Romanian Journal of Petroleum & Gas Technology
VOL. VI (LXXVII) • No. 1/2025

PICK AND PLACE ROBOTIC CELL SIMULATION USING RoboDK SOFTWARE

Traian-Andrei Pană [1]

Gabriela Bucur [1]

[1] Petroleum-Gas University of Ploiesti, Romania
e-mail: gbucur@upg-ploiesti.ro

DOI: 10.51865/JPGT.2025.01.14

ABSTRACT

In the era of advanced technology, the automation of industrial processes has become an essential element for improving efficiency and productivity. One of the most relevant aspects of this phenomenon is the development of robotic systems capable of performing repetitive tasks with a precision and speed superior to human ones. The objective of the work is the design and implementation of a robotic arm intended to perform the "pick and place" function, an essential process in numerous industrial fields, from product manufacturing to logistics and assembly. Pick and place involves picking an object from a specific location and placing it in another position. This task, while seemingly simple, is fundamental to the production flow of many industries. For example, in automobile manufacturing, components must be handled with care to ensure the quality and integrity of the final product. In logistics, robotic arms are used to organize goods in warehouses, reducing handling time and human error.

The article presents the structure and operating specifications for the Doosan Robotics M1013 robot, as well as the RoboDK software, with its main features and application areas. After defining the specifications and selecting the hardware components, such as servo motors, sensors and control modules of the manufacturing cell, we move on to the design stage, where a 3D model of the robotic arm will be created. This model will allow the simulation of the robot's operation and the anticipation of possible problems, before the physical construction.

Keywords: robotic flexible cell, pick and place, offline programming, RoboDK software, Doosan Robotics M1013 robot

INTRODUCTION

Robotics is an interdisciplinary field that combines engineering, computer science, artificial intelligence, and many other disciplines to develop machines capable of performing automated tasks. A key component of robotics is robotic arms, which are used in a variety of applications, from industrial assembly to material handling [13, 27, 30, 33]. Pick and place projects are among the most common uses of these devices, with the goal of improving operational efficiency and reducing costs. The literature highlights the vast potential of robotic arms in various industrial fields, as well as the challenges they face [2, 12, 14, 15, 24, 25].

Despite the growing demand in industry for flexible part handling solutions, the design of flexible robotic pick and place operations – where multiple or frequently changing parts that arrive in bulk in the work cell need to be handled – remains a challenge [4, 17, 18, 20]. Automating processes with parts in undetermined positions requires advanced robotic pick and place solutions based on sensory information. At the same time, flexibility must be maintained, even when economic considerations allow only minimal modification of the initial work cell [19, 26, 36, 37, 38].

In traditional robot cells, operations assume that parts are placed in a known location in a structured manner (e.g., on a pallet), from where they can be gripped and moved to a predefined target position. [39, 40]. To ensure reliable handling of industrial parts, robust operation is important, along with minimal computation and execution times.

Robot trajectories must be collision-free with respect to both the robot-environment relationship and the robot itself. In addition, parts must be accurately grasped and placed using a general-purpose tool, while allowing for complex geometries and multiple different part types. The selection and placement of positions, as well as the approach directions, can be complicated. Furthermore, work cell designs - often initially optimized for human operations - introduce strict constraints on the robot trajectory [1, 3, 5, 6, 8, 9]. Finally, the collision-free robot path must be planned in the common robot configuration space.

Process planning for robotic pick-and-place operations involves sequencing parts, selecting the appropriate grip for each part, planning robot trajectories, etc. These decisions are closely related to each other and together determine the performance of the robotic pick-and-place operation. Moreover, they must be performed online, with as little computational time as possible [7, 16, 21, 22]. A new concept is called bin-picking. Bin picking (also referred to as random bin picking) is a core problem in computer vision and robotics. The goal is to have a robot with sensors and cameras attached to it pick-up known objects with random poses out of a bin using a suction gripper, parallel gripper, or other kind of robot end effector [36].

Offline Programming means programming robots outside the production environment. Offline Programming eliminates production downtime caused by shop floor programming. Simulation and Offline Programming allows studying multiple scenarios of a robot work cell before setting up the production cell [23, 28, 29, 32, 34, 35]. Mistakes commonly made in designing a work cell can be predicted in time. Offline Programming is the best way to maximize return on investment for robot systems and it requires appropriate simulation tools. The time for the adoption of new programs can be cut from weeks to a single day, enabling the robotization of short-run production.

PICK AND PLACE ROBOTIC CELL DESIGN

The robotic arm studied and presented in this article is a Doosan Robotics M1013. The designed structure also includes a linear translation rail as well as a conveyor driven by an electric motor.

Doosan Robotics M1013 robotic arm [41] is a 6-axis articulated robot with top-notch force sensitivity and collision detection. This robot has a payload capacity of 10 kg and a reach of 1.30 m. A versatile model, suitable for all work processes. The 6 torsion sensors ensure a safe working environment for any type of application. The robot has a cockpit function, which allows the robot to move, learn waypoints and save coordinates via buttons located near the tool flange.

Doosan robots support the latest communication technologies (RS232/485, TCP/IP, Modbus TCP (Parent/Child), Modbus RTU (Parent), PROFINET IO Device, Ethernet/IP Adapter), even without a gateway, allowing seamless connections with various types of industrial equipment and systems [41]. The main applications of the Doosan Robotics M1013 robotic arm are: assembly, CNC feeding for loading/unloading objects on pallets, welding, grinding and deburring, palletizing. The main characteristics of the robotic arm are those presented in Table 1.

