Vol. 06, No. 2 (2024) 875-884, doi: 10.24874/PES06.02A.014

Proceedings on Engineering Sciences

www.pesjournal.net

MACROS AS A PROGRAMMING TOOL FOR SYNCHRONIZATION OF TWO NON-SYNCHRONIZABLE INDUSTRIAL ROBOTS

Bojan Z. Knezevic[1]
Radovan Jalic
Dragan Erceg

Received 02.06.2023.
Received in revised form 02.09.2023.
Accepted 07.11.2023.
UDC – 621.865.8

Keywords:

Industrial robot, Welding robot, Handling robot, Macro Jobs, Synchronization.

A B S T R A C T

The main goal of this paper was to connect two controllers of the older generation into one robotic cell, then software implements their synchronized work on servicing and welding certain objects. Two controllers manufactured by Yaskawa Motoman were used with the associated manipulators. The hardware solution involved connecting the controllers via General I/O circuit board. The software solution involved parallel programming of two robots for synchronized operation on a common task. During programming, it was necessary to create macro jobs that implemented repetitive actions, such as calling and waiting for robots, grabbing and releasing objects, as well as the actions of starting and stopping the program. The conclusion of presented research is that the robotic cell, formed by two robots that are not intended for synchronization, meets the requirements that until now could only be solved with robots of the newer generation. Proposed solution was confirmed by experimental verification.

1. INTRODUCTION

The rapid development of the industry and the distinguished demands of customers have led to working conditions in which man cannot meet the required precision and speed. Consequently, there was a need to introduce robots into all branches of industry. According to the ISO 8373:2021 standard, "an industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or fixed to a mobile platform for use in automation applications in an industrial environment" (ISO 8373:2021). Such robots have brought many advantages in production processes, some of them are better precision and repeatability, work in dangerous conditions, higher speed, efficiency, etc. (Al Mamun & Buics, 2022; Tan et al., 2023).

Depending on their function and purpose, industrial robots are divided into several groups. Two important groups are robots for performing processes such as welding, painting, cutting, and a group of manipulation robots, i.e. robots for serving, packing and sorting (Grau

[1] Corresponding author: Bojan Knezevic
Email: bojan.knezevic@mf.unibl.org

et al., 2017). In this research focus is on robots from both groups, robots for welding and robots for handling.

There are many robot manufacturers, and among the most prominent are Japan's Yaskawa and Fanuc, and Germany's Kuka. Together they make up between 60% and 80% of the total production in the world (Zhang & Zeng, 2022; Nebot, 2018).

Considering that industrial robots are intended for very long-term operation companies still have a large number of robots of the older generation whose age reaches over 15 years. Such robots have significantly modest characteristics compared to robots of newer generations.
In order to use older generation robots in the automation of modern production processes or modern but cheaper robots, it is necessary that they work in synchronisation with other parts of the production system. Exactly one such synchronization was performed in this research between two robots from the Yaskawa company. On one side is a welding robot consisting of a Motoman NX100 controller with a Motoman HP6 manipulator, while on the other side is a service robot consisting of a Motoman XRC controller with a Motoman UP6 manipulator.

2. EXPERIMENTAL SETUP

As already mentioned, the aim of research is to synchronize the work of two robots of an older generation and thus form a robotic cell in which one robot serves another for welding certain parts. It is a combination of two controllers with manipulators. A Motoman NX100 controller with a Motoman HP 6 manipulator was used as a welding robot, while a Motoman XRC controller with a Motoman UP 6 manipulator was used as a service robot. The robot cell experimental setup is shown in Figure 1.

Figure 1. The robot cell experimental setup

In order to be able to fully understand the aim and results of the presented research, the mentioned robotic cells are described below and their basic specifications are given.

2.1 Handling robot

The service robot consists of a controller, manipulator, the teaching pendant and an external control.

The Motoman XRC controller used a RISC (Reduced Instruction Set Computer) processor, which provided it with much better performance compared to its Pentium-based predecessors. This controller, as well as the NX100, has the ability to control up to 4 robots.

Teaching Pendant with 5.7-inch display is used for programming. Connection is possible via the RS 232 protocol as well as via digital inputs and outputs. The programming language used for programming is Inform II (Zhan & Han, 2022; Chung et al., 2008; Zhang et al., 2023).

The characteristics of the Motoman UP6 manipulator are shown in Table 1. The manipulator has 6 degrees of freedom and a maximum load of 6 kg. In Table 1. the maximum speeds for each of the axes are also shown. (UP6 Manipulator Manual, 2001).

More about external control and the safety light curtain will be discussed in the section that describes security.

