Virtual Prototyping and Commissioning of Manufacturing Cycles in Robotic Cells

Published Version:
Raffaeli, R., Bilancia, P., Peruzzini, M., Pisu, S., Berselli, G., Pellicciari, M. (2024). Virtual Prototyping and Commissioning of Manufacturing Cycles in Robotic Cells. Cham : Springer [10.1007/978-3-031-58094-9_43].

Availability:
This version is available at: https://hdl.handle.net/11585/972698 since: 2024-06-25

Published:
DOI: http://doi.org/10.1007/978-3-031-58094-9_43

Virtual prototyping and commissioning of manufacturing cycles in robotic cells

Roberto Raffaeli[1[0000-0003-0301-454X]], Pietro Bilancia[1[0000-0002-4931-1745]], Margherita Peruzzini[2[0000-0003-2260-0392]], Simona Pisu[2[0009-0009-5996-3235]], Giovanni Berselli[3[0000-0003-0093-3006]] Marcello Pellicciari[1[0000-0003-2578-4123]]

[1] Dept. Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia, Italy
[2] Dept. Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy
[3] Dept. of Mechanical, Energy, Management and Transportation Engineering, University of Genova, Genova, Italy
roberto.raffaeli@unimore.it

Abstract. In the context of Industry 4.0, industrial robots are experiencing wider application fields due to improved capability of executing flexible and diversified manufacturing cycles. The implementation of mechatronic automation systems remains a critical task, since it must cope with many heterogeneous domains, from layout definition to design of mechanical, actuating, and sensing devices, control logic coding, testing and optimization of the whole system. This paper leverages a Python-based connection between a simulation software for robotic cells, i.e. RoboDK, and a PLC system, i.e. Beckhoff TwinCAT, to realize a holistic virtual prototyping environment able to support the design and virtual commissioning of automation systems. The proposed approach is demonstrated with a case study comprising a robotic deburring cell. The resulting application shows the ability to effectively debug logic code, optimize the sequence of manufacturing tasks, and monitor the primary kinematic quantities.

Keywords: Virtual prototyping, Robotic cell, Virtual Commissioning, TwinCAT, RoboDK.

1 Introduction

Industry 4.0 (I4.0) pushes towards flexible and reconfigurable production systems to cope with highly customized products and ever smaller production batches [1]. Robotic cells made of anthropomorphic manipulators, exchangeable devices, general purpose tooling, and suitable sensing capabilities are feasible setups to reach the requested level of variability in manufacturing cycles [2].

Industrial robots are utilized in diversified applications. Traditional ones are handling, picking and placing, welding, painting, and all types of operations based on predefined movements and paths in which the robots replace humans in highly repetitive and alienating duties. In the context of I4.0, new tasks are investigated which include milling and deburring [3], assembling, inspection, material deposition such as gluing and wire arc additive manufacturing [4]. Such applications require complex and highly variable paths to be followed which depend on the geometrical characteristic of the product being processed. Furthermore, the production mix may vary continuously in response to customization needs.

Therefore, two major needs emerge. The first one is related to the impossibility of teaching the robot with all the exact positions to be reached. As is well known, robots are highly repeatable in positioning but not so accurate. Therefore, the level of accuracy of robots must be increased as well as errors correction strategies must be devised to overcome the intrinsic limits of the manipulators which often hinder the possibility to use them in the new contexts. The second aspect is linked to the need to exploit the possibilities of programmability and flexibility of the robot with respect to continuously variable tasks. It is necessary to be able to exploit virtual environments in which to design, simulate and validate complex work cycles to ensure robust and efficient sequencing of operations, but also the achievement of optimized performances. This second aspect is faced in this work.

Flexible and reliable programming approaches are needed to cope with the limited development times and the required logic complexity. In industrial configurations, Programmable Logic Controllers (PLCs) and robot controllers must be coordinated and integrated [5]. Then, there is the need to design the robotic cell, simulate its behavior for operations sequence verification, performance measurement, and optimization of the working cycle by evaluating the correctness of both PLC and robot control codes.

The pursuit of this objective is founded on the principles of digital manufacturing, wherein production data management systems and simulation technologies are jointly used for optimizing manufacturing systems before starting the production and supporting the ramp-up phases [6]. In particular, Virtual Commissioning (VC) is the process by which control code can be evaluated, tested and optimized in a virtual environment that includes simulated hardware devices, especially mechatronic systems such as robotic cells [7]. This hybrid environment enables realistic robot and device validation through the signal mapping of robot programs from simulation software to the PLC. However, as pointed out by Makris et al. [8], although many commercial tools are available, the realization of VC is far from being fully automated and straightforward.

