sensors

MDPI

Article

Internet of Things-Cloud Control of a Robotic Cell Based on Inverse Kinematics, Hardware-in-the-Loop, Digital Twin, and Industry 4.0/5.0

Dan Ionescu [1,2], Adrian Filipescu [1,2,*], Georgian Simion [1,2] and Adriana Filipescu [1,*]

[1] Department of Automation, "Dunarea de Jos" University of Galati, 800008 Galati, Romania; dan.ionescu@ugal.ro (D.I.); georgian.simion@ugal.ro (G.S.)
[2] Doctoral School of Fundamental Sciences and Engineering, "Dunarea de Jos" University of Galati, 800008 Galati, Romania
* Correspondence: adrian.filipescu@ugal.ro (A.F.); adriana.filipescu@ugal.ro (A.F.)

Abstract: The main task of the research involves creating a Digital Twin (DT) application serving as a framework for Virtual Commissioning (VC) with Supervisory Control and Data Acquisition (SCADA) and Cloud storage solutions. An Internet of Things (IoT) integrated automation system with Virtual Private Network (VPN) remote control for assembly and disassembly robotic cell (A/DRC) equipped with a six-Degree of Freedom (6-DOF) ABB 120 industrial robotic manipulator (IRM) is presented in this paper. A three-dimensional (3D) virtual model is developed using Siemens NX Mechatronics Concept Designer (MCD), while the Programmable Logic Controller (PLC) is programmed in the Siemens Totally Integrated Automation (TIA) Portal. A Hardware-in-the-Loop (HIL) simulation strategy is primarily used. This concept is implemented and executed as part of a VC approach, where the designed PLC programs are integrated and tested against the physical controller. Closed loop control and RM inverse kinematics model are validated and tested in PLC, following HIL strategy by integrating Industry 4.0/5.0 concepts. A SCADA application is also deployed, serving as a DT operator panel for process monitoring and simulation. Cloud data collection, analysis, supervising, and synchronizing DT tasks are also integrated and explored. Additionally, it provides communication interfaces via PROFINET IO to SCADA and Human Machine Interface (HMI), and through Open Platform Communication—Unified Architecture (OPC-UA) for Siemens NX-MCD with DT virtual model. Virtual A/DRC simulations are performed using the Synchronized Timed Petri Nets (STPN) model for control strategy validation based on task planning integration and synchronization with other IoT devices. The objective is to obtain a clear and understandable representation layout of the A/DRC and to validate the DT model by comparing process dynamics and robot motion kinematics between physical and virtual replicas. Thus, following the results of the current research work, integrating digital technologies in manufacturing, like VC, IoT, and Cloud, is useful for validating and optimizing manufacturing processes, error detection, and reducing the risks before the actual physical system is built or deployed.

Keywords: VC; DT; A/RC; Industry 4.0/5.0; Digital Twin; HIL; PLC; SCADA; HMI; OPC-UA; STPN

check for updates

Academic Editors: Virginia Pilloni and Francesca Marcello

Received: 8 February 2025
Revised: 1 March 2025
Accepted: 12 March 2025
Published: 14 March 2025

Citation: Ionescu, D.; Filipescu, A.; Simion, G.; Filipescu, A. Internet of Things-Cloud Control of a Robotic Cell Based on Inverse Kinematics, Hardware-in-the-Loop, Digital Twin, and Industry 4.0/5.0. *Sensors* **2025**, *25*, 1821. https://doi.org/10.3390/s25061821

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the manufacturing and automation industry, adopting new Industry 4.0/5.0 concepts like advanced manufacturing technology and intelligent digital technology integra-

tion has become essential for achieving success in a fast-evolving and competitive market. Industry 4.0/5.0 plays a key role in engineering digital transformation by introducing smart manufacturing technologies like IoT and digitalization into manufacturing and industrial processes.

Digital Twin technology involves a virtual transformation of the field, by shifting and replacing traditional, physical factories to a more digital-focused approach model. It achieves this by incorporating and integrating advanced software tools and technologies to enhance business factory operations and processes. As part of the DT technology, the VC technique is used, employing a virtual environment process, for simulating and testing the system off-site, typically in the context of manufacturing or industrial automation.

Digital transformation and smart factories are fundamental concepts in modern-day industrial manufacturing and automation, defined by the integration of digital technologies such as the IoT, DT, VC, Cloud robotics, and storage in big data analytics [1]. DT is a dynamic, clear, and understandable representation, a virtual replica of the physical system that offers a detailed, real-time perspective on asset information, mirroring system behavior, properties, and interactions. In the manufacturing industry, a DT is a virtual model and process visualization platform that integrates and accurately replicates the characteristics, performance, and operations of the actual physical system [2,3].

The integration of twins into engineering processes is rapidly accelerating, primarily driven by Industry 4.0/5.0. This advancement brings "smart" manufacturing system capabilities into the digital realm, enhancing connectivity and facilitating improvements across performance indicators, production, system support, prediction, efficiency, and quality and reducing operational costs and risks [4–6].

As part of the DT technology, the VC tool and techniques are used to increase production and performance on a long-term basis by optimizing the operations tasks, to improve the traceability by monitoring, analyzing, and tracking every aspect of all manufacturing processes like costs, risks, and reducing delays associated with the real on-site commissioning. It is well known that a variety of potential problems can arise when the manufacturing automation system is first deployed, during integration with IO Field and actual controller (PLC): software bugs or obsolete/unnecessary functions, machine sequence and operations faults, sensors, or communication issues. Therefore, DT is used to perform commissioning for equipment and systems virtually first, to minimize the downtime risks and errors during real, on-site commissioning. The main goal is to identify potential problems at an early stage that can be solved at the virtual environment level before the real process is put into operation.

In this paper, we focus mainly on the design, modeling, and software implementation of the DT robotic cell platform application, as seen in Figure 1, by developing a virtual replica for an A/DRC assisted by ABB 120 IRM for assembly and disassembly of a predefined five-component product [7], evaluating through extensive testing its accuracy, reliability, effectiveness, and applicability in real-world applications. The primary motivation for developing and testing a DT robotic cell platform application is to ensure that the virtual design and behavior accurately reflect the dynamics of the physical system. This is achieved by validating the 3D model implementation and the analytical methods of the IRM inverse kinematics model. The objective is to implement a complex virtual environment development using cutting-edge digital technology for remote monitoring, analyzing, designing, and optimizing engineering processes and operations without requiring direct intervention on the physical equipment. Consequently, the presented manufacturing technology facilitates the integration of the VC concept by leveraging the bi-directional data exchange flow between the dynamic digital representation of the physical system. By allowing the integration of IoT into Cloud infrastructure, a stable, flexible tech backbone is

built, this digital platform provides a safe and robust tool for real-time remote monitoring with optimized performance, simulating scenarios, and predicting maintenance needs or potential outcomes, ultimately leading to smarter decision making and improved overall efficiency [8].

