IFERP
ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 12, Issue 01, January 2025

Virtual Commissioning in Robotic Welding Cell

[1] Someshwar Rodge, [2] Vinay Kumar
[1] [2] John Deere India Pvt Ltd
Corresponding Author Email: [1] RodgeSomeshwarS@Johndeere.com, [2] KumarVinayB@JohnDeere.com

***Abstract*— Robotics virtual commissioning uses an automation simulation environment, rather than the physical robotics system, to test the system's behaviour based on the actual control software. Robotics virtual commissioning is performed ahead of production-floor deployment and commissioning Intelligent robotic manufacturing cells must adapt to constantly changing running conditions, autonomously developing the best manufacturing strategies to achieve superior quality and productivity. This is carried out using distributed controllers, where complex control logic must interact and process various input/output signals. Specifically, coordination and integration between programmable logic controllers (PLCs) and robot controllers are essential. To ease this, it is necessary to simulate the behaviour of robotic cells for performance verification and optimization, evaluating the impact of both PLC and robot control codes. This work proposes a method and its implementation into an integrated tool that uses FANUC Roboguide software as a virtual prototyping platform for robotic cells. In this setup, real robot control codes are executed on a virtual controller and integrated with the PLC environment.**

A PLC Smart Component was developed as an extension of Roboguide functionalities to enable signal exchange with a PLC. This new module allows for the virtual commissioning of an entire robotic cell, assessing the effects of control logic on overall productivity. The solution is shown in a robotic Welding cell, highlighting its feasibility and effectiveness in improving final performance.

***Index Terms*— VC (virtual commissioning), Weld robotic cell, Roboguide, PLC, ADS (Automated device Specification), SC (smart components), SFC.**

I. INTRODUCTION

virtual commissioning is a technique designed to confirm the control software of a manufacturing system using a simulation model within a virtual environment during the first stages of the commissioning process. One significant advantage of virtual commissioning is that it enables the parallelization of many design activities, allowing multiple engineers to collaborate simultaneously, thereby reducing overall design time. Additionally, this approach helps the identification and correction of potential errors, enhancing the performance of the entire robotic cell before it is installed.

an essential factor for effective virtual commissioning is Virtual commissioning has the capability to integrate technologies from various engineering disciplines, creating a comprehensive environment where all aspects of manufacturing systems are simultaneously considered. Despite recent advancements, there is still a significant lack of integrated simulation-based platforms in standard industrial practices. digital plant models often separate the geometric and physical aspects of the system from the PLC control program and signals, which are assessed using control software development tools without a direct view of the process behavior.

one notable improvement offered by this software, compared to Process simulate, is the ability to simulate PLCs from any vendor. However, a common drawback is the limited robot simulation capabilities; robot movements can only be approximated as there is no actual virtual controller working within the application, only a generic emulator. Literature reviews show that a major limitation of VC solutions for robotic cells is the lack of realism in simulating automated plants. Consequently, the most effective strategy for developing exceptionally dependable models appears to be the integration of specialized commercial platforms.

Multi-software frameworks have been extensively explored by academic researchers in recent years, not just in relation to virtual commissioning but also for the design of servo-actuated mechanisms, dynamic characterization of robotic systems, and tuning of robot controllers. Within this context, the primary goal of this work is to develop and test a virtual commissioning tool that uses the integration of Roboguide and PLC software. PLC software transforms a PC into a real-time controller, the proposed approach can be applied to test a real PLC system, i.e., realizing the so-called hybrid commissioning as a part of the hardware-in-the-loop approach & simulate its behaviour on a standard PC, i.e., realizing a full virtual commissioning with a software-in-the-loop approach.

II. ASSOCIATED TASK

A. Virtual Commissioning Methodology

the proposed software architecture is described starting from a brief introduction of the two main software systems (Roboguide and PLC. Then, details about the software part and its logic, which were developed to enable the integration between Roboguide and PLC. A demonstrative Fig and Picture of the proposed VC tool can be viewed at supplementary.