***Table 1**. Main features of the Doosan Robotics M1013 robotic arm*

Weight	33 kg
Payload in working range	10 kg
Maximum working radius	1300 mm
Number of axles	6
Maximum TCP speed	Over 1 m/s
Positional repeatability (ISO 9283)	±0.1mm (±0.05mm)
Degree of protection	IP54
Noise	< 65 dB
Installation direction	Floor / Wall / Ceiling
Controllers	Doosan & TP Controller
Vibration and acceleration	$10 \leq f < 57$Hz - amplitude 0.075mm $57 \leq f \leq 150$Hz - 1G
Impact	Maximum amplitude: 50m/s² (5G) * Time: 30ms, Pulse: 3 of 3 (X,Y,Z)
Operating temperature	0 °C ~45 °C (273K to 318K)
Storage temperature	-5 °C ~50 °C (268K to 323K)
Moisture	20%~80%

The kinematic scheme of the robot is presented in Figure 1 and the variation ranges of the generalized coordinates and joint velocities are presented in Table 2. The layout plan of the designed structure is presented in Figure 2.

The pallet with boxes was placed at the end of the conveyor, so that the distance between the farthest box and the box placement point on the conveyor was no more than 1300 mm, and the M1013 robotic arm was placed next to the pallet, so that the distance from the vertical translation rail to the farthest box or to the box placement point did not exceed 1300 mm.

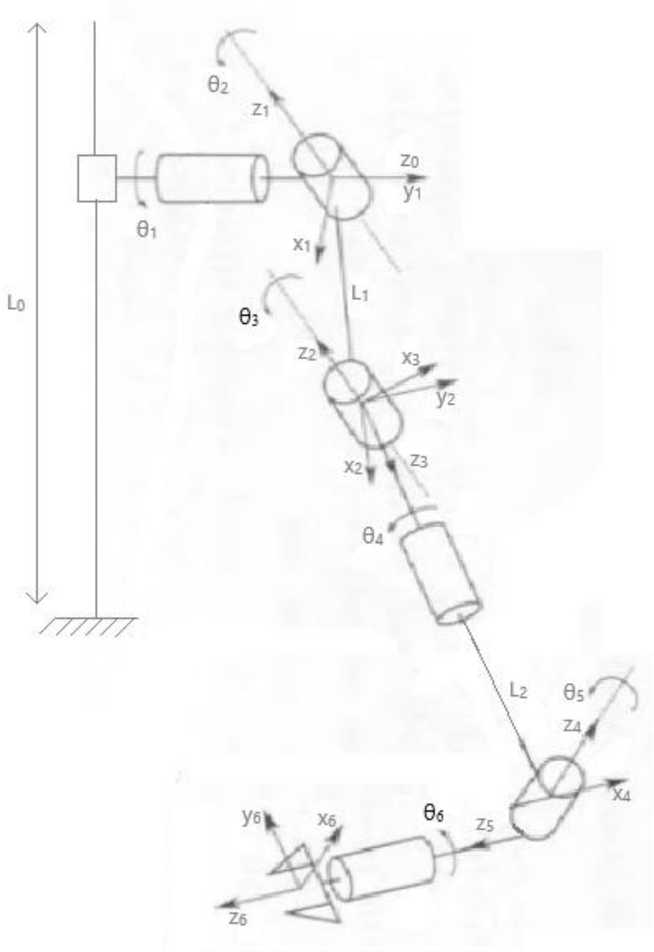

Figure 1. *Kinematic diagram of the Doosan Robotics M1013 robotic arm*

Table 2. *Doosan Robotics M1013 robotic arm joint parameter values*

Parameter (degrees)	**Minimum value**	**Maximum value**	**Default value**	**Tolerance**
θ_1	-360	360	-360~360	3/-3
θ_2	-360	360	-95~95	3/-3
θ_3	-160	160	-135~135	3/-3
θ_4	-360	360	-360~360	3/-3
θ_5	-360	360	-135~135	3/-3
θ_6	-360	360	-360~360	3/-3

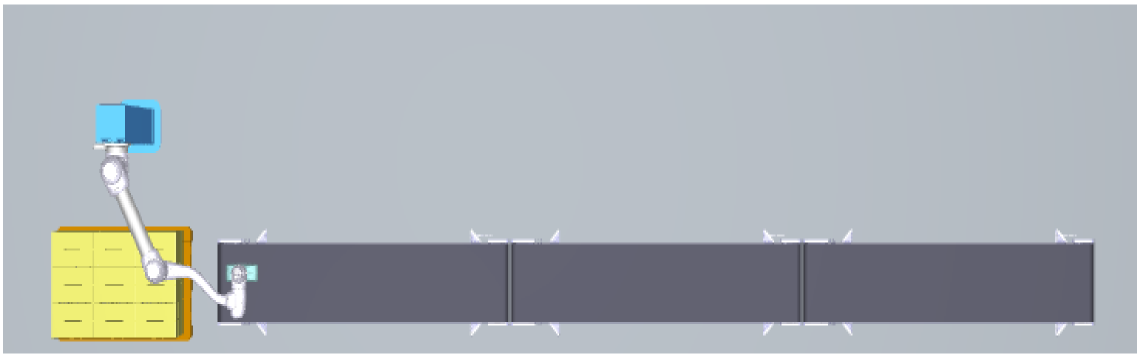

***Figure 2**. Layout plan*

Figures 3, 4 show side and front views of the assembly. Figure 5 shows a 3D image of the designed assembly.

