam, which is distorted upon impact with objects or people in their path and is recorded by the infrared camera. From the read data, a room image is created and the user or users are identified. A set of microphones is used to collect voice information that can be used as control commands.

KINECT V2 "sees" a person as a set of 25 points (Figure 2b). Their interpretation allows to track and read the movements of a person or group of up to 6 people. Thanks to the points it is possible to assess in which position a person is currently in front of the device. From these coordinates, it is possible to separate the required coordinates ( $X_{n-1}, X_n, Y_{n-1}, Y_n$ ) that corresponds to the right and left hand. Vector equations (1, 2, 3) can be used to convert these coordinates to joint angles [18].

$$A = X_n - X_{n-1} \quad (1)$$

$$B = Y_n - Y_{n-1} \quad (2)$$

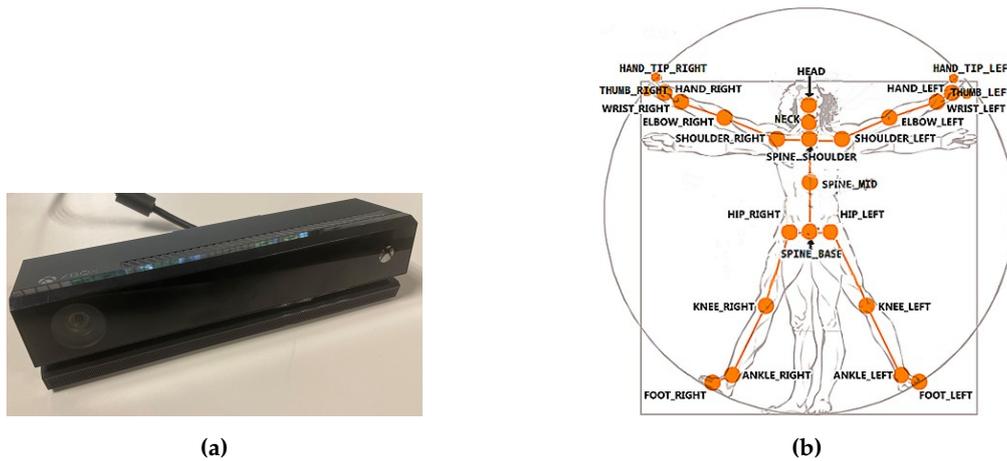
$$\theta = \text{ArcTan}(A, B) \quad (3)$$

Where  $\theta$  - angle between two adjacent joints,  $X_{n-1}, X_n$  - X coordinates of the joints,  $Y_{n-1}, Y_n$  - Y coordinates of the joints.

## 2.3. PC

A computer with Windows 10 was used to control an industrial robot with gestures and voice commands. Thanks to developed control programs, the computer provided support for the KINECT sensor (Visual Studio) and communication with the robot controller (RobotStudio). Kinect for Windows SDK 2.0 (Microsoft.Kinect library) was used to operate the KINECT sensor, while PC SDK 6.07 (ABB.Robotics.Controllers) libraries were used to operate the robot controller.

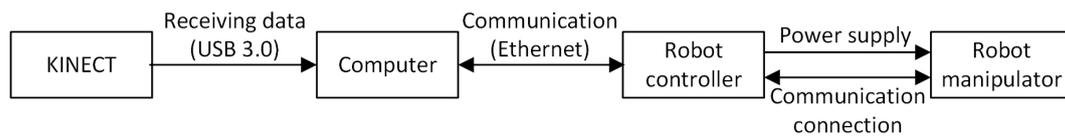
The key issue for this project was the detection of human body points using Microsoft KINECT sensor and the use of microphones. A full image processing logic was developed to control the industrial robot using gestures. In addition, a set of microphones was used to control the robot using voice commands.