In this context, a general approach to realize an effective virtual prototyping of mechatronic systems for flexible manufacturing cycles driven by industrial PLCs is introduced. In particular, this work focuses on a solution to exploit the potential of RoboDK software as a virtual prototyping platform for robotic cells, in which robot instructions and devices behaviors are modelled to allow realistic simulations of an industrial setup. In addition, the work proposes the method implementation into an integrated tool, able to define and optimize the desired manufacturing operations into the virtual environment. The execution logic is developed using an instance of the PLC environment, i.e., Beckhoff TwinCAT, which is the real PLC to operate in the industrial shopfloor.

To test, validate and optimize the code in the conceptual design phase, a connection between the control logic and the virtual prototype has been realized. For this purpose, Python routines have been developed and integrated in the RoboDK environment to exchange I/O signals with a TwinCAT PLC instance. The established link between the two software allows the VC of the whole robotic cell to be performed in a holistic way, assessing the control logics effects on the overall functionality and production performance. Moreover, product processing and material flow in the shop floor, involving sequence of operations and interdependencies between the production phases can be analyzed, which is becoming increasingly crucial to the realization of effective digital twins of production plants, as highlighted in [9].

Specifically, the solution is demonstrated on a robotic cell performing deburring operations on cast components, showing its feasibility and effectiveness in defining the manufacturing operation and optimizing the related performances. The contributions of the presented work can be recognized as follows:

- The definition of a design-oriented environment for the holistic design of the robotic cell, compared to the more programming- or simulation-oriented approaches;
- The creation of a software connection between RoboDK and TwinCAT systems, for a commissioning of the PLC code through the virtual prototype;
- The integration through Python coding to ease platform customization, analysis, and optimization;
- The extension of RoboDK functionalities by means of Python scripts to monitor specific kinematic outputs.

The remainder of the paper outlines the architecture of the system and the main steps in its realization. Then, an application is presented to showcase the obtained results and outline other potential outputs of the simulation tool.

2 Approach for virtual prototyping of robotic cells

The proposed method consists of an integrated virtual environment for robotic cell design, manufacturing planning, programming and, finally, simulation and optimization, as reported in **Fig. 1**. The envisaged design workflow moves from the definition of the geometric model of the automation devices by a CAD tool (system design). In the subsequent phase, the 3D models are imported in the simulation environment where the system designer can enrich the geometries with kinematic behaviors, mainly defining translational and rotational joints, as well as kinematic chains (virtual prototyping). The virtual prototype is then completed with the definition of elementary operations, such as picking and manufacturing operations, or enabling specific devices. The operations are defined in the virtual environment taking advantage of the specific features of the employed simulation software. Their activation is based on the values assumed by specific signals, i.e. variables whose values are modified at runtime by the control logic, normally incorporated in the PLC. Therefore, the virtual environment building strategy must reflect the physical structure and emulate in a realistic way the functionalities of the real devices, including the distributed control systems, that are the local control units incorporated within some devices. Moreover, the dialog with the PLC logic to be tested needs to be ensured by the same set of signals as in the physical automation system to allow the substitution of the virtual system with the physical one at the commissioning phase, without any change in the controlling PLC project.

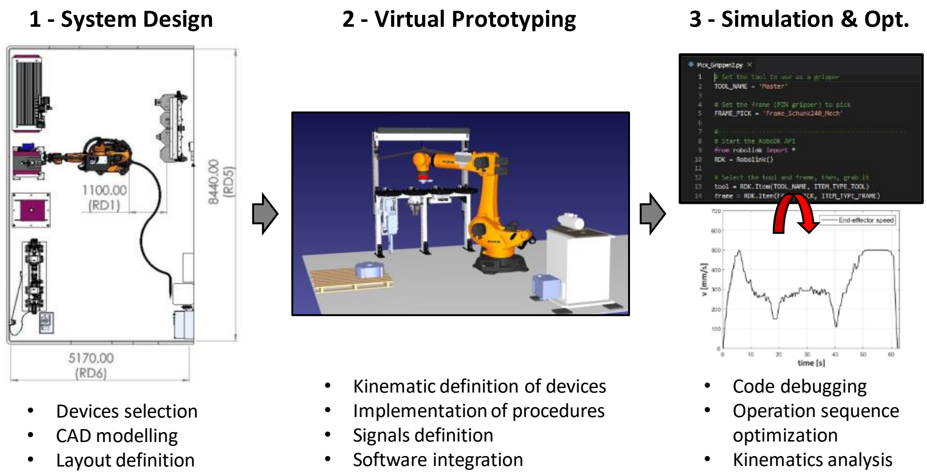

Fig. 1. Approach for the realization of the virtual prototype.