B. Integrated Software Tools

Roboguide has been selected as the software tool to model

ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 12, Issue 01, January 2025

the virtual cell, as it provides functionalities to stand for the geometric layout, kinematics of devices, physical behaviour, and control logic. Originally developed as offline programming software for Fanuc robots, Roboguide utilizes virtual controller technology to perform highly realistic simulations of Fanuc robot movements and execute complex Karel and Micro robot programs. The software running on real robot controllers (Robot Neighbourhood) is identical to that used by the virtual robot controllers.

Roboguide also allows users to model or import 3D geometries of other devices, Roboguide can be combined to form more complex configurations. additionally, Fanuc provides a development tool which enables users to create new applications using Microsoft Visual Studio. this includes add-ins and customized SCs, thereby enhancing the software's capabilities.

On the other hand, PLC is used for configuring and programming Beckhoff devices, AB devices including servo drives and PLCs. PLC programs can be developed using various programming languages, such as ladder diagram, instruction list, function block diagram, structured text, and sequential function chart (SFC). These programs control the overall task execution flow of the robotic cell, helping signal exchange with the controls of robots, motors, linear axes, and other devices. The PLC works on a hardware system designed for real-time performance, typically a dedicated PC that ensures prompt execution of cycles.

C. General Architecture

The core idea of the proposed approach is to replicate the architecture of the physical automation solution in a virtual environment, specifically utilizing Roboguide, to establish a one-to-one mapping between real physical components and their digital counterparts. As shown in the *Figure 1*, a typical robotic cell in industrial settings is controlled by a single PLC that communicates with various automation modules, including robot controllers, motors, and other devices. The signals exchanged can include various data types, such as Boolean values, integers, and real numbers. This communication is facilitated through a field bus, which is physically implemented via cables connecting the controllers and devices in a series configuration.

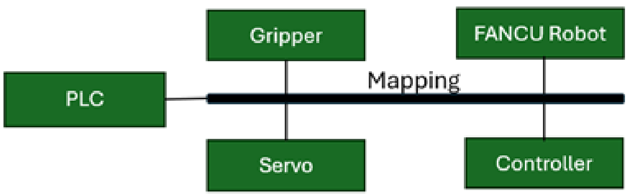

Figure1. Typical Hardware

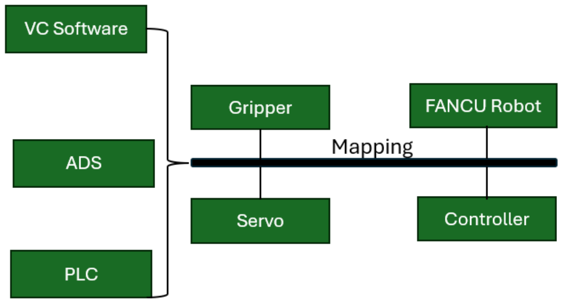

Figure 2. virtual environment of the control architecture physical and virtual assets

Figure2 represents the proposed virtual commissioning (VC) solution. In this configuration, the PLC system serves as the primary control tool for the cell this can also be fully replaced by a PC running an emulated PLC for a complete virtual strategy. Each physical device is represented in the prototyping software as a virtual module, designed to replicate the interface of its real counterpart, including the exchanged signals and expected behaviours. These virtual models are developed as smart components (SCs) that leverage Roboguide' s capabilities, enabling users to create reusable blocks that encompass geometric definitions, kinematics, sensors, and control logic.

The level of detail in representing each device achieves a balance between required realism and overall simulation performance. Key factors that influence PLC performance, such as dynamic or synchronized activities, must be modelled accurately. However, physically based computations that realistically depict system performance can be resource-intensive, so they should be restricted to areas where they are essential for realism and consistency with the desired behaviour.

