

On the possibilities and the advantages of robot off-line programming

Teaching of off-line programming methods as a tool to further industrial implementation utilization

[Radovan Holubek, Karol Velišek, Michala Šimúnová, Nina Vetricová, Daynier Rolando Delgado Sobrino]

Abstract— Current trend in automation area is focused to the innovation in all fields of the technologies and technological devices. Probably industrial robots and manipulators belong to development too. Industrial robots and manipulators belong to progressive elements of automation. Modern concept uses the offline programming as a tool integration of the virtual CAD models into programming environment and implementation to the education process. Offline programming carries a lot of advantages such as possibility of the programming without real robot system, possibility of the collision state debugging. Implementation of CAD models in virtual environment of Robot studio achieves better workplace station of creating robotic system and also it generates rather trajectories of the robot path in this system.

Keywords— Industrial robot, off-line programming, CAD model, education,

I. Introduction

Contribution deals with possibilities use of the robotic simulation mainly oriented to the offline programming with integrated CAD 3D models created by students during education process. It is used mostly to the automation the production applications and process. A robotic workstation includes a robot system, containing one or more robotic tools or grippers used for the technological operation or manipulation with semi-products or final products. Industrial robotics and automation represent the best solution in the framework of the productivity and flexibility. Nevertheless, the programming of industrial robotic system for a specific application is still very difficult, time-consuming, and expensive. Nowadays, in practical industrial applications, there are two main possibilities of robotic programming methods, which are, online programming (including lead-through and walk-through) and offline programming (OLP) [1].

Radovan Holubek, Karol Velišek, Michala Šimúnová, Nina Vetricová,
Daynier Rolando Delgado Sobrino
Institute of Production Technologies,
Faculty of Materials Science And Technology in Tmava
Slovak University of Technology

In general, robot programming methods are normally classified to:

- On-line programming methods, where a robot is used to generate the program,
- Off-line programming methods, used where there is no need to access the robot in order to develop the control program. Well at least, no need until the final test of the program. It is basically consisting of writing a program using a text-based robot programming language like ‘Siemens Process Simulate’ or ‘ABB RobotStudio Figure 1 shows the main methods used in the robot programming [2].

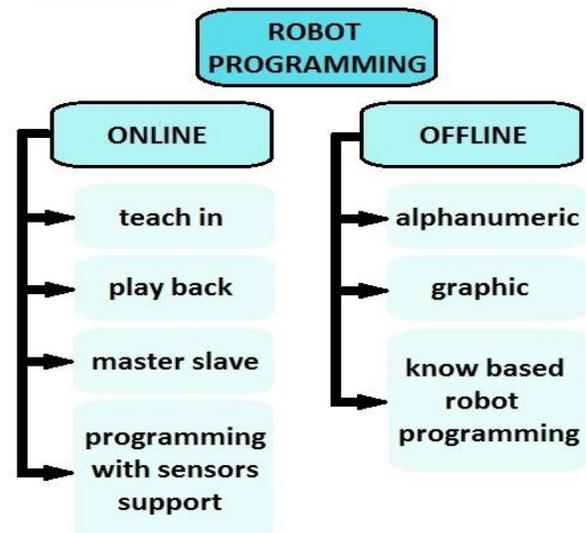


Figure 1 The main methods of the robot programming

The basic element of industrial robot offline programming is to design and create the main proposal of the surrounding peripheral in the appropriate CAD software. In this case, it is to create the CAD models in the CATIA software. This mentioned proposal is realized by students in the educational process during the study exercises/lessons focused on the designing of 3D model with using CATIA software.

Conventional robot simulation and offline programming of the robotic tools is mainly oriented to the skilled programmers rather than a person without too much skilled background focused to this problematic. In this contribution, it is primary contained creation of the virtual workplaces for offline

programming and simulation by using CAD models imported to the programming environment [3]. Very important at the offline programming of the robot system is to need to be considered the aspects of modeling a geometrical entity; generating robot targets and paths; adjusting robot targets; planning path configuration; positioning snap objects etc.