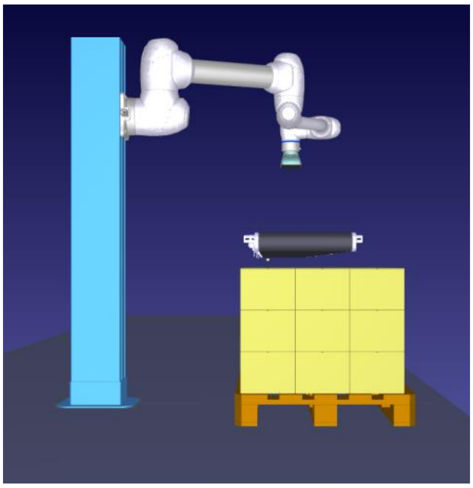

***Figure 3**. Side view of the assembly*

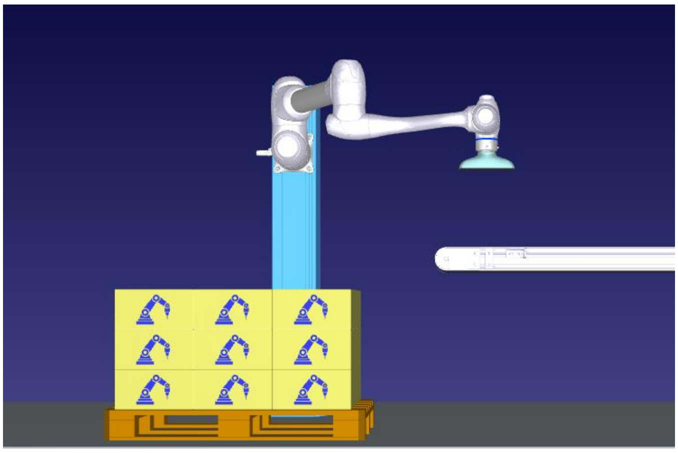

***Figure 4.** Front view of the assembly*

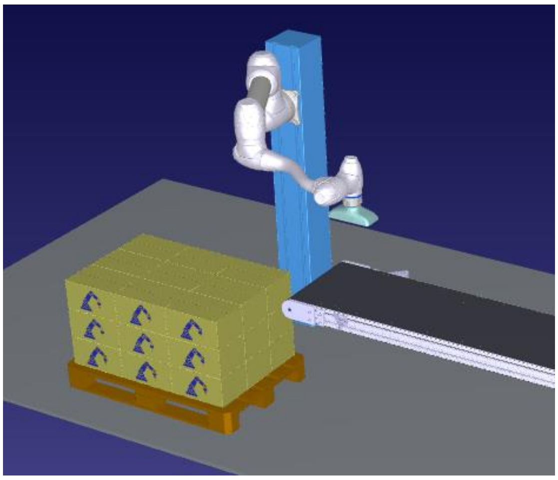

Figure 5. *Three-dimensional view of the assembly*

SOFTWARE IMPLEMENTATION

RoboDK is an offline simulation and programming application for industrial robots and automation. With RoboDK, users can create and test robot programs before deploying them on real equipment. This software allows for the simulation of robot movements in a virtual environment, which helps identify and correct problems before they enter the production line. RoboDK is compatible with a wide range of industrial robots and is widely used in industry to optimize processes and reduce the costs associated with deploying and operating robots.

Main features of RoboDK software:

- Advanced 3D Simulation: RoboDK enables 3D simulation of industrial robots, including their movements, trajectories, and interactions with the environment. Users can visualize robot performance in real time, identifying potential collisions or motion errors before they occur in reality. This detailed simulation helps optimize processes and increase operational safety.
- Compatibility with a wide range of robots: RoboDK is compatible with a variety of industrial robots from different industries. The software supports models from leading manufacturers such as ABB, FANUC, KUKA, Universal Robots and others. This universal compatibility allows users to work with various types of robots without having to learn different software for each type of equipment.
- Offline Programming: One of the most important features of RoboDK is offline programming. This means that robot programs can be developed and tested in a virtual environment, without having to shut down real equipment or interrupt production.
- Process automation: RoboDK provides tools for automating manufacturing processes such as welding, painting, part handling, or assembly.

- User-friendly interface: RoboDK's interface is designed to be easy to use, even for those with no experience in robot programming. The software offers a wide range of functionality, including a program editor and robot motion simulation, all in an accessible and intuitive environment.
- Import and export CAD files: RoboDK allows the import of CAD files in various formats, including STEP, IGES, STL and more, to create 3D models of the parts and equipment that the robot will interact with. This allows users to test and simulate manufacturing processes using accurate models of real parts.
- Path calculation and optimization: RoboDK includes an advanced path calculation system that helps generate the most efficient paths for robot movements. This helps optimize cycle times and reduce equipment wear, increasing the efficiency of manufacturing processes.
- Integration with external systems: RoboDK can be integrated with external systems such as PLCs (Programmable Logic Controllers), machine vision systems or other industrial equipment. This offers great flexibility in terms of integration with various types of automation and technologies.

RoboDK is used in a wide variety of industries, including automotive, aerospace, electronics manufacturing, packaging industries, and many more. Here are some examples of RoboDK applications:

- Automotive: In the automotive industry, RoboDK is used to optimize welding, assembly, part handling, and painting processes. Users can test manufacturing processes on a virtual robot before implementing them on real robots, which saves time and costs.
- Electronics Manufacturing: RoboDK is also used in the manufacturing of electronic components, including printed circuit board (PCB) assembly, part handling, and visual inspection. It helps create high-speed, high-precision automated processes.
- Welding and Material Handling: RoboDK is very useful in robotic welding and material handling processes. The software can be used to create precise welding paths and simulate robot movements in heavy or bulky part handling processes.
- Industrial Painting: RoboDK is used to simulate part painting processes in a virtual environment, optimizing the trajectories of painting robots to reduce paint loss and ensure uniform coverage.
- Testing and Inspection: RoboDK can be used to simulate part testing and inspection processes, such as dimensional inspection or quality verification of manufactured parts.

IMPLEMENTING THE DESIGNED STRUCTURE IN RoboDK

After placing the M1013 robotic arm, the relative position of the pallet in relation to the position of the linear translation rail was calculated, using the parameters in Figure 6.

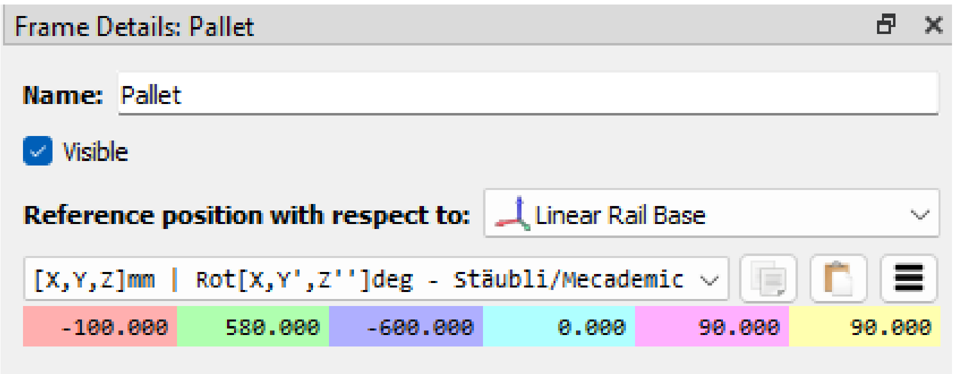

Figure 6. *Relative position of the pallet in relation to the linear translation rail*

Similarly, the relative position of the conveyor was also calculated, resulting in the parameters in Figure 7. The next step was to position the boxes on the pallet surface in a 3x3x3 box arrangement. For each box, the relative position to the pallet (Figure 8) and implicitly to the linear translation rail (Figure 9) was calculated.