Similarly, the flow of signals on the field bus is seamlessly integrated with a flow of virtual signals in the virtual environment. Communication between the PLC and the virtual environment is facilitated by a developed connection interface known as PLC, which simulates the functionality of the real PLC within Roboguide. The connection to the actual PLC is established through the Automation Device Specification (ADS) protocol, used by the PLC to connect devices within a control chain. With the availability of ADS Advanced Programming Interfaces (API) for the MS .NET Framework platform, C# was utilized to develop the PLC component, enabling it to read and write the values of variables defined in the PLC program. From the PLC's perspective, it functions as a genuine part of the control chain, exchanging actual signals on the field bus. In contrast, this component is integrated into the virtual environment of Roboguide, allowing it to read and write virtual signals that control the synthetic environment.

Using smart components (SCs) in Roboguide is straightforward, as it enables users to add the signals, they

IFERP
Explore Your Research Journey...
ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 12, Issue 01, January 2025

wish to exchange with the PLC simply by naming them as they appear in the PLC code. For instance, *Figure 3* illustrates how the two signals, Roboguide_Input1 and Roboguide_Output1, defined in the robot controller Controller_R1, are linked to the PLC variables PLC_Output1 and PLC_Input1, respectively. These variables are identified within the running TC code, and their values are automatically synchronized by the SC.

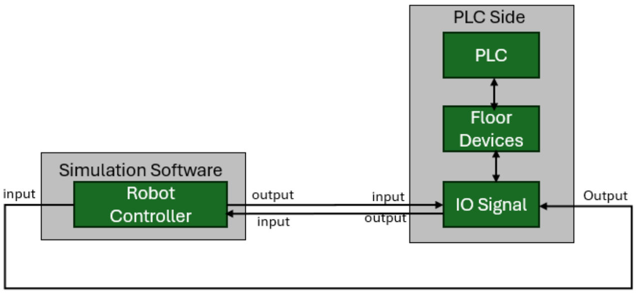

Figure 3. PLC, Robot, Roboguide standard block

D. PLC- Smart Component Interface

The PLC Smart Component Interface is designed to facilitate seamless communication and integration between the PLC system and various robotic or automation components within a manufacturing environment. Here are the key features and functionalities of this interface.

There are two categories interface 1) I/O section 2) Properties.

In the I/O section, four default signals are presented as pulsed types, displayed to the user as buttons for managing signals: Create Signal, Remove Signal, Start, and Stop. These signal names are reserved and thus excluded from the ADS communication exchange mechanism. They are utilized to operate and configure the Smart Component (SC) for creating a new signal with a designated name, removing an existing signal, initiating an ADS connection, and terminating the connection, respectively.

The Properties section features two critical fields: Ads Address and Port, which are necessary for the ADS protocol to establish a connection with the PLC and facilitate communication. After entering the required information in these fields, clicking the Start button attempts to establish the connection. If the connection fails, an error message will appear in the Roboguide output messages window.

The property "Disconnect When Simulation Stops" is a Boolean value that determines whether the ADS connection should be terminated when the Roboguide simulation is stopped. The last two properties, Signal Name and Signal Type, are utilized for creating or deleting signals. After inputting the necessary information, clicking the Create Signal button will instantiate a signal of the specified type and add it to the Smart Component's (SC) custom signals list. To remove a signal from this list, the user can enter the signal's name in the Signal Name property and then click the Delete Signal button. The setup phase is completed by ensuring that each PLC variable intended for exchange with Roboguide is added to the SC.

E. Interfacing and Signal Exchange Mechanism

As previously stated, the user adds the required I/O signals based on the devices utilized within the cell and the necessary signals by providing names that correspond to the desired variables in the PLC system. For instance, a variable might be named “MAIN.Robot1. busy,” indicating that within the PLC code, a variable named “busy” is sought in the instance “Robot1” of a function block that defines a specific robot type. The instance “Robot1” is declared in the program organization unit (POU) named “MAIN.” This approach effectively maps the signal names in Robot guide directly to PLC variables, the following outlines the initialization of communication through the mapping of Roboguide signals to PLC variables. This involves using an Ethernet/IP Scanner with PLC, which is connected to an Ethernet/IP Adapter with the R784 option on the Fanuc Roboguide side.

The method for IO data mapping on the Fanuc Roboguide side is also described, where enable and reset signals are transmitted from the PLC side.

