engineering proceedings

MDPI

Proceeding Paper

Programming Industrial Robots in the Fanuc ROBOGUIDE Environment †

Boryan Vladimirov *, Stiliyan Nikolov and Stanislav Tsolov

Faculty of Mechanical Engineering, Technical University of Sofia, 1000 Sofia, Bulgaria; st_nikolov2@tu-sofia.bg (S.N.); st.tsolov90@gmail.com (S.T.)

* Correspondence: bvladimirov@tu-sofia.bg; Tel.: +359-886-839-018

† Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES'24), Kavala, Greece, 19–21 June 2024.

Abstract: Descriptions of the main CARC environments for programming industrial robots are given, describing the main used programming environments for various robot manufacturers such as ROBOGUIDE developed by FANUC Robotics, KUKA Sim and Kuka Work Visual developed by KUKA ROBOTICS, Robot Studio developed by ABB Robotics, K-ROSET and K-ROSET LITE developed by Kawasaki Robotics, Visual Component, DELMIA ROBOTICS of Dassault Systems, Tecnomatix Robotics & Automation Simulation of SIEMENS PLM Software/Simatic Robot Integrator, Visual Components, etc. A methodology describing the main stages, when working with computer systems, of off-line programming of industrial robots is proposed. The features characterizing the implementation of the stages defined in the methodology have been specified. The created methodology has been applied when working with the Fanuc ROBOGUIDE computer system. When using the given example of the Fanuc ROBOGUIDE, the emphasis is also on expanding the working space of the robot (Robot Envelope) by adding a 7th axis. The general software options that are added when performing this task are described, and two sample programs are given for the implementation of the given example—and a 3D simulation is made for moving a part (box). A control program has been generated for an industrial robot Fanuc LR Mate 200 iD/7L that shall perform "Pick and Place" operations and shall service a conveyor for the transportation of cartons and their arrangement on pallets.

Keywords: industrial robots; off-line programming; computer programming systems; robotic production systems; Fanuc ROBOGUIDE

check for updates

Citation: Vladimirov, B.; Nikolov, S.; Tsolov, S. Programming Industrial Robots in the Fanuc ROBOGUIDE Environment. *Eng. Proc.* **2024**, *70*, 20. https://doi.org/10.3390/engproc2024070020

Academic Editor: Grigor Mihaylov

Published: 1 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Industrial robots are increasingly used in modern industry. They are the basis for bringing the manufacturing equipment in compliance with the requirements of Industry 4.0 [1]. As a result of that, the range of tasks performed by industrial robots has surged and is thus reflected on the complexity of the programs executed by them.

The increased complexity of programs executed by industrial robots and their inclusion in various robotic manufacturing systems creates prerequisites for the widespread introduction in the engineering practice of specialized computer systems for the programming of industrial robots. These include CARC (Computer-Aided Robot Control) [2–7] systems, also known as OLPE (Off-line Programming Environments), which incorporate the latest achievements in the field of computing technology and engineering computer graphics. CARC systems employ industrial robots' 3D models and other manufacturing equipment that enable the construction of robotic manufacturing systems, simulate and analyze their operations, and generate control programs for the industrial robots included in them.

Currently, all industry leaders offer users CARC systems for the programming of their industrial robots: ROBOGUIDE developed by FANUC Robotics, KUKA Sim and Kuka

Work Visual developed by KUKA ROBOTICS, Robot Studio developed by ABB Robotics, K-ROSET and K-ROSET LITE developed by Kawasaki Robotics, etc. [7].

The main advantage of these systems is that they include the algorithms used by the real controllers of the industrial robots offered by the specific company, which minimizes the possibility of errors when the industrial robot executes the programs generated by the system.

There are also CARC systems created by companies that specialize in the development of software products, but these companies, such as Robot master, Robot Interface Robo DK, etc. [8], are not directly involved in the manufacturing of industrial robots. The said systems use a post-processor to generate a program for a specific industrial robot. The post-processor produces a program (code), considering the specifics of the robot controller. In this case, the chance of any errors emerging while the generated program is being executed by the industrial robot depends on the post-processor's settings and the extent to which the algorithms embedded in the post-processor correspond to the ones used by the robot's controller, which are a company secret.

With the introduction of the engineering practice of PLM (Product Lifecycle Management) systems, CARC systems were also integrated into their structure, which, in addition to programming, can also be used in the design of robotic cells and robotic manufacturing systems. Such systems include DELMIA ROBOTICS of Dassault Systems, Tecnomatix Robotics & Automation Simulation of SIEMENS PLM Software/Simatic Robot Integrator, SIMIT, Visual Components, etc. [9,10].

Regardless of the variety of CARC systems currently offered, the main steps in the operation of these systems largely overlap. The present document proposes a methodology that describes the CARC systems' basic work stages necessary for the creation of a control program for an industrial robot.

2. Materials and Methods

The core stages in the creation of a control program for an industrial robot, no matter what CARC system is being used, can be defined in the following manner.

2.1. Selection of an Industrial Robot

At this stage, the user must select the industrial robot and the controller for which the control program will be generated. An industrial robot is selected from the system's library containing 3D models of industrial robots. If necessary, a 3D model for an industrial robot with no existing data in the system could be developed. This is most often achieved by using a CARC system developed by an independent software company, as the company's CARC system libraries support 3D models of the industrial robots being produced.

When creating new 3D models, one should meet the requirements of the CARC system. Three-dimensional models may be produced by employing a CAD system, following which they shall be imported into the CARC system in consideration of the file formats supported by it.

The controller is selected from the system's library, considering the characteristics of the real robot controller. In the case of CARC systems developed by independent software companies, a post-processor matching the respective controller shall be picked up. Post-processors may be developed or modified to maximize the generated code's compliance with the requirements of the industrial robot's controller. When designing or editing the post-processors, the applicable rules must be followed.

The selected industrial robot shall be positioned in the graphic area of the system, but its position can be changed when adding extra equipment during the design of a robotic system.

2.2. Selection of End-Effector

Depending on the operation performed by the industrial robot, at this stage there shall be an end-effector selected, with which the robot will be working. When choosing an

end-effector, the considerations made in the first stage of selecting 3D models for industrial robots shall be valid. Some CARC systems support parameterized 3D models of the end-effector, which allow the user to enter, in dialog mode, the real end-effector's exact characteristics the robot will be working with.

