Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 17592–17597

A proposal to introduce digitalization technologies within the automation learning process

N. Iriondo, D. Orive, O. Casquero, M. Marcos

Systems Engineering and Automatic Control Department, University of the Basque Country (UPV/EHU), Bilbao, Spain (e-mail: {nagore.iriondo, dario.orive, oskar.casquero, marga.marcos}@ehu.eus)

Abstract: Although the digital factory (DF) concept has raised high expectations since its inception, it is still missing industrial impact. One of the problems attributed to this issue is the lack of education curricula for enhancing the related digital competences of the future professionals. Higher education institutions, as major stakeholders in education, should introduce the new technologies for DF in practical courses. However, it is difficult to deal with the complexity of those technologies in a time-limited environment such us a bachelor or a master course. Instead of providing complete knowledge, this paper proposes to focus on the methodological aspects that allow students to acquire the skills needed to handle those technologies. Specifically, this paper illustrates this approach for teaching virtual commissioning (VC) within the automation learning process. The goal is to show the students how to use powerful industrial tools for performing VC through a set of methodological steps that help students manage the complexity of the VC process regardless of the specific tools used for it.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0)

Keywords: education, digital factory, product lifecycle management, digital twin, virtual commissioning.

1. INTRODUCTION

The increasing complexity of the manufacturing systems for Industry 4.0 entails engineering challenges at planning, scheduling, execution control and shop floor configurations. The digital factory (DF) (VDI 4499, 2011) is a mainstream technological concept that addresses those challenges by means of different digitalization technologies for simulating and optimizing the interactions among mechatronics, automation programs and operation sequences in a production process across the whole product lifecycle management (PLM). At the foundation of the DF is the digital twin (DT), which involves the virtual modelling of components, machines, cells and plants, as well as product and processes, throughout different phases of the PLM. Thus, each phase of the PLM can have each own DT with a different level of detail (Oppelt et al., 2014). There are several works in the literature that propose using DTs for different purposes. Tao et al. (2019) emphasized the potential of the DT along the product design, manufacturing and service phases. Macchi et al. (2018) and Negri et al. (2017) evaluated the role of the DT within the management of the physical asset. Kritzinger et al. (2018) performed a categorical review remarking the need to develop more industrial case studies to analyze the benefits of DTs.

Manufacturing systems are often composed of highly customized manufacturing cells with complex automation programs that control them. One of the challenges of the DF is to verify the assembly and automation programs of the manufacturing system before the real commissioning. To that end, it is crucial to optimize the design, development and verification phases of manufacturing systems by means of efficient prototyping environments. This has led to the notion of virtual commissioning (VC), a digitalization technology proposed by VDI 4499 (2011) that aims at the early validation of an automation project against a DT of the plant at the end of the automation engineering phase of the PLM. According to the terminology proposed by Oppelt et al. (2014), the controller running the automation program can be emulated (i.e., software-in-the-loop, SIL) or physically included (i.e., hardware-in-the-loop, HiL) within the VC. This virtual prototyping leads to a full verification and a better refinement of the automation program and, thus, reduces real commissioning time, since many possible issues have already been corrected during the simulation. In this sense, Ayani et al. (2018) concluded that simulation and optimization performed during VC provides a reduction of the commissioning time up to 30%. Orive et al. (2019) used VC to test the response of the control logic to process faults by extending the DT to provide fault injection support. Their methodology confirms the advantages of using DT and VC to validate the control logic of a manufacturing system.

Although DF has raised high expectations since its inception, it is still missing industrial impact (Mortensen et al., 2018), which can be mainly attributed to three reasons: a) the modeling effort required for developing DT (Lee et al., 2014; Oppelt et al., 2014; Schamp et al., 2018); b) the complex, proprietary and hardware vendor dependent PLM software platforms; and c) the lack of education curricula for enhancing the digital competences of the professionals of the future that will use those tools (Fradl et al., 2017; Kockmann, 2019; Oppelt & Urbas, 2014).