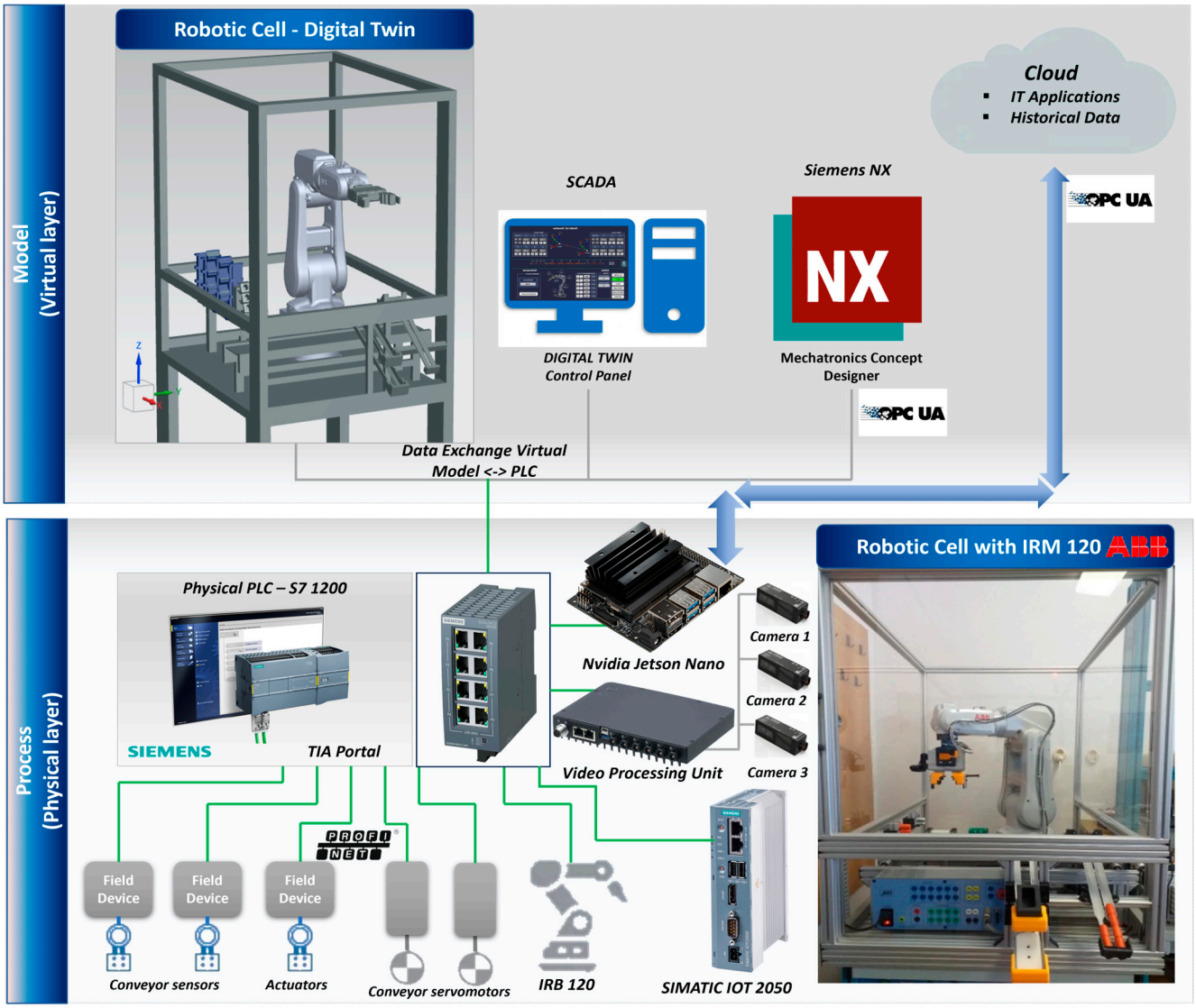


Figure 1. A/DRC with ABB 120 IRM. DT Physical and Virtual Layer Communication topology.

The PLC software design introduces a new approach, beyond the logic for IRM and A/DRC automation control, referred to as the physical layer. It includes a set of PLC function blocks for kinematic computation, translating models from MATLAB R2021a to PLC code for use in the digital simulation environment of the complete engineering process. Additionally, it provides communication interfaces via PROFINET IO to SCADA and HMI, and through OPC UA for Siemens NX MCD with a DT virtual model. A closed-form inverse kinematics solution for the ABB 120 IRM is implemented in the PLC, using the geometric approach, which generates and selects the right solution according to robot configuration input in a reasonably faster time due to the analytic nature of the computation method.

The primary contribution aspects and originality of this research refer especially to the software development of an HIL simulator to emulate a robotic cell which allows a real-time interface with its physical counterpart through advanced analytics. Our primary focus was on software implementation and optimization [9], IoT-Cloud Control and data

acquisition, as well as 3D virtual modeling methods of A/DRC physical process dynamics, followed by simulation using MATLAB and ABB RobotStudio. Additionally, we integrated the bi-directional data exchange flow between the digital model and the physical process. In this research, the DT approach for mirroring a flexible robotic cell is an extremely powerful tool for predictive maintenance and optimization to improve performance. It can be used additionally as a design and planning tool for logging, reporting, and control. We discuss several system elements that we designed in the laboratory:

- Siemens 1200 PLC program for controlling ABB 120 IRM and mobile visual servoing to be able to perform movements, positioning, and storage of disassembled parts;
- Development of a Siemens NX 3D virtual model as a dynamic replica of the DT's physical counterpart;
- Virtual model control, based on inverse kinematics computation, has been designed to facilitate a digital simulation environment for the entire physical process, from modeling in MATLAB to PLC code implementation;
- Additionally, path planning for linear motion, using the MoveL command from ABB(ABB Ltd., Zurich, Switzerland), from an initial to a desired final position is implemented in the PLC, utilizing a smooth Cubic Polynomials function;
- Implementation and optimization of Siemens 1200 PLC software to execute emulated robotic cell assembly and disassembly functions for HIL-structured DT design;
- Development of an IoT cloud-based DT for real-time sensor data logging and reporting in MongoDB 8.0;
- OPC UA interface connectivity, virtual and real mapping, real-time interaction with the 'real-world' between DT and A/DRC related parameters;
- SCADA interactive graphic interface programming in Siemens TIA Portal;
- Interactive graphic interface programming of HMI KTP700.

Simulation results will be compared with real-world experimental data to evaluate the DT's predictability, reliability, and overall performance. Moreover, this study explores the complex communication and control of A/DRC, assessing the integration and compatibility of IoT and Cloud-based infrastructures. It also addresses DT scalability and adaptability, ensuring its applicability to larger industrial systems and multiple applications.

The rest of the paper is organized as follows: hardware structure, basic design concept, and architecture of the A/DRC assisted by ABB 120 IRM, IoT integrated within the DT application, and the Cloud environment, is presented in Section 2. The theoretical framework of 6-DOF RM kinematics is addressed in Section 3, highlighting the analytical methods for both the calculation and implementation of inverse kinematics. Robotic cell task operations and modeling using Petri Nets are also discussed. This section covers 3D virtual model implementation as well, using Siemens NX MCD software. Section 4 focuses on the design, implementation, and testing of a DT application including PLC program design. In this section, VC-"smart" manufacturing systems as new concepts of Industry 4.0/5.0 are introduced and the integration with DT for manufacturing systems is discussed. SCADA serves as a DT operator control interface for collecting, monitoring, and visualizing live data or Cloud history within the framework of the presented DT technology is described in Section 5. In this section, the integration of remote solutions via Cloud or VPN is also presented, exploring Node-RED application for building powerful, smart IoT integrated automation system. A/DRC testing and simulations for virtual model operations, as well as real-time control results for DT and Cloud solution implementation, are essential key points and will be discussed in Section 6. Section 7 is focused on discussing the outcomes, analyzing data, and reporting findings. The last section, Section 8, outlines the conclusions of the proposed approach.