The proper selection of an end-effector and its connection to the industrial robot's 3D model are extremely important because they determine the position of the gripper's endpoint or the central point of the tool TCP (Tool Center Point), for which the generated program will set positions in the work area and will define trajectories with a certain speed. At this stage, one should consider the end-effector's compatibility with the industrial robot's controller to secure the program control of its functions.

2.3. Choosing Additional Equipment

The system's graphic area shall now be supplemented with 3D models of additional equipment to be used by the industrial robot or equipment that falls within its operational zone and needs to be considered during the development of the control program to avoid any collisions in the robot's work. At the same time, the considerations made in the first stage of selecting 3D models for industrial robots shall be valid.

The additional equipment's 3D models shall be positioned towards the industrial robot already selected. The accuracy of this positioning must correspond to the actual location to ensure trouble-free operation of the system. It is possible to add equipment to add additional mobility to the selected industrial robot to expand its working area. At this stage, one must also check if the additional equipment's control is compatible with the industrial robot's controller to ensure the program control of its functions.

2.4. Development of Control Program

The robotic system's 3D model developed in the previous stages will now be used for the off-line programming of the industrial robot. During this programming, the gripper's endpoint or the central point of the tool provided by the CARC systems shall be positioned in the desired locations, and the end-effector shall be oriented towards the preferred poses. The information about the industrial robot's configurations in desired locations is saved in the system. The set of tools provided by the CARC systems shall set the gripper's endpoint or the instrument's central point movement between memorized poses. The robot's preferred poses may be reached under each of the following methods:

- Direct guiding of the robot's 3D model in the system's graphic area;
- Direct setting of robot's joint coordinates;
- Direct setting of coordinates in any of the robot's coordinate systems;
- Controlling the robot from a virtual console (Teach pendant), a copy of the real one.

When using CARC systems developed by manufacturers of industrial robots, a robot can be controlled through a code entered by the user directly into the system.

2.5. Simulating the Developed Program

At this stage, the robot's work and the entire robotic system shall be simulated with the purpose of detecting collisions and singularities during the implementation of the developed program. Any issues that have been established shall be fixed by returning to the previous level to change the end-effector's poses or movement trajectories.

After eliminating all issues, one should use the set of tools provided by the CARC system to analyze the created program and eventually optimize it in a way that shall reduce the time necessary for the performance of required operations.

2.6. Implementation of the Developed Program

Once the detected problems have been dealt with and the program has been optimized at the previous stage, the CARC systems' program, developed by independent software companies, will have to be post-processed to produce a code executable by the robot's

controller. Company CARC systems will automatically generate the code, taking into consideration the industrial robot controller selected at the first stage.

The executable code shall be uploaded in the controller of the actual robot through some of the ports supported by it. The industrial robot can then start executing the program. The methodology developed will be used for the programming of an industrial robot in the ROBOGUIDE environment.

ROBOGUIDE is an in-house CARC system developed by FANUC for the programming of FANUC industrial robots. The system provides the ability to assist in the design of robotic systems and is also capable of creating multiple 3D simulations. Fanuc ROBOGUIDE can exchange IGES, STL, OBJ, and other files. Data movement thought the different stages of the system's operation is shown in Figure 1 (the gray arrows). The main system modules needed to create a virtual robot are marked with * in the Figure 1.

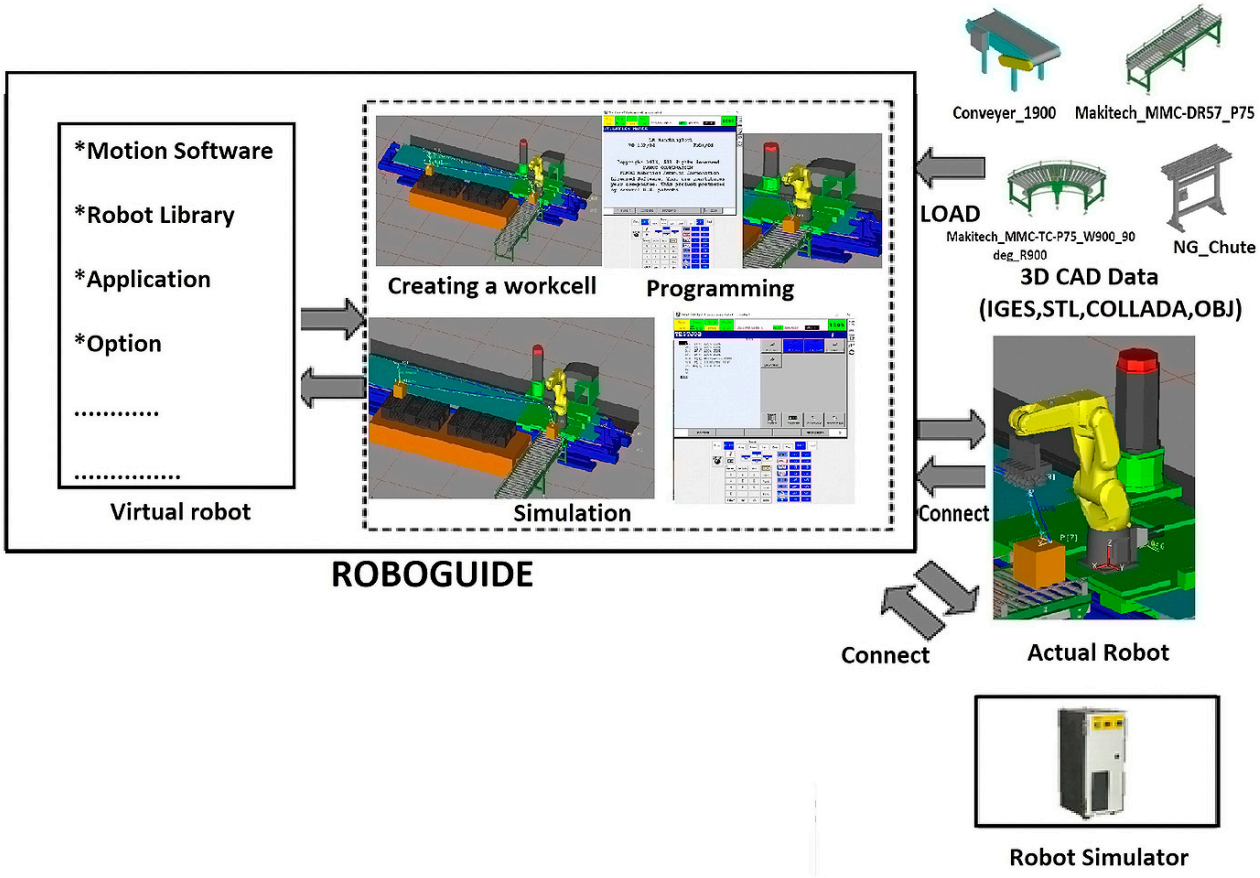


Figure 1. Data processing in Fanuc ROBOGUIDE.