A. *Creating of a robot trajectories by means of the Off-line method with integration CAD models*

OLP programming method has a many advantages against the online method. First, the programming process does not require the real robot, minimizing the production robot down time. Robot programs can be developed before in the design / production cycle and programming can be carried out in parallel with production rather than in series with it. Second, programs generated offline are more flexible than jog-and-teach method. OLP changes can be incorporated quickly by only substituting the necessary part of the program and previously developed routines can be easily included in new programs. Third, simulation is usually incorporated into the OLP method. As a result, programs can be pre-checked, hereby confirming the robots' movements, minimizing the chance of error and therefore improving productivity and safety. There is also a greater possibility for optimization of the workspace layout and the planning of robot tasks [4].

The main advantage of OLP programming is the use of the possibility of predisposition visualization resources which facilitate subsequent implementation of robotized workstation. Creating workstation in a virtual environment will allow us to better:

- perceive conceptual design as complex workplace,
- possibility of interactive correction of the position of the workstations,
- integration of CAD models into Robot Studio environment and recognition of particular edges and points for then define the exact targets of robot,
- visualization of kinematic motion of the robot, as long as the simulation is based on the use of virtual controller, which is identical to the real character controller,
- simulation of material flow in the workplace,
- whole simulation of the workplace, ensuring the elimination of the collision and adverse influence during assembly, or technological operation,
- the creation of OLP and the whole workplace can reduce the total cost of investment, as long as is possible to determine in advance the optimal solution in terms of material flow and the overall layout of particular workplaces.

OLP starts from a 3D CAD model (Catia, Solid Works etc.) primary starts of the creation workstation and

workpieces. Very important part of the individual modeling of workpieces, components and supporting of peripherals create their proper modeling in software CAD, which is software CATIA in our case.

In our case, it is necessary to save model as STL format and follow it with the second software designed to convert 3D models to IGS format, however Catia doesn't know directly export this format. Consequently, it is possible to insert modeled parts into Robot Studio, but we must remember to tick to show all edges and points on the given model.

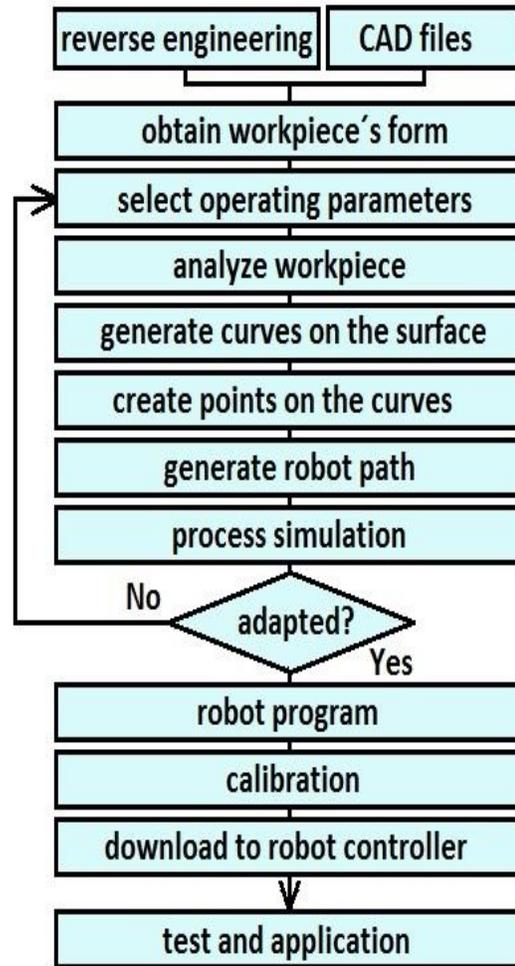


Figure 2 Procedures of the generated trajectories of the robot by method OLP from CAD file [6]

OLP methods, which utilize 3D CAD data Figure 2 of a work piece to generate and simulate robot programs, are widely used for automation system with large product volumes. Herein the complete robotic work place is modelled in 3D in Robot Studio. The user can test the reachability, fine-tune properties of robot movements and handle process related information before generating the programs that can be downloaded to the robot controller [5].