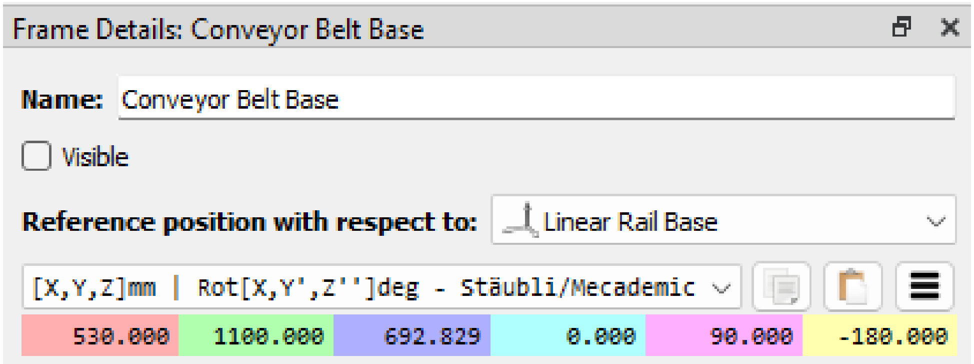

Figure 7. *Relative position of the conveyor in relation to the linear translation rail*

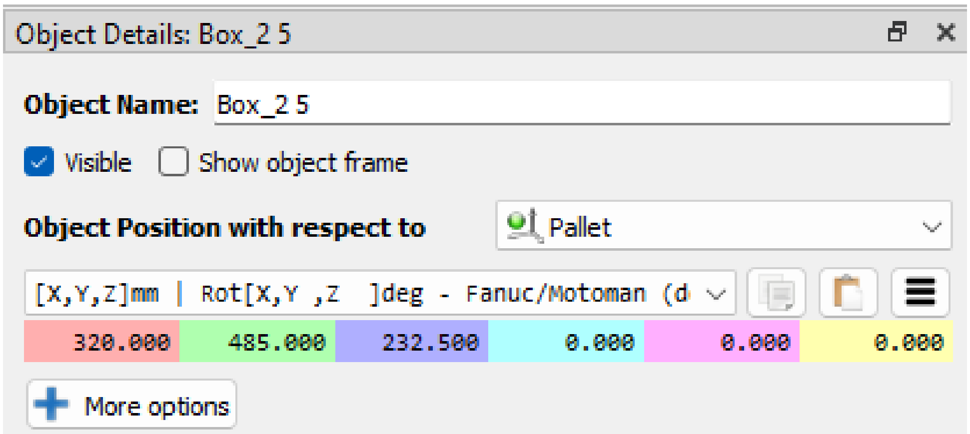

Figure 8. *Relative position of the box on row 2, column 2, layer 2 in relation to the pallet*

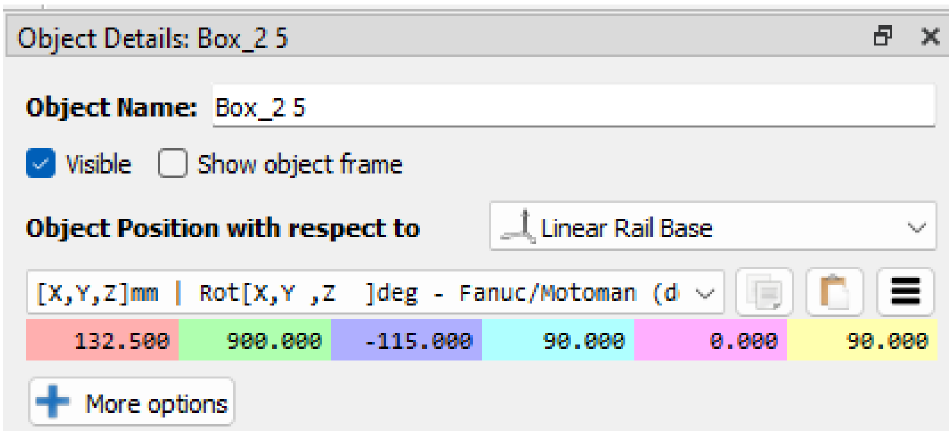

Figure 9*. The relative position of the box on row 2, column 2, layer 2 in relation to the linear translation rail*

After creating all the boxes, it was necessary to memorize the positions where the tool can attach each box separately (Figure 10) and memorize an approach position to the pallet of boxes from which the robot will depart to all 27 boxes (Figure 11). The approach position was chosen depending on the order in which the boxes are removed from the pallet or added, so that the tool's trajectory is not blocked by another box.