2. A/DRC System Characterization

2.1. Hardware Structure of the A/DRC

Regarding flexible manufacturing technology within an A/DRC environment, a system has been designed to allow the complete product assembly and disassembly process on the station assisted by ABB 120 IRM [10]. The entire process was designed to meet the requirements for adapting the initial manufacturing system to a flexible one, allowing A/D tasks for two technical different workpieces, in the same cell.

The robotic cell, shown in Figure 1, comprises a Siemens S7-1200 PLC-controlled A/D unit, equipped with SCADA supervision with IO field data collection and monitoring capabilities. This station manages various tasks at the operational level, from storing and supplying subcomponents to delivering the fully assembled product onto the cell's conveyor unit, which functions as a transport line. Although the presented robotic cell is a laboratory setup, it employs the ABB 120 IRM to recreate, mirror, and simulate a real industrial process by executing multiple A/D tasks [11]; the last, is to place the final built product (assembly) or to recover disassembled components and place them into the respective storage units on the side of the cell station (disassembly). Each final product comprises five components for assembly: base pallet, body, top, and two inserted cylinders, which can be either plastic or metal [12]. The assembly process is carried out through IRM repetitive operations of picking up and handling/transporting components to the designated assembly area within the cell. Thus, the system is equipped with storage areas for the components needed for assembly; ABB 120 IRM picks up the necessary components and completes each assembly operation in full. In the case of the disassembly task, the reverse assembly process occurs, PLC is handling operations while the disassembled components are picked up and placed by RM to their corresponding storage areas from the robotic cell. The complete system setup is built on a centralized PLC architecture, synchronized with the DT environment, to operate the ABB 120 IRM equipped with an electric gripper, enabling coordinated management, motion planning, and control of all subsystems. A/DRC control strategy employs IRM feedback of joint angles to the S7-1200 PLC, receives signals from the sensors located on the conveyor, processes camera images, and transmits control signals to the actuating elements, as well as interfaces the controller of the conveyor motor.

As mentioned, the basic design concept of the hardware architecture integrates three major subsystems, which operate, communicate, and synchronize together: PLC, SCADA, and DT 3D-virtualization of the complete process using Siemens NX MCD [13], using PROFINET and OPC UA interfaces, acting as a complete flexible manufacturing cell with DT functionality that performs synchronized tasks for operations and functionality of the engineering process.

PROFINET interface enables HMI and SCADA to communicate with PLC, providing greater interoperability [14]. Therefore, several assembly tasks can be assigned to the A/DRC station, followed by the disassembly operation of the subcomponents using ABB 120 IRM. Process monitoring sensors are connected to the digital inputs of the PLC, components such as solenoid valves and actuators are connected to the PLC's digital outputs while Jetson Nano using Node-RED [15], helps integration of cell's sensors to IoT application within DT and Cloud platform using OPC UA. Complex execution components, such as servomotor drivers, are also connected and controlled by the PLC via the PROFINET protocol. Thus, this system can be classified as an IoT system from Industry 4.0/5.0 perspectives. This type of communication enables quick and easy information data exchange between the robotic cell system devices, sensors, and the transport system, enabling communication technologies and applications over the Internet.

An HMI linked to the system's network facilitates the interconnection between the manufacturing system and the external decision-making environments. This allows the operator to quickly and efficiently input the necessary production data setpoints—such as the number of products and the configuration of the intermediate layer of parts for each product. The HMI also enables the operator to control and monitor the production process, providing access to information regarding the status of robotic cell workstations and production progress. Moreover, the HMI connection to the system's PROFINET network enables fast and precise communication with all other cell workstations and integrated devices. This ensures swift communication, as well as good synchronization between workstations, thus, avoiding potential issues and errors that may arise during production.

To identify the type and position of components, A/DRC uses a visual identification system in the control loop, with integrated video cameras into the IRM workspace, a system called visual system (VS) [12,16,17]. Data captured by the visual system is processed within the robot controller, to pick up the necessary components and to position them in the desired locations. Following the processed information from the VS 3D HD cameras, and from the robotic cell IO Field Devices, the control, positioning, and tracking tasks are synchronized and executed by the ABB 120 IRM using PLC commands [12].

2.2. ABB 120 IRM

ABB 120 IRM is used in this research. Designed for industrial environments, robots feature specific types of capabilities, like advanced motion control technology with an open-source Robot automation Framework including Rapid programming language, which simplifies complex motion paths and operations providing communication and easy integration in a variety of industrial automation system setups [18]. The six-axis configuration gives it high flexibility, with each axis offering a specific range of motion and a great Smooth Path Control for precise path following. This configuration creates a comprehensive 3D workspace. Full rotational reach and extended flexibility in end-effector orientation provides usability in compact environments, particularly when mounted on various surfaces. The robot is controlled with the IRC5 ABB compact controller and control software, RobotWare 6.15.07 build 7032, which supports every aspect of the robot system, such as motion control, system integration, and communication with third-party devices, enabling simulation and offline programming development, which minimizes setup time and optimizes performance [19]. ABB 120 IRM performances and product detailed specifications per axes are shown in Table 1.

Table 1. Angle limits and velocities for ABB 120 IRM axes.

ABB 120 IRM Axes	Range	Max Speed (T)
Axis 1 Rotation	+165° to −165°	250°/s
Axis 2 Arm	+110° to −110°	250°/s
Axis 3 Arm	+70° to −110°	250°/s
Axis 4 Wrist	+160° to −160°	320°/s (420°/s)
Axis 5 Bend	+120° to −120°	320°/s (590°/s)
Axis 6 Turn	+400° to −400°	420°/s (600°/s)

The end-effector robot arm position data and target to where is heading are communicated by robot IRC5 controller at any cycle time to PLC using a *robtarget* data type format. *Robtarget* data type consists of four components:

- Translation—position (P_x, P_y, P_z) of the end effector expressed in mm;
- Rotation—orientation, expressed in the form of quaternions (q_1, q_2, q_3, q_4);

- Robot configuration—axis configuration of the robot (cf_1, cf_4, cf_6, cf_x). The configuration supervision will check axes 1, 4, and 6, will not move more than 180 degrees;
- External axes—position of the external axes (eax_a, eax_b, eax_c, eax_d, eax_e, eax_f). The robot controls up to six external axes, in addition to its six internal axes, each logical axis can be connected to a physical axis, in this case, the connection must be activated in the system parameters.