**Figure 2.** KINECT V2: (a) general view; (b) recognizable body points.

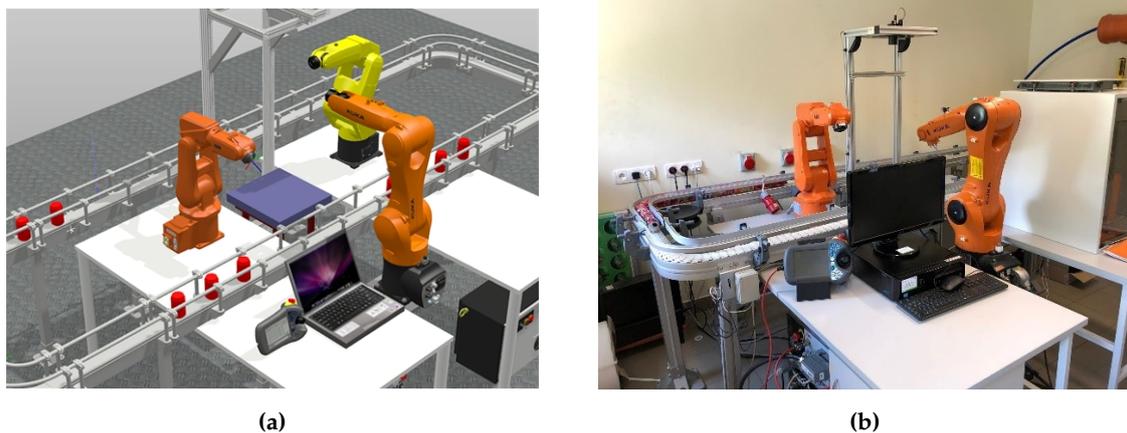
#### 2.4. Test station and its digital twin

Flow chart of the station is shown in Figure 3.



**Figure 3.** Flow chart of the station.

One of our aims was to develop the digital twin of the real robotic position (Figure 4). This enabled the creation, modification and testing of software in a safe way, without the need to work on-line with the real robot.

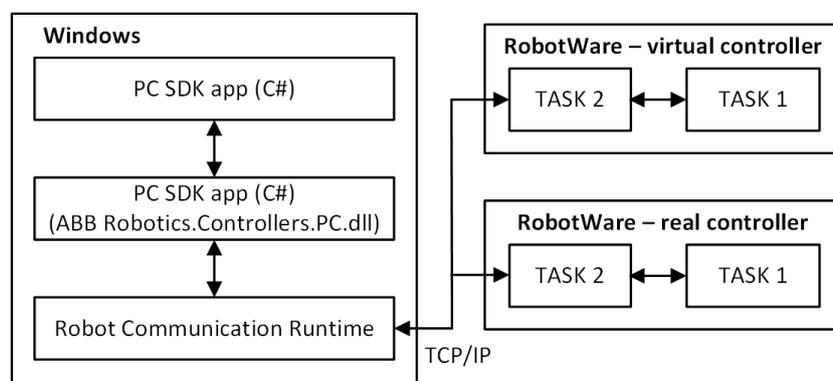


**Figure 4.** General view of the robotized station: (a) virtual; (b) real.

The backup of real robot was used to create a digital twin system. In the RobotStudio environment, the backup generated a robot system that runs on a virtual controller. Such an approach allowed to reproduce the real robot controller with almost 100% accuracy. This means that the robot model in a virtual environment has the functionality and features of a real device. This enables the robot model to be programmed taking into account all the capabilities and limitations of the real robot and the results obtained during the simulation are reliable.

## 2.5. Application design

The integration of devices included in the station (IRB 120 robot, KINECT V2 sensor, computer) required the development of applications enabling the operation of KINECT sensor and the implementation of an industrial robot control process as well as ensuring communication between devices (Figure 5).



**Figure 5.** General diagram of communication between the application and the robot controller.

A control application in C# language has been developed which is responsible for image processing, communication between the Kinect controller, computer and robot controller [35,36]. The application is also equipped with a graphical user interface, allowing for free operation without the need to interfere with the program code.

Microsoft.Kinect library was used to operate the sensor, which allows to detect and connect to the connected sensor [37]. After recognizing the body of operator, i.e. the position of individual points in relation to each other (Figure 2b) – elbows, shoulders, head and the state of hand (open, closed), control using gestures was developed [38]. The operator's skeleton was displayed on the user interface as a mirror image to facilitate control.

To implement voice control, KINECT V2 is connected to a computer and set in Windows 10 as the default microphone (another external microphone can be used). The software was developed using the System.Speech.Recognition library [39]. It was necessary to build a database of words and assign them the appropriate signals to control an industrial robot.

The application developed in the Visual Studio environment communicates with the robot via Ethernet using the TCP/IP data transmission protocol (Figure 5). The ABB.Robotics.Controllers and System.Net libraries were used to create the application [40]. This application uses ABB libraries to send data to the Robot Communication Run Time module, which is responsible for code synchronization and data transfer in both directions. The first of these libraries allows direct access to the robot controller before starting the program. This allows to download and set the values of various data and parameters of the robot control program from the computer application, e.g.: start points of the robot program, manipulator speed, accuracy. It is also possible to start or stop the robot control program. The second library allows to influence the robot's operation also while it is performing its program. This gives the possibility of sending and receiving signals, e.g. sending control signals and receiving the current position of robot's manipulator.