The PLM software market is developing exceptionally rapidly, providing an increasing number of tools that allow the construction of a DF over the different scenarios and phases of the PLM. This forces companies and employees to keep pace on the multiple, heterogeneous and complex digitalization

2405-8963 Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2020.12.2674

technologies that made up the DF. Thus, the DF concept requires a holistic view in education in which higher-education institutions (HEI) should be involved (Oppelt et al, 2014). Thus, it is important to introduce practical courses in engineering education whose objective is the use of those new technologies for digital transformation (Mortensen et al., 2018). However, the complexity of these digitalization technologies cannot be fully addressed in a time-limited environment such us a bachelor or a master course (Fradl et al., 2017). Instead of providing complete knowledge, HEI should focus on the methodological aspects that allow students to acquire the skills needed to learn how to use industrial tools for DF through real-world situations, but trying to manage their complexity through a methodologically guided teaching regardless of the specific tools used.

In this sense, many HEI are considering the DF and the PLM as a fundamental part of their engineering curricula (Fradl et al., 2017). Bedolla et al. (2017) used project-based learning and open source tools to teach a PLM strategy composed of the product design, the process design and the virtual factory simulation stages of the product lifecycle. Vrabič et al. (2018) presented an architecture focusing on different functionalities of DTs for PLM and defined a common space to manage the interaction between DTs throughout the product lifecycle. Mortensen et al. (2018) provided a VC learning platform to support training on control programing and physical device modelling. Other works show different approaches to the educational case study of a robotic cell by means of different tools: open source tools such as Simumatik3D (Azkarate et al., 2019) and Gazebo (Verner et al., 2019), and proprietary tools such as Tecnomatix (Villagómez et al., 2014), RobotStudio (Martins et al., 2019) and DELMIA (Vergnano et al., 2017).

This paper tries to contribute to the training of engineers by means of a methodology that provides the students an introduction to the new technologies in automation, specifically to those related to VC based on a DT of the plant. The goal is to show the students complex and powerful tools by means of a set of methodological steps that help them manage complexity. Students belong to the master in Robotics, Automation and Control. They have previously attended subjects in which they learned to solve automation problems in a methodological way (Alvarez et al., 2013), as well as to configure the hardware architecture and the industrial communications. In this laboratory subject, they learn to use complex tools to build DTs for VC in order to test and verify an already designed automation system. In particular, tools from Siemens PLM will be used to model the process. Students learn how to use complex commercial tools employed in real industry applications and how to integrate them with the PLC automation project. The goal is twofold: to introduce students in these advanced tools (thus, they can contribute to achieve industry digitalization) and to teach them how to create DTs for validating the whole automation project.

The rest of the paper is structured as follows: Section 2 describes the steps for building a DT for VC. Section III is dedicated to the steps for performing VC using HiL. Section 4 presents a case study for performing VC against the DT of a robotic cell. Finally, the paper ends with some conclusions.

2. HOW TO BUILD A DT FOR VC

The VC phase consists on validating the automation system. To do that, it is not necessary to have a precise model of the manufacturing plant, but a model whose outputs behave as the real system does in response to the commands of the automation program. Thus, the DT is built by means of the complete definition and location in the scenario of the different components (mechanical parts and robots), and the definition of the different operations the process performs, as well as the signals involved from the start to the end of the operation. In order to automate the process, it is also necessary to include the sensors that notify the controller of relevant events. The following steps describe the methodology the student must follow to build the DT of a robotic cell using an industrial modelling tool.

Step1. Definition of the scenario

First, it is necessary to specify the elements (tables, feeders, I/O zones, conveyors belts…) that are part of the process to be modeled. Usually, DT tools for manufacturing systems support importing CAD files of those elements. For every mechanical element of the cell it is necessary to: 1. Define the element, which, in general, is constituted by a set of components. 2. Import the CAD file. 3. Locate the element within the scenario. 4. If an element contains moving parts, it is also necessary to define different groups of components specifying the relative movement between the groups (i.e., to define the kinematics of the element). 5. If the latter step is needed, the operation of every moving part must be also defined (e.g., the linear movement of a belt or the rotation of a rotary table). These operations will be activated by signals (input signals to the DT).