Rotation quaternions or Euler parameters arranged in quaternion form, as seen in ABB *robtarget* position data type, describe the axis–angle representation, a commonly used method for defining any 3D rotation (or sequence of rotations). This representation can serve as an alternative to rotation matrices in the three-dimensional space, by specifying only two parameters: a unit vector for the rotation axis and the rotation angle about an axis. Therefore, for a given unit vector axis (x, y, z), to which a rotation angle, ω, is applied, the Euler parameters can be defined using the following conversion:

$$q_1 = \cos\frac{\omega}{2}; q_2 = x \cdot \sin\frac{\omega}{2}; \; q_3 = y \cdot \sin\frac{\omega}{2}; \; q_4 = z \cdot \sin\frac{\omega}{2} \tag{1}$$

Given the rotation quaternion for the end effector orientation, as defined in ABB:

$$q = (q_1, \; q_2, \; q_3, \; q_4) \tag{2}$$

$$q_1 = \omega; \; q_2 = \; P_x; \; q_3 = \; P_y; \; q_4 = \; P_z, \tag{3}$$

The corresponding transformation matrix, T, for robot direct kinematics, can be calculated in terms of the unit quaternion components:

$$\begin{bmatrix} \omega^2 + x^2 - y^2 - z^2 & 2\cdot(xy \; - \; \omega z) & 2\cdot(\omega y \; + \; xz) & P_x \\ 2\cdot(\omega z \; + \; xy) & \omega^2 - x^2 + y^2 - z^2 & 2\cdot(yz \; - \; \omega x) & P_y \\ 2\cdot(xz \; - \; \omega y) & 2\cdot(\omega x \; + \; yz) & \omega^2 - x^2 - y^2 + z^2 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

When calculating the corresponding axis positions (joint angles) through inverse kinematics, for the six-axis robot, there will be eight possible solutions generated [20–22]. This means that the robot can reach the same end effector position and orientation, with several different link angles or configurations for the RM axes [23]. For reducing/selecting the desired joint angles solution $(\theta_1, \theta_2, \theta_3, \; \theta_4, \theta_5, \theta_6)$ out of the eight possible, the RM controller provides a subset of data through the *robtarget* data type; therefore, the robot configuration data type is specified using axis 1, 4, and 6 quadrant values. For a rotating axis, the configuration values define the quadrant of the robot axis meaning a 90° quarter revolution in a positive or negative direction. Component cf_x is used to select one of eight possible robot configurations numbered from 0 through 7. When the RM movement is finalized, the RM controller checks that the robot has reached the programmed configuration.

3. Modeling and Simulation of the A/DRC

3.1. ABB 120 IRM Kinematics. Modeling, Analysis, and Validation

3.1.1. Forward Kinematics

The direct kinematics general approach is based on linear algebra, to compute the joint variables function of the RM mechanical structure for the initial, reference frame [23].

For kinematic chain representation in a more compact way, the homogeneous transformation matrix is preferred to represent a frame, translated to a new position P and rotated over another frame, described by the fundamental rotational matrices [24], as follows:

$$ {}^{n-1}_{n}A = \begin{bmatrix} {}^{n-1}_{n}R(\theta_n) & {}^{n-1}_{n}P(\theta_n) \\ 0_{1X3} & \sigma \end{bmatrix}, \quad (5) $$

where σ is the scale factor, which usually is 1.

For the forward kinematics approach and calculation of the 6-DoF RM position and orientation, the homogenous transformation matrix A, representing how a frame Σ must be displaced to achieve the next frame, will be renamed as T matrix, to describe further the complete kinematic chain by using post-multiplication proprieties. The Denavit–Hartenberg (D-H) convention is used [25,26] to describe the open kinematic chain structure, which introduces four parameters (scalars). These parameters vary with the RM joint motion, to represent a coordinate frame, Σ, displacement in space to achieve the next one.

The graphical representation and interpretation of the DH parameters for describing each robot link are as follows:

- θ_i—angle/ rotation between axis X_{i-1} and axis X_i, about the axis Z_{i-1}, which varies with the joint motion;
- d_i—coordinate of the origin $\mathbf{O_i}$ translation distance along the Z_{i-1} axis of the previous frame;
- α_i—the rotation angle between axis Z_{i-1} and axis Z_i about the axis X_i;
- a_i—the translation distance between origin coordinates O_i and O_{i-1}, along axis X_i.

According to the dimensions provided by ABB in the product specifications for ABB 120 IRM, D-H kinematic parameters are shown in Table 2, while the D-H link coordinate system and frame assignment for the robot joints are presented further in Figure 2. To represent the relation between two frames Σ and Σ_{i-1} such as the representation or specification of the orthogonal coordinate frames, by translating or rotating the frame Σ_{i-1} to achieve the next frame Σ_i, the joint-to-joint homogenous transformation matrix T_i is used, where $i > 1$. For every joint, i, greater than 1, associated with the degree of freedom, each transformation matrix is a function of θ_i Angle, which gives the relationship between the frame Σ_i over the frame Σ_{i-1}, starting with the initial frame, Σ_0. Therefore, using the D-H parameters, for every link or joint coordinate i, representing each of the 6 links, such that $1 < i \leq m$ where m is 6, T_i can be written, by successive post multiplication of the single transformations [27].

Table 2. ABB 120 IRM D-H parameterization used to determine the homogeneous transformation matrix for each link (also known as the Denavit–Hartenberg matrix).

Joint	α_i [°]	a_i [mm]	θ_i [°]	d_i [mm]
$T_1(\theta_1)$	$-\pi/2$	0	θ_1	$d_1 = 290$
$T_2(\theta_2)$	0	$a_1 = 270$	$\theta_2 - \pi/2$	0
$T_3(\theta_3)$	$-\pi/2$	$a_1 = 70$	θ_3	0
$T_4(\theta_4)$	$\pi/2$	0	θ_4	$d_4 = 302$
$T_5(\theta_5)$	$-\pi/2$	0	θ_5	0
$T_6(\theta_6)$	0	0	$\theta_6 + \pi$	$d_6 = 72$

