APPLIED INFORMATION TECHNOLOGY SESSION

ROBOT MOVEMENT PROGRAMMING FOR FLEXIBLE CELL IN "OPEN CIM SCREEN"

Petar Jakovljević[1]*,
Đorđe Dihovični[2],
Nada Ratković Kovačević[2]

[1]Polytechnic – school for new technologies,
Belgrade, Serbia

[2]Academy of Applied Technical Studies Belgrade,
Belgrade, Serbia

Abstract:

Industrial robots have become an important part of every technical institution. The experiment was held on the flexible manufacturing cell for robot movement and cell control. The executed robot movements are transferred to the software for the model used called SCORBASE. The software is able to control the entire flexible cell, including the robot movement, the control of the machine door and the clamping device, and the transport of the parts with pneumatic feeding. The robot is mounted on a mobile axis, which allows additional movement of the robot, as the robot is located in a flexible cell with two machines (CNC Mill and CNC Lathe) and pallets for picking up and placing objects.

In this paper, the movement of the robot in its workspace is described using the positions created with the teach pendant device for manual robot movement. These stored positions are used to program part priorities for transferring individual parts from the part feeders to the machine for the manufacturing process and then back to the pallet for the finished parts. The part priorities described in this paper are used to supply machines with six parts, four parts on the CNC lathe and two parts on the CNC Mill.

Keywords:

Robotics, Artificial intelligence, Programming, Automation, Education.

INTRODUCTION

Flexible manufacturing cells are integrated manufacturing systems that consist of processing machines and automated systems for workpiece manipulation. A flexible manufacturing cell is used to machine one type of machine part. When a product is changed, the entire cell must be changed in terms of the arrangement of the machines, the automated systems, and the programs with which this cell operates. It is necessary to machine six workpieces according to certain priorities: Four on the CNC lathe and two on the CNC mill. Depending on the priority, the robot picks up the workpieces from a pneumatic or a gravity feeder. In the pneumatic feeder, the parts are prismatic while in the gravity feeder the parts are cylindrical. The robot inserts the workpiece into the machine and waits until the machine has finished the process. After processing, the robot takes the processed part out of the machine and places it on the pallet place.

Correspondence:
Petar Jakovljević

e-mail:
pjakovljevic51@gmail.com

DOI: 10.15308/Sinteza-2023-235-241

The robot is programmed on two computers using SCORBASE software for Scorbot ER - 4. The first computer is used for programming the part priorities, while the second computer is responsible for moving the robot, activating the part feeder, opening and closing the machine doors, and opening and closing the machine clamps [1].

2. CELL SPECIFICATION

A flexible cell on which the process is carried out consists of (Figure 1):

1. Emco Concept 55 Mill;
2. Emco Concept 55 Turn;
3. Robot SCORBOT ER-4;
4. Robot mobile axis;
5. Pneumatic Feeder;
6. Gravity Feeder;
7. Palette for mill parts;
8. Palette for turn parts;
9. Computer for part priority programming;
10. Computer for Robot and cell control; and
11. Air compressor.

The robot is connected to two computers that have a program to control the robot and the entire flexible cell. The entire flexible cell is connected to a single air compressor that supplies compressed air to the machines for opening and closing the door and the clamps, and to the pneumatic feeder for pushing the workpieces. The pneumatic feeder consists of a cylinder that is pulled out with compressed air. The cylinder pushes the part into the pick-up position. The parts are stacked on top of each other, when the cylinder returns the next part is ready to be pushed into the pick-up position. SCORBOT - ER 4 is an anthropomorphic configuration with 5 rotary joints. With the gripper attached, the robot has 6 degrees of freedom of movement [2]. This allows unrestricted positioning and orientation in a large working area. It has exceptional flexibility, reflected in its ability to avoid obstacles and enter inaccessible areas. It can move at high speed but has lower accuracy and variable resolution [3]. The cell is also equipped with a device called Teach Pendant. Teach Pendant is used to move the coordinate system of the gripper or robot joints throughout the workspace and save the positions in the SCORBASE software using the online mode [4], [5]. Online mode represents controlling the robot via cable connected device or manually guiding the gripper by hand.