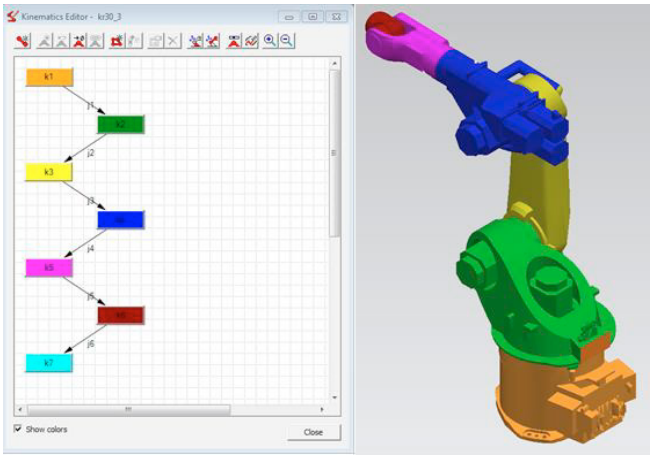


Fig. 1. Model of a Kuka robot.

Step2. Modelling of robots

Generally, robot manufacturers provide models that include the kinematics of their robots. Fig. 1 shows the mechanical model of a Kuka robot and its associated kinematic diagram imported into Tecnomatix Process Simulate (PS) modeling tool. It is necessary to: 1. Define the element (robot type). 2. Import the robot model and 3. Locate it in the scenario. The latter can be done by dragging the robot model to the desired point of the scenario or by defining the coordinates. 4. Tooling. Once located in the scenario, the robot tool must be defined,

i.e., the type of tool, the tool center point (TCP) and the base frame which is used to associate the tool to the robot. Finally, the tool must be mounted on the robot. 5. Once the robot is located and the tool is mounted, the robot operations can be defined. Every operation may involve a different tool and consists of defining the set of coordinates describing the tool movement. Then, the code for the robot controller is generated. If the virtual CPU is the same as in the real robot, the generated code can be executed directly on the real robot. The different operations will be activated by commands issued to the robot.

In order to use the DT for VC, two more steps are needed.

Step 3. Modelling of sensors

Typically, sensors will allow detecting parts arriving to some point and/or to detect concrete situations. DT Modelling tools generally allow inserting elements that simulate the behavior of sensors in order to automate the process. Hence, sensor modelling must include the signal to be read by the controller as well the sensor activation function. As the sensor is a new element in the DT model, for every sensor it is necessary to: 1. Define an element of sensor type. Usually, the modelling tool offers a set of sensor types from which the one that simulates the behavior of the real sensor must be selected. 2. Locate the sensor in the scenario. 3. Define the activation function (e.g., by collision with the part or by proximity).

Step 4. Identification of control commands/signals

The commands to the robot (see Step 2.5) and/or the signals to the DT (see Step 1.5) constitute the input commands and signals that operate the cell. Thus, they will be used during VC phase.

Step 5. DT validation

Once the DT model is completed, it is necessary to validate its behavior. Generally, this is performed through the generation of timed sequence of orders that allows checking the behavior as well as whether sensor signals activate accordingly. Similarly, events can be created to test different parts of the DT. At this point, the DT is ready to be connected to the control system for performing VC.

3. USING DT FOR VC

The automation project is loaded on the real PLC. When the CPU starts up, instead of finding the real IO devices, it finds the SIMIT UNIT card that emulates them.

Step1. Creation the DT of the process

This can be performed following the methodology explained in Section 2.

Step 2: Configuration of the distributed hardware in the automation project

The VC phase assumes that the automation project has been designed and debugged assuring its logical correctness (including the HMI). In this step, the distributed IO devices are included in the automation project, assigning the corresponding addresses in the IO address space. In order to perform the VC phase, it is important to consider the following aspects: 1. The input and output signals/commands associated to the operations that the process, represented by the DT, can perform. 2. The address space configured in the automation project for every IO. Both, 1 and 2, are necessary to map the signals/commands between the DT and the controller (see Step 4). 3. Extension, if necessary, of the HMI to facilitate the testing.

Step 3. Configuration of the communications in the automation project

DT generally does not have a direct signal exchange method with the elements in the automation project. In this approach, SIMIT for Siemens is used for HiL, but OPC UA can be used in general.

Step 4. Mapping the DT and controller IOs

This step is performed on the DT simulation tool. It is necessary to map the IO address configured in the automation project to the signals/commands of the DT.

Step 5: Executing the test plan

Once the automation project is downloaded to the controller and the DT is configured, it is possible to verify a test plan that must be previously defined.