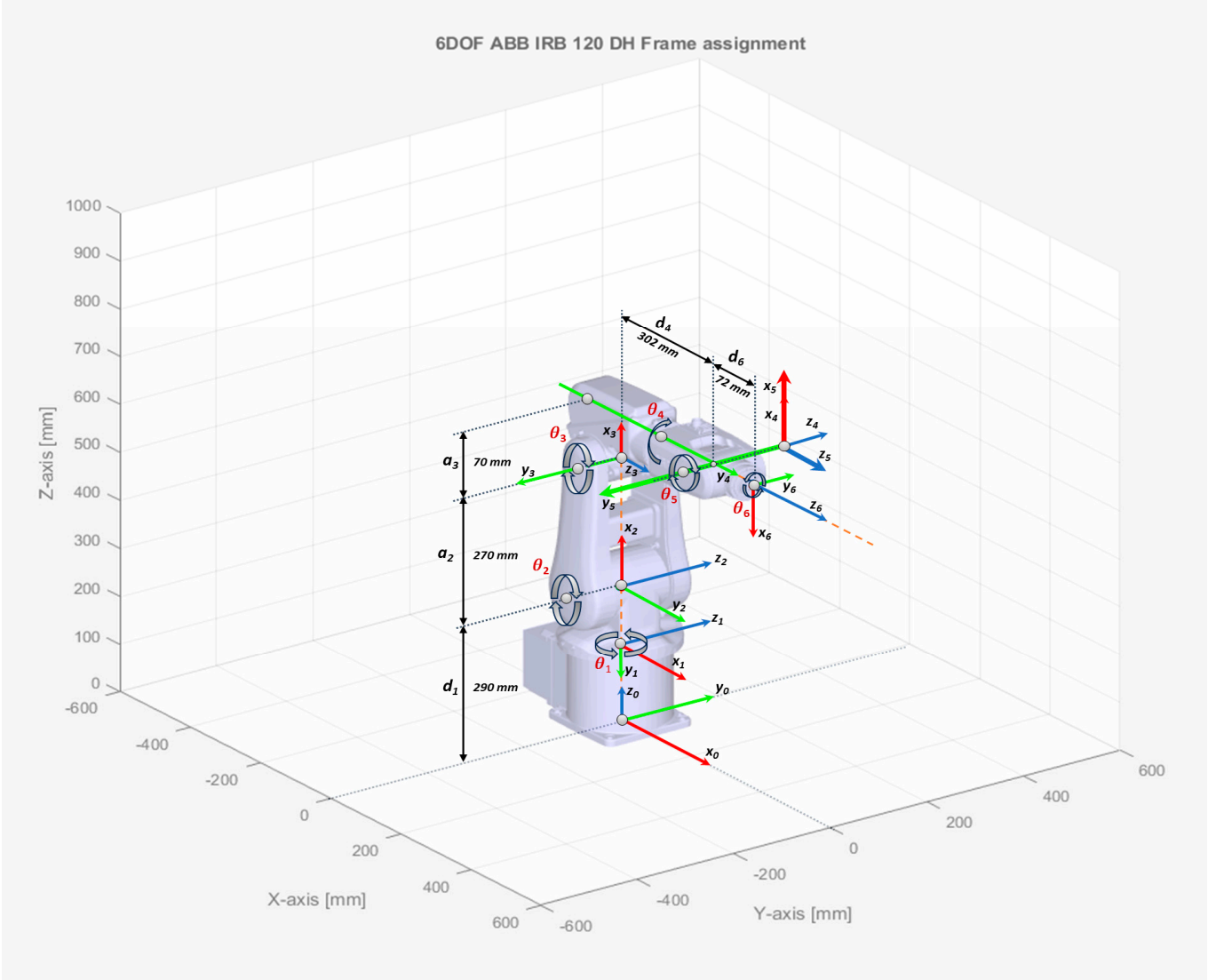


Figure 2. D-H link coordinate system and frame assignment for the ABB 120 IRM.

The D-H parameters from Table 2, defined by the manipulator kinematic parameters, are substituted into the homogeneous transformation matrices, sometimes called relative transforms, for generating the joint-to-joint transformation matrices for each frame and link dimension [24], which maps the coordinates of adjacent links $i-1$ over the link i. The initial frame, Σ_0, is moved to achieve the next frame Σ_1, origin O_1, and rotated with α_1 about the x-axis. The translation distance along the z-axis is L_1, as pointed in Figure 3a.

$$ {}^{0}_{1}\mathrm{T}(\theta_1) = \begin{bmatrix} {}^{0}_{1}\mathrm{R}_z(\theta_1) & {}^{0}_{1}\mathrm{P}(\theta_1) \\ 0_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

Frame Σ_2 is achieved by shifting Σ_1 with L_2 along the z-axis and rotated with θ_2 about the z-axis (Figure 3b).

$$ {}^{1}_{2}\mathrm{T}(\theta_2) = \begin{bmatrix} {}^{1}_{2}\mathrm{R}_y(\theta_2) & {}^{1}_{2}\mathrm{P}(\theta_2) \\ 0_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7} $$

Frame Σ_3 is achieved by shifting Σ_2 with L_3 along the z-axis and rotated with α_3 about the x-axis (Figure 3c).

$$ {}^{2}_{3}T(\theta_3) = \begin{bmatrix} {}^{2}_{3}R_y(\theta_3) & {}^{2}_{3}P(\theta_3) \\ 0_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_3) & 0 & \cos(\theta_3) & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8) $$

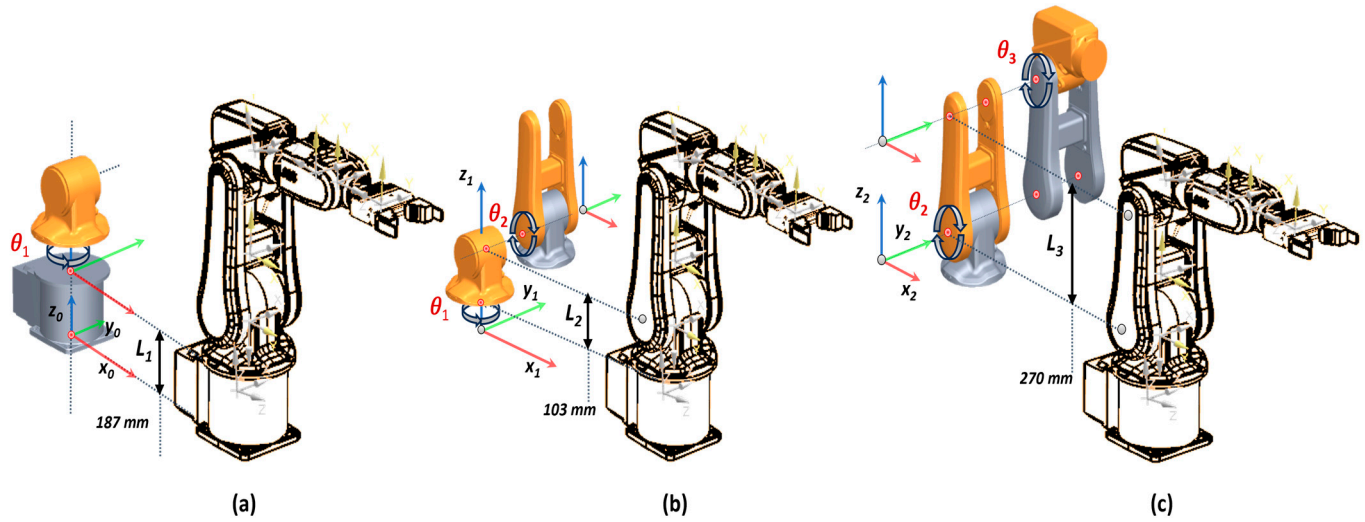


Figure 3. ABB 120 IRM frame transformations according to D-H convention: (**a**) frame Σ_0 to frame Σ_1; (**b**) frame Σ_1 to frame Σ_2; (**c**) frame Σ_2 to frame Σ_3.

Frame Σ_4 is achieved by shifting Σ_3 with L_4 along the x-axis, shifting with L_5 along the z-axis and rotated with α_4 about the x-axis (Figure 4a).

$$ {}^{3}_{4}T(\theta_4) = \begin{bmatrix} {}^{3}_{4}R_x(\theta_4) & {}^{3}_{4}P(\theta_4) \\ 0_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & \cos(\theta_4) & -\sin(\theta_4) & 0 \\ 0 & \sin(\theta_4) & \cos(\theta_4) & L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9) $$