Exact information obtained from CAD models will be used in the next time to generate trajectories of the robot. Through the CAD interface, any user with basic CAD skills will be able to define the robot working paths and organize them in the

desired sequence (definition and parameterization of robot positions/orientations, reference frames and trajectories). After completing the design, a program converts it into robot programs (off-line robot programming) (Figure 3). The generated programs can be immediately tested for detailed tuning and a set of tools is then available to speed up corrections, if necessary. Depending on the complexity of the robotic cell, this process can be completed in just a few minutes, representing a huge reduction in programming and set up time.



Figure 3 Education concept by OLP method. Working in an office environment, the user can generate robot programs without interrupt production. [7]

B. Interesting steps of the OLP method

OLP is more complex than online programming because the programming method should have 3D robot targets but also it should have the trajectory of robot motion and it should optimize and generate the sequence of the path process [8]. The key steps of OLP are shown at Figure 4.

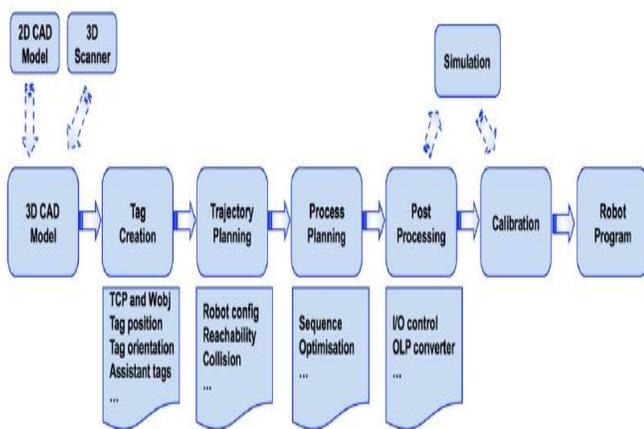


Figure 4 Procedures of the generated trajectories of the robot by method OLP from CAD file [8]

II. Offline programming of an ABB robot using imported CAD models in the RobotStudio

On the basis of previous analyzes, the required issues were determined by a specific task, which have been aimed to verify the applicability of the theoretical creation of the workstation by offline method [7]. At the beginning it was necessary to create a workpiece who having a rugged character as shown in Figure 6. Workpiece has projections which characters are straight, and curved surfaces. It is therefore possible to deduce that when creating individual TARGET, respectively PATH will be generated motion in a linear, but and curve move. The process of CAD model importing created in CAD software CATIA is mentioned above [9]. In RobotStudio it was created a new workstation as is shown in Figure 6, where the imported suitable type of the robot is shown. In the next step it was selected a work table from the library on which the aid of command "SET POSITION" gives workpiece into a suitable position. On the basis of appropriate positions of the workpiece we can create individual targets. Very important is to take into account in what position the tool comes to the workpiece (coming to him perpendicular or at an angle). Target orientation should be alignment oriented to the axis of the tool. Correct orientation of the target can be verified by the command "JUMP TO TARGET". If a problem occurs when we applied this command, we must re-orientate target with respect to the tool. The correct orientation of Target allows us to generate a sequence of "PATH". Verify the correctness of motion is carried out using the command "MOVE ALONG PATH". Important is to check before creating offline continuous simulation of the PATH (feasibility of individual "PATH") with the command "CHECK reachability". Important is to check before creating offline continuous simulation of the PATH (feasibility of individual "PATH") with the command "CHECK reachability". Setting of the simulation rests in choosing the sequence of movements and their gradual steps Figure 5.