Table 1. Basic characteristics of the Motoman UP6 manipulator (Zhang & Jia, 2009)

Operation Mode	Vertically Articulated
Degree of Freedom	6
Payload	6 kg
Repetitive Positioning Accuracy	±0.08 mm
Mass	130 kg
Power Capacity	1.5 kVA
Motion Range	
S-Axis (turning)	±170°
L-Axis (lower arm)	+155°, -90°
U-Axis (upper arm)	+190°, -170°
R-Axis (wrist roll)	±180°
B-Axis (wrist pitch/yaw)	+225°, -45°
T-Axis (wrist twist)	±360°
Maximum Speed	
S-Axis	2.44 rad/s, 140°/s
L-Axis	2.79 rad/s, 160°/s
U-Axis	2.97 rad/s, 170°/s
R-Axis	5.85 rad/s, 335°/s
B-Axis	5.85 rad/s, 335°/s
T-Axis	8.73 rad/s, 500°/s
Allowable Moment	
R-Axis	11.8 N*m
B-Axis	9.8 N*m
T-Axis	5.9 N*m
Allowable Inertia	
R-Axis	0.24 kg*m²
B-Axis	0.17 kg*m²
T-Axis	0.06 kg*m²

2.2 Welding robot

The robotic welding cell consists of a controller NX100, a Motoman HP6 manipulator and a teaching pendant. In addition, complete welding equipment is necessary, but since it is not the subject of presented research, it will not be discussed further.

The NX100 controller with the associated touch-sensitive teaching pendant represented a real revolution at the time when its production began. The programming language Inform III is used for programming. Fast data processing, a large memory that supports up to 60,000 steps, the ability to control multiple robots (maximum of 4 robots and 36 axes, including robotic and external axes), and high path precision are some of the advantages of this controller. What particularly characterizes this controller is its ability to connect via Ethernet and RS 232 protocol, as well as a large number of digital inputs and outputs (Abhishek et al., 2022; Thamilarasi et al., 2010).

The characteristics of the Motoman HP 6 manipulator are given in Table 2. where you can see that it has 6 degrees of freedom of movement, and you can see the maximum range and speed in all 6 axes and other important characteristics. The maximum load that the manipulator can carry is 6 kg (Motoman HP6 Instructions 2007).

Table 2. Basic characteristics of the Motoman HP 6 manipulator (Motoman HP6 Instructions, 2007)

Operation Mode	Vertically Articulated
Degree of Freedom	6
Payload	6 kg
Repetitive Positioning Accuracy	±0.08 mm
Mass	130 kg
Power Capacity	1.5 kVA
Motion Range	
S-Axis (turning)	±170°
L-Axis (lower arm)	+155°, -90°
U-Axis (upper arm)	+250°, -175°
R-Axis (wrist roll)	±180°
B-Axis (wrist pitch/yaw)	+225°, -45°
T-Axis (wrist twist)	±360°
Maximum Speed	
S-Axis	2.62 rad/s, 150°/s
L-Axis	2.79 rad/s, 160°/s
U-Axis	2.97 rad/s, 170°/s
R-Axis	5.93 rad/s, 340°/s
B-Axis	5.93 rad/s, 340°/s
T-Axis	9.08 rad/s, 520°/s
Allowable Moment	
R-Axis	11.8 N*m
B-Axis	9.8 N*m
T-Axis	5.9 N*m
Allowable Inertia	
R-Axis	0.24 kg*m²
B-Axis	0.17 kg*m²
T-Axis	0.06 kg*m²

3. HARDWARE CONFIGURATION FOR PROPOSED SYNCHRONIZATION

The previously described robots need to be connected so that the functions of object manipulation and welding in synchronization. It is stated that both the XRC controller and the NX100 controller have the ability to control multiple manipulators (maximum 4). At first, this problem seems easily solvable, but the problem is actually much more difficult.

The fact is that many companies in countries in transition have started with partial automation of production facilities in accordance with their possibilities. The consequence of the above is that no consideration was given to the later connection of individual parts of the production plant, and robots were purchased according to the current needs and possibilities. This has led to the situation that companies have several robots that are of different generations and different manufacturers and that are not compatible for connecting to one common controller (Ferreira et al., 2022).

Now that companies want to raise the level of automation, it is necessary to overcome the problem of incompatibility. In order to solve the problem, it is necessary to design a universal way of communication between the controllers.

An attempt was made to overcome the problem using offline programming, but since the XRC as well as newer NX100 controller does not supports this type of programming, as well as lot another type of robots at the market, it was necessary to find a more universal solution.

Starting from the assumption that robots should communicate like humans, the solution to the problem presented in this paper is that controllers can communicate via digital inputs and outputs according to the Master-slave principle, where one controller will be the master and the other is subordinate to it (slave) - who will execute the received commands (Chen et al., 2022).

Considering that the handling robot is the one intended for automation in the company, it was chosen as the Master controller. For communication with the Slave robot (the welding robot), digital inputs and outputs are used. Another type of communication and connection is not possible, for example via communication interfaces, because they are also incompatible.

The XRC and NX100 controllers have many digital inputs and outputs. For this communication, on the XRC controller, the General I/O circuit board - connector CN10 was used (Figure 2.). A bus is derived from this connector and components that communicate through digital signals are connected to it.Some of these inputs and outputs have already been used for various functions such as start, stop and light sensor curtain buttons, for communication with the gripper and safety elements, etc. For communication with the NX100 controller, output A14 (#OT7 in the program) and input A2 (#IN4) were used. Also, in order for the system to function, it is necessary to equalize the reference potentials, and for this purpose connector B7 was used.

Now the two controllers are connected so that the signal from the digital output of the A14 Master controller is sent to the digital input of the A4 Slave controller, and the signal from the digital output of the A9 Slave controller is sent to the digital input of the A2 Master controller.