3. Results

Fanuc ROBOGUIDE shall be used to program an industrial robot performing "Pick and Place" operations in the servicing of a conveyor for the transportation of boxes, sizing 150 × 150 × 150 mm and weighing 5 kg, and their arrangement on pallets.

3.1. Stage 1—Robot Selection

A Fanuc LR Mate 200 iD/7L [11] robot was selected to service the conveyor. The robot was selected through the menu in Figure 2.

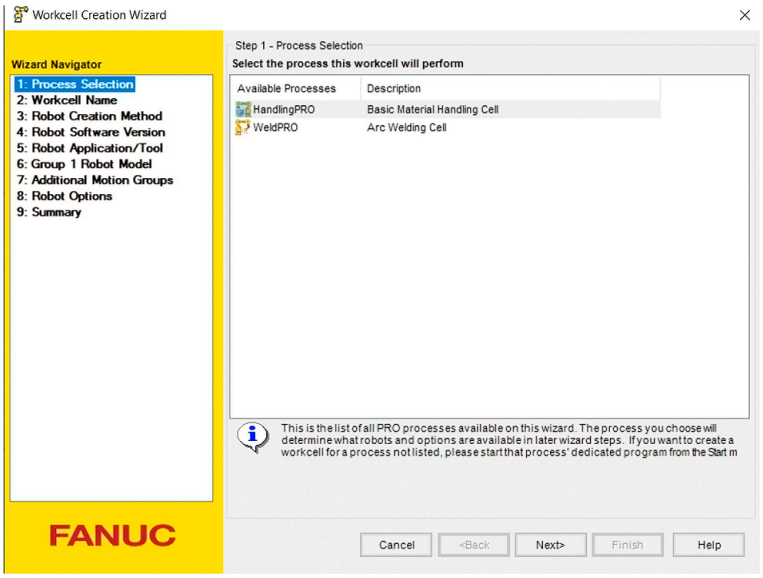
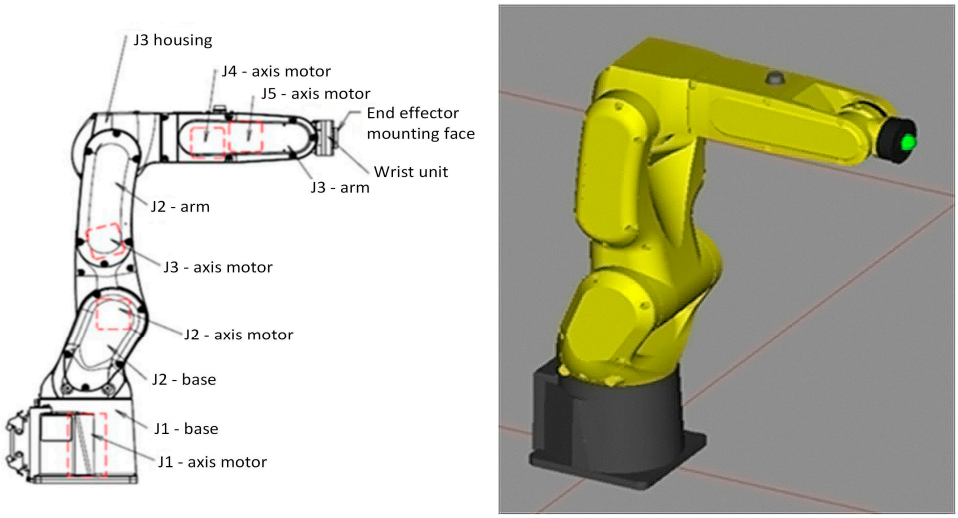


Figure 2. Robo guide—robot selection menu.

File → New.
Next → Step 1 Create a new robot with the default HandlingPro config
Next → Step 2 Robot Software Version → select a method.
Next → Step 3 Robot Application/Tool → Handling Tool(H552)
Next → Step 4 Group 1 Robot Model → LR Mate 200iD/7L
Next → Step 5 Additional Motion Group → Empty
Next → Step 6 Robot Option -

- Extended Axis Control (J518)
- Multi-Group Motion (J601)

The general appearance of the actual robot and its 3D model provided by ROBOGUIDE are shown on Figure 3.



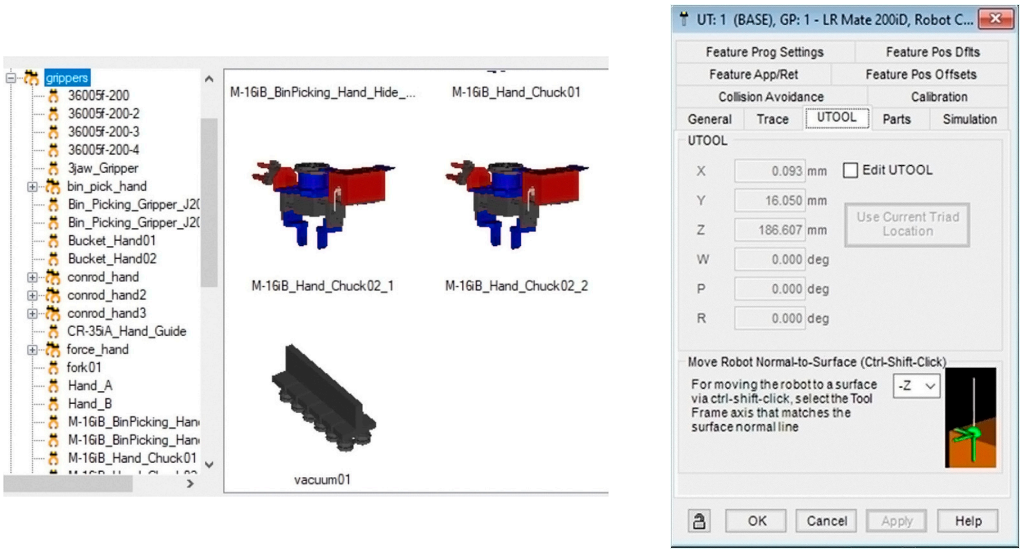
(**a**) (**b**)

Figure 3. Fanuc LR Mate 200 iD/7L [11]: (**a**) actual robot; (**b**) 3D model provided by ROBOGUIDE.