4. CASE STUDY: A ROBOTIC CELL

This section applies the methodology to a case study consisting of a robotic soldering cell for manufacturing the bottom cover of a boiler. The scenario in Fig. 2 shows a conveyor belt through which the part arrives, while a second conveyor belt moves the operated part to the output. Both conveyor belts are made of rollers and driven by motors. Two Kuka robots with six degrees of freedom are used: one for manipulating the part and the other in charge of the soldering. A rotating table driven by an asynchronous motor moves the part to the soldering place and back, performing 180º turns.

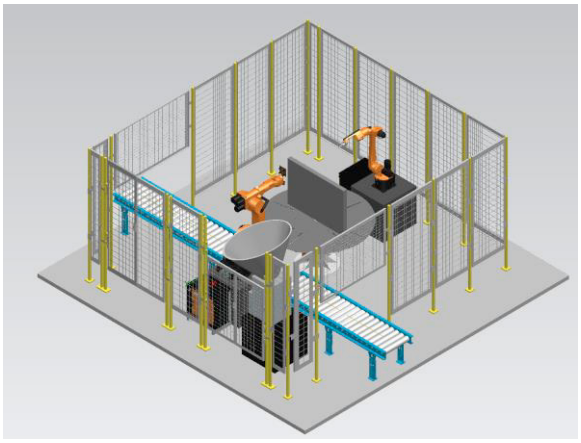


Fig. 2. The soldering cell scenario.

Step1. Creating the DT of the process

The DT has been developed using Tecnomatix Process Simulate (PS) tool from the Siemens PLM platform.

DT Step 1. Definition of the scenario

The elements to be modelled as part of the scenario of the cell are a rotary table, a tube feeder, two conveyor belts, two robots (one for handling and one for soldering), the safety fence and the doors.

Every element is modeled as described in Section 2. The rotary table requires: 1. Define the element as a table type. 2. Import the CAD file. 3. Locate the element within the scenario. 4. Define the groups of kinematics: The rotary table consists of a fixed base and a rotary part, each of which is made up of a set of elements. The rotary part makes 180º turns in the horizontal plane. Giving movement to the rotary part involves defining the kinematics or relative movement between the elements that form it. Fig. 4 shows the steps for the definition of the rotary table: the CAD model and the kinematics definition. 5. The last step is to bind the moving part to the operations associated to it; i.e., the signals activating the operations and those generated once operations are completed (see Fig. 4). Specifically, two are the variables related to each rotary movement.

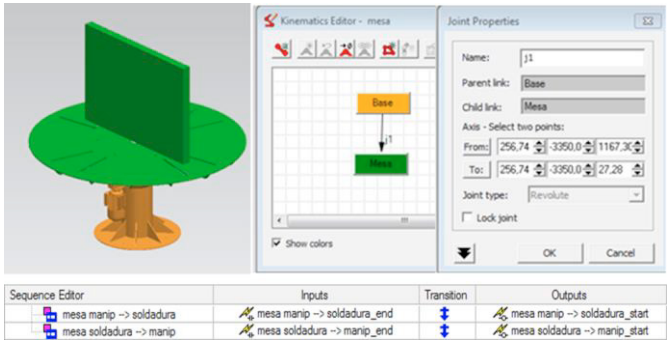


Fig. 4. Modeling of the rotary table.

DT Step 2. Modelling of robots

The manufacturing plant includes two Kuka robots: model KR30-3 for handling and model KR8 2100 for soldering. Both are controlled by KR C4 control units, which can be downloaded from the website of the manufacturer.

PLC Signal Name	Robot Signal Name	I/O	Signal Function	HW Type
kr30_3_startProgram	startProgram	Q	Starting Program	BOOL
kr30_3_programNumber	programNumber	Q	Program Number	BYTE
kr30_3_emergencyStop	emergencyStop	Q	Program Emergency Stop	BOOL
kr30_3_programPause	programPause	Q	Program Pause	BOOL
kr30_3_programEnded	programEnded	I	Ending Program	BOOL
kr30_3_mirrorProgramNumber	mirrorProgramNumber	I	Mirror Program Number	BYTE
kr30_3_robotReady	robotReady	I	Robot Ready	BOOL

Fig. 5. Signals of the handling robot.