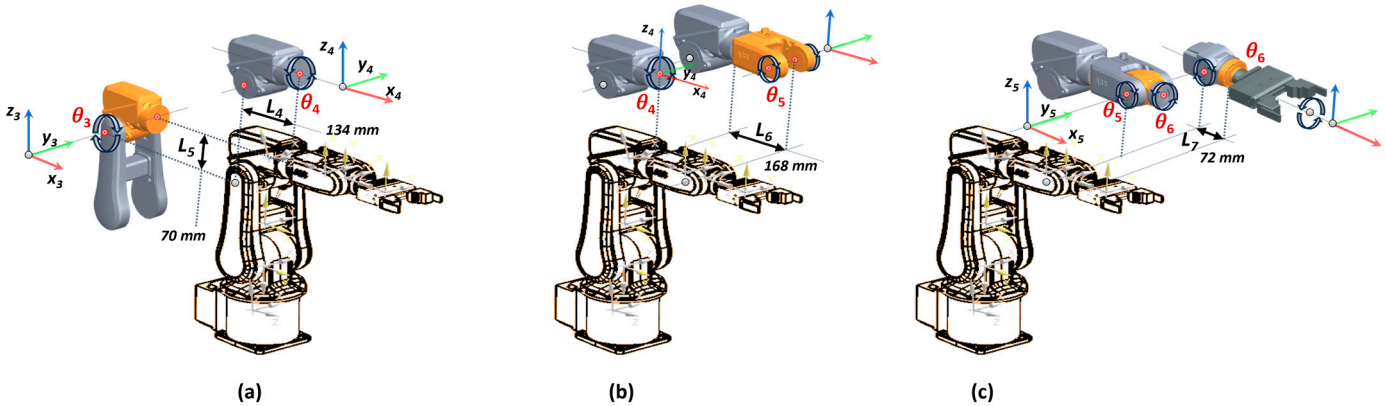


Figure 4. ABB 120 IRM frame transformations according to D-H convention: (**a**) frame Σ_3 to frame Σ_4; (**b**) frame Σ_4 to frame Σ_5; (**c**) frame Σ_5 to frame Σ_6.

Frame Σ_5 is achieved by shifting Σ_4 with L_6 along the *x*-axis and rotated with α_5 about the *x*-axis (Figure 4b).

$$ {}^{4}_{5}T(\theta_5) = \begin{bmatrix} {}^{4}_{5}R_y(\theta_5) & {}^{4}_{5}P(\theta_5) \\ 0_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) & L_6 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10) $$

Frame Σ_6 is achieved by shifting Σ_5 with L_7 along the *x*-axis and rotated with θ_6 about the *z*-axis (Figure 4c).

$$ {}^{5}_{6}T(\theta_6) = \begin{bmatrix} {}^{5}_{6}R_x(\theta_6) & {}^{5}_{6}P(\theta_6) \\ 0_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & L_7 \\ 0 & \cos(\theta_6) & -\sin(\theta_6) & 0 \\ 0 & \sin(\theta_6) & \cos(\theta_6) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11) $$

Matrix ${}^{i-1}_{i}T$ denotes the homogeneous transformation, by relating the description of a point in the frame Σ_i to the description of the same point in the frame Σ_{i-1}. The algebraic approach of forward kinematics can be expressed as a matrix successive concatenation, by multiplying all six transformation matrices according to the D-H convention [27–29]. This allows the complete transformation from the origin of the base robot to the RM end effector by splitting the transformation matrix ${}^{0}_{6}T$ into matrices, which generate a rotation matrix and a translation vector to calculate the position and orientation of a kinematic chain from the joint variables.

$$ {}^{0}_{6}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12) $$

By extracting the fourth column of T, the end effector position vector p is achieved. Furthermore, n is the normal vector, s the sliding vector, and a is the approach vector:

$$ p = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \; n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \; s = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}, \; a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (13) $$

By defining the D-H parameter table and calculating the homogeneous transformation matrix, T, a mathematical representation of the robot forward kinematic structure is achieved. To calculate the inverse kinematics solutions, the homogeneous transformation matrix, T, is the starting point, referred to as the input for the inverse kinematics problem. In this scenario, for reaching a desired position and orientation of the end-effector, each joint coordinates are calculated, associated with the degree of freedom denoted by the set of unknown variables $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$.

3.1.2. Inverse Kinematics

There are several ways to calculate the inverse kinematics solution, which can be tested concerning each other, the geometric approach—used further in this paper, as well as the algebraic method which can be solved by equation systems using numerical methods [21]. The analytical solution derives a closed-form mathematical expression to directly compute joint angles or positions based on the desired end-effector position and orientation providing explicit, precise expressions for joint configurations without approximation,

enabling rapid computation, unlike numeric solutions. The motion algorithms embedded in most industrial robot controllers avoid implementing numerical solutions of Jacobian Inverse Kinematics Algorithms. Instead, they employ analytical inverse solutions, which are computationally more efficient.

To simplify the problem, the kinematic decoupling method is implemented by splitting the inverse kinematics general problem into two separate directions using the relative position and orientation of the intersection of the three reference systems in one point: first homogenous matrix ${}^{0}_{3}\mathrm{T}$ defines the position and second matrix ${}^{3}_{6}\mathrm{T}$ for solving the orientation problem of the spherical wrist. The geometric solution for the first joint angle is determined by reducing the three-dimensional problem into an XY-plane, taking into consideration only the first three joints and keeping in mind that the wrist motion of the fourth coordinate system is not influencing the equation. The first angle, θ_1, is calculated by computing the position of the wrist, P_4, from the rotation matrix, T, by using the approach vector, a, and position vector, p, of the end effector position (Figure 5). The geometric solution for the inverse kinematics of the first link angle, θ_1, admits two possible solutions, although it is not always physically or mechanically possible. θ_1 and $(\theta_1 + \pi)$ are calculated as below:

$$P_4 = p - d_6 \cdot a, \tag{14}$$

$$p = \begin{bmatrix} P_x \\ P_y \end{bmatrix}, \; a = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{15}$$

$$\theta_1 = \mathrm{atan2}(P_4(y), \; P_4(x)) \tag{16}$$

$$\gamma = \arccos \frac{{a_2}^2 + R^2 - {L_4}^2}{2R \cdot a_2} \tag{17}$$

$$\beta = \mathrm{atan2}(P_4(z) - d_1, r) \tag{18}$$

$$\eta = \arccos \frac{{a_2}^2 + {L_4}^2 - R^2}{2a_2 \cdot L_4} \tag{19}$$

$$\varphi = \mathrm{atan2}(d_4, a_3) \tag{20}$$

$$L_4 = \sqrt{{a_3}^2 + {d_4}^2} \tag{21}$$

$$r = \sqrt{P_4(x)^2 + P_4(y)^2} \tag{22}$$

$$R = \sqrt{r^2 + \left(P_4(z) - d_1\right)^2} \tag{23}$$

Geometric solution for the inverse kinematics of the joint angle θ_2 and θ_3, as drafted in Figure 6, admits two solutions.