Many specific software tools available in the market can be chosen and connected as described in this paper to design the system (mostly mechanical CAD systems), realize the virtual prototype (for instance robot offline programming systems), and develop and test its controller (typically industrial PLCs). The hybrid environment being realized allows the PLC code to be tested, debugged, and improved (simulation and optimization).

From a system design point of view, many advantages emerge. The designer can benefit of a detailed analysis of the kinematic behavior of the automation system. Moreover, the virtual environment can provide trajectories, possible collisions, speed, acceleration of the moving parts according to diverse actions commanded by the control logic. The same analysis on physical setups would require complex and lengthy tests. Finally, several alternatives and sequences of tasks can be evaluated, also taking advantage of optimization packages, to identify optimized manufacturing cycles.

3 Use case: a robotic deburring cell

The proposed approach has been applied to a flexible manufacturing robotic cell, available in the University facilities. The aim of the cell is to provide on-demand production cycles according to I4.0 production paradigm. As shown in **Fig. 2**, the cell is equipped with a high payload KUKA industrial robot, a storage with several tools to grab components or perform manufacturing operations, and a spindle for material removal by chip forming.

The specific application case is the deburring of a cast component, i.e. a cast iron case which presents burrs to be eliminated. The burr removal is a process which is normally performed manually, given the complexity of the geometry of the parting lines in such components. Robotic deburring can be extremely advantageous, given the high dexterity of manipulators and the possibility of handling parts and deburring them in an optimized sequence of operations [3]. Moreover, such a solution based on a robotic cell allows variable parts to be processed as well as reconfiguration of the manufacturing cycle according to the specific shapes of the parts and the extension of the burrs.

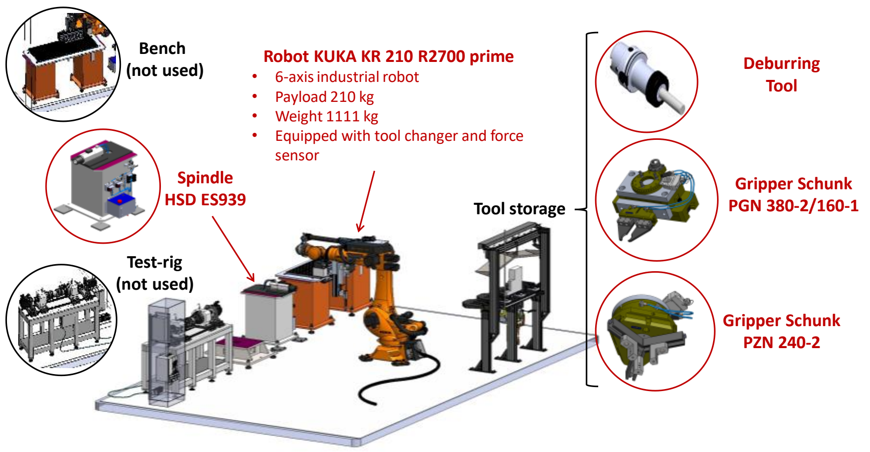

Fig. 2. Main components of the robotic cell: KUKA robot, spindle, grippers, and tools for flexible manufacturing cycles.

The virtual prototyping of the robotic cell and the commissioning of the control code to perform the required sequence of operations has been accomplished respectively in RoboDK and TwinCAT environments. The communication between the two systems has been realized via an external routine that allows continuous exchange of signals during simulation, as depicted **Fig. 3** and outlined in [10]. The communication flow is based on the identification of elementary procedures, which are combined to compose the manufacturing cycle. Procedures are labelled with a code and are requested by the PLC by an execution call. The PLC monitors the correct and complete execution of each procedure and dictates the whole sequence accomplishment.

The implemented communication interface is used to exchange the signals with the PLC, where an analog list is defined, and will be used in the actual implementation of the physical cell. The connection to the TwinCAT instance is provided through the Automation Device Specification (ADS) interface, which is implemented thanks to the *pyads* library, i.e. a Python wrapper for TwinCAT ADS library. A Sequential Function Chart (SFC) has been used to structure the logic code in the PLC, while single procedures are implemented and stored in RoboDK for each slave device, i.e. robot controller, distributed controllers for specific devices (e.g. spindle and tool storage).