The steps needed to model the robots are: 1. Define the element as a robot type. 2. Import the CAD file, which includes the kinematics. 3. Locate the element within the scenario. Fig. 1 shows the CAD model and the kinematics of the Kuka KR30-3 robot. 4. Define the tools for both robots. The handling robot needs a double clamping system for holding a vacuum cup to take the base of the boiler, and a pneumatic gripper to take the tubes. These tools are of *gripper* type. The soldering robot needs an element of *gun* type that simulates the soldering tool. Finally, the tools are mounted on the robot and they define the new TCP of the robot. A set of signals must also be assigned to both robots to control their movements (see Fig. 5). 5. The robot operations are structured by tasks or operations. The handling robot has four operations: place base, place tube1, place tube2 and remove soldered set. The soldering robot has two operations: weld tube 1 and weld tube 2. Each operation has its own control code. In both cases, a main program selects one program accordingly by means of a code received from the controller (see Fig. 6).

```
PROG MAIN Soldador

   START_OPERATION MAIN Soldador
     # Send  robotReady 1
     # While (startProgram) Do
     # If  (ProgramNumber == 2) Then
     # CallPath Soldadura_izquierda
     # Elsif (ProgramNumber == 3) Then
     # CallPath Soldadura_derecha
     # Endif
     # Endwhile
   END_OPERATION

END_PROG
```

```
PROG MAIN_Manipulador

   START_OPERATION MAIN_Manipulador
     # Send  robotReady 1
     # While (startProgram) Do
     # If (ProgramNumber == 2) Then
     # CallPath kr30_3_Base_mesa
     # Elsif (ProgramNumber == 3) Then
     # CallPath kr30_3_cil_izquierda
     # Elsif (ProgramNumber == 4) Then
     # CallPath kr30_3_cil_derecha
     # Elsif (ProgramNumber == 5) Then
     # CallPath kr30_3_PNP_dejar_conjunto
     #Endif
     #Endwhile
   END_OPERATION

END_PROG
```

Fig. 6. Control code of the robots.

Once all the elements and the relationships between them have been adequately defined, it is possible to simulate the DT and to check the kinematics and possible collisions between cell elements.

In order to use the DT for VC, two more steps are needed.

DT Step 3. Modelling of the sensors

The nine sensors used in this cell are illustrated in Fig. 7. Sensor 1 detects an arrival piece. Sensors 2 and 3 detect the existence of tubes at the feeder output. Sensors 4, 5 and 6 are related to the location on the part and the tubes on the table. Sensor 7 detects the presence of the base and tubes in front of the soldering robot. Sensors 8 and 9 are related to the evacuation of the soldered part.

Following the modeling steps 1. Sensors are defined as elements of sensor type. 2. Sensors are located in the scenario and 3. Sensors are configured to be activated by collision. Fig. 7 shows the location of sensors as well as the interface windows that allow the creation of the sensor 1, named *light_sensor_New_Base*, as photoelectric type that activates by collision with the arrival of the part. Along with this, the detection range to activate the signal must be also defined. The bottom right side of such figure also illustrates the signals associated to sensors, one for each sensor, created at sensor definition. These signals will be part of the interface of the model with the controller.

DT Step 4. Identification of control commands/signals

There will be signals or commands related to the different components of the model that will form the interface of the DT with the controller. Robots receive from the controller a command containing the number of the program related to the operation they have to execute. Robots generate the corresponding signals when they initiate an operation and when operation is finished.

To perform VC with distributed I/O in HiL mode, the real controller must be connected to a DT of the process. The DT of the soldering cell is developed following the steps of the methodology of Section 2, and with the characteristics indicated in Section 4.1. In this phase, all the I/O signals of each model component are also defined.