The control program for IRB 120 has been developed in RAPID language in RobotStudio environment. The program is a two-threaded application in which:

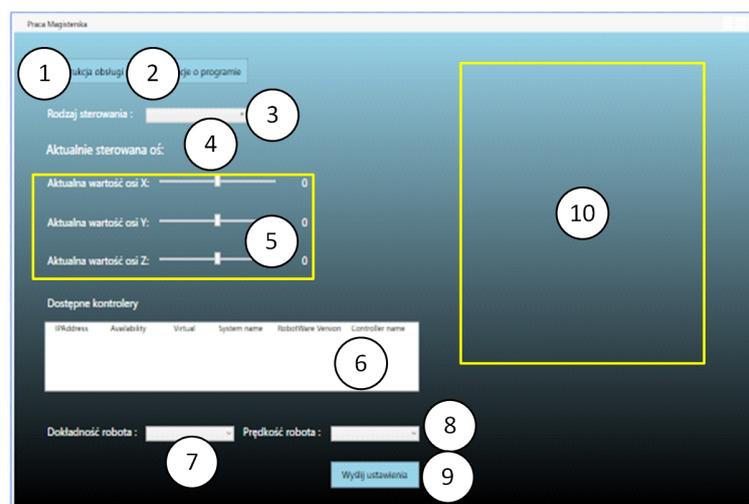
- Thread 1 (Main) – responsible for setting the parameters and interpreting the robot's direction of motion and controlling the robot's movements.
- Thread 2 – responsible for TCP/IP communication between the robot and the computer and transferring the data received from the KINECT sensor to the main thread.

This made it possible to make the robot's motions independent of actions responsible for data exchange. The threads operate asynchronously based on the same global data (PERS – persistent data). Thus, if the value of data in one thread changes, it is automatically updated in the other thread. This is of key importance in the control process, because this solution allows to control an industrial robot using gestures and voice commands without visible delays.

## 2.6. User interface

In addition to the full functionality of this application, which allows for controlling the industrial robot by means of gestures and voice commands, an intuitive graphical interface has been developed, which allows for easy startup and control of the station.

The user interface has been developed in Microsoft Visual Studio 2017 in C# language. This application enables the user to control an industrial robot by hand movement in 3D space (hand tracking and gestures) (Figure 6) and to control the robot by voice commands.



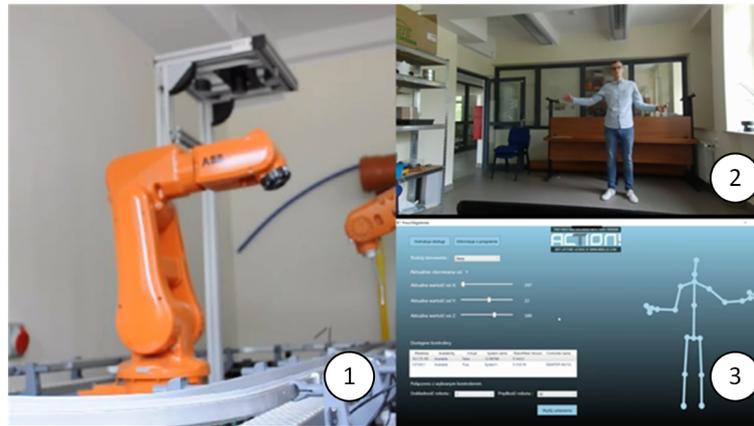
**Figure 6.** Graphical interface of the developed application: 1 – help button, 2 – program information button, 3 – control method list, 4 – current controlled axis, 5 – current x, y, z axis position field, 6 – list of available robots, 7 – robot accuracy window, 8 – robot speed selection window, 9 – start button, 10 – operator's "skeleton" field.

## 2.7. Application tests

The startup tests of the application were performed using digital twin in RobotStudio environment (Figure 4a). The developed digital twin allowed for:

- launching all prepared applications on one computer. The virtual station model was launched in the RobotStudio environment, communication with the application supporting the KINECT sensor and with the user interface was carried out using the localhost,
- launching the prepared applications on two computers. The virtual station model was launched in the RobotStudio environment on one computer – simulation of the operation of a real station. The application supporting the KINECT sensor and the user interface was launched on the second computer. The computers were connected to the local ethernet.