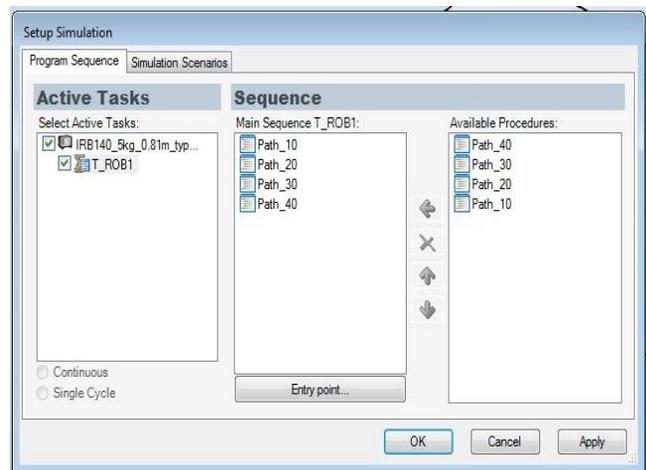


Figure 5 Setting of the Simulation setup in OLP – RobotStudio

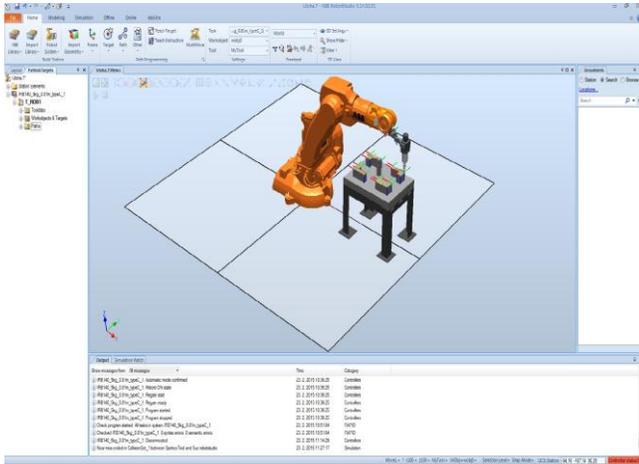


Figure 6 3D CAD model integrated to the Robot studio environment

As the simulation is continuous and the robot goes into all TARGET respectively. "Path" we can still adjust "Zone", which tells us how accurate robot has come to that target. Subsequently, using the commands "Offline" getting to the generated program, which can subsequently be used as a backup for the control of real robot trajectories have been generated using the offline programming [11].

The generated program is written in the RAPID language, which represents the interface between the CAD model and the program writing. In our educational process, we try to create CAD models with all functional edges and surfaces to ensure problem-free integrity of live models after their importing to Robotstudio. Usually it has been arisen a problem of edge recognition and in that case it has been also difficult or sometimes impossible to create the targets in some special items (it is possible to talk about death model).

These unwanted facts should be eliminated during the importing to the mentioned software RobotStudio, where it is necessary to define/confirm "Surface Model- render both sides of surfaces". If this step would be forgotten or it hasn't been realize by this procedure, whole process of the model importing would be dysfunctional because we would have not the functional surfaces and edges for the creating of accurate points for the robot navigation in the process of target creation of individual model parts/devices.

The process of the individual target and Path creating has been noted above. The complete final output of defined Target/Patch can be subsequently converted by this RobotStudio software to Rapid language, which is the main base of the real robot movement used under the real conditions in the production.

In the figure 7, it is shown the generated program in Rapid language. This program has been generated on the base of imported model of work table with the metal plate where the welding process of single lugs is simulated as is shown in Figure 6. The individual sequences of the movement series have been sequential created in Figure 5. In the depending of the access to the individual purposes/targets it has been necessary to define the angle and slope of the end effector to

get closer to the individual targets. In the final phase, there is this defining generated also in Rapid language. The simulation in the offline environment is such a great advantage in a real program debugging in the real environment. It may have prevented the collision situation, and unwanted factors thanks to accurate modelled CAD models.