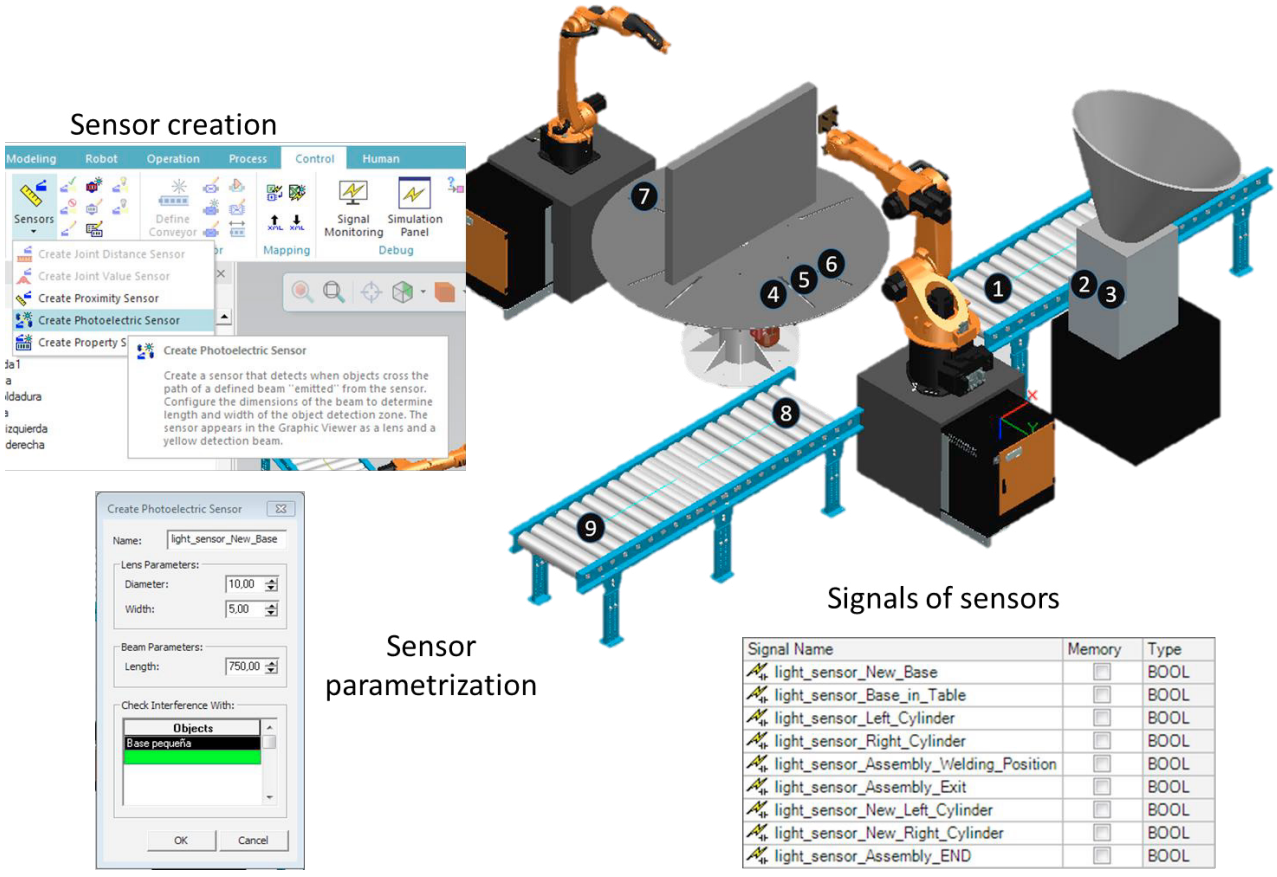


Fig. 7. Location of the sensors.

Step 2: Configuration of the distributed hardware in the automation project

The automation project will run in a Siemens SIMATIC S7 1516 PLC. In addition to the CPU I/O modules, the control system also has an HMI. Students should be trained previously in the use the TIA Portal engineering tool for the configuration of hardware, software and industrial communications, as well as to develop and validate the automation project.