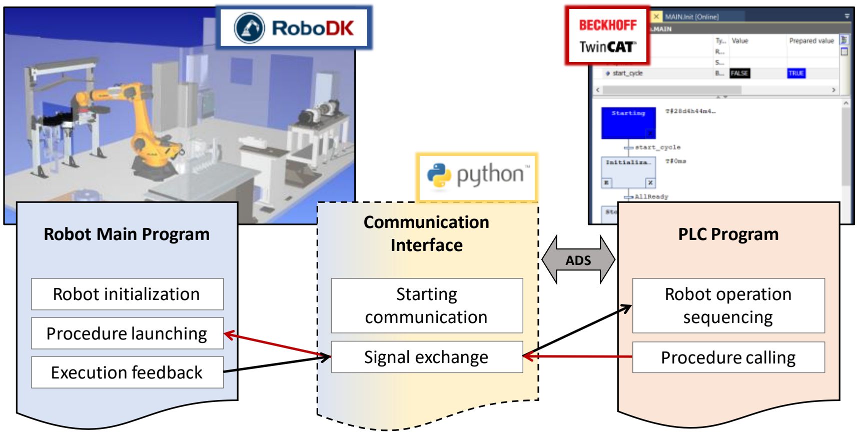

Fig. 3. Software architecture to enable the communication between RoboDK virtual environment and TwinCAT PLC.

As shown in **Fig. 4**, the communication interface has been developed in Python and embedded in RoboDK as a specific program. RoboDK allows a custom list of signals to be defined by the user. Such signals are used to command operations and to provide feedback related to the state of the tasks.

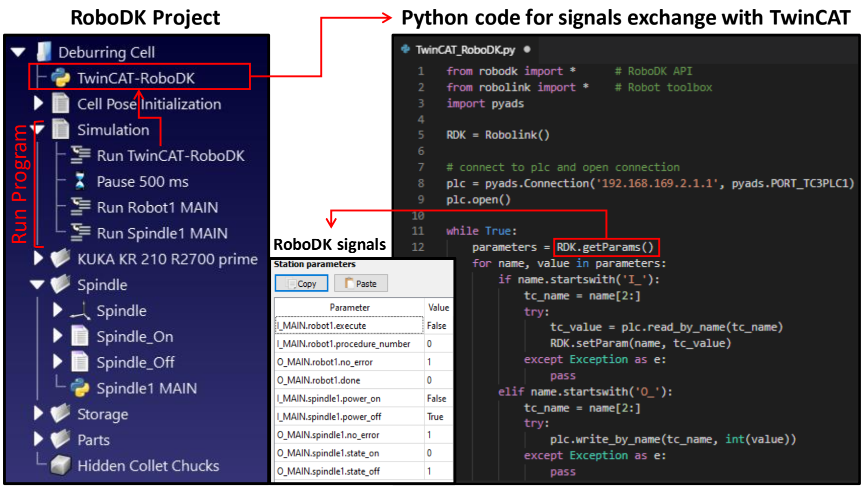

Fig. 4. Implementation of the signals exchange routine in RoboDK. Structure of the deburring application (left); List of the exchanged signals defined as station parameters (middle); Python application to connect and exchange signals between RoboDK and TwinCAT (right).

Considering the virtual prototyping environment, the procedures have been defined using a combination of native instructions of the RoboDK platform, combined with Python routines when more complex behaviors are to be reproduced. In the physical implementation of the cell, the procedures have been implemented within the specific controllers of the employed devices.

In the considered deburring application, the process simulation within the virtual environment has allowed several aspects of the cell design to be assessed. First, the reachability of the different locations has been verified, including the tools in the storage, the part to be processed and its positioning in front of the spindle. The geometrical model of the cast part has been used to derive the exact paths to be followed in the deburring process verifying the absence of kinematic singularities of the robotic arm. Moreover, the process simulation has allowed computing the accomplishing times of the single procedures (**Fig. 5**). Finally, the resulting speeds as well as maximum acceleration levels have been assessed. A specific Python script has been implemented to trace the positions assumed by the Tool Center Point (TCP), which are then plotted and verified during post-processing (**Fig. 6**). A numerical derivation algorithm has been implemented to assess speed and acceleration laws, and the results are compared against the maximum allowable limits of the robot.

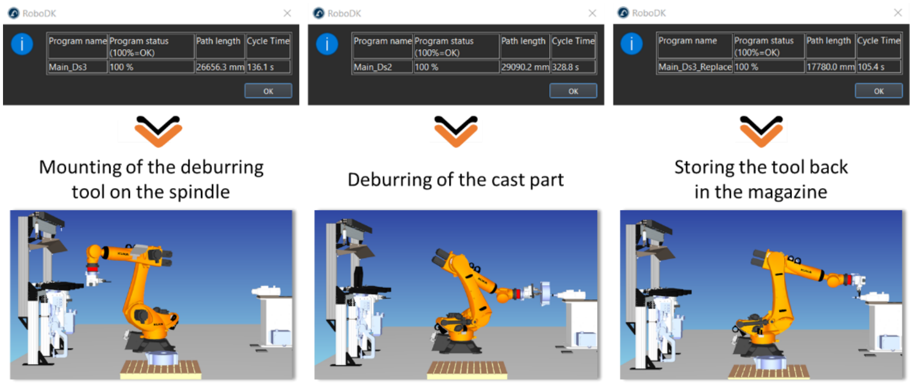

Fig. 5. Examples of robot programs run to complete the deburring cycle and Python program to monitor their effective cycle time.