- R784 options are needed to use the ethernet/IP Adapter
- Host Communication configuration

Figure 4. Fanuc Protocol setup

- Setup Protocol & Set Robot IP address
- Port IP address-Set Port Address, port IP address as the network Adapter on the PC

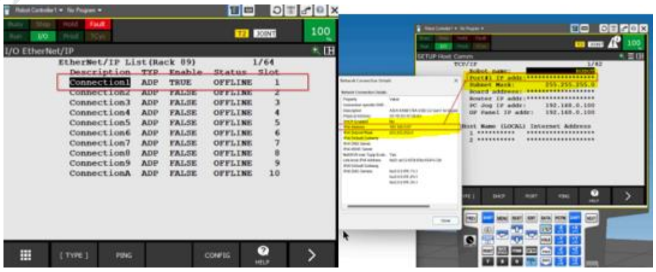

Figure 5.0. Fanuc Ethernet

- PC Jog / OP Panel IP address set as per panel & restart the robot Controller
- After restart ping the Function to test the connection
- Set the Ethernet /IP Setting – to change the adapter function of the Fanuc Robot must first disable connection & select the enable field click false
- Validate the enable state of connection is now False

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

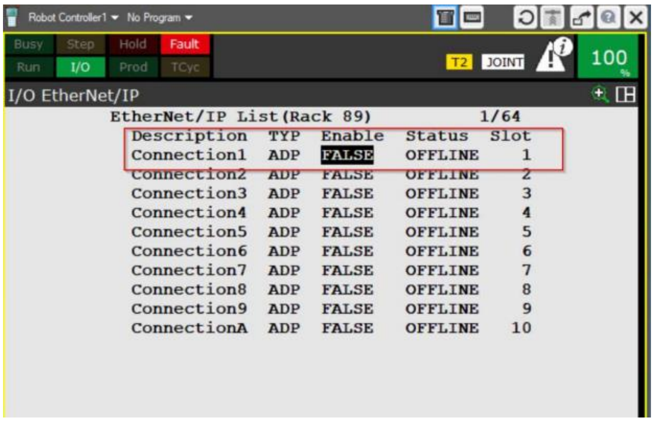

Figure 5.1. Fanuc Ethernet/IP

- If want to change the setting for connection select config & do not forget to set adapter back to enable at the end
- System variable side modification needed
- Select variable and go to $EIP_ENBL_IO variable makes it 0 to 1
- Select system / config and go to remote / Local setup OP Panel option by hitting choice make it Remote
- Enable UI signal – check & Conform True
- Configure UOP Mapping settings and check the I/O UOP Monitor screen appears
- No 89 Ethernet/IP Rack & STAT changes to PEND when the RACK of UO is changed if the RACK slot & start values are set correctly the device name will appear on the Ethernet/IP

Figure 6. Input out UOP signal

- Similarly click on in / Out and configure UI Restart the Robot Controller
- Export EDS file to build the Ethernet/IP Network
- Click Backup option, set the Com.Conf backup option export it.
- Now click on file > explore folder to open the location roboguide project.
- PLC side – To prove communication and mapping of UIO signal.
- Store EDS File Exported from Robo guide folder.
- Add Ethernet/IP Scanner – add a new fieldbus driver under I/O > device > Add added item.

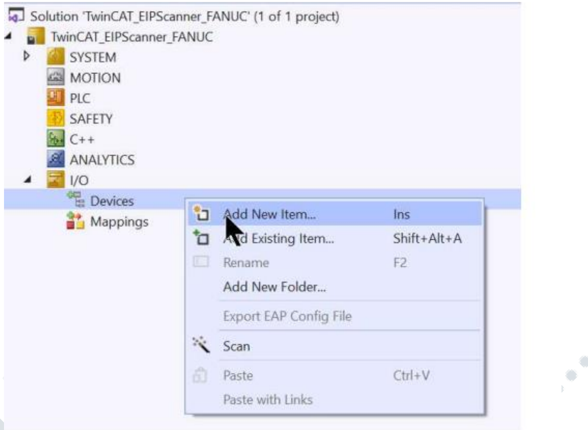

Figure 7.0. PLC EIP Scanner Setting

- Open the adapter tab & click the search button to set the scanner network interface to be used as the Ethernet/IP Scanner
- PLC will list the network interface currently available on IPC.
- Select Sync Task & create new task I/O Task enter task name.
- Set the IP Address of the Ethernet/IP Scanner
- After add Ethernet/IP Scanner complete then need to add Fanuc Ethernet/IP adapter by right click on scanner, click & select fanuc robot controller.