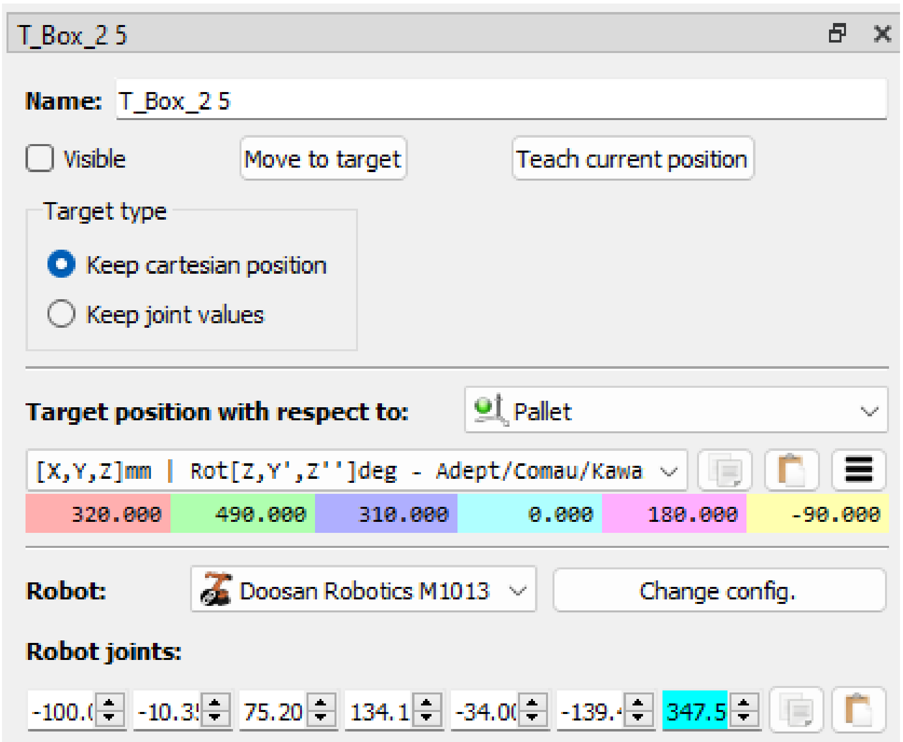

Figure 10*. The position where the tool can attach/detach the box on row 2, column 2, layer 2*

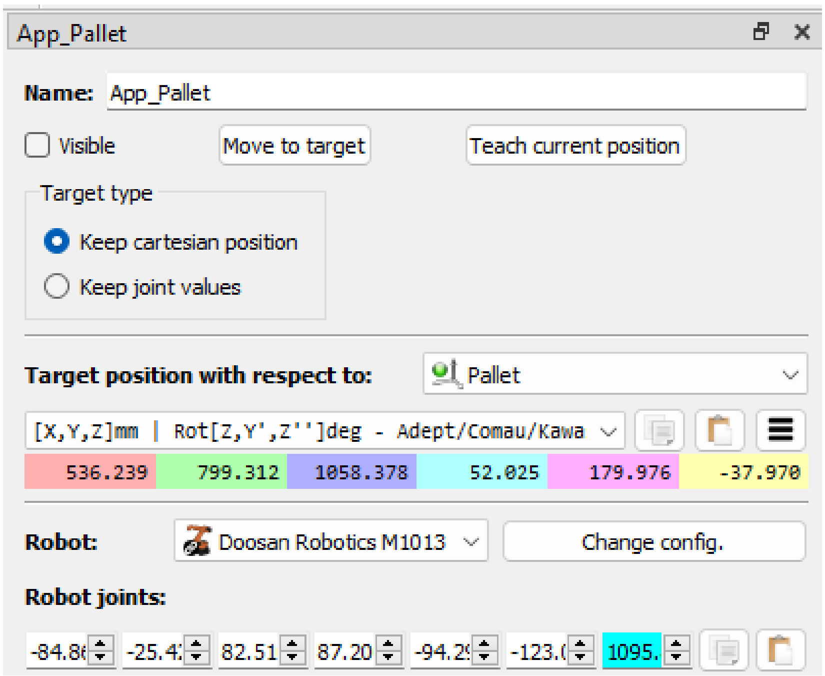

***Figure 11**. Approach position to the pallet*

For placing on or lifting from the conveyor, a placing position (Figure 12) and an approaching position (Figure 13) were calculated.