The memory space in the robot controller is reserved for the definition of positions, so that recording positions is possible, which is defined using points, the number of points can be between 1 and 100. The points that define the robot positions are:

- 100-149 – Movement towards the Mill;
- 150-199 – Movement from the Mill;
- 200-249 – Movement towards the Turn;
- 250-299 – Movement from the Turn;

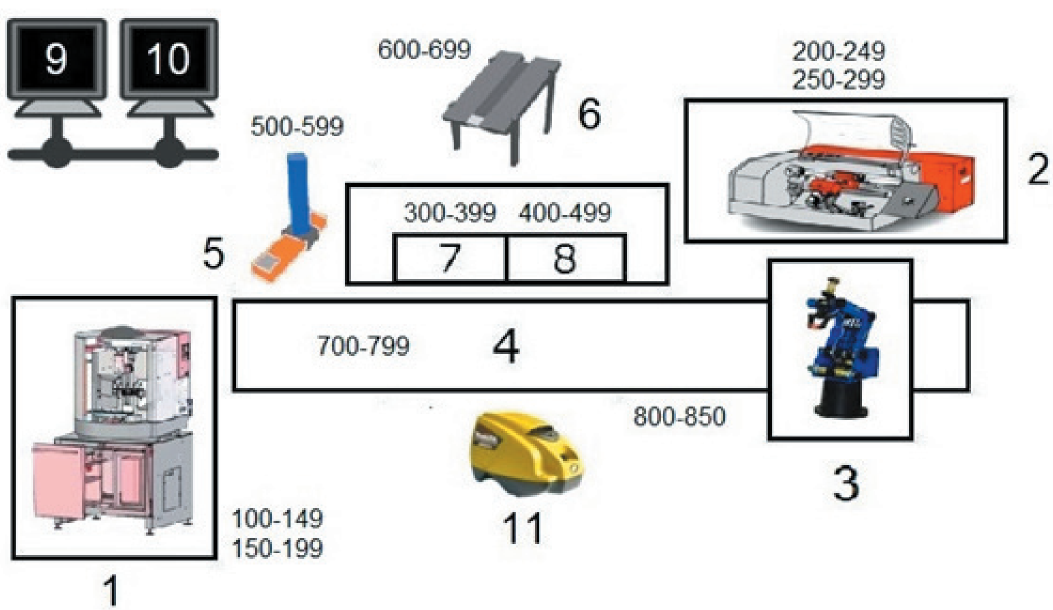

Figure 1 – Cell Specification.

- 300-399 –Palette for finished Mill parts;
- 400-499 – Palette for finished Turn parts;
- 500-599 – Picking the parts from the Pneumatic Feeder;
- 600-699 – Picking the parts from the Gravity Feeder
- 700-799 – Movement on the mobile axis and
- 800-850 – Waiting state.

3. ROBOT MOVEMENT AND CELL CONTROL

Planning the robot movement and controlling the cell operation is a complex and demanding task which can be solved in different ways [6], [7], [8].

To service the machines, the robot moves along the points on the mobile axis (700 - 799) to reach the pneumatic feeder for milling parts or the gravity feed for turning parts. After moving to one of the feeders, the robot opens the gripper, moves to the removal position, and removes the part (500 - 599) from the pneumatic feeder or (600-699) from the gravity feeder.

The gripper closes and the robot returns to its position after moving to one of the feeders. The robot moves towards the machine (100 - 149) to the CNC mill or (200 - 249) to the CNC lathe, the machine door opens, and the gripper is released, the robot takes the position to place the part in the machine, then places the part in clamping device, waits two seconds for the clamping device to close and moves away (150 - 199) from the mill or (250 - 299) from the lathe.

The machine door closes and the robot returns to its position after reaching the machine. After machining the part, the machine door opens, the robot enters the machine work area, grips the finished part, the clamp releases and the robot returns to the position where it reached the machine. The machine door closes, and the robot moves to pallet 1 or pallet 2 and places the finished part on a pallet (300 - 399) for mill finished parts and (400 - 499) for turn parts.