During the tests, the correctness of robot's tasks performance was verified both on the basis of performed gestures and voice commands. The adopted solution allowed execution of the tests. Validation of the application's correctness was carried out on the real station (Figure 4b, Figure 7).



**Figure 7.** Views of the test station: 1 – ABB IRB 120 robot, 2 – operator, 3 – operator interface window.

### 3. Results and Discussion

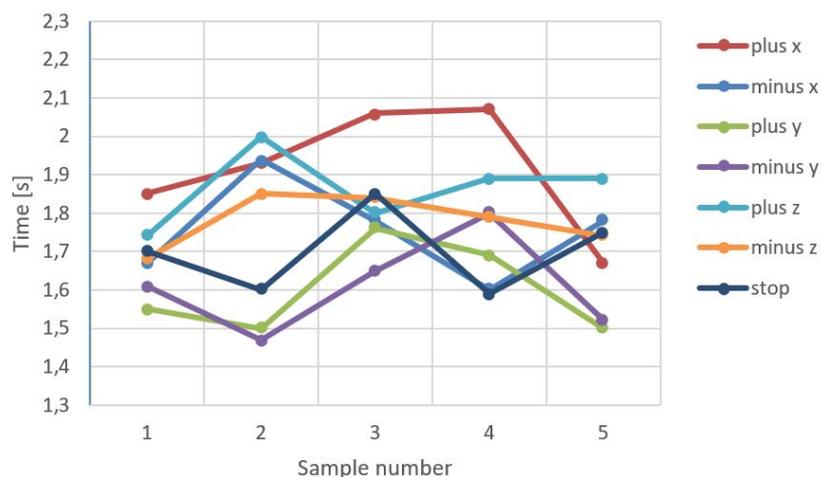
A number of tests were carried out to verify the correctness of robot's reaction to gestures and voice commands. Tests were conducted using a digital twin and a real robot. Both single movements and complex tasks (e.g. pick and place) were tested.

In case of robot control by means of gestures, tests were carried out on the performance of commands by the robot in a form of motion in selected directions (Table 3), following the operator's hand (Figure 7), performing a complex task and controlling the robot's input signals. No communication errors or unrecognized commands were observed. The dynamics of KINECT sensor and, above all, the acceleration and speed capability of the robot are limited when the robot follows the hand movement. This type of control seems to be useful when tasks need to be carried out remotely (in a place that is difficult to reach or far away), and the time of task completion does not play a significant role here. Gestures seems to be a more promising control method than hand tracking, because it may execute complex tasks for the robot (e.g. assembly, sorting). The main advantage in this case is that the operator can start the robot, indicate the task, or turn off the robot without having to use an additional control panel.

**Table 3.** Gestures used in robot control.

|           | Robot action          | Operator's body position                                     |
|-----------|-----------------------|--|
| Command 1 | Axis X move forward   | Right hand on the left, left hand below shoulder             |
| Command 2 | Axis X move backwards | Right hand on the right, left hand below shoulder            |
| Command 3 | Axis Y move forward   | Right hand on the left, left hand between shoulder and head  |
| Command 4 | Axis Y move backwards | Right hand on the right, left hand between shoulder and head |
| Command 5 | Axis Z move forward   | Right hand on the left, left hand above head                 |
| Command 6 | Axis Z move backwards | Right hand on the right, left hand above head                |
| Command 7 | Confirm movement      | Close left hand  |

Tests of the delay of robot's reaction to voice commands were conducted. Selected test results are presented below (Tables 4 and 5).



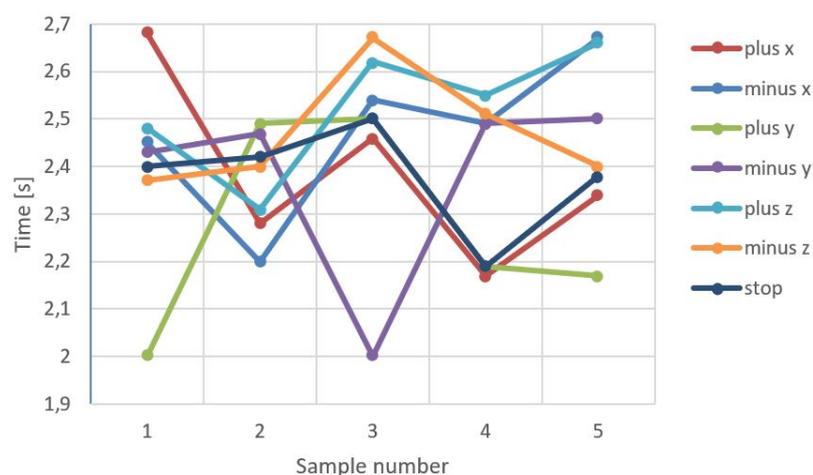
**Figure 8.** Measurement of the delay of robot's reaction to the voice command - real station .