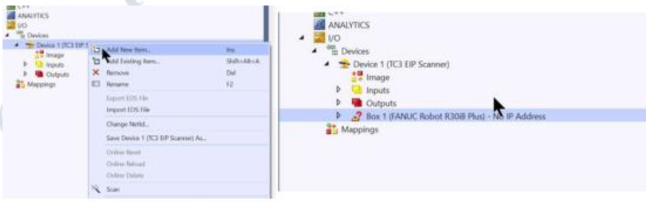

Figure 7.1. PLC EIP Scanner Setting

- Set the IP address of the Fanuc Robot from setting & configure ethernet connection between PLC and Fanuc
- Right click one Fanuc Robot Adapter select and Add Fanuc IO Slot right slot & create Variable as defined in EDS-File
- Input and output match with Fanuc Roboguide

ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 12, Issue 01, January 2025

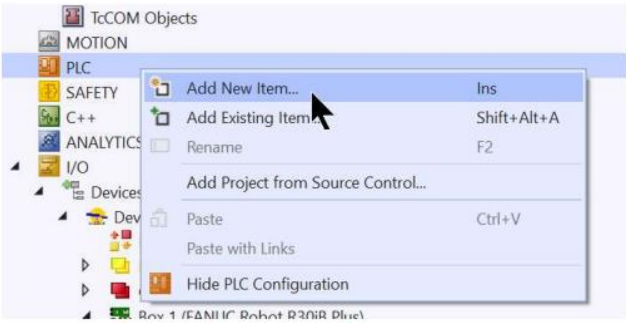

Figure 7.2. PLC configuration setting

- Add PLC -GO To PLC click and Add PLC & select Standard PLC project.
- Create standard and structure Function Block program in PLC.

When the Start button is clicked, the ADS connection is established, and the handle variables are automatically created as part of the initialization procedure described in the logic above. Once this setup is complete, the Roboguide simulation can be initiated. If the simulation is active, an event handler known as "on Simulation Step" is triggered for each elementary simulation step in Roboguide. The code executed at each step iteration includes two methods:

- ROBOGUIDE READ TO PLC WRITE is the method that updates the values of variables in PLC based on signal changes in Roboguide.
- PLC READ ROBOGUIDE WRITE is the method that updates the status of signals in Roboguide when a variable value is changed in PLC

III. CASE STUDY- ROBOTIC WELDING CELL

The approach described the virtual model of the cell was realized in Fanuc Roboguide starting from a CAD representation of the real implemented prototype. The developed case allowed the VC capabilities of the proposed approach to be tested in a complex scenario made of two robots and various other systems. In particular, the interaction mechanism between the PLC and the virtual representation of the cell devices is reported here in detail. Finally, an extract of the PLC code is shown and discussed, as well as the capability of the VC approach to improve the overall performance of the system.

A. Description of the Robotic Welding Cell

The robotic welding cell includes two Fanuc industrial robots working in the same workspace. The aim of the cell is to weld small trinket's part and other robot load for welding.

See in *table 01* list of hardware main device employed which to be used in creating virtual environment for VC. The two robots in the cell are equipped with different tools: the 100IC and the M10IA 2 (Robot 1). Robot 1 picks the part from the cart and loads it onto the weldment, while the other robot performs the welding operation. Robot 1 is designed to lock the appropriate gripper based on the object that needs to be picked and subsequently placed in the weldment.