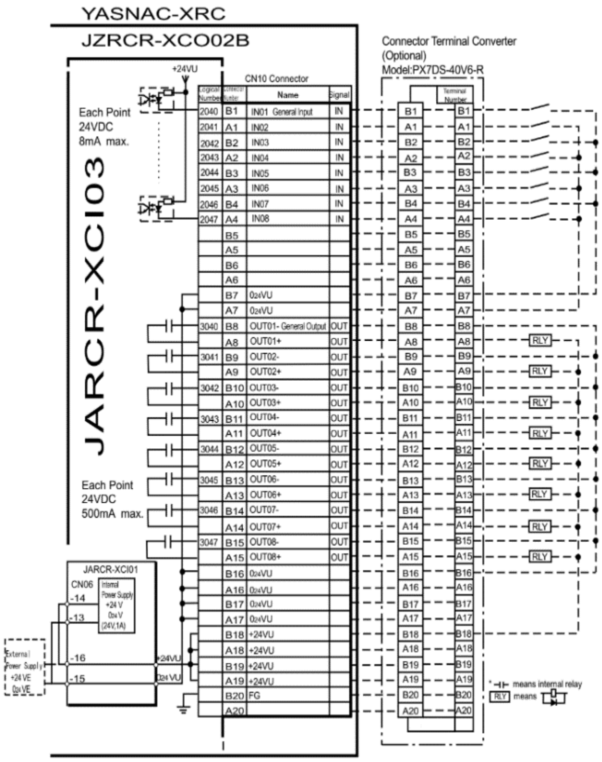

Figure 2. General I/O circuit board, connector CN10 (XRC Instruction Manual, 2001)

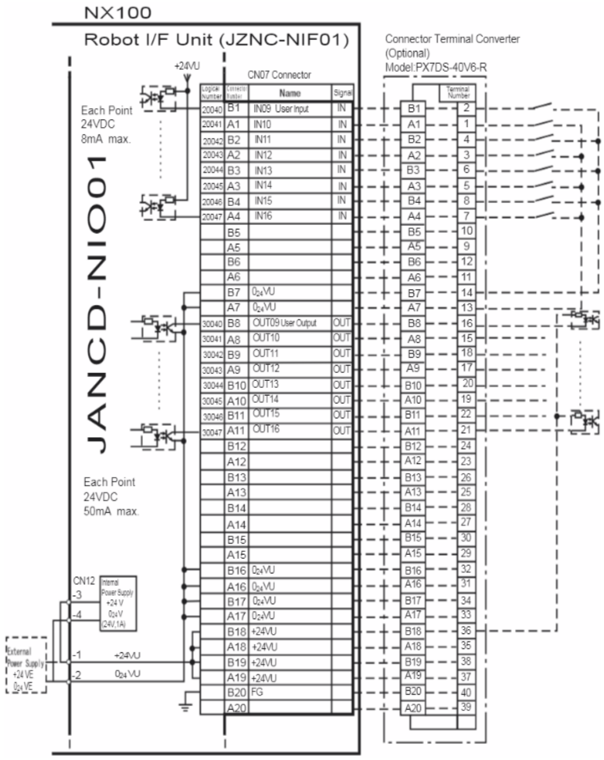

Figure 3. General I/O circuit board, connector CN07 (NX100 Controller Manual, 2004)

The same was done on the NX100 controller. It also uses the General I/O circuit board for communication, but this time the CN07 connector (Figure 3.). On the derived bus, it can be clearly seen that output A9 (#OT 4) and input A4 (#IN8) were used for communication with the XRC controller. Connector B7 was used for common grounding.

In Figure 4. the communication method is illustrated where the controllers of two robots communicate with each other through low-voltage signals, and each of these controllers further sends the information to its manipulator.

With this, the hardware part of the problem has been solved and we are going to the software realization of the synchronization.

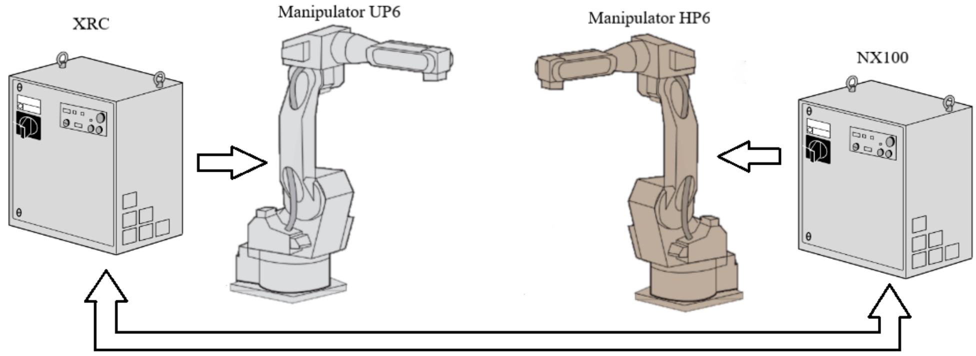

Figure 4. Controllers and manipulators connection scheme

4. SOFTWARE IMPLEMENTATION FOR PROPOSED SYNCHRONIZATION

After the hardware connection, it is necessary to program the robots to work on a common task. The Master robot takes the lead and at some moments he calls the Slave robot and instructs him to perform his part of the task. After completing that task, it returns information to the Master robot, which calls it again when necessary. This kind of programming would lead to very complicated programs because, depending on the complexity of the task, these programs would be long, with a large number of repeated instructions.