3.2. Stage 2—Selection of End-Effector

In ROBOGUIDE [12], the vacuum gripper shall be selected from the following menu: Library → gripper → vacuum01.

Setting the vacuum gripper's size based on the dimensions of the processed workpiece and its generated 3D model are shown in Figure 4.



(**a**) (**b**)

Figure 4. Vacuum gripper: (**a**) 3D model provided by ROBOGUIDE; (**b**) dialog for the selection of gripper's parameters.

Figure 5 shows the dialog used in the setting of an end point for the selected vacuum gripper, for which point there shall be assigned positions in the workspace and movement trajectories.

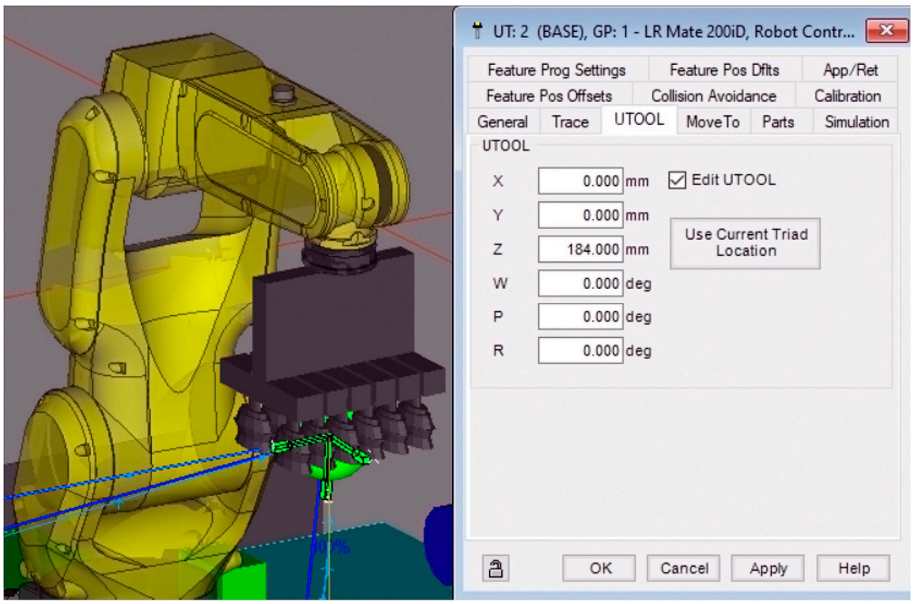


Figure 5. Setting the gripper's end point.

3.3. Stage 3—Selection of Additional Equipment

Using the ROBOGUIDE libraries, the following shall be added for the creation of a robotic cell's 3D model to the 3D model picked up for the industrial robot.

3.3.1. Manipulated Workpiece

Bearing in mind the shape, dimensions, and weight of boxes the robot will be working with by employing the ROBOGUIDE toolkit, a new 3D model shall be generated, as shown in Figure 6.

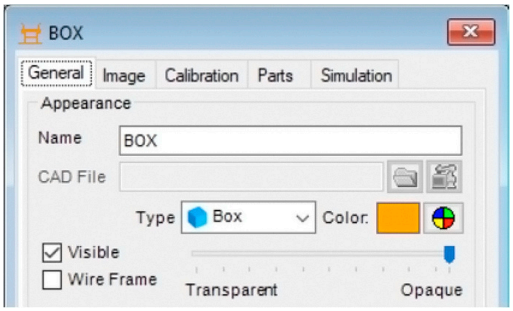
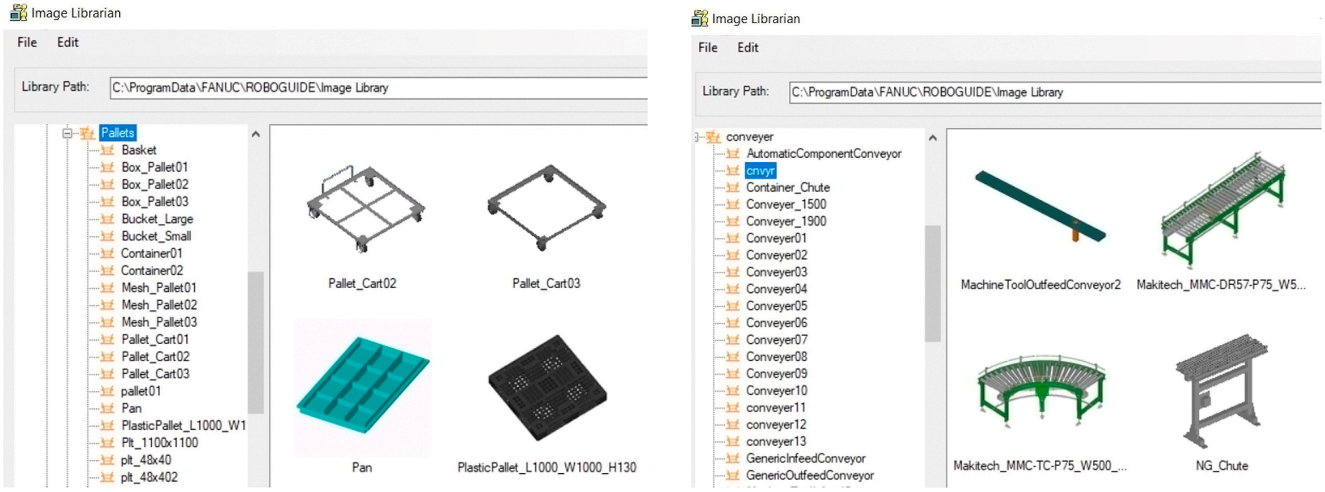


Figure 6. Dialog for the generation of a 3D model of the manipulated workpiece.