Fanuc 100IC (Robot2) mounts a welding torch its function is to weld part which is pick and place on weldment by Robot 1

Table I: Details main device employed on the cell

Name	Description	Qty
Fanuc 100IC/8L	Serail manipulator	01
Fanuc 10IA/12	Serail manipulator	01
Quick Change system	Tool changer	01
Schunk gripper	Parallel gripper	01
Welding Torch	Welding equipment	01

The virtual model of the cell was developed from a 3D representation created in PTC Creo, incorporating both the models designed for the components and those provided by the manufacturers of the devices. Subsequently, the geometry was imported into Roboguide. This process was followed by a standard workflow that involves defining movable devices, referred to as mechanisms, which in this specific case include the gripper and the welding torch.

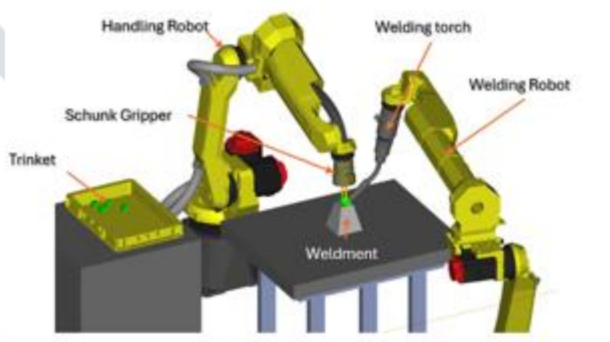

Figure 8.0. Full view of welding cell reproduced in Robot Guide 3D Environment

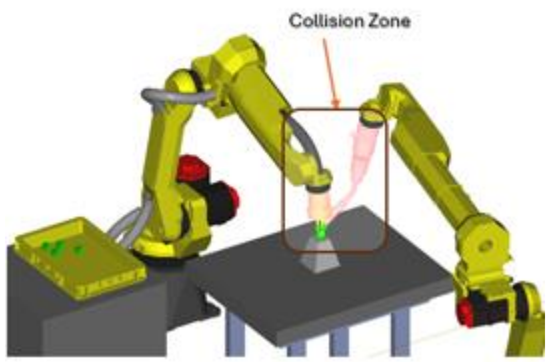

Figure 8.1. Full view of welding cell reproduced in Robot Guide 3D Environment VC Analysis

B. Communication pattern between PLC and Cell devices

The role of the PLC is to coordinate the activities of the robots and the devices within the cell. In the proposed approach using the Roboguide virtual model, a dedicated Smart Component (SC) is created for each physical device. The SC represents the behaviour of the physical asset by utilizing the functionalities offered by Roboguide. However, the SC's interface regarding exchanged signals must match that of the real devices to facilitate an accurate virtual commissioning (VC) of the cell.

A communication pattern was established between the PLC and other devices using a standardized set of signals, which are summarized in *Table 2*. The listed signals represent a minimum core set necessary to ensure efficient interaction between the PLC and the cell devices.

Table II: list of the IO Signal exchange between Robot/device and PLC

Signal Name	From Output	To Input	Description
Procedure name Program name Target position	PLC	Robot or Device	Contain coded data to identify action that the device has carried out
Execute		Robot or Device	Trigger the execution whose code has been transmitted to the device
Done	Robot or Device	PLC	Raised when device complete the procedure
Error	Robot or Device	PLC	Error code execute when fault generate

The actions carried out by the robots and devices in the cell are divided into elementary procedures, which are then combined to execute more complex tasks. The PLC determines the necessary sequence of operations & the potential simultaneity of actions based on the overall assembly task of the cell and the constraints of space sharing. Consequently, the PLC sends identifiers for the procedures to be executed to the cell devices, referred to as "procedure Number" for manipulators and "program Number" for the press.

For the pick-and-drop gripper, it transmits numeric values for opening and closing the gripper and shares coordinate values with the welding robot to indicate where the part should be placed on the weldment, i.e., the target position. Finally, the communication pattern is enhanced with additional error signals to ensure the required safety and robustness in industrial applications.

Table 02. illustrates the PLC connected to the cell devices, indicating that the configuration stage of the cell has been completed in accordance with the previously explained signal pattern. The signal exchange operates on a simple remote procedure call (RPC) framework, which involves client-server interaction through request-response messages. This pattern effectively manages complex scenarios in a standardized and safe manner.