In addition, there are also difficulties such as the increased possibility of errors, difficulties when eliminating them, and more difficult creation and understanding of programs by the operator (Bouteraa & Ghommam, 2009; Sato et al., 2007).

In order to avoid all the listed problems during programming, Macro jobs were created. To begin with, it is crucial to familiarize oneself with macro programming.

In robot programming, routinely performed tasks such as welding inside and outside corners, initiating programs where specific conditions must be met, and control peripheral devices are commonplace. Storing these actions as macros enables their reuse in various programming tasks. The use of a well-organized macro library, sorted by tasks, results in reduced programming time, improved programming efficiency, and enhanced program structure (Twarog & Zeslawska, 2018).

Macros in robot programming refer to customized sequences of instructions created by users. These sequences encompass path data, technology information, welding parameters, and logical instructions. Figure 5. illustrates common types of macros utilized in robot programming.

The structure of the macro module, depicting how macros are created, deleted, and called in the program, is illustrated in Figure 6.

From illustration on Figure 6, it can be observed that it is possible to create a library of macro tasks and easily invoke them during programming.

The main purpose of macro tasks in programming robots for welding is to simplify the creation of welding programs that involve frequently repeated paths. One of the fundamental challenges of macro programming is that geometric measurements and tool orientations stored in macro paths can differ from those needed for other workpieces. Therefore, geometric operations are necessary to adjust macro data. Possible geometric operations include: Move, Rotate, Mirror, Scaling, Inversion, and Adaptation0 (Figure 7.).

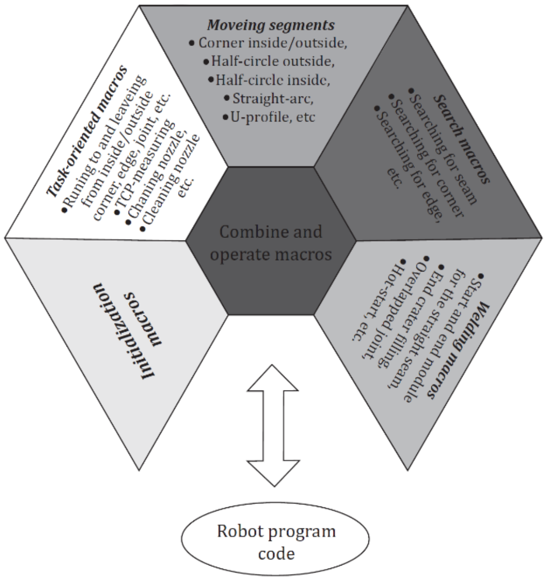

Figure 5. Types of macros utilized in robot programming

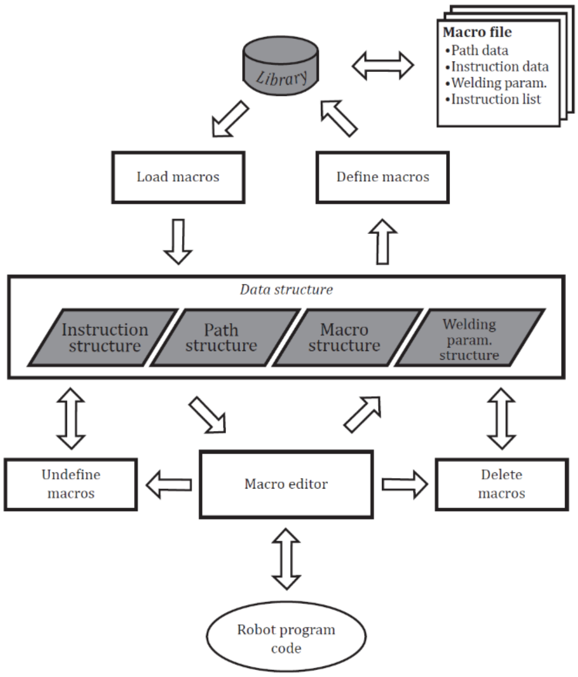

Figure 6. Structure of the macro module

In proposed solution, given the challenge of achieving synchronized operation of robots not designed for such collaboration, and considering the communication method via digital inputs and outputs, macro tasks were employed. These tasks were used for robot communication, gripper operation, as well as checking the fulfilment of conditions for starting and stopping.

They bring simplicity of programming, transparency of the program and easier debugging (Tan et al., 2023). Programming and editing of Macro jobs is possible only when the controller is in Management mode. The name of the Macro program can be entered among the instructions in the Inform list, and this is also done in

Management mode. This solves the availability of Macro jobs during programming.