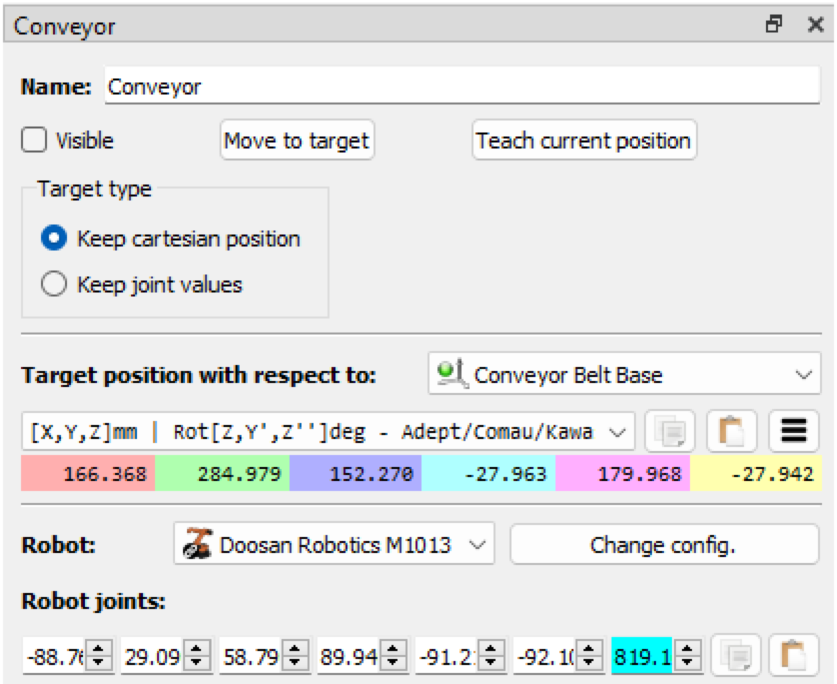

***Figure 12**. The position where the tool can detach/attach a box from the conveyor*

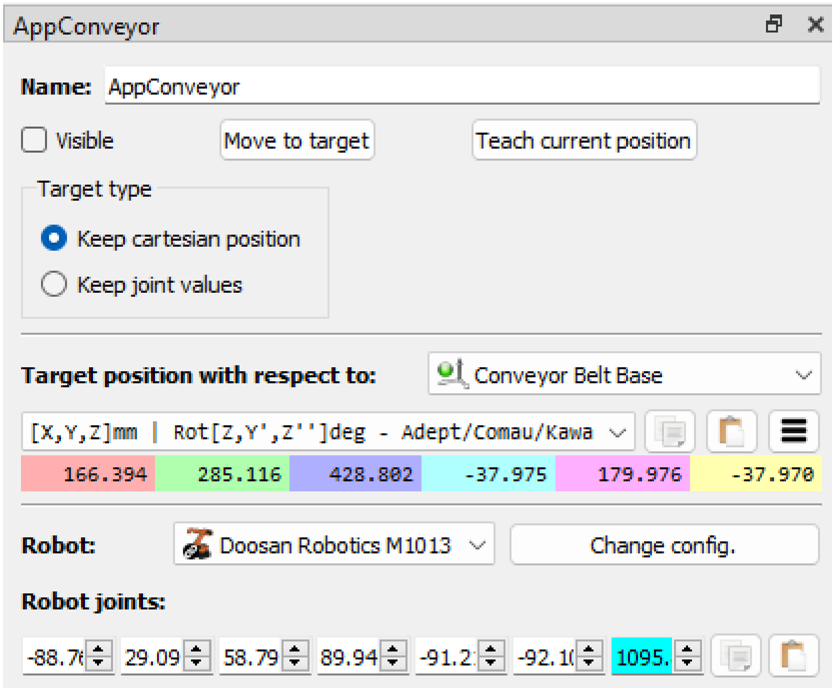

Figure 13. *Approach position to the conveyor*

Because the instruction to place or lift a box on the conveyor is the same regardless of the box, the "Place_Conveyor" program was built, which has the following instructions: set the conveyor reference system, set the required tool, move the tool to the approach position to the conveyor, move the tool to the attachment/detachment position, execute the box detachment/attachment command, and move the tool to the approach position to the conveyor (Figure 14).

Figure 14. *Program for placing/lifting a box from the conveyor*

There cannot be a single program for the instruction to place or lift a box off the pallet, because the position of each box is different, so a program was created that only has the attachment/detachment instruction (Figure 15).

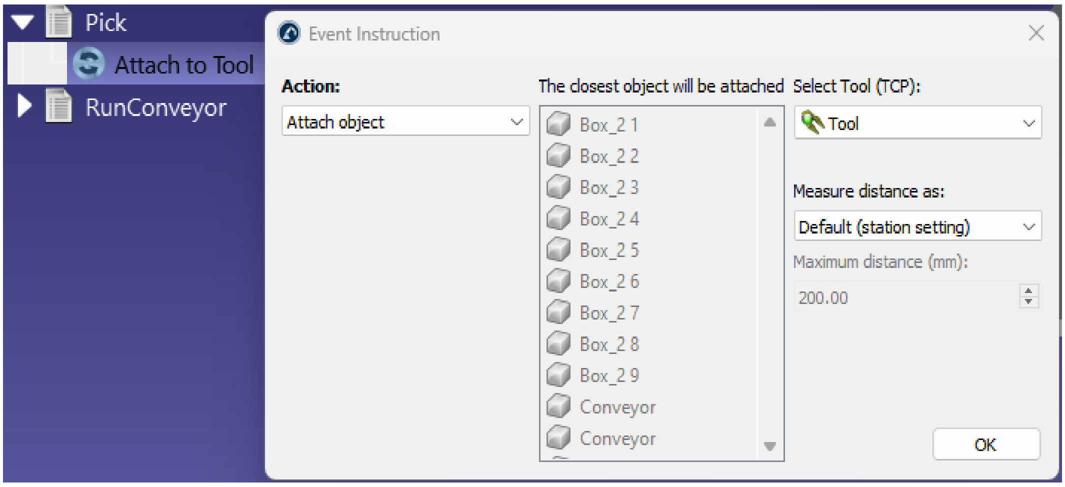

Figure 15. *Simple program with attach/detach instruction*

In the main program, the sequence of instructions necessary to lift and place each box was implemented in an order such that, on the tool's trajectory from the position approaching the pallet, to the box to be lifted/placed, it is not obstructed (Figure 16). The program's operating logic diagram is presented in Figure 17.

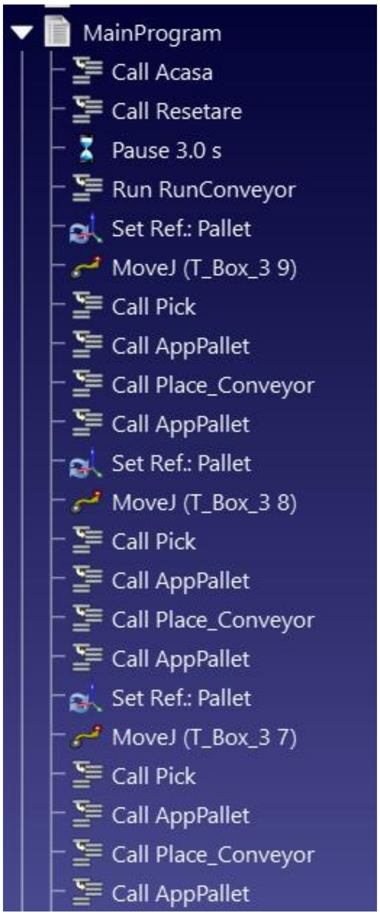

Figure 16. *The main program with the instructions for picking up and placing the first 3 boxes on the pallet on the conveyor*