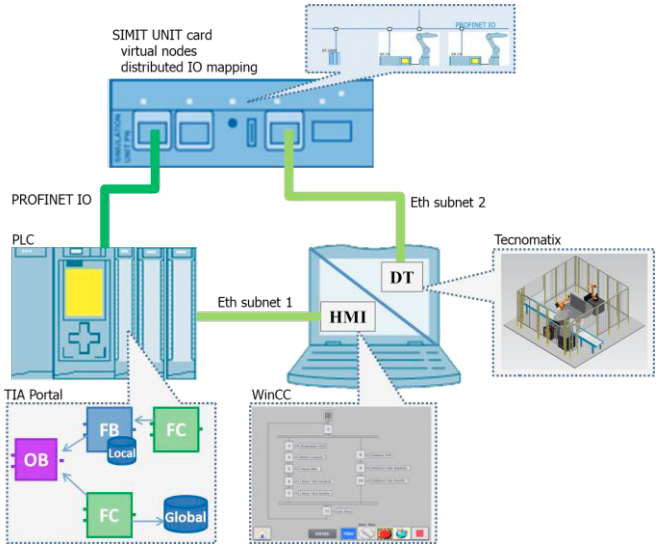


Fig. 8. Demonstrator for VC with HiL and distributed I/O.

In this subject, students will learn what additional hardware is necessary to connect the real controller to a virtual model of the process. The whole configuration is depicted in Fig. 8. Besides the PLC, the hardware architecture is composed of a PC to run the DT and the HMI of the control system. Particularly, Tecnomatix PS V14.1 is used to develop and simulate the DT. There is also an intermediate card that simulates the functionality of the distributed IO of the virtual nodes and IO devices related to the DT. In this case, the SIMIT UNIT PN card acts as the interface between the PLC and the DT, allowing the simulation of the distributed I/O of the robot control units and the I/O devices related to other elements of the DT, such as the belt motors and the rotary table. For do that, it is necessary to load on the card information exported from the automation project relative to the devices it simulates.

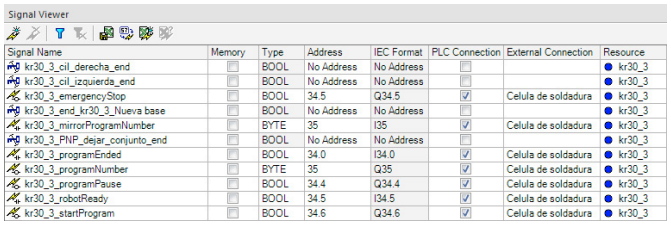
Step 3. Configuration of the communications in the automation project

With regard to the communications, the PC needs two Ethernet network cards, one to connect the HMI to the PLC, and the other to connect the signals of the DT to the PLC. Additionally, it is necessary to configure a Profinet IO network to communicate the PLC with the SIMIT UNIT card.

Step 4. Mapping the DT and controller IOs

Simulation Unit V9.1 tool is used for loading the hardware of the IO devices and the robot control units into the SIMIT UNIT PN card as well as mapping the I/O of the control program with the signals of the DT in Tecnomatix. Fig.9 shows part of the

I/O mapping of the elements defined in the DT, concretely those related to the handling robot previously shown in Fig. 5.



Signal Viewer

Signal Name	Memory	Type	Address	IEC Format	PLC Connection	External Connection	Resource
kr30_3_cil_derecha_end		BOOL	No Address	No Address			kr30_3
kr30_3_cil_izquierda_end		BOOL	No Address	No Address			kr30_3
kr30_3_emergencyStop		BOOL	34.5	Q34.5	☑	Celula de soldadura	kr30_3
kr30_3_end_kr30_3_Nueva base		BOOL	No Address	No Address			kr30_3
kr30_3_mirrorProgramNumber		BYTE	35	I35	☑	Celula de soldadura	kr30_3
kr30_3_PNP_dejar_conjunto_end		BOOL	No Address	No Address			kr30_3
kr30_3_programEnded		BOOL	34.0	I34.0	☑	Celula de soldadura	kr30_3
kr30_3_programNumber		BYTE	35	Q35	☑	Celula de soldadura	kr30_3
kr30_3_programPause		BOOL	34.4	Q34.4	☑	Celula de soldadura	kr30_3
kr30_3_robotReady		BOOL	34.5	I34.5	☑	Celula de soldadura	kr30_3
kr30_3_startProgram		BOOL	34.6	Q34.6	☑	Celula de soldadura	kr30_3

Fig. 9. Distributed I/O mapping

Step 5. Executing the test plan.

Finally, once the demonstrator is configured, connected and ready to run, it is possible to verify the test plan that must be previously defined.