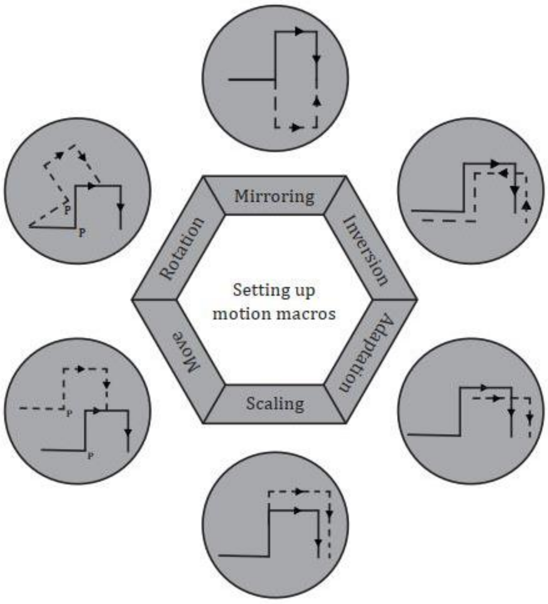

Figure 7. Geometrical operations

The proposed solution implies the creation of 6 new original Macro jobs were created on the Master robot: CALL_R2, WAIT_R2, CATCH, RELEASE, START, STOP (Figure 8).

For easier programming and marking, in the following text, the Master robot will be marked with R1, while the Slave robot will be marked with R2.

The first Macro job, CALL_R2, (Figure 8 a), represents the initial part of the protocol between robots, where the Master robot calls the Slave robot to perform its part of the task. The proposed initial part of the protocol is defined as a pulse signal with an amplitude of 24 VDC and a duration of 0.1 seconds.

In Figure 8 b the Macro job named WAIT_R2 is shown, which realizes the function of waiting for a signal from the Slave robot. Program execution does not continue until the Slave robot sends a signal to the Master.

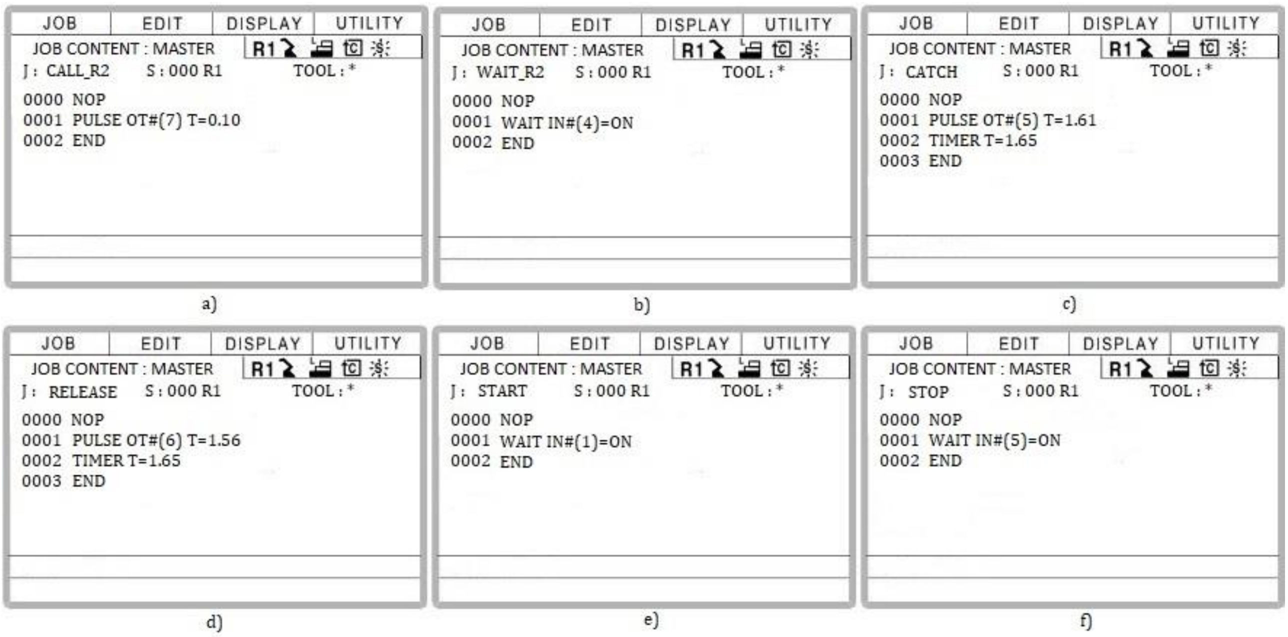

Figure 8. Macro jobs that were created originally for the proposed solution

These two Macro jobs were also created on the Slave robot, only their names are CALL_R1 and WAIT_R1. Depending on the requirements, the program developer can implement three modes of operation:

- The master robot executes part of the program while Macro job WAIT_R1 is activated on the slave robot.
- The slave robot executes part of the program while Macro job WAIT_R2 is activated on the Master robot.
- Both robots execute parts of the program, with the robot that finishes its task first, activates Macro job WAIT.

Macro jobs CATCH and RELEASE shown in Figures 8 c and 8 d refer to the operation of the gripper. CATCH Macro job realizes the function of catching an object by sending an impulse signal to the gripper motor. If it is necessary to define the gripping time for the operation of the gripper, it is possible to introduce a timer in the instruction that will ensure that the duration of the signal is within a certain time interval. Also, in this case, it is necessary to introduce an additional timer, which will prevent further execution of the program and thus ensure a safe grip.

Macro job RELEASE accomplish the function of releasing objects, by sending a reverse polarity signal to the gripper motor, which is realized using a built-in relay.