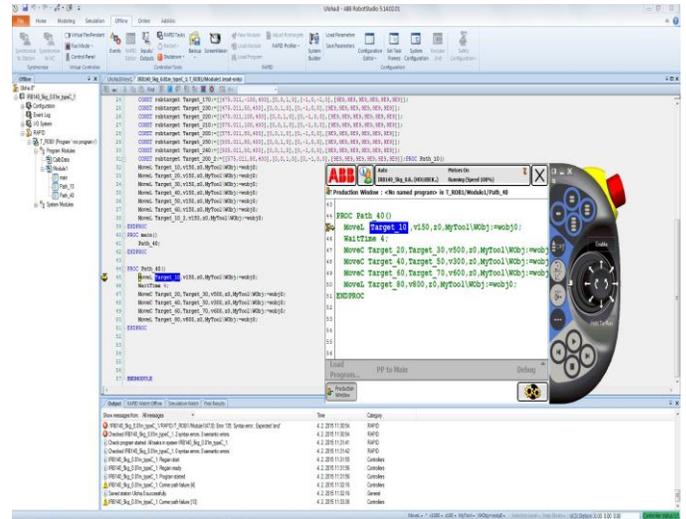


Figure 7 Generated Rapid language and its visualization with flexpendant

III. Collision detection state between the robot tool and CAD models

A. Debugger collision state

When creating a robotic workplace with the integration of CAD models by the method OLP during the teaching process, it is necessary to take into account the tool placed in the robot. It may occur to the several collisions such as conflict states of robot arm and surrounding peripheries or instrument versus CAD model during the implementation of the various technological or handling operations [12]. The possibility to simulate the trajectories of the robot or respectively its tool on CAD model of the real technological process / handling operations can be considered as a great advantage. At first, it should be created the individual targets and paths according to the appropriate incoming tool under the most appropriate angle to CAD model by the offline programming method in the workplace of RobotStudio. As first the bonds where can be some collision states detected should be selected via the "Collision Set". In our case it is the tool (welding end effector) and CAD model, where the welding of four lugs is simulated as is shown in Figure 8.

After the defining of the conflict bonds between the tool and the CAD model it is necessary to set at what distance of the tool and the material the simulation of collision will be indicated. Within the simulated application, we selected the 3 mm distance where we defined the dependences by two colors. Yellow color represents (in this case) the value of 3 mm, which is the limit but not critical. The other targets which are

below the value of 3 mm are evaluated as critical and diagnosed as collision state.

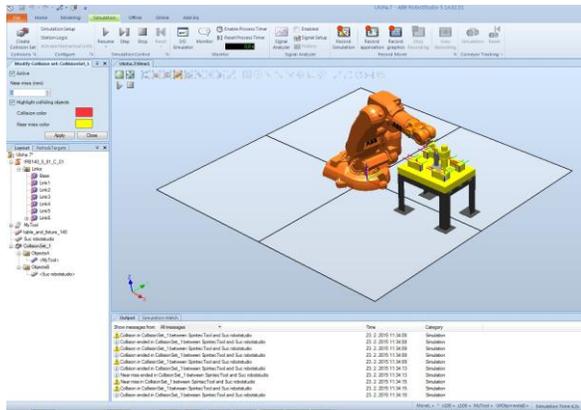


Figure 8 RobotStudio interface showing the interaction of the robot and its targeted object without collision (the ones in yellow color)

In the Figure 9, there is possible to see the detected collision situation in the simulation, where the selected target shows the collision state and the instrument and CAD model have a lower value than the value of 3 mm. It is necessary to modify the target and re-verify the critical points [10].

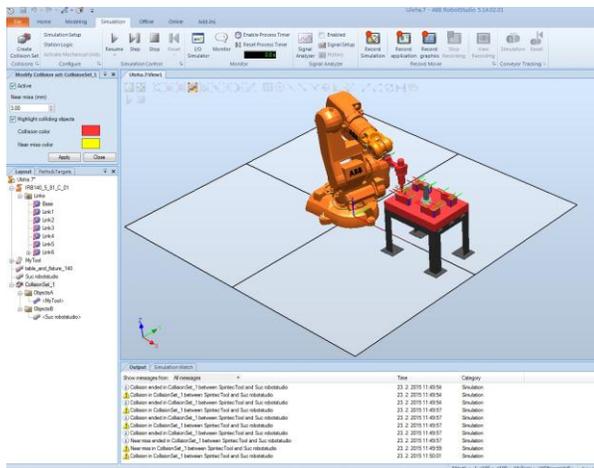


Figure 9 RobotStudio interface showing a collision detection between the robot tool and its targeted object (the ones in red color)