For elbow up:

$$\theta_2 = \frac{\pi}{2} - \beta - \gamma \tag{24}$$

$$\theta_3 = \pi - \varphi - \eta \tag{25}$$

For elbow down:

$$\theta_2 = \frac{\pi}{2} - \beta + \gamma \tag{26}$$

$$\theta_3 = \pi - \varphi + \eta \tag{27}$$

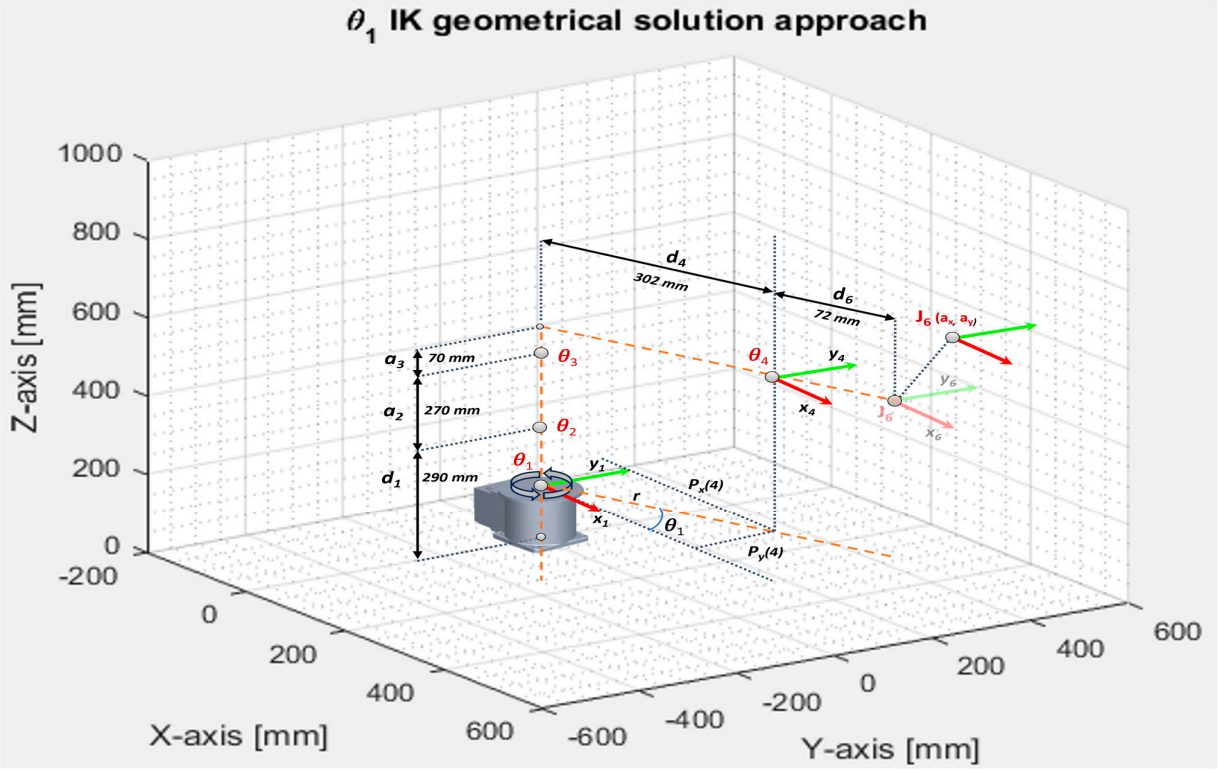


Figure 5. Joint θ_1 geometrical solution approach by computing and reducing into an XY-plane the position of the wrist, P_4.

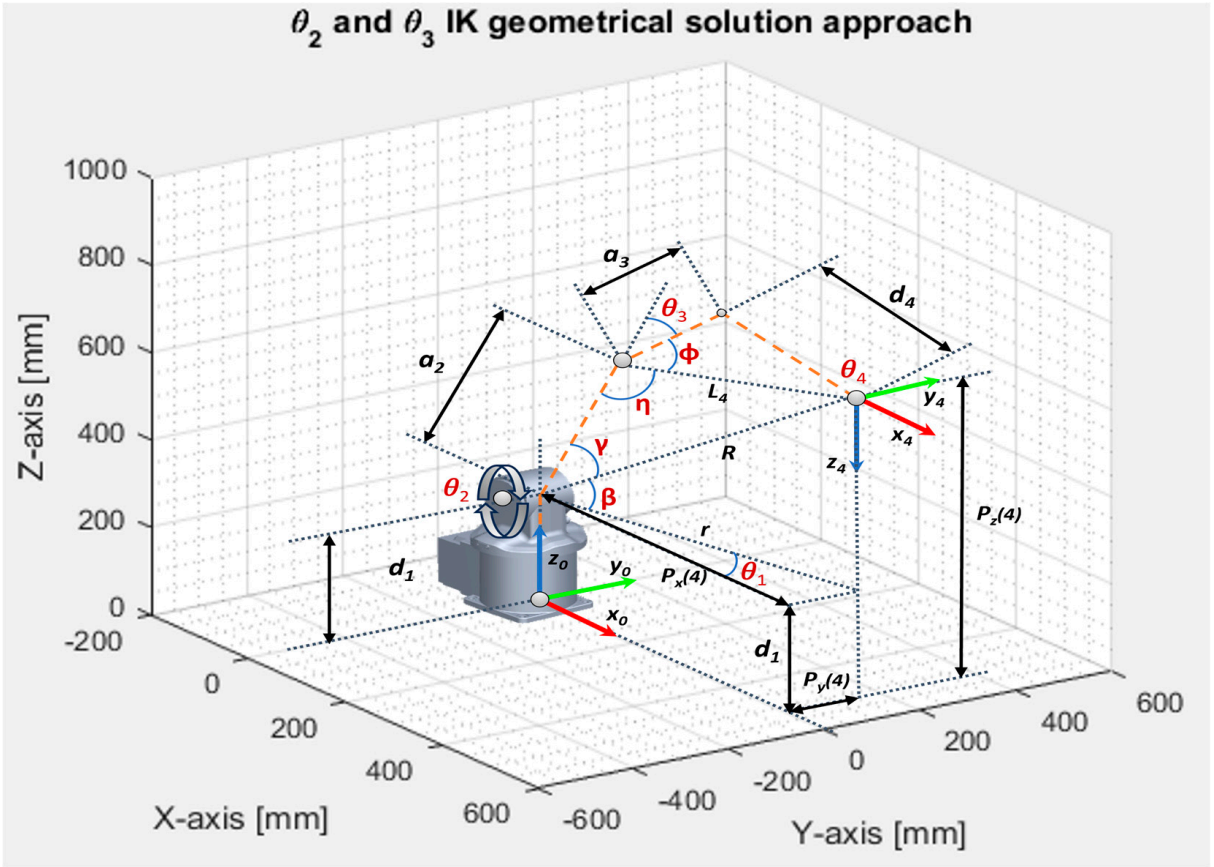


Figure 6. Joints θ_2 and θ_3 geometrical solution approach starting from the computation of the wrist position P_4 and joint angle θ_1.