Macro jobs START and STOP (Figures 8 e and 8 f) are used when starting and stopping the program, and are implemented using the wait function. Since the buttons are placed on the external control, at the beginning of each program it is necessary to use START to start the program execution. When the start button is pressed, the controller receives a signal at input A1, and thus the Macro job START is executed.

An example of calling Macro jobs is shown in Figure 9 where the initial part of the program, which realizes the function of bringing and welding items, is separated. Calling Macro jobs is very simple, because they can be added among the standard instructions in the Inform List.

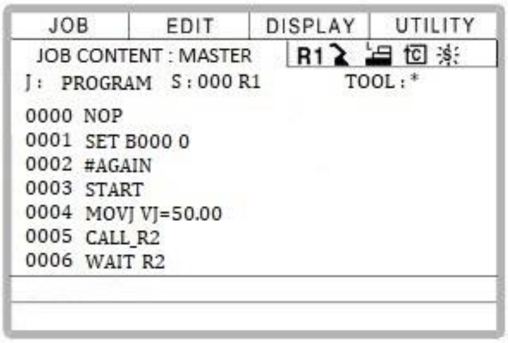

Figure 9. An example of calling a Macro job in a program

5. EXPERIMENT AND RESULTS

To verify the proposed solutions, a program was created by which the Handling robot places the parts on the work table, and the Welding robot welds them. For these needs, the parts that the robots manipulate were modelled and manufactured (Figure 10.).

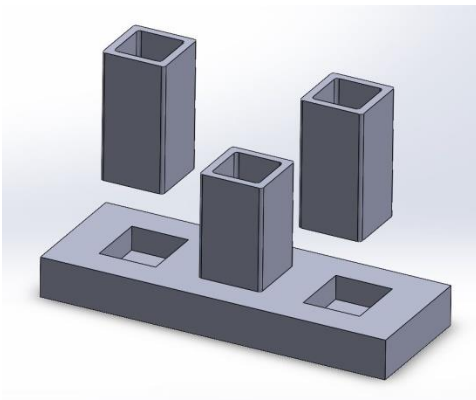

Figure 10. Models created to validate proposed solutions

During programming, a parallel approach was used where both robots were programmed at the same time. In Figure 11. a diagram of the operation of the robot is shown, and based on this diagram, the mode of operation was created and will be explained here. Points marked with numbers 1 to 11 represent places where robots communicate with each other.

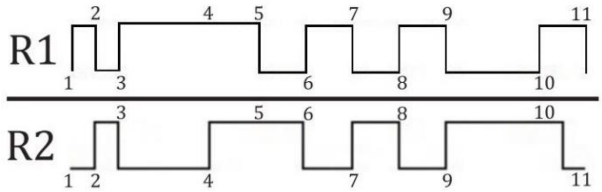

Figure 11. Comparative diagram of two robots working in synchronization

In point 1, the program starts by pressing the Start button located on the external control calling the Macro job START.

Then R1 goes to the parking position and after that, calling the Macro job CALL_R2 (point 2), it sends a signal to another robot that should also position itself in the parking place. While R1 is being positioned, macro job WAIT_R1 is active on R2 and vice versa.

At point 3, R2 sends a signal to robot R1 that it is ready and then R1 starts working. First, he brings the base item and places it in position on the workstation, and then brings an additional part that is welded to the base part.
At point 4, R1 signals that R2 can start welding by calling the Macro job CALL_R2.

During this time, R1 goes to get the second part and since this operation is shorter than the welding operation, he calls the Macro job WAIT_R2 in a safe position, so that the two manipulators do not collide (point 5).

Point 6 marks the moment when robot R2 finishes welding, and from a safe point sends a signal to robot R1 that it can place the second part as well.

After placing the second part, R1 sends a signal to the robot R2 that it can start welding (Point 7).

When he finishes welding the closer sides (point 8), robot R1 rotates the assembly by 180° around the Z-axis, so that R2 can proceed to welding the remaining sides of the objects that were previously inaccessible (point 9).
At point 10, robot R2 has finished welding and sends a signal to robot R1. Then he goes to the parking position, while robot R1 takes the welded assembly from the workstation.

In point 11, the JUMP function is used, which returns the program to the beginning. The idea is to bring the objects into the grasping position using batch loading so that the Handling robot always picks up one type of object from the same position.

When welding the created assembly, considering that it is a quadrangular model, three sides were welded on each of them, in accordance with the defined welding technology (Figure 12.).

In the first position, the sides marked in red are welded in numbered order. Sides 1 and 2 are welded in part 4-6 of the diagram in Figure 11., while side 3 is welded in part of diagram 7-8. After rotating the assembly along the z-axis, sides 4, 5 and 6 are welded, which is the period between points 9 and 10 on the diagram.