3.3.2. Pallet

A 3D model of a Plastic Pallet L1000_W1000 H130 in Figure 7a has been added to arrange the cartons in the 3D model of the robotic cell.



(**a**) (**b**)

Figure 7. Three-dimensional models embedded in the robotic cell: (**a**) pallet; (**b**) conveyor.

3.3.3. Conveyor

The robot will service conveyor Makitech MMC-DR57-P75_W500_2P4 [13]. Its 3D model embedded in the 3D model of the robotic cell is shown on Figure 7b.

3.3.4. Mobile Platform

The robot's working area will have to be expanded given the real distance between the conveyor and the pallet station. Therefore, the selected robot will be placed on a mobile platform [14–16].

The following function will be used for this purpose:

Tools → Rail Unit Creator → Rail Unit Creator Menu

The dialogue for the creation of the mobile platform's 3D model is shown in Figure 8. The result from this stage is a 3D model of the robotic cell used for the cartons' unloading from the conveyor and their arrangement on pallets is shown in Figure 9.

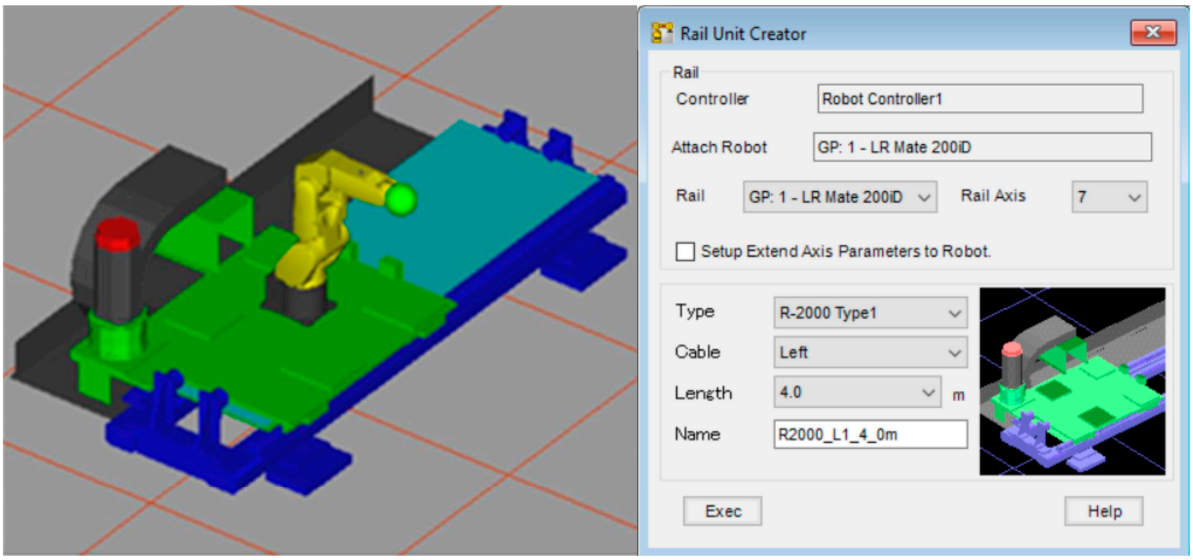


Figure 8. Dialog generating the mobile platform's 3D model with a length of 4 m.