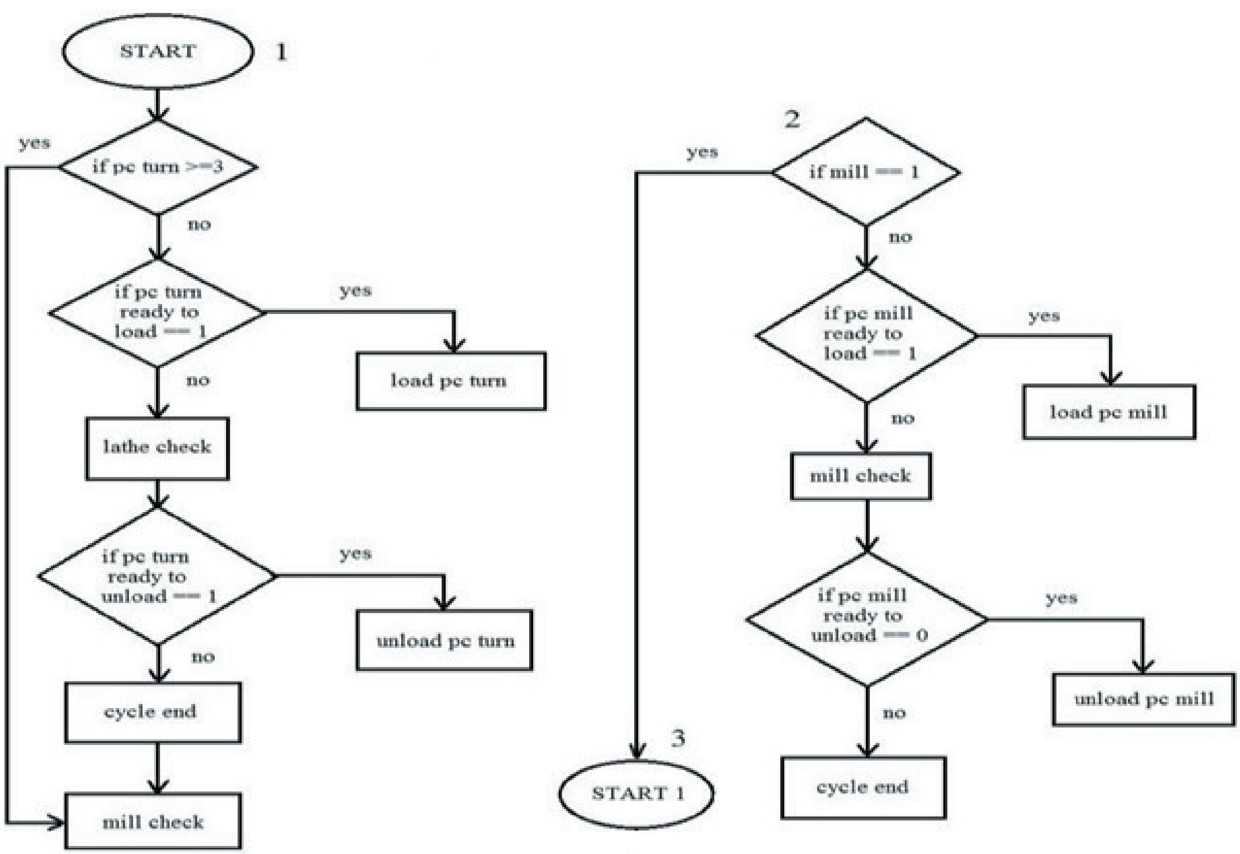

Figure 2 – Algorithm 1 and algorithm 2.

4. PART PRIORITIES

After defining the points that the gripper must reach, it is necessary to define the priorities of the parts, based on the established priorities of the parts. The best way to manage the priorities is to create an algorithm based on which the robot movement is displayed according to the number of parts being processed [9].