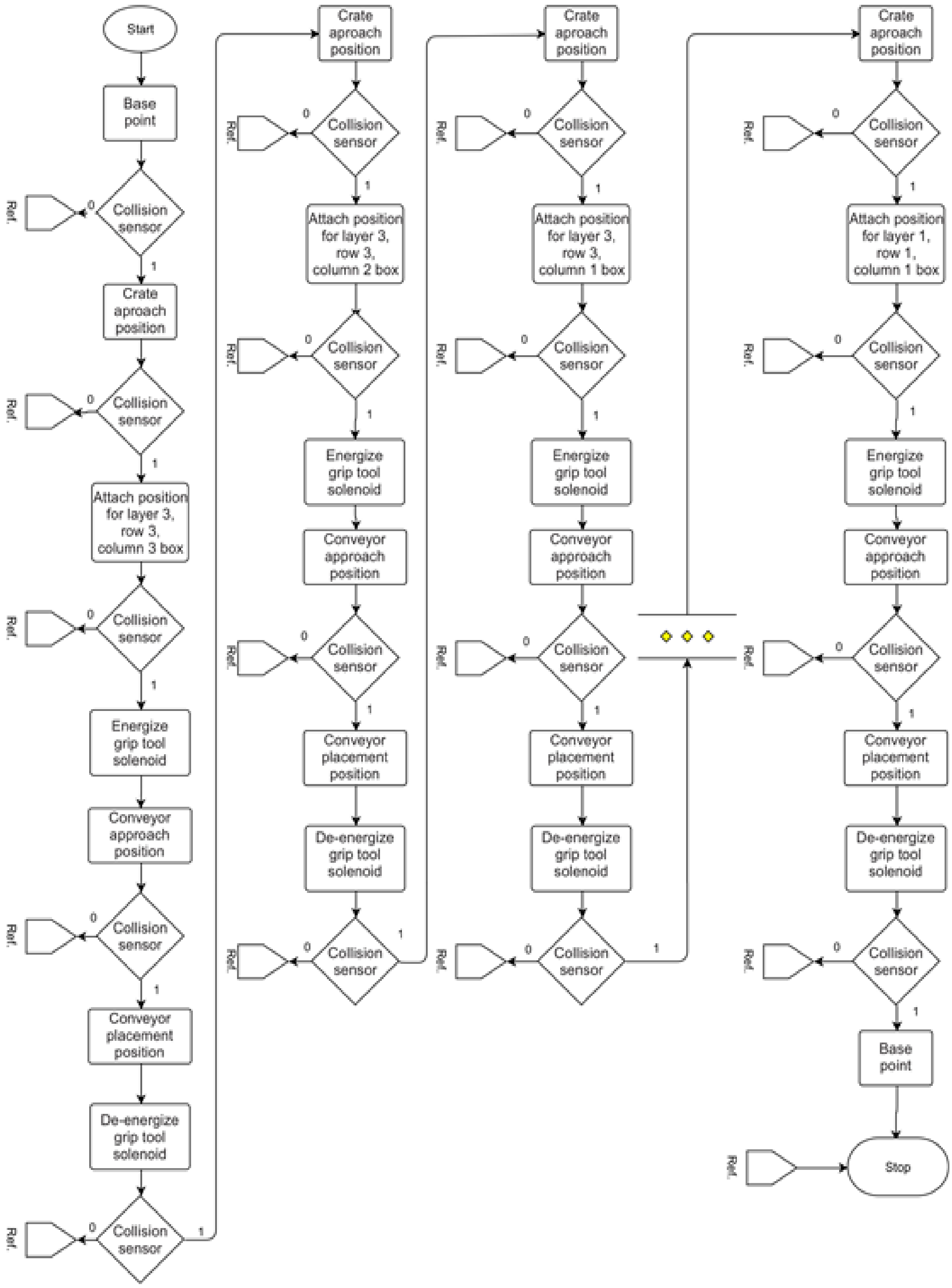

Figure 17*. Logical diagram of the program's operation*

CONCLUSIONS

RoboDK is an extremely useful software for simulating and programming industrial robots, offering a wide range of functionalities that help optimize manufacturing processes, including Doosan Robotics arms such as the Doosan M1013 collaborative robot (cobot). Here's how RoboDK can be used specifically with a Doosan M1013: program M1013 robot without having it physically present; create complex paths, loops, and conditional logic in RoboDK's visual interface; generate Doosan-compatible code (Python or specific script formats depending on Doosan's robot controller); simulate robotic processes such as pick-and-place, welding, painting, and inspection; test robot reachability, singularities, and collision avoidance before deploying on the real robot; import CAD files (STEP, IGES, STL) and generate robot paths directly from surfaces or curves; useful for tasks like polishing, milling, or inspection; integrate vision systems or other sensors; use RoboDK's API (Python, C#, etc.) to dynamically control robot behavior based on external data; use RoboDK to create a digital twin of your M1013 robot, useful for remote monitoring and predictive planning.

REFERENCES

[1] Alberto, C. et al., Collaborative pick and place using industrial robots, simulation and deployment, Vision Electronica vol.14, no. 1, 2020

[2] Almobaied, M., Metair, AA, Thabet, B.M., Prototype of 3-DOF Pick and Place Arm Robot, International Research Journal of Innovations in Engineering and Technology (IRJIET) ISSN (online): 2581-3048, Volume 7, Issue 3, pp. 1-9, March 2023.

[3] Ang, J., Tan, K., Forward kinematic solutions for a class of serial robots using the DH parameter method, IEEE transactions on robotics and automation, vol. 13, no. 5, pp. 796–805, 1997.

[4] Brooks, R.A., Planning collision free manipulation for pick and place operation, The International Journal of Robotic Research, 1983

[5] Chirikjian, G.S., Triangulation method for inverse kinematics, The International Journal of Robotics Research, vol. 11, no. 3, pp. 331–341, 1992.

[6] Chen, J., Wang, Y., Inverse kinematics solution of a robotic manipulator based on triangulation method, Journal of Control, Automation and Electrical Systems, vol. 19, no. 2, pp. 146–152, 2008.

[7] Corke, PI, Khatib, O., Robotics, vision and control: fundamental algorithms in MATLAB. Springer, vol. 73, 2011.

[8] Cui, Y., Shi, P., Hua, J., Kinematics analysis and simulation of a 6-DOF humanoid robot manipulator, 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), vol. 2. IEEE, 2010, pp. 246–249.

[9] Dahari, M., Tan, J.D., Forward and inverse kinematics model for robotic welding process using KR-16KS KUKA robot, Fourth International Conference on Modeling, Simulation and Applied Optimization, Kuala Lumpur, Malaysia, IEEE, 2011, pp. 1-6.