In practice, as depicted in *Figure 09*, an operation is activated by the execute signal, and it gives feedback when activity ends by the done signal. Once the PLC has received the done signal, it resets the execute signal. Finally, the device resets the done signal as soon as it receives the information that the execute signal has been reset

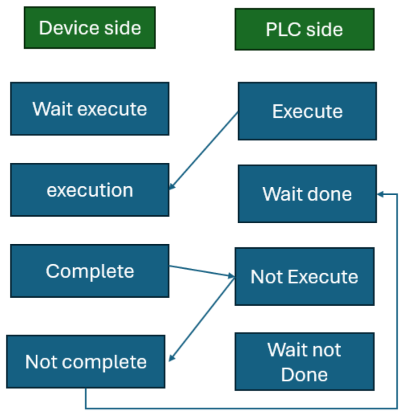

Figure 9. Remote procedure call

The same logic is applied to devices and robots, using respective controllers' coding means to handle the signals' exchanges. For instance, in the specific case of a Fanuc robot, the RPC scheme is managed by a main loop using the Karel code, i.e., the code running in the robot controller. As shown in *Figure 10*, the robot acts like any other device: it receives an execute command from the PLC and it returns a done signal when the requested procedure is finished.

Other code sections are then implemented to detail the action required by the specific operation procedure, such as a joint movement, gripper attaching, object grabbing. Such procedures were defined according to the subdivision of the robot tasks in elementary operations carried out in the first design phase of the assembly process.

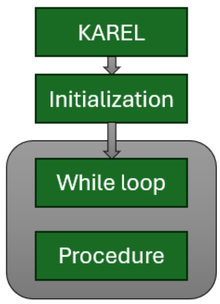

Figure 10. Schematic of the structure of the Karel program added in the virtual controller of both robots included in the cell

C. PLC Programming

The Ladder language is easy to understand, as some basic interpretation rules are provided and can be combined with other PLC programming languages. In *Figure 11*, a simple diagram is shown as an example to recall the basics of SFC. The Init block with the double outline is the entry point of the program. The little rectangle below, linked by a vertical line, stands for a transition, i.e., a step forward in the graph execution flow; the related condition is pointed out by the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

label beside it. Until the transition condition is not verified, the program cyclically executes the code included in the specific block. When the condition expressed in the label is met, the execution of the program will pass to the next block, and so on.

Figure 11. Ladder programming language for PLC

in the robotic welding cell test case, the Ladder programming language has been employed to program the PLC. Figure 11 reports a part of the program to show the logic used to build it. Each bit of the diagram corresponds to an operation of a certain device. The features of the Ladder. For instance, virtual Field device sensor data transfer. Robot sync logic, cycle time, Auto Cycle, safety signal validate in the VC environment by creating standard block or ladder logic in PLC & other side supposed to be moved simultaneously to reach the robot target position of the table or fixture

IV. VALIDATION AND RESULTS

The implemented integration between Roboguide and PLC was evaluated against the possibility to perform effective VC of the cell. Series of welding and pick & place tasks were defined and simulated to show the best welding and pick and place sequence for the proposed trinket assembly task. For a better understanding of the executed work, a video showing the VC of the welding cell is provided as added material to this paper. In the video, the PLC, and the virtual cell in Roboguide are displayed side-by-side. An example of the welding, pick and place sequence was shown that has been recorded to illustrate the command activation from the PLC and the variation in the signals' status, both in PLC and Roboguide. The whole pick and place – welding task was subdivided in a list of elementary operations to be arranged in proper sequences.