Algorithm 1 checks if the CNC turn has produced three or more parts (Figure 2 on the left):

- Answer YES: The algorithm checks the state of the CNC Mill (algorithm 2);
- Answer NO: The algorithm asks if the CNC Turn is ready to accept the part.

Is CNC turn ready to accept the part:

- Answer YES: The robot places the part in the CNC Turn;
- Answer NO: Checking whether the CNC Turn is ready to remove the part (in case it has been previously processed).

Is the CNC Turn ready to remove the part:

- Answer YES: The robot removes the part from the CNC Turn;
- Answer NO: the cycle is completed and the CNC Mill is checked (algorithm 2).

Algorithm 1 is implemented as a part of the SCOR-BASE code in the provided example (Listing 1):

```
If TURN >= 3 Jump to MILL_CHECK
If PCTURN55_READY_TO_LOAD == 1 Jump to LOAD PCTURN55
LATHE _CHECK:
If PCTURN55_READY_TO_UNLOAD == 0 Jump to UNLOAD PCTURN55
Jump to CYCLE_END
```

Listing 1 – Algorithm 1.

Algorithm 2 checks if the CNC Mill has manufactured one part (Figure 2 on the right).

- Answer YES: A new cycle begins (algorithm 3);
- Answer NO: The algorithm checks if the CNC Mill is ready to accept the part.

Is the CNC Mill ready to accept the part:

- Answer YES: The robot places the part in the CNC Mill;
- Answer NO: The algorithm checks the condition of the CNC Mill.

Is the CNC Mill ready to remove the part:

- Answer YES: The robot removes the part from the CNC Mill;
- Answer NO: The cycle ends.

Algorithm 2 is implemented as a part of the SCOR-BASE code in the provided example (Listing 2):

```
MILL_CHECK:
If MILL == 1 Jump to START1
If PCMILL55_READY_TO_LOAD == 1 Jump to LOAD PCMILL55
MILL_CHECK1:
If PCMILL55_READY_TO_UNLOAD == 0 Jump to UNLOAD PCMILL55
Jump to CYCLE_END
```

Listing 2 – Algorithm 2.

Algorithm 3 checks if the CNC Turn produced four or more parts (Figure 3 on the left):

- Answer YES: The production of parts on the CNC Turn is finished and the state of the CNC Mill is being checked;
- Answer NO: Checking if the CNC Turn is ready to accept the part.

Is the CNC Turn ready to accept the part:

- Answer YES: The robot places the part in the CNC Turn;
- Answer NO: The algorithm checks the state of the CNC Turn.

Is the CNC Turn ready to remove the part:

- Answer YES: The robot removes the part from the CNC Turn;
- Answer NO: The cycle ends.

Algorithm 3 is implemented as a part of the SCOR-BASE code in the provided example (Listing 3):

```
START1:
If TURN >= 4 Jump to MILL_CHECK2
If PCTURN55_READY_TO_LOAD==1 Jump to LOAD PCTURN55
LATHE _CHECK1:
If PCTURN55_READY_TO_UNLOAD==0 Jump to UNLOAD PCTURN55
Jump to CYCLE_END1
```

Listing 3 – Algorithm 3.

Algorithm 4 checks if the CNC Mill manufactured two or more parts (Figure 3 on the right):

- Answer YES: The cycle is complete;
- Answer NO: The state of the CNC Mill is being checked.

Is the CNC Mill ready to accept the part:

- Answer YES: The robot places the part in the CNC Mill;
- Answer NO: The state of the milling machine is being checked.

Is the CNC Mill ready to remove the part:

- Answer YES: The robot removes the part from the CNC Mill;
- Answer NO: The cycle ends.

Algorithm 4 is implemented as a part of the SCORBASE code in the provided example (Listing 4):

```
MILL_CHECK2:
If MILL >= 2 Jump to ORDER_COMPLETED
If PCMILL55_READY_TO_LOAD == 1 Jump to LOAD PCMILL55
MILL_CHECK3:
If PCMILL55_READY_TO_UNLOAD == 0 Jump to UNLOAD PCMILL55
Jump to CYCLE_END1
```

Listing 4 – Algorithm 4.