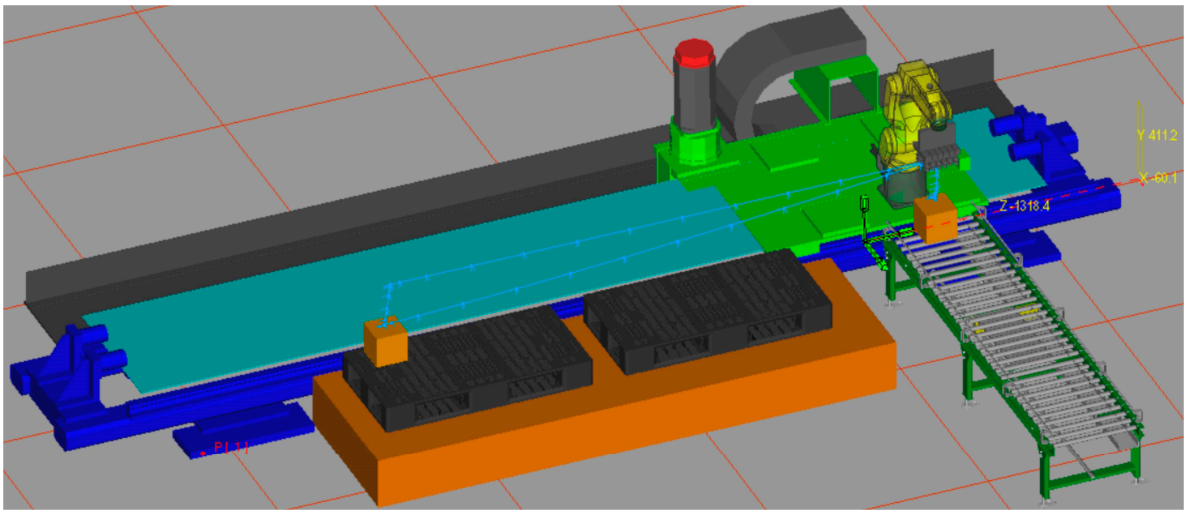


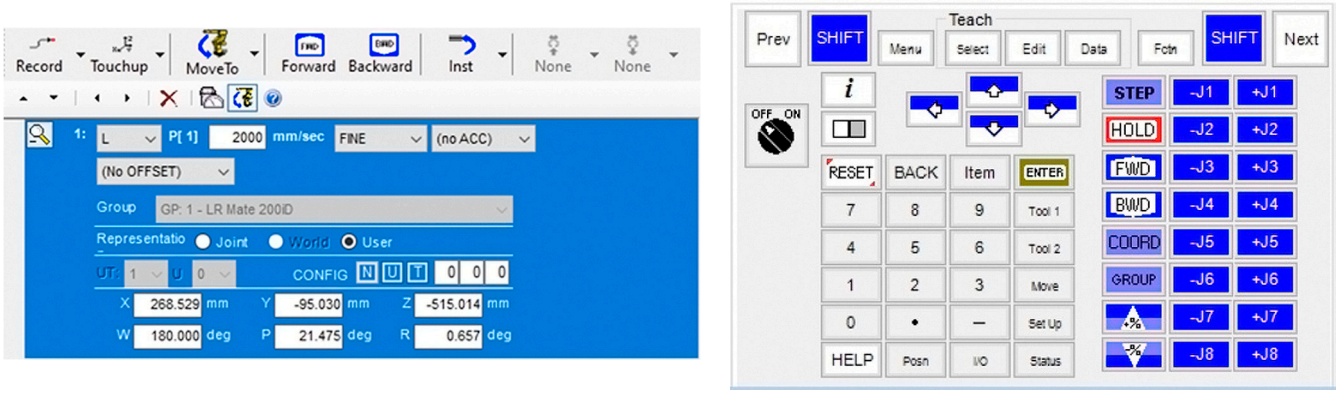
Figure 9. Three-dimensional model of the robotic cell.

The cell's working cycle is as follows:

1. The industrial robot takes a box or a package from the conveyor;
2. The industrial robot moves to the pallet station via the mobile platform on which it is mounted;
3. The industrial robot places the box on the pallet, starting from the outermost pallet (position on Figure 9);
4. The industrial robot moves back to the conveyor.

3.4. Stage 4—Development of the Control Program

When developing the industrial robot's necessary control program, the coordinates shall be directly set in one of the robot's coordinate systems and in the Teach pendant virtual console in Figure 10.



(**a**) (**b**)

Figure 10. Tools for the development of a control program: (**a**) setting coordinates in program simulation editor; (**b**) Teach pendant virtual console—including 7th (−J7/+j7) and 8th (J8/+J8) axes in the controller.

3.4.1. Control of the Mobile Platform—7th Axes on the Robot Programming/Creating

Controlling the additional seventh axis [14–16] (the movement of the mobile platform on which the robot is mounted) requires the following software options:

1A05B-2600-J518—Extended Axis Control (J518)

1A05B-2600-J601—Multi-Group Motion (J601)

Provided these options are available, the MASTER functions shall need the following variables—SYSTEM Variable.

$DMR_GRP

1 [1] DMR_GRP_T

2 [2] DMR_GRP_T

By using a sub-variable:

$MASTER_COUN → $DMR_GRP [1].$MASTER_COUN

It is possible to increase the robot's controllable axes from 6 up to 9 (six robot axes and up to three additional controllable axes).

3.4.2. Example Program—Two Variants

First version:

The program generated is as follows:

UFRAME_NUM [1]—frame of the conveyor

UFRAME_NUM [2]—frame of pallet 1

UFRAME_NUM [3]—frame of pallet 2

UTOOL [1]—TCP of the instrument (the vacuum gripper)

/UFRAME—USER FRAME; UTOOL—USER TOOL/

Program for the movement of a single workpiece:

1. UTOOL [1]
2. UFRAME_NUM = 1
3. PAYLOAD [1]
4. J P [1] 40% FINE
5. J P [2] 40% FINE
6. L P [3] 2400 mm/s FINE
7. Pickup ('BOX') From ('Makitech_MMC-DR57_W500_2P4')

   With ('GP:1-UT:1(Eoat1)')// RO [1] = ON
8. L P [4] 2400 mm/s FINE
9. J P [5] 40% FINE
10. UTOOL [1]
11. UFRAME_NUM = 2

12. L P [6] 2400 mm/s FINE
13. Drop(‘BOX’) From(‘GP:2-UT:1(Eoat1)’)

    On (‘PlasticPallet_L1000_W1000_H130_1’) // RO [1] = OFF

14. L P [7] 2400 mm/s FINE
15. J P [8] 40% FINE

Second version

The controller’s program—the program recorded on the controller consists of the following sub-programs:

1. Sending the robot to the starting position.

----------Program HOME----------
1: L P [1] 4500 mm/s FINE
2: L P [2] 4500 mm/s FINE
3: CALL CONVEYOR
4: L P [2] 4500 mm/s FINE
5: L P [3] 1800 mm/s FINE
6: L P [4] 1800 mm/s FINE
7. CALL PALLET1 or PALLET2
8. L P [4] 1800 mm/s FINE
9. L P [1] 4500 mm/s FINE

2. Control of the conveyor.

----------Program CONVEYOR----------
1: L P [1] 1500 mm/s FINE
2: L P [2] 1500 mm/s FINE
3: Pickup Pickup(‘BOX’) From (‘Makitech_MMC-DR57_W500_2P4’)
With (‘GP:1-UT:1(Eoat1)’) / RO [1] = ON
4: L P [3] 1500 mm/s FINE

3. Placing the box on pallets.

----------Program PALLET1/UF2/ and PALLET2/UF3/----------
P [1] GP2 (UF2 and UF3) UT1
J1 0.000 mm
Position Detail
1: L P [1] 1500 mm/s FINE
2: Drop(‘BOX’) From(‘GP:2-UT:1(Eoat1)’)On (‘PlasticPallet_L1000_W1000_H130_1’)/
RO [1] = OFF
3: L P [1] 1500 mm/s FINE

3.5. Types of Box Configurations on the Pallet with Some Coordinates

In total, 18 boxes measuring 150 × 150 × 150 mm can be placed on one pallet. An example layout is presented (shown) in Figures 11–14 (below), with coordinates of the location of the end boxes.