As discussed, by applying the kinematic decoupling approach and using the desired orientation of the end effector, the last three joint angles can be calculated by finding a set of Euler angles corresponding to the desired rotation matrix, ${}^{3}_{6}\mathrm{R}$, which is equal to the rotation matrix for the Euler angles [28,29].

$$\mathrm{T} = {}^{0}_{3}\mathrm{R} \cdot {}^{3}_{6}\mathrm{R} \tag{28}$$

$${}^{3}_{6}\mathrm{R} = ({}^{3}_{6}\mathrm{R})^{-1} \cdot \mathrm{T} = ({}^{2}_{3}\mathrm{R})^{-1} \cdot ({}^{1}_{2}\mathrm{R})^{-1} \cdot ({}^{0}_{1}\mathrm{R})^{-1} \cdot \mathrm{T} \tag{29}$$

$${}^{3}_{6}\mathrm{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c_4 \cdot c_5 \cdot c_6 - s_4 \cdot s_6 & -c_4 \cdot c_5 \cdot s_6 - s_4 \cdot c_6 & c_4 \cdot s_5 \\ s_4 \cdot c_5 \cdot c_6 + c_4 \cdot s_6 & -s_4 \cdot c_5 \cdot s_6 + c_4 \cdot c_6 & s_4 \cdot s_5 \\ -s_5 \cdot c_6 & s_5 \cdot c_6 & c_5 \end{bmatrix} \tag{30}$$

For the nonsingular case, meaning $\theta_5 \neq 0$, which implies $\cos(\theta_5) \neq 1$, there are two possible geometric solutions for the inverse kinematics orientation:

$$r_{33} = \cos(\theta_5) \tag{31}$$

$$\sin(\theta_5) = \pm\sqrt{1 - (\cos(\theta_5))^2} = \pm\sqrt{1 - {r_{33}}^2} \tag{32}$$

$$\theta_5 = \mathrm{atan2}\left(r_{33}, \pm\sqrt{1 - {r_{33}}^2}\right) \tag{33}$$

For wrist up:

$$\theta_4 = \mathrm{atan2}(r_{23}, r_{13}) \tag{34}$$

$$\theta_6 = \mathrm{atan2}(-r_{32}, r_{31}) \tag{35}$$

For wrist down:

$$\theta_4 = \mathrm{atan2}(r_{23}, r_{13}) + \pi \tag{36}$$

$$\theta_6 = \mathrm{atan2}(-r_{32}, r_{31}) + \pi \tag{37}$$

For the singular configuration $\theta_5 = 0$, meaning $\cos(\theta_5)$ in the matrix ${}^{3}_{6}\mathrm{R}$ is 1, it generates an infinite number of solutions (degenerate solution, in this case, θ_4 cannot be determined), only the sum of $(\theta_4 + \theta_6)$ can be found; thus, it is common to arbitrarily set θ_4 to null and solve the θ_6 joint angle.

$$\theta_4 + \theta_6 = \mathrm{atan2}(r_{21}, r_{11}) \tag{38}$$

$${}^{3}_{6}\mathrm{R} = \begin{bmatrix} c_4 \cdot c_6 - s_4 \cdot s_6 & -c_4 \cdot s_6 - s_4 \cdot c_6 & 0 \\ s_4 \cdot c_6 + c_4 \cdot s_6 & -s_4 \cdot s_6 + c_4 \cdot c_6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{39}$$

For wrist up:

$$\theta_4 = 0 \tag{40}$$

$$\theta_6 = \mathrm{atan2}(r_{12} - r_{21}, -r_{11} - r_{22}) \tag{41}$$

For wrist down:

$$\theta_4 = -\pi \tag{42}$$

$$\theta_6 = \mathrm{atan2}(r_{12} - r_{21}, -r_{11} - r_{22}) + \pi \tag{43}$$

Figure 7 presents the simulation results of the inverse kinematics motion spaces, specifically for RM joint angles θ_3, θ_4, θ_5, and θ_6 when translating linear motion from the A/DRC Pick Cylinder position to the RM home position.

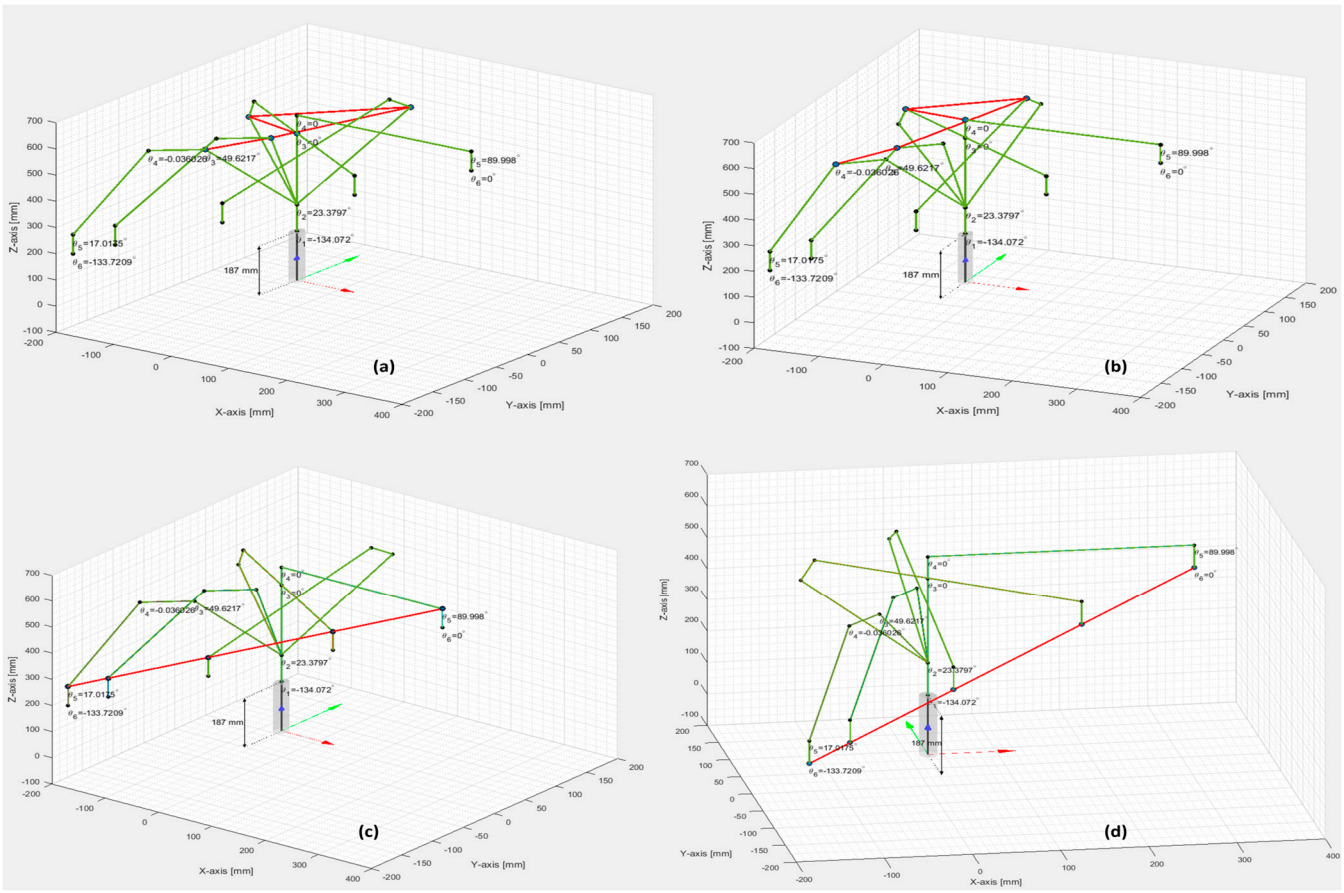


Figure 7. ABB 120 IRM Linear motion space simulation with Inverse Kinematics: (**a**) θ_3 joint angle; (**b**) θ_4 joint angle; (**c**) θ_5 joint angle; (**d**) θ_6 joint angle.

3.2. Assembly Tasks Planning and STPN Model

3.2.1. Planning Assembly Tasks

A/DRC assisted by ABB 120 IRM performs assembly and disassembly operations of a predefined 5-component product for two workpieces called WP1 and