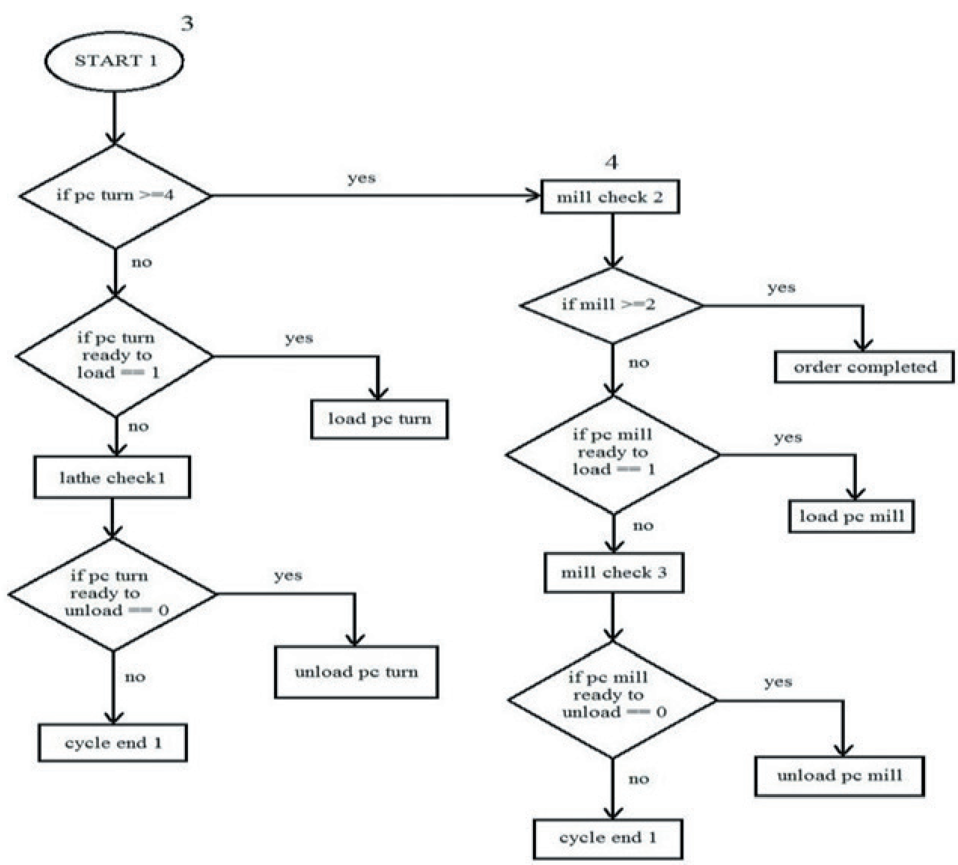

Figure 3 – Algorithm 3 and algorithm 4.

5. STATE DIAGRAMS

The state diagrams (Figures 4 and 5) represent the completed tasks of the robot in a unit of time. The symbol M represents the start of the program for tending the CNC Mill and the symbol T represents the start of the program for tending the CNC Turn.

The diagrams show that the robot spends the least amount of time removing parts from the feeders. This is because the parts to be removed are not placed in the fixture, but are already on the removal rack, so the robot can quickly remove a part from the rack without much precision or care if it collides with the fixture.

The machine tending takes most of the time as greater precision is required. Another process that can take a lot of time is waiting for the machine to finish machining. The robot's waiting time depends on the part's priority and the part's machining time. A mistake can cause the part to fly out of the clamping device and damage the machine.