5. CONCLUSIONS

This paper presented a well-defined methodology for teaching VC within the automation learning process. This methodology will be implemented next year in the practical part of a new subject for the introduction of digital transformation technologies. Instead of providing complete knowledge, the goal is to show the students how to use powerful industrial tools for performing VC through a set of methodological steps that help students to acquire the skills to perform the VC phase while managing the complexity of industrial VC tools.

ACKNOWLEDGMENT

This work was financed by Erasmus+, UE (grant number 2018-1-FR01-KA203-048175) and by GV/EJ (grant numbers IT1324-19 and KK-2019-00095).

REFERENCES

Alvarez, M. L., Estévez, E., Sarachaga, I., Burgos, A., & Marcos, M. (2013). A novel approach for supporting the development cycle of automation systems. *The International Journal of Advanced Manufacturing Technology*, *68*(1), 711–725.

Ayani, M., Ganebäck, M., & Ng, A. H. C. (2018). Digital Twin: Applying emulation for machine reconditioning. *Procedia CIRP*, *72*, 243–248.

Azkarate, I., Ayani, M., & Eciolaza, L. (2019). Virtual commissioning of a robotic cell: an educational case study. *ETFA 2019 - 24th IEEE International Conference on Emerging Technologies and Factory Automation*, 820–825.

Bedolla, J. S., D'Antonio, G., & Chiabert, P. (2017). A Novel Approach for Teaching IT Tools within Learning Factories. *Procedia Manufacturing*, *9*, 175–181.

Fradl, B., Sohrweide, A., & Nyffenegger, F. (2017). PLM in Education – The Escape from Boredom. In J. Ríos, A. Bernard, A. Bouras, & S. Foufou (Eds.), *Product Lifecycle Management and the Industry of the Future* (pp. 297–307).

Kockmann, N. (2019). Digital methods and tools for chemical equipment and plants. *Reaction Chemistry & Engineering*, *4*(9), 1522–1529.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, *51*(11), 1016–1022.

Lee, C. G., & Park, S. C. (2014). Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, *1*(3), 213–222.

Macchi, M., Roda, I., Negri, E., & Fumagalli, L. (2018). Exploring the role of Digital Twin for Asset Lifecycle Management. *IFAC-PapersOnLine*, *51*(11), 790–795.

Martins, A., Costelha, H., & Neves, C. (2019). Shop Floor Virtualization and Industry 4.0. *ICARSC - 2019 IEEE International Conference on Autonomous Robot Systems and Competitions*, 1–6.

Mortensen, S. T., & Madsen, O. (2018). A Virtual Commissioning Learning Platform. *Procedia Manufacturing*, *23*, 93–98.

Negri, E., Fumagalli, L., & Macchi, M. (2017). A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*, *11*, 939–948.

Oppelt, M., & Urbas, L. (2014). Integrated virtual commissioning an essential activity in the automation engineering process: From virtual commissioning to simulation supported engineering. *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2564–2570.

Orive, D., Iriondo, N., Burgos, A., Sarachaga, I., Alvarez, M.L. & Marcos, M. (2019). Fault injection in Digital Twins as a means to test the response to process faults at virtual commissioning. *ETFA 2019 - 24th IEEE Conference on Emerging Technologies and Factory Automation*, 1230-1234.

Schamp, M., Hoedt, S., Claeys, A., Aghezzaf, E. H., & Cottyn, J. (2018). Impact of a virtual twin on commissioning time and quality. *IFAC-PapersOnLine*, *51*(11), 1047–1052.

Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering*, *5*(4), 653–661.

Vergnano, A., Berselli, G., & Pellicciari, M. (2017). Interactive simulation-based-training tools for manufacturing systems operators: an industrial case study. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *11*(4), 785–797.

Verner, I., Cuperman, D., Gamer, S., & Polishuk, A. (2019). Training Robot Manipulation Skills through Practice with Digital Twin of Baxter. *International Journal of Online and Biomedical Engineering*, *15*(9), 58–70.

Villagómez, L., Vásquez, V., & Echeverry, J. (2014). Virtual Commissioning with Process Simulation (Tecnomatix). *Computer-Aided Design and Applications*, *11*(sup1), S11–S19.

Vrabič, R., Erkoyuncu, J. A., Butala, P., & Roy, R. (2018). Digital twins: Understanding the added value of integrated models for through-life engineering services. *Procedia Manufacturing*, *16*, 139–146.