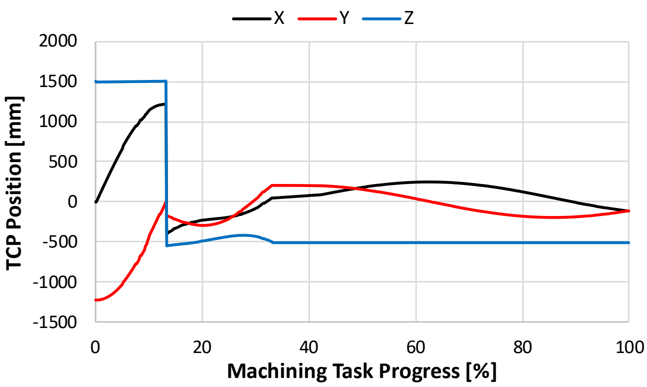

Fig. 6. Simulation of the TCP positioning in X, Y, Z directions of the robot base frame during deburring phase.

4 Conclusions

Virtual prototyping can be extended from product to manufacturing sphere, allowing effective means to pursue effective custom made and on-demand production. The integration of existing tools can provide comprehensive platforms to simulate flexible automation systems in the view of the digital factory, including the mechanics, logics, coding and dynamics domains. The analysis of the state of the art showed that currently available tools may offer partial or limited functionalities. Therefore, the presented work has aimed to complement simulating robotic cells environments with a PLC to coordinate the robotic cell and sequencing the tasks. A use case demonstrated how to create a solid communication flow between RoboDK and TwinCAT software systems and realize a hybrid simulation environment to design, validate and optimize design solutions. Future work will address experimental tests to validate spatial locations reached, execution times, speeds, etc. against the simulated ones.

Funding. This research was funded by the European Community's HORIZON 2020 program under grant agreement No. 958303 (PeneloPe).

References

1. Oztemel, E., Gursev, S.: Literature review of Industry 4.0 and related technologies. Journal of Intellignet Manufacturing 31, 127–182 (2020).
2. Noga, M., Juhás, M., Gulan, M.: Hybrid Virtual Commissioning of a Robotic Manipulator with Machine Vision Using a Single Controller. Sensors 22, 1621 (2022).
3. Schmidt, J., Grandi, F., Peruzzini, M., Raffaeli, R., Pellicciari, M.: Novel Robotic Cell Architecture for Zero Defect Intelligent Deburring. In: Procedia Manufacturing 51, pp. 140-147 (2020).
4. Jiang, J., Newman, S.T., Zhong, R.Y.: A review of multiple degrees of freedom for additive manufacturing machines. International Journal of Computer Integrated Manufacturing 34(2), 195–211 (2021).
5. Pérez, L., Rodríguez-Jiménez, S., Rodríguez, N., Usamentiaga, R., García, D.F.: Digital Twin and Virtual Reality Based Methodology for Multi-Robot Manufacturing Cell Commissioning. Applied Sciences 10, 3633 (2020).
6. Dumitrașcu, A., Nae, L., Predincea, N.: Virtual Commissioning as a Final Step in Digital Validation of the Robotic Manufacturing Systems. In: Proceedings in Manufacturing Systems 9(4), pp. 215-220 (2014).
7. Lee, C. G., Park, S. C.: Survey on the virtual commissioning of manufacturing systems. Journal of Computational Design and Engineering 1(3), 213-222 (2014).
8. Makris, S., Michalos, G., Chryssolouris, G.: Virtual Commissioning of an Assembly Cell with Cooperating Robots. Advances in Decision Sciences, ID 428060 (2012).
9. Aivaliotis, P., Arkouli, Z., Georgoulias, K., Makris, S.: Methodology for enabling dynamic digital twins and virtual model evolution in industrial robotics - a predictive maintenance application. International Journal of Computer Integrated Manufacturing, (2023).
10. Raffaeli, R., Bilancia, P., Neri, F., Peruzzini, M., Pellicciari, M.: Engineering Method and Tool for the Complete Virtual Commissioning of Robotic Cells. Applied Sciences 12, 3164 (2022).