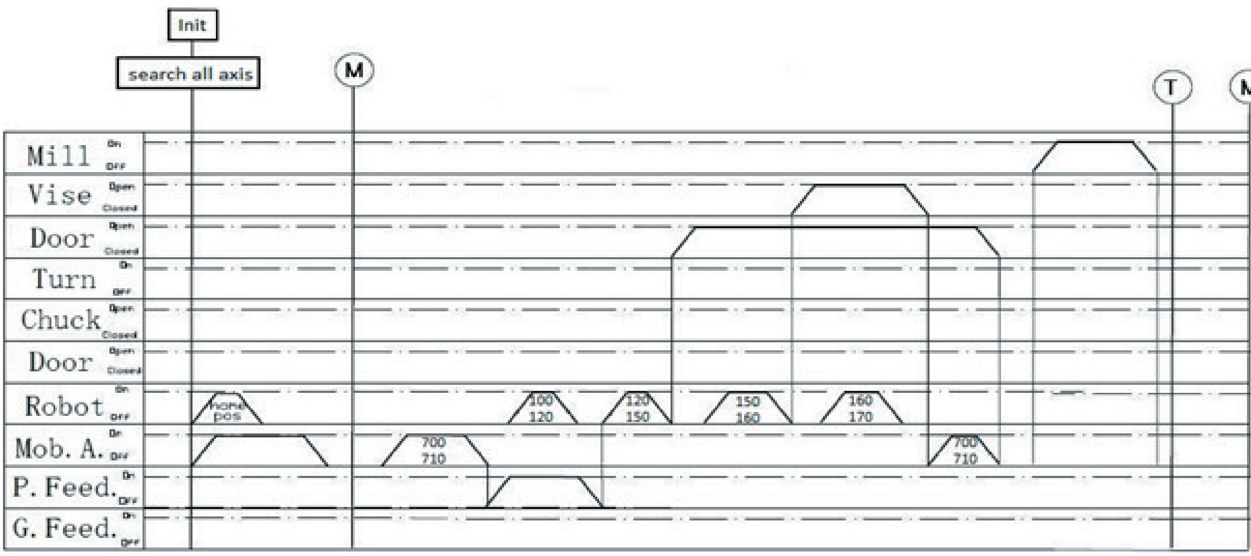

Figure 4 – State Diagram for Mill tending.

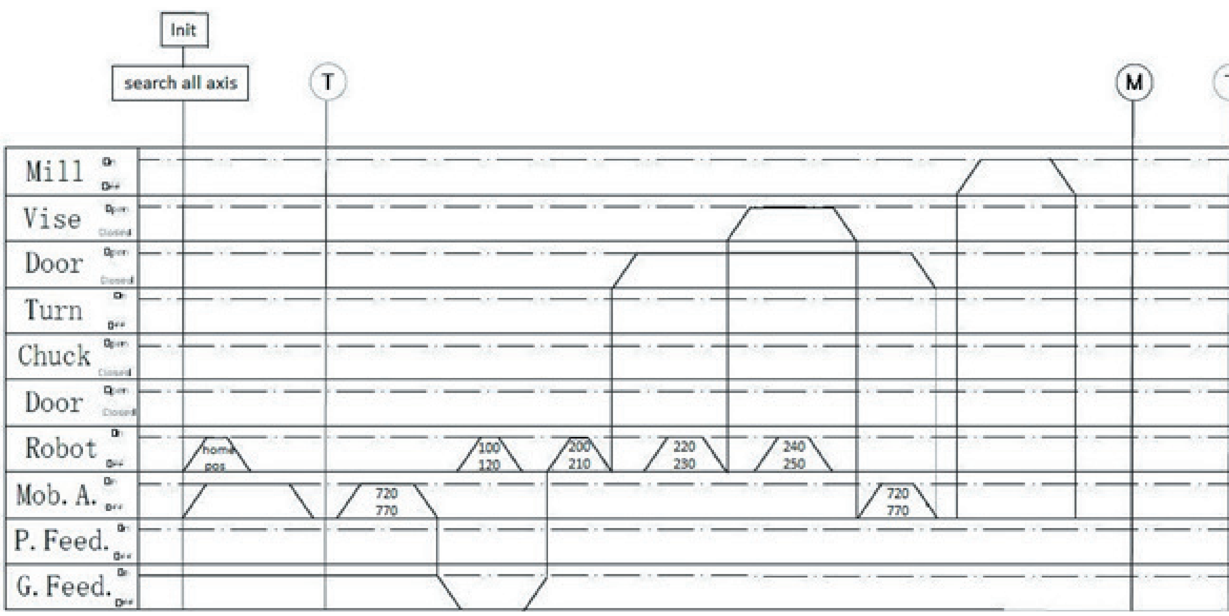

Figure 5 – State Diagram for Turn tending.