The examples of considered operations include:

- Pick a trinket from cart by using end effector gripper
- Drop: detach a trinket from gripper and find in the weldment where to be weld
- Position: transfer trinket position to welding robot where trinket is to be drop
- Weld: tack trinket by welding robot

following the architecture described in the earlier section, the code in the controllers of the robots and other cell devices oversees activating the operations according to the signals received by the PLC. The actual sequence of operations is decided by the PLC program. Therefore, several PLC programs, including different sequences of operations were generated and tested. From the experimental activity, it appears that, in the virtual environment, various aspects can be verified and perfected. At first, the proper sequence of actions and the correctness of signals exchange between PLC and controllers must be verified and improved. Therefore, PLC program robustness is verified and tested in normal operating conditions as well as in fault events caused by errors, safety alarms or unexpected situations. the reachability of the desired locations, the quality and fluency of the movements, the absence of collisions, as well as the overall time needed to conduct the tasks are other significant tasks that can be verified before commissioning the real cell to ensure a successful result. Finally, the VC was analyzed to refine the sequence of the operations coordinated by the PLC. The absence of collisions between two robots sharing their workspace (see Figure 12), the interchangeability of the single assembly operation, and the possibility to parallelize actions performed in different zones of the cell make the solution space of all the possible operating sequences quite vast. This opens the possibility of improving the order of the operations requested of the two robots and other devices in order minimize the overall accomplishing time.

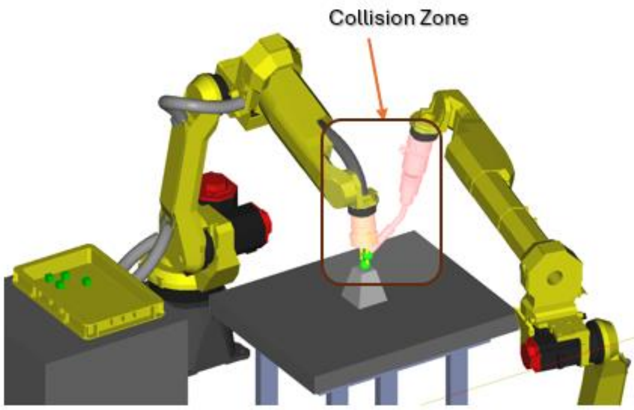

Figure 12. The sharing of the workspace of the two robots makes it essential to study the proper sequences of actions in a virtual environment to avoid collisions. In the figure, Robot1 locating a trinket, collides with Robot 2 while it is welding.

V. CONCLUSION

Virtual Commissioning technology serves as an effective tool for organizations to maintain competitiveness by enabling the commissioning activities of a production system to be anticipated during the design phase. This approach allows for identifying errors and enhancing the project when investments are still relatively low.

Additionally, since both Roboguide and PLC are proprietary solutions tailored for vendor-specific devices, the proposed method can be replicated using other commercial platforms or even expanded to general-purpose platforms,

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

where products from multiple vendors can be simulated.

The primary advantage is that both PLC and robot programs can be concurrently verified and refined before implementation in the physical system. Any modifications made at the code level can be tested across the entire system within a single environment, without the need to simplify or idealize the behaviour of components.

Looking ahead, future work could focus on enhancing the application's performance, particularly by developing more efficient communication code to reduce latency.

REFERENCES

[1] Lee, C. G. & Park, C. S., 2014. Survey on the virtual commissioning of manufacturing system, Toronto: Toronto University

[2] Oztemel, E.; Gursev, S. Literature Review of Industry 4.0 and Related Technologies. J. Intell. Manuf. 2020 Literature review of Industry 4.0 and related technologies | Journal of Intelligent Manufacturing (springer.com)

[3] Fanuc India Pvt Ltd FANUC – Leaders in Factory Automation (fanucindia.com)

[4] Noga, M.; Juhás, M.; Gulan, M. Hybrid Virtual Commissioning of a Robotic Manipulator with Machine Vision Using a Single Controller. Sensors 2022, 22, 1621. https://www.mdpi.com/1424-8220/22/4/1621

[5] Min, B.-K.; Huang, Z.; Pasek, Z.J.; Yip-Hoi, D.; Husted, F.; Marker, S. Integration of real-time control simulation to a virtual manufacturing environment. https://www.worldscientific.com/doi/abs/10.1142/S0219686702000076

[6] PLC Pluggable solution for automation without control cabinet Beckhoff | New Automation Technology | Beckhoff Worldwide