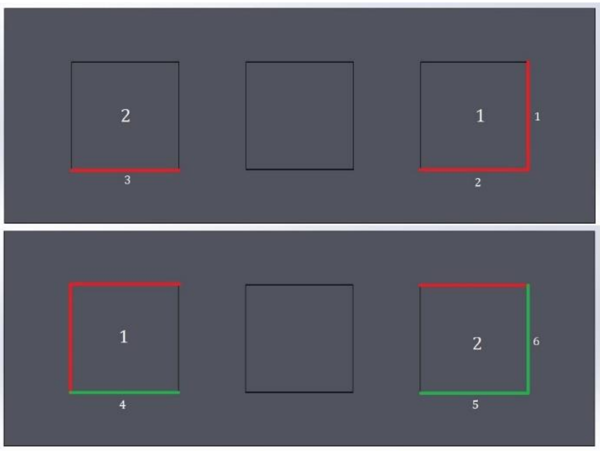

Figure 12. Welding sequence

After demonstrating the working principle on a simple example of welding basic models, it is necessary to show that the proposed solution also meets more challenging requirements typically encountered in the industry. To prove this, a simulation of welding a more complex model, representing a support structure for a device produced by a local company, was performed. Due to the significant heat released during the welding process, it is essential to securely clamp the object being welded to prevent shifting and bending of assembled parts. To ensure all of the above, a construction with screws holding all parts of the welding subject is necessary.

In Figure 13, the support being welded and the mentioned clamp (a structure for holding all parts of the model) mounted on the manipulator of R1 robot are shown.

To fully weld this complex model, it is necessary to weld 22 edges. Since it is difficult to determine what the bracket actually consists of from the previous edge, the mentioned object is modelled in the 3D modelling software package, SolidWorks. In Figure 14, the modelled object is shown from two angles with marked edges to be welded (green lines).

Figure 13. The welding item mounted on the manipulator

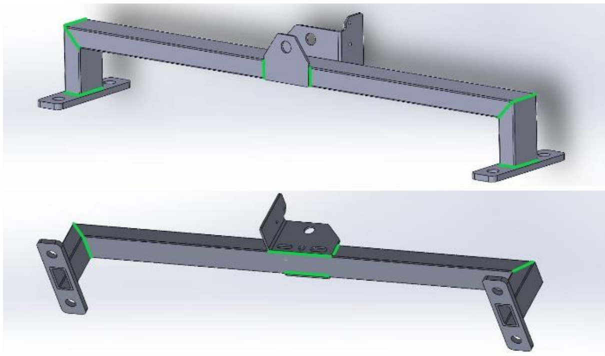

Figure 14. Model with highlighted welding edges

To weld all the mentioned edges, considering that the construction ensuring the stability of the model during welding hinders the access of the welding robot, it is necessary for the servicing robot to take 8 suitable positions during the welding process, enabling access to individual edges. This is evident from the diagram shown in Figure 15.

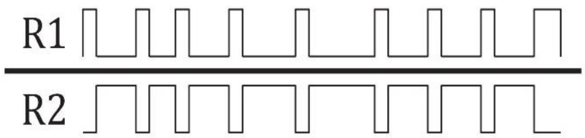

Figure 15. Comparative diagram during the welding process of a complex model

The upper positions on the diagram represent the robot's active states, while the lower ones indicate its idle states, as was the case in the previous example. Each change involves calling a macro for communication (CALL_R2), followed by waiting for a return signal while invoking the WAIT_R2 macro.

The first activation of robot R1 signifies moving to the first position from the parking position. The next 7 activations achieve positioning in the corresponding positions to allow R2 to access specific edges. The ninth activation indicates returning to the parking position.

In the first activation and at the initial position of robot R1, robot R2 welds two edges located on the bottom side of the bracket. In the second activation, spot welding is performed at two locations, ensuring the stability of the construction. The third position of robot R1 allows welding 2 edges, the fourth 4, the fifth 6, the seventh and eighth 3 each, and the ninth 2 edges, followed by returning to the parking position. Robot R2 begins welding after receiving the signal sent by the CALL_R2 macro, and upon completion of the welding, it sends a signal to the digital input of robot R1, deactivating the WAIT_R2 macro.

4. CONCLUSION

The solution proposed in the paper realizes the synchronization of two robots that do not have a built-in synchronous operation option. In the research, a joint cell was realized that works in harmony on the manipulation and welding of certain parts. The robots are connected in a Master-Slave relationship. The Handling robot have a role of the Master robot in the Master-Slave control system, while the Welding robot is in the role of Slave robot.

The interface is made using digital inputs and outputs where the robots communicate with each other using low voltage impulse signals. For the successful implementation of synchronous programming, it is suggested to use Macro jobs, which define the methods of communication between robots. One Macro job is called when a robot sends a pulse signal to another robot, while another Macro job is used when the robot is waiting for a signal from another robot. In addition to these, Macro jobs were created for the start and stop functions, as well as for gripper management.

Proposed solution based on macro programming is capable to effectively respond to various tasks that was shown on the example of welding an assembly consisting of three parts.

Synchronization of the operation of several robots is inherent only to expensive controllers of the newer generation, which, with their performance in the form of connecting and managing a large number of manipulators, ensure easy programming and solving complex problems.

The robots described in this paper are not compatible with other controllers, they do not have the possibility of offline programming, while the XRC controller does not even have an Ethernet connector. With the proposed solution, problem of using robots of the older generation into industry automation has been conceptually and practically solved. By connecting and programming the robot in the manner previously described, it is possible to achieve identical goals as with newer controllers. It is only necessary, depending on the manufacturer and purpose of the robot, to determine free and appropriate digital inputs and outputs, physically connect them, create Macro jobs and program the robots to operate on a specific task.