Figure 8 shows delay of the robot's reaction to the voice commands on real station.

**Table 4.** Measurement of the delay of robot's reaction to the voice command -real station.

| Sample Nb         | Commands |           |          |           |          |           |        |
|-------------------|----------|-----------|----------|-----------|----------|-----------|--------|
|                   | "plus x" | "minus x" | "plus y" | "minus y" | "plus z" | "minus z" | "stop" |
| Sample 1 [s]      | 1.85     | 1.67      | 1.55     | 1.61      | 1.74     | 1.68      | 1.70   |
| Sample 2 [s]      | 1.93     | 1.94      | 1.50     | 1.47      | 2.00     | 1.85      | 1.60   |
| Sample 3 [s]      | 2.06     | 1.78      | 1.76     | 1.65      | 1.80     | 1.84      | 1.85   |
| Sample 4 [s]      | 2.07     | 1.60      | 1.69     | 1.80      | 1.89     | 1.79      | 1.59   |
| Sample 5 [s]      | 1.67     | 1.78      | 1.50     | 1.52      | 1.89     | 1.74      | 1.75   |
| Average delay [s] | 1.92     | 1.75      | 1.60     | 1.61      | 1.86     | 1.78      | 1.70   |

Figure 9 shows the delays expressed in seconds for voice control commands on the virtual station.



**Figure 9.** Measurement of the delay of robot's reaction to the voice command -virtual station.

**Table 5.** Measurement of the delay of robot's reaction to the voice command -virtual station.

| Sample Nb         | Commands |           |          |           |          |           |        |
|-------------------|----------|-----------|----------|-----------|----------|-----------|--------|
|                   | "plus x" | "minus x" | "plus y" | "minus y" | "plus z" | "minus z" | "stop" |
| Sample 1 [s]      | 2.68     | 2.45      | 2.00     | 2.43      | 2.48     | 2.37      | 2.40   |
| Sample 2 [s]      | 2.28     | 2.20      | 2.49     | 2.47      | 2.31     | 2.40      | 2.42   |
| Sample 3 [s]      | 2.46     | 2.54      | 2.50     | 2.00      | 2.62     | 2.67      | 2.50   |
| Sample 4 [s]      | 2.17     | 2.49      | 2.19     | 2.49      | 2.55     | 2.51      | 2.19   |
| Sample 5 [s]      | 2.34     | 2.67      | 2.17     | 2.50      | 2.66     | 2.40      | 2.38   |
| Average delay [s] | 2.39     | 2.47      | 2.27     | 2.38      | 2.52     | 2.47      | 2.39   |

Tables 4 and 5 indicate a slower robot response in the RobotStudio environment. There is a difference of 0.92 s between the fastest average response of the real robot as compared to a virtual robot. In both cases, the robot best responds to commands to control along y-axis. The average response rate for the actual device is 1.74 s and for the virtual 2.41 s. This may be due to insufficient computing capacity of the computer used.

Next study evaluates the influence of sound interference intensity (20dB, 40dB, 60dB) on the ability to control the robot by voice commands. The "Sound Master" application was used to measure the interference. Measuring device is located next to the KINECT sensor. A music track without vocals was used as a disturbance. The commands for the robot were given in a calm voice of 70dB intensity. A number of tests were carried out on a virtual and real station, changing the robot's parameters. For 20dB and 40dB interference there were no problems with the performance of commands. Tables below show results for 60dB motion commands.

**Table 6.** Effects of 60dB interference on robot's operation.

| Sample Nb            | Commands |           |          |           |          |           |        |
|----------------------|----------|-----------|----------|-----------|----------|-----------|--------|
|                      | "plus x" | "minus x" | "plus y" | "minus y" | "plus z" | "minus z" | "stop" |
| Sample 1             | ok       | ok        | ok       | ok        | ok       | ok        | ok     |
| Sample 2             | ok       | ok        | ok       | ok        | ok       | ok        | ok     |
| Sample 3             | ok       | ok        | ok       | ok        | ok       | ok        | ok     |
| Sample 4             | ok       | ok        | x        | ok        | ok       | ok        | ok     |
| Sample 5             | ok       | ok        | x        | ok        | ok       | ok        | ok     |
| Sample 6             | ok       | ok        | x        | ok        | ok       | ok        | ok     |
| Sample 7             | ok       | ok        | x        | ok        | ok       | ok        | ok     |
| Sample 8             | ok       | ok        | ok       | ok        | x        | ok        | ok     |
| Sample 9             | ok       | ok        | ok       | ok        | ok       | ok        | ok     |
| Sample 10            | ok       | ok        | x        | ok        | ok       | ok        | ok     |
| Recognition rate [%] | 100      | 100       | 50       | 100       | 90       | 100       | 100    |

Results presented in Table 6 show that commands of 70dB intensity can be incomprehensible for a robot system equipped with KINECT sensor if the interference is at 60dB. The achieved recognition rate of voice commands is equal to 91.43%. This value might be enough for performing laboratory tests but is definitely not acceptable in industrial environment. However, it should be noted that in the case of simple commands (for virtual and real system) the "plus y" command caused the problem, other commands were recognized correctly. Therefore, it is important that the selected commands are easily and clearly identified by the system, especially in industrial conditions (with noise).

#### 4. Conclusions

Human interaction with an industrial robot using a vision system equipped with microphones seems to be an interesting and promising solution for future robot programming. Teaching the trajectory of industrial robots with gestures and voice commands may prove to be one of the methods

of robot programming in the future. The combination of gestures and voice commands also seems to be an interesting issue in terms of robot control, especially since the advantage of using gestures and voice commands in interaction with the machine is the ability to adapt dynamically. For humans, this form of communication seems to be the most intuitive, as we have been learning from childhood to use gestures for more accurate expression of our feelings, emotions, and needs. In our lives (in direct contacts) speech and gestures coexist, and therefore they should also be combined in interaction with machines. Expanding the systems of industrial robots and their control programs with the possibility to control by means of gestures and voice commands will make it easier for operators of production stations to communicate with machines and will make them able to operate extended stations after a short training.

This paper focuses on the presentation of prepared software, which allows to control an industrial robot by means of gestures and voice commands. ABB IRB 120 robot was used as an example, but the developed software can be easily integrated with other robots. For the IRB 120 robot, we have developed a multithreaded control program that allows for a flexible tasks realization. Our aim was to develop a cheap and universal solution. Tests have shown that gesture control allows for closer cooperation between the man and machine. Additional voice commands extend the possibilities of controlling the robot without a need to focus all the operator's attention on it. The presented test results prove that the recognition of voice commands depends on the noise level. It has been noted that the 10dB difference between command and noise provides 91.43% recognition of the commands.

The paper also presents the possibility of performing tests based on digital twins. Already today such an approach allows to create control programs for robots and conduct analyses of the work of robotic stations, and even entire production lines and factories. The simulation results are particularly important for integrators (at the start of production) and managers (who manage production and conduct analyses of market changes). The research carried out, using a real robotic station and a virtual robotic system, has shown that in the case of voice handling, the average difference between real and virtual robot performance is 0.67s. This difference may result from the hardware parameters of the computer, operating system used or communication protocol parameters. Such analyses allow for a quick response to market changes and the introduction of changes in production. The use of digital twins allows to prepare and test the introduced changes without stopping production.

In the next stage of research, the industrial robot will be controlled by means of gestures and voice commands taking into account the robot's safety zones. Moreover, with the use of digital twins, tests of controlling the robot by means of gestures and voice commands in virtual reality with HTC VIVE Cosmos goggles will be conducted.

**Author Contributions:** Conceptualization, W.K. and P.B.; methodology, W.K. and P.B.; software, W.K. and P.B.; validation, P.B., S.B. and J.P.; formal analysis, W.K. and J.P.; investigation, P.B. and W.K.; resources, S.B., W.K. and J.P.; data curation, W.K., P.B. and J.P.; writing—original draft preparation, S.B., W.K. and P.B.; writing—review and editing, S.B., W.K. and J.P.; visualization, W.K. and S.B.; supervision, W.K. and J.P.; project administration, W.K.. All authors have read and agreed to the published version of the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Peursum, P.; Bui, H.; Venkatesh, S.; West, G. Human action segmentation via controlled use of missing data in HMMs. *Proc. 17th Int. Conf. Pattern Recognition, 2004. ICPR 2004. IEEE, 2004*, pp. 440–445 Vol.4. doi:10.1109/ICPR.2004.1333797.
2. Wortmann, A.; Barais, O.; Combemale, B.; Wimmer, M. Modeling languages in Industry 4.0: an extended systematic mapping study. *Softw. Syst. Model.* **2020**, *19*, 67–94. doi:10.1007/s10270-019-00757-6.
3. Bect, P.; Simeu-Abazi, Z.; Maisonneuve, P.L. Identification of abnormal events by data monitoring: Application to complex systems. *Comput. Ind.* **2015**, *68*, 78–88. doi:10.1016/j.compind.2014.12.008.
4. Borys, S.; Kaczmarek, W.; Laskowski, D. Selection and Optimization of the Parameters of the Robotized Packaging Process of One Type of Product. *Sensors* **2020**, *20*, 5378. doi:10.3390/s20185378.

5. van der Aalst, W. *Process Mining*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2016. doi:10.1007/978-3-662-49851-4.
6. Vera-Baquero, A.; Colomo-Palacios, R.; Molloy, O. Towards a Process to Guide Big Data Based Decision Support Systems for Business Processes. *Procedia Technol.* **2014**, *16*, 11–21. doi:10.1016/j.protcy.2014.10.063.
7. Esmailian, B.; Behdad, S.; Wang, B. The evolution and future of manufacturing: A review. *J. Manuf. Syst.* **2016**, *39*, 79–100. doi:10.1016/j.jmsy.2016.03.001.
8. Prusaczyk, P.; Kaczmarek, W.; Panasiuk, J.; Besseghieur, K. Integration of robotic arm and vision system with processing software using TCP/IP protocol in industrial sorting application. 25th Int. Conf. Eng. Mech. 2019, 2019, p. 020032. doi:10.1063/1.5092035.
9. Kaczmarek, W.; Panasiuk, J.; Borys, S. *Środowiska programowania robotów*; Wydawnictwo Naukowe PWN: Warszawa, 2017.
10. Kaczmarek, W.; Panasiuk, J. *Robotyzacja Procesow Produkcyjnych*; 2017.
11. Kaczmarek, W.; Panasiuk, J. *Programming industrial robots (in Polish)*; PWN Scientific Publishing House: Warsaw, 2017.
12. Park, C.B.; Lee, S.W. Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter. *Image Vis. Comput.* **2011**, *29*, 51–63. doi:10.1016/j.imavis.2010.08.006.
13. Faudzi, A.M.; Ali, M.H.K.; Azman, M.A.; Ismail, Z.H. Real-time Hand Gestures System for Mobile Robots Control. *Procedia Eng.* **2012**, *41*, 798–804. doi:10.1016/j.proeng.2012.07.246.
14. Rosenstrauch, M.J.; Pannen, T.J.; Krüger, J. Human robot collaboration - using kinect v2 for ISO/TS 15066 speed and separation monitoring. *Procedia CIRP* **2018**, *76*, 183–186. doi:10.1016/j.procir.2018.01.026.
15. Rosen, R.; von Wichert, G.; Lo, G.; Bettenhausen, K.D. About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine* **2015**, *48*, 567–572. doi:10.1016/j.ifacol.2015.06.141.
16. Bouteraa, Y.; Ben Abdallah, I. A gesture-based telemanipulation control for a robotic arm with biofeedback-based grasp. *Ind. Robot An Int. J.* **2017**, *44*, 575–587. doi:10.1108/IR-12-2016-0356.
17. Bernier, E.; Chellali, R.; Thouvenin, I.M. Human gesture segmentation based on change point model for efficient gesture interface. 2013 IEEE RO-MAN. IEEE, 2013, pp. 258–263. doi:10.1109/ROMAN.2013.6628456.
18. Ghonge, E.P.; Kulkarni, M.N. Gesture based control of IRB1520ID using Microsoft's Kinect. 2017 2nd Int. Conf. Commun. Electron. Syst. IEEE, 2017, pp. 355–358. doi:10.1109/CESYS.2017.8321298.
19. Liu Jingbiao.; Xu Huan.; Li Zhu.; Sheng Qinghua. Dynamic gesture recognition algorithm in human computer interaction. 2015 IEEE 16th Int. Conf. Commun. Technol. IEEE, 2015, pp. 425–428. doi:10.1109/ICCT.2015.7399873.
20. Popov, V.; Ahmed, S.; Shakev, N.; Topalov, A. Gesture-based Interface for Real-time Control of a Mitsubishi SCARA Robot Manipulator. *IFAC-PapersOnLine* **2019**, *52*, 180–185. doi:10.1016/j.ifacol.2019.12.469.
21. Cueva C., W.F.; Torres, S.H.M.; Kern, M.J. 7 DOF industrial robot controlled by hand gestures using microsoft kinect v2. 2017 IEEE 3rd Colomb. Conf. Autom. Control. IEEE, 2017, pp. 1–6. doi:10.1109/CCAC.2017.8276455.
22. Moe, S.; Schjolberg, I. Real-time hand guiding of industrial manipulator in 5 DOF using Microsoft Kinect and accelerometer. 2013 IEEE RO-MAN. IEEE, 2013, pp. 644–649. doi:10.1109/ROMAN.2013.6628421.
23. Jha, A.; Chiddarwar, S.S.; Alakshendra, V.; Andulkar, M.V. Kinematics-based approach for robot programming via human arm motion. *J. Brazilian Soc. Mech. Sci. Eng.* **2017**, *39*, 2659–2675. doi:10.1007/s40430-016-0662-z.
24. Duque, D.A.; Prieto, F.A.; Hoyos, J.G. Trajectory generation for robotic assembly operations using learning by demonstration. *Robot. Comput. Integr. Manuf.* **2019**, *57*, 292–302. doi:10.1016/j.rcim.2018.12.007.
25. Jha, A.; Chiddarwar, S.S.; Andulkar, M.V. An integrated approach for robot training using Kinect and human arm kinematics. 2015 Int. Conf. Adv. Comput. Commun. Informatics. IEEE, 2015, pp. 216–221. doi:10.1109/ICACCI.2015.7275612.
26. Chen, C.S.; Chen, P.C.; Hsu, C.M. Three-Dimensional Object Recognition and Registration for Robotic Grasping Systems Using a Modified Viewpoint Feature Histogram. *Sensors* **2016**, *16*, 1969. doi:10.3390/s16111969.
27. Zhou, Y.; Chen, M.; Du, G.; Zhang, P.; Liu, X. Intelligent grasping with natural human-robot interaction. *Ind. Robot An Int. J.* **2018**, *45*, 44–53. doi:10.1108/IR-05-2017-0089.

28. Wang, F.; Liang, C.; Ru, C.; Cheng, H. An Improved Point Cloud Descriptor for Vision Based Robotic Grasping System. *Sensors* **2019**, *19*, 2225. doi:10.3390/s19102225.
29. Panasiuk, J.; Kaczmarek, W.; Besseghieur, K.; Borys, S.; Siwek, M.; Gołowski, K. Study of the 3D scanning accuracy using industrial robot. 24th Int. Conf. Eng. Mech. 2018, 2018, pp. 641–644. doi:10.21495/91-8-641.
30. Dybedal, J.; Aalerud, A.; Hovland, G. Embedded Processing and Compression of 3D Sensor Data for Large Scale Industrial Environments. *Sensors* **2019**, *19*, 636. doi:10.3390/s19030636.
31. Ullo, S.L.; Zarro, C.; Wojtowicz, K.; Meoli, G.; Focareta, M. LiDAR-Based System and Optical VHR Data for Building Detection and Mapping. *Sensors* **2020**, *20*, 1285. doi:10.3390/s20051285.
32. Perkins, B.; Hammer, J.V.; Reid, J.D. *Beginning C# 7 Programming with Visual Studio® 2017*; Wiley, 2018. doi:10.1002/9781119549550.
33. *Product manual IRB 120*; ABB AB Robotics: Västerås, Sweden, 2014.
34. Kinect v2 Introduction and Tutorial, 2020.
35. Albahari, J.; Albahari, B. *C# 7.0 in a Nutshell: The Definitive Reference*; O'Reilly Media Inc.: Canada, 2017.
36. Troelsen, A.; Japikse, P. *C# 6.0 and the .NET 4.6 Framework*; Apress: Berkeley, CA, 2015. doi:10.1007/978-1-4842-1332-2.
37. Webb, J.; Ashley, J. *Beginning Kinect Programming with the Microsoft Kinect SDK*; Apress: Berkeley, CA, 2012. doi:10.1007/978-1-4302-4105-8.
38. Pagliari, D.; Pinto, L. Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. *Sensors* **2015**, *15*, 27569–27589. doi:10.3390/s151127569.
39. Greenberg, S.; Ainsworth, W.; Popper, A.; Fay, R. *Speech Processing in the Auditory System*; Vol. 18, *Springer Handbook of Auditory Research*, Springer-Verlag: New York, 2004. doi:10.1007/b97399.
40. Communication Developer Support, 2019.