iv. Conclusion

The possibility to define the critical places and debugging respectively adjusting of the individual targets of the tool to CAD model under the different angle and other is one of the biggest advantages of the simulation of the individual robot movements with the individual peripheries. It is necessary to mentioned also financial aspect because different technological devices form large costs and the possible collision conflicts could have the fatal and destructive consequences on the individual devices in the real life. That's why is necessary to see the debugging and elimination of the collision conflicts in the virtual environment as an advantage and see the reason why the program with the debugged positions bring to the real workplace. The another advantage is

saving a time, real application deployment, most collisions have been debugged in a virtual environment by method OLP and thus for the real deployment it can be used already increased speed of the robot movement. Statistics show that most time is spent during debugging collisions and position correcting of the robot/tool to the components by deploying in the real process, as it works with minimum speed of the robot. This minimum speed has the consequences such is an increase of the real deployment in the industry. This time is significantly reduced by using OLP method.

Acknowledgment

This work was supported by KEGA 027STU-4/2014: Establishment virtual laboratory of robotics and manipulation techniques.

References

- [1] Ang M.H. Jr, L. Wei, Lim Ser Yong: As industrial application of control of dynamic behaviour of robots - a walk-through programmed welding robot. In: Pro-ceedings of the IEEE international conference on robotics and automation. San Francisco, CA; April 2000.
- [2] OSHA Technical Manual (OTM), Section IV: Chapter 4, Industrial robots and robot system safety
- [3] Liwei Q., Xingguo Y., Haipeng W.,Tao L.: Virtual engineering: challenges and solutions for intuitive offline programming for industrial robot. RAM 2008: 12-17.
- [4] Pan Z., Polden J., Larkin N., Van Duin S., Norrish J.: Robotics and Computer-Integrated Manufacturing. Published/Hosted by Elsevier Science. ISSN: 0736-5845.
- [5] Liwei Q., Xingguo Y., Haipeng W.,Tao L.: Virtual engineering: challenges and solutions for intuitive offline programming for industrial robot. RAM 2008: 12-17.
- [6] Si Hao D., Zhen Hua C., Fang Dan D., Han Lin L., Ghislain M.: Application of robot offline programming in thermal spraying. Surface and Coatings Technology, Volume 206, Issues 19–20, 25 May 2012, Pages 3875-3882.
- [7] P Neto, JN Pires, AP Moreira: CAD-based off-line robot programming. Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on, 516-521
- [8] Pan Z., Polden J., Larkin N., Van Duin S., Norrish J: Recent progress in programming methods for industrial robots. In: Robotics and Computer-Integrated Manufacturing, Volume 28, Issue 2, April 2012, Pages 94–94
- [9] Neto P., Mendes N.: Direct off-line robot programming via a common CAD package. In: Robotics and Autonomous Systems Volume 61, Issue 8, August 2013, Pages 896–910
- [10] Holubek, Radovan - Delgado Sobrino, Daynier Rolando - Košťál, Peter - Ružarovský, Roman. Offline programming of an ABB robot using imported CAD models in the RobotStudio software environment. In Applied Mechanics and Materials. Vol. 693 (2014), s. 62-67. ISSN 1660-9336. Projekt: 1/0285/12 113.
- [11] F. Pauker, I. Ayatollahi, B. Kittl: OPC UA for machine tending industrial robots; 2nd International Conference On Advances In Robotics Engineering - AMRE 2014, Zürich (eingeladen); 25.10.2014 - 26.10.2014; in: "Proc. of the 2nd Intl. Conf. on Advances In Robotics Engineering", (2014), ISBN: 978-1-63248-031-6; S. 79 - 83.
- [12] HOLUBEK, Radovan - VELÍŠEK, Karol. Incorporation, programming and use of an ABB robot for the operations of palletizing and depalletizing at an academic-research oriented to Intelligent manufacturing cell. In Applied Mechanics and Materials : 11th International Conference Industrial, Service and Humanoid Robotics, ROBTEP 2012; Strbske Pleso, High Tatras;14-16 November 2012. Vol. 282 (2013), s.127-132. ISSN 1660-9336