[10] Goswami, A., Robotics: Theory and Industrial Applications, Springer Science Business Media, 2010.

[11] Geraerts, R., De Schutter, J., Swevers, J., A survey on motion planning for robots, Mechanism and Machine Theory, vol. 42, no. 5, pp. 547–567, 2007.

[12] House, B., Malkin, J., Bilmes, J., The voicebot: a voice controlled robot arm, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2009, pp. 183–192.

[13] Han, B., Kim, K., Choe, Y., Forward kinematics solution of a 3-DOF serial robot using the DH convention, Journal of Mechanical Science and Technology, vol. 25, no. 1, pp. 99–105, 2011.

[14] Jangam, S.A., Patil, A.A., Patil, S.M., Modeling and Development of Multi-Arm Robot Using CATIA for Pick and Place Operation to Adapt Ease in Work, International Research Journal of Engineering and Technology (IRJET),Vol. 08, Iss. 05, May 2021

[15] Jahnavi, K., Sivraj, P., Teaching and learning robotic arm model, International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, 2017, pp. 1570–1575.

[16] Kalimullah, I.Q., Desikan, V., Kalaichelvi, V., Analysis of a proposed algorithm for point to point control of a 3-DOF robot manipulator, 3rd International Conference on Control, Automation and Robotics (ICCAR), IEEE, 2017, pp. 289–292.

[17] Koyuncu, B, Guzel, M., Software development for the kinematic analysis of a lynx 6 robot arm, International Journal of Computer and Information Engineering, vol. 1, no. 6, pp. 1575–1580, 2007.

[18] Karthikeyan, S., et al., ROS based pick-and-place robot simulation using MOVEit, International Journal of Research in Engineering, Science and Management, Volume-3, Issue-6, June 2020.

[19] Li, F., Wang, L., Trajectory planning for robot manipulators: a review, Robotics and computer integrated manufacturing, vol. 26, no. 3, pp. 201–212, 2010.

[20] Lucak, D., Milic, M., Simulation of a pick-and-place cube robot by means of the simulation software Kuka Sim Pro, Electronic ETF, 2018

[21] Megalingam, R.K., Boddupalli, S., Apuroop, K., Robotic arm control through mimicking of miniature robotic arm, 4th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2017, pp. 1–7.

[22] Mathew, E.B., et al., Robotic arm control through human arm movement detection using potentiometers, International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE). IEEE, 2015, pp. 298–303.

[23] Marcu, C., Lazea, G., Robotin, R., An openGL application for industrial robots simulation, International Conference on Automation, Quality and Testing, Robotics, vol. 2. IEEE, 2006, pp. 254–259.

[24] Moran, E.M., Evolution of robotic arms, Journal of Robotic Surgery. Vol. 1, p. 103-111, June 2007, https://doi.org/10.1007/s11701-006-0002-x

[25] Naser Alanabi, J.S., Shrivastava, J., Performance comparison of robotic arm using Arduino and Matlab Anfis, International Journal of Scientific & Engineering Research, vol. 6, no. 1, 2015.

[26] Ohara, K. et al., Component-based robot software design for pick-and-place task described by SysML, 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Incheon, Korea (South), 2011, pp. 124-127.

[27] Olukayode, O., Adeboye, B., Alawode, K., Design and fabrication of a manually controlled electromechanical manipulator for educational purpose, Journal of Engineering and Environmental Sciences, 2022

[28] Qian, W., et al., Manipulation Task Simulation using ROS and Gazebo, 2014 IEEE International Conference on Robotics and Biomimetics December 5-10, 2014, Bali, Indonesia.

[29] Qassem, M.A., Abuhadrous, I., Elaydi, H., Modeling and simulation of 5-DOF educational robot arm, 2nd International Conference on Advanced Computer Control, vol. 5. IEEE, 2010, pp. 569–574.

[30] Popescu, C., Neuro-fuzzy control of mobile robots, UPG Ploieşti, 2015

[31] Pei, S., Cui, F., Chu, F., Application of solid works in mechanical design and drafting courses, 3rd International Conference on Science, ICCSR 2014.

[32] Suhaimin, S.C., et al., Analysis of point-to-point robotic arm control using pid controller, 7th International Conference on Mechatronics Engineering (ICOM). IEEE, 2019, pp. 1–6.

[33] Shafiee, A., Pirbazari, M., Singularity analysis of redundant robots using polynomial geometry and numerical optimization, Robotics and Computer Integrated Manufacturing, vol. 34, pp. 67–80, 2015.

[34] Spong, M.W., Hutchinson, S., Vidyasagar, M., Robot modeling and control, Jon Wiley & sons Inc, ISBN-100-471-649, 2005.

[35] Salozer, M., Portero, P., CAD design and modeling of a robotic arm for automated harvesting, Electronic ETF, 2018

[36] Torres, P., et al., Bin-Picking Solution for Randomly Placed Automotive Connectors based on Machine Learning Technique, Electronics, February 2022

[37] Tsai, CY, et al., A Hybrid Switched Reactive-Based Visual Servo Control of 5-DOF Robot Manipulators for Pick-and-Place Tasks, IEEE System Journal, 2015.

[38] Zhang, Y., et al., A conveyor-belt based pick-and-sort industrial robotics application, Second IEEE International Conference on Robotic Computing (IRC). IEEE, 2018, pp. 17–24.

[39] Zhang, H., Xie, X., Identification and analysis of singularities in robot manipulators, Robotics and Computer-Integrated Manufacturing, vol. 26, no. 6, pp. 615–623, 2010.

[40] Wong, K., Kress-Gazit, H., High-level Task Specification to Robot Operating System (ROS) Implementation, IEEE IRC 2017

[41] https://cobot-smx.jp/info/wp-content/uploads/2023/08/Doosan_Robotics_User_Manual_V2_10_v2_8_EN_A_Series.pdf

Received: April 2025; Revised: May 2025; Accepted: May 2025; Published: May 2025