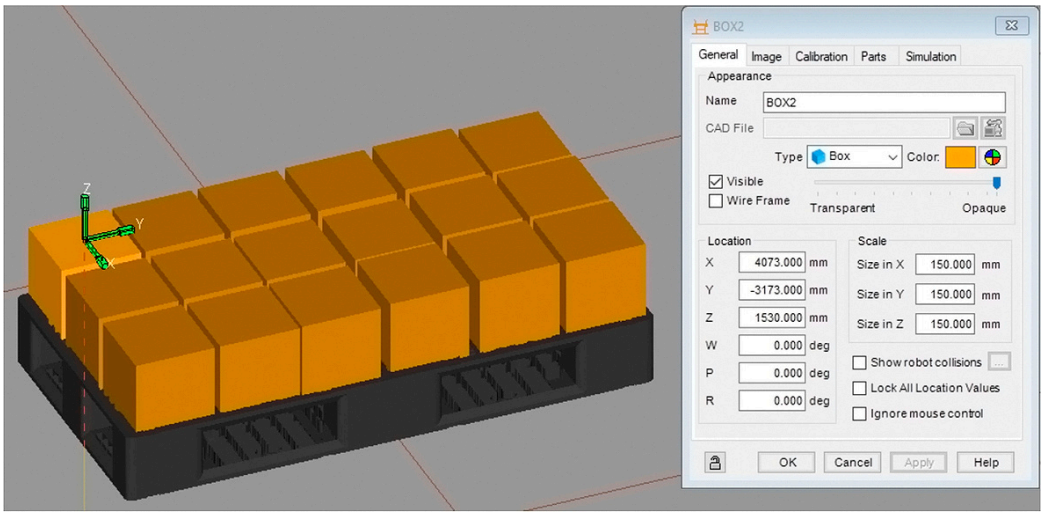


Figure 11. First left box on pallet with example coordinates.

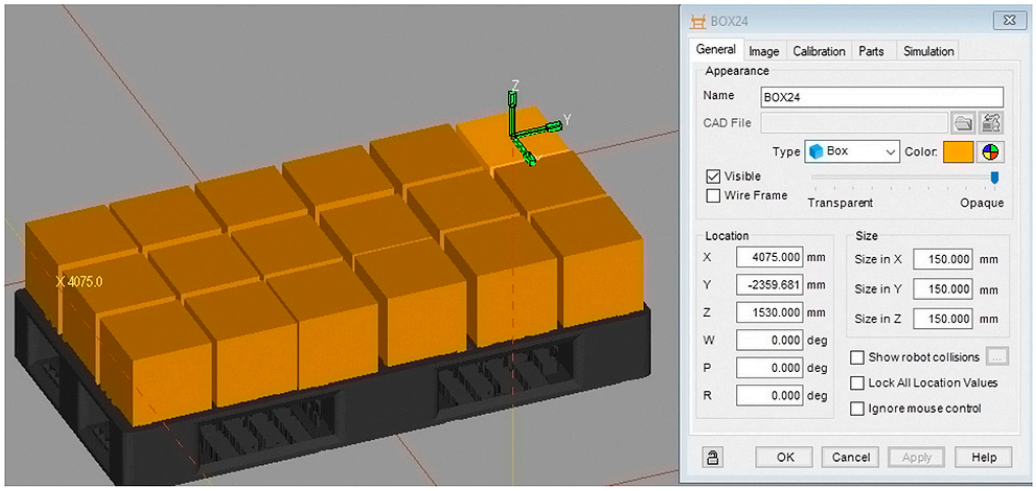


Figure 12. First right box on pallet with example coordinates.

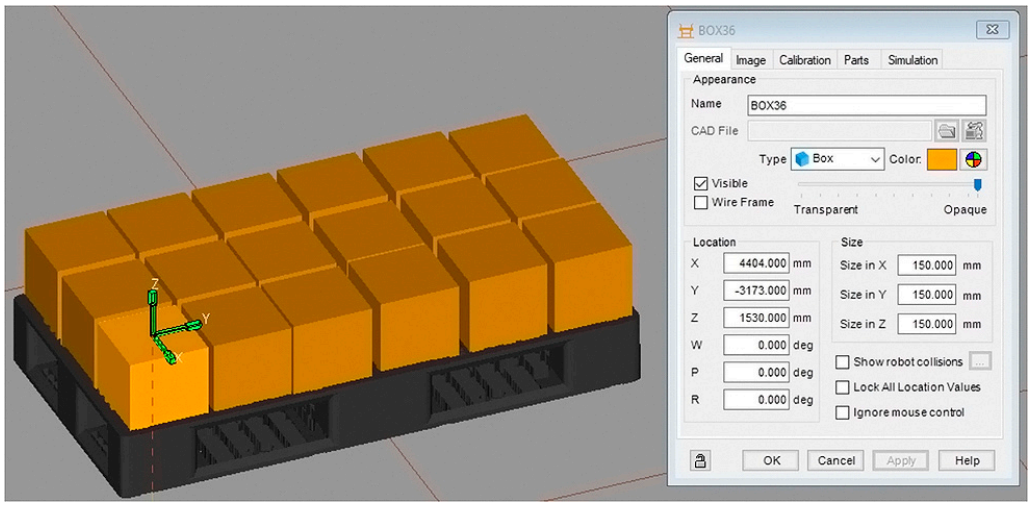


Figure 13. Third left box on pallet with example coordinates.

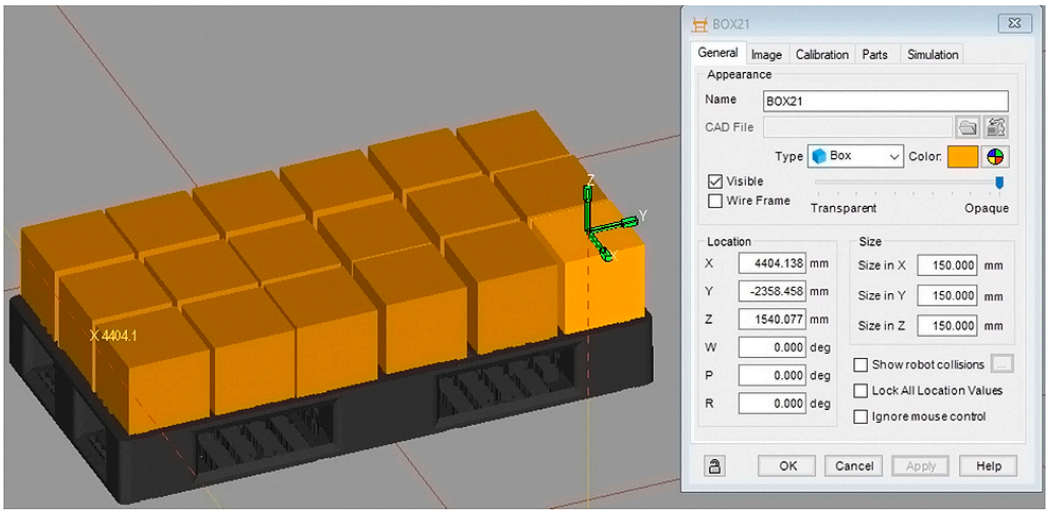


Figure 14. Third right box on pallet with example coordinates.

3.6. Stage 6—Simulation of the Developed Program

The ROBOGUIDE toolkit has been used to simulate how the robotic cell operates in accordance with the generated program. No collisions and singularities have been found while simulating operation under the generated program.

4. Conclusions

The proposed methodology, describing the main steps in the operation of CARC systems, which are needed to generate a control program for an industrial robot, is universal and can be used when working with various CARC systems.

The defined individual stages are interconnected, and the results obtained from the simulation of the developed program may require changing the decisions taken at previous stages.

The functionality of the proposed methodology is proven by the control program generated for a Fanuc LR Mate 200 iD/7L industrial robot performing "Pick and Place" operations in the servicing of a conveyor for the transportation of boxes or packages and their arrangement on pallets.

Simulating the work of robotic systems by using their 3D models created in a CARC system environment allows us not only to detect possible collisions and singularities, but also to obtain data on the systems' technical and functional characteristics.

Author Contributions: B.V., S.N. and S.T. were involved in the full process of producing this paper, including conceptualization, methodology, modeling, validation, visualization, and preparing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Scientific and research sector of Technical University of Sofia Contract No. 241ПД0024-06 Examination the possibilities of programming industrial robots using API.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are displayed in the article.

Acknowledgments: The authors wish to thank Scientific and research sector of Technical University of Sofia.

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

References

1. Totev, D.; Dimitrova, R.; Dimitrov, S. Main Steps in the Process of Upgrade of Existing Systems for Automation and Control of Industrial and Manufacturing Processes in Order to Fulfill the Requirements of the Concept "Industry 4.0", EEPES 2023. *AIP Conf. Proc.* **2024**, *63*, 060007. [CrossRef]
2. Novak, J.; Kuzmiakova, M. Computer Aided Production Engineering. In Proceedings of the International Conference on Manufacturing Systems—ICMaS, Bucharest, Romania, 5–6 November 2009; University Politehnica of Bucharest, Machine and Manufacturing Systems Department: Bucharest, Romania, 2009; Volume 4, pp. 311–314.
3. Computer-Aided Design of Robot Controller Handling and Real-Time Control. Available online: https://www.semanticscholar.org/paper/Computer-Aided-Design-of-Robot-Controller-Handling-Simon-Espiau/da1567ba85474382766d01eaf4fa42857755f78f (accessed on 12 March 2024).
4. Computer-Aided Design of a Generic Robot Controller Handling Reactivity and Real-Time Control Issues. Available online: https://ipn.elsevierpure.com/en/publications/computer-aided-design-of-a-generic-robot-controller-handling-reac (accessed on 12 March 2024).
5. Design of Intelligent Robot Control System Based on Human–Computer Interaction. Available online: https://link.springer.com/article/10.1007/s13198-021-01267-9 (accessed on 22 March 2024).
6. Application of the Virtual Reality Modelling Language to Computer Aided Robot Control System ROANS. Available online: https://www.semanticscholar.org/paper/Application-of-the-Virtual-Reality-Modelling-to-Novak-Marcincin-Doli%C3%A1k/0389d7a146df141ab0e9a60f5d442ff0f48db79b (accessed on 12 April 2024).
7. Wu, H.; Deng, H.; Yang, C.; Guan, Y.; Zhang, H.; Li, H. A Robotic Off-line Programming System Based on SolidWorks. In Proceedings of the 2015 IEEE Conference on Robotics and Biomimetics, Zhuhai, China, 6–9 December 2015; pp. 1711–1716.
8. Novak-Marcincin, J.; Brazda, P.; Janak, M.; Kocisko, M. Application of Virtual Reality Technology in Simulation of Automated Workplaces. *Tech. Gaz.* **2011**, *18*, 577–580.
9. Dimitrova, R.; Dimitrov, S.; Totev, D.; Bahchevanov, B. Electrical Control, Program Code and Experimental Studies of the Gauss Curve in a Specialized Automated Mechatronic System. In Proceedings of the 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Plovdiv, Bulgaria, 23–25 November 2023; pp. 1–7. [CrossRef]
10. Saaksvuori, A.; Immonen, A. *Product Lifecycle Management*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2005.
11. Industrial Robot Fanuc LR Mate 200iD/7L. Available online: https://www.fanuc.eu/se/en/robots/robot-filter-page/lrmate-series/lrmate-200id-7l (accessed on 9 April 2024).
12. Fanuc Roboguide, Operator Manual, B-83234EN/03. Available online: https://kupdf.net/download/roboguide-operator-manual-b-83234en-02_5d32fb49e2b6f5b75a4c0937_pdf (accessed on 23 January 2024).
13. Conveyor Makitech MMC-DR57-P75. Available online: https://www.makitech.co.jp/conveyor/4/mmc-dr.html (accessed on 9 April 2024).
14. Global 7-Axis Robot Market Insights, Forecast to 2029. Available online: https://www.industryresearch.co/global-7-axis-robot-market-25770153 (accessed on 12 April 2024).
15. World's First Composite Concrete 7th Axis Used for the First Time in Series Production at Car Manufacturer. Available online: https://www.roboticstomorrow.com/article/2019/07/worlds-first-composite-concrete-7th-axis-used-for-the-first-time-in-series-production-at-car-manufacturer/13885 (accessed on 9 April 2024).
16. Collaborative Synchronization of a 7-Axis Robot. Available online: https://www.researchgate.net/publication/338091284_Collaborative_Synchronization_of_a_7-Axis_Robot (accessed on 9 April 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.