6. CONCLUSION

At the end of the flexible cell tending experiment, it can be stated that the flexible cell is working properly. There are a few problems with setting the priorities on the computer for partial priority. If one enters the value 0 or 1 in a certain line of the program, the program does not work, whereas if one enters the values 0 and 1 the program works. The advantage of the machine operation and control of a flexible cell is the control of the robot in real-time via the training device and the peripherals [10]. Controlling the robot in real-time via a training device can improve precision and make logical decisions for better control of the robot and the cell. Also, the ability to review the program, correct errors and control the robot through the program leads to the conclusion that the program can be executed online and offline. The online method is more recommended, as the robot's position can be viewed close up to avoid collisions with machines and peripherals.

7. REFERENCES

[1] P. Jakovljevic, “Role and application of industrial robots in education,” M.S. thesis, Dept. Comput. Mech. Eng., The Academy of Applied Technical Studies, Belgrade, Serbia, 2022.

[2] T. Saric, Robot Programming, Accessed: Apr. 30, 2023. [Online], Avaliable: https://dokumen.tips/documents/programiranje-robota.html

[3] D. Milutinovic, Industrial robots, Accessed: Apr. 30, 2023. [Online]. Avaliable: https://dokumen.tips/documents/profdr-dragan-milutinovic-industrijski-roboti-predavanja.html

[4] Intelitek , Derry, NH, USA. *Manual SCORBASE version 7 and higher for SCORBOT-ER 4U SCORBOT-ER 2U ER-400 AGV mobile robot*, Catalog No. 100342 (2016). Accessed: May. 1, 2023. [Online]. Avaliable: https://downloads.intelitek.com/Manuals/Robotics/ER-4u/Scorbase_USB_I.pdf

[5] Intelitek, Derry, NH, USA. *Manual on RoboCell for Motoman MHJF*, Catalog #34-8000-0012 (2018). Accessed: May. 1, 2023. [Online]. Avaliable: https://www.intelitekdownloads.com/Manuals/Robotics/MHJF/RoboCell_for_MHJF_v10.1_user_manual_revC.pdf

[6] Y. He., and L. Liang, "Application of Robotics in Higher Education in Industry 4.0 Era," *Universal Journal of Educational Research*, Vol. 7, No. 7, pp. 1612–1622, Jul. 2019, doi: 10.13189/ujer.2019.070715.

[7] A. Lobbezoo, and H.-J. Kwon, “Simulated and Real Robotic Reach, Grasp, and Pick-and-Place Using Combined Reinforcement Learning and Traditional Controls,” *Robotics*, vol. 12, no. 1, p. 12, Jan. 2023, doi: 10.3390/robotics12010012.

[8] M. Dinary, A.-H. El-Sayed, A. B. Sharkawy, and G. Abouelmagd, “A Control Scheme for Industrial Robots Using Artificial Neural Networks,” *International Journal of Materials, Mechanics and Manufacturing*, vol. 3, no. 2, pp. 80–85, May 2015, doi: 10.7763/ijmmm.2015.v3.171.

[9] O. Mubin, C.J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, “A Review of the Applicability of Robots in Education,” *Technology for Education and Learning*, vol. 1, no. 1, pp. 1–7, Jun. 2013, doi: 10.2316/journal.209.2013.1.209-0015.

[10] E. Mihret, “Robotics and Artificial Intelligence,” *International Journal of Artificial Intelligence and Machine Learning*, vol. 10, no. 2, pp. 57–78, Jul. 2020, doi: 10.4018/ijaiml.2020070104.