References:

Abhishek, R., Hasa, S., Ranjan, A., Kumar, N., & Ghosh, S. (2022). Research Paper on the Application of Robot Welding Technology. *International Journal of Creative Research Thoughts (IJCRT), 10*(8), 683–689.

Al Mamun, A. A., & Buics, L. (2022). Automated assistance in hazardous manufacturing processes to improve productivity and reduce workers bodily risks: A literature review. *International May Conference on Strategic Management IMCSM 2022, 18*(1), 193–209, Bor.

Bouteraa, Y., & Ghommam, J. (2009). Synchronization control of multiple robots manipulators. *6th International Multi-Conference on Systems, Signals and Devices*. doi:10.1109/ssd.2009.4956742

Chen, H., Ma, H., Jiang, H., Lv, S., Li, Y., & Liu, H. (2022). The Influence of Control Parameters on Precision of Welding Seam Tracking in Manually Control Master-slave Robot Remote Welding System. *Journal of Physics, Conference series 2218*, 1-6. doi:10.1088/1742-6596/2218/1/012045

Chung, G.-J., Kim, D.-H., & Park, C.-H. (2008). Analysis and Design of Heavy Duty Handling Robot. IEEE Conference on Robotics, Automation and Mechatronics, Chengdu, 774-778, China. doi: 10.1109/RAMECH.2008.4681502

Ferreira, J. P., Simões, I., Coimbra, A. P., & Crisóstomo, M. (2022). Human–Machine Interfaces to NX100 Controller for Motoman HP3L Robot in Educational Environment. In book: Applied Technologies. doi: 10.1007/978-3-031-03884-6_3

Grau, A., Indri, M., Bello, L. L., & Sauter, T. (2017). Industrial Robotics in Factory Automation: From the Early Stage to the Internet of Things. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, 6159–6164.

ISO 8373:2021. (2021). Industrial robot – Part 3.6: Automatically controlled, reprogrammable multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or fixed to a mobile platform for use in automation applications in an industrial environment. International Organization for Standardization. Geneva.

Motoman HP6 Instructions (2007). Retrieved from https://www.manualslib.com/manual/2147614/Yaskawa-Motoman-Hp6-Yr-Hp6-A00.html

Nebot, E. (2018). Robotics: From Automation to Intelligent Systems. *Engineering*, *4*(4), 446–448.

NX100 Controller Manual (2004). Retrieved from https://dokumen.tips/documents/nx100-controller-manual.html?page=107

Sato, T., Hashimoto, M., & Tsukahara, M. (2007). Synchronization based control using online design of dynamics and its application to human-robot interaction. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 652-657, Sanya, China. doi: 10.1109/ROBIO.2007.4522239.

Tan, A. H., Bejarano, F. P., Zhu, Y., Ren, R., & Nejat, G. (2023). Deep Reinforcement Learning for Decentralized Multi-Robot Exploration With Macro Actions. *IEEE Robotics and Automation Letters*, *8*(1), 272-279. doi: 10.1109/LRA.2022.3224667

Thamilarasi, P., Ragunathan, S., & Mohankumar, E. (2010). Automation of robot welding using Machine vision - A review. *International Conference on Communication and Computational Intelligence* (INCOCCI), 266-268, Erode, India.

Twarog, B., & Zeslawska, E. (2018). Synchronization of distributed robotics control system. ITM Web of Conferences, 21, 00007. doi:10.1051/itmconf/20182100007

UP6 Manipulator Manual (2001). Retrieved from https://www.manualslib.com/manual/2013782/Yaskawa-Motoman-Up6.html#manual.

XRC Instruction Manual (2001). Retrieved from https://icdn.tradew.com/file/201606/1569362/pdf/7055804.pdf

Zhang, Y., & Jia, Y. (2009). The Control of Archwire Bending Robot Based on MOTOMAN UP6. *2nd International Conference on Biomedical Engineering and Informatics.* doi:10.1109/bmei.2009.5305333

Zhang, Z., & Zeng, J. (2022). A Review on Development and Application of Industrial Robot. *Academic Journal of Science and Technology*, *2*(2), 78-81

Zhan, Z., & Han, X. (2022). Design and Research of 5 DOF Material Handling Robot. *5th World Conference on Mechanical Engineering and Intelligent Manufacturing* (WCMEIM), 800-805, Ma'anshan, China. doi: 10.1109/WCMEIM56910.2022.10021498.

Zhang, Z., Guo, F., Tang, Q., & Chen, J. (2023). Handling robot non-fixed high-velocity handling method. *Journal of Physics: Conference Series*, 1–7. doi:10.1088/1742-6596/2492/1/012025

Bojan Z. Knezevic
Faculty of Mechanical engineering,
University of Banja Luka,
Banja Luka,
Bosnia and Herzegovina
bojan.knezevic@mf.unibl.org
ORCID 0000-0002-5048-6446

Radovan Jalic
Faculty of Mechanical engineering,
University of Banja Luka,
Banja Luka,
Bosnia and Herzegovina
radejalic@gmail.com
ORCID 0009-0008-4750-1634

Dragan Erceg
Faculty of Mechanical engineering,
University of Banja Luka,
Banja Luka,
Bosnia and Herzegovina
dragan.erceg@mf.unibl.org
ORCID 0009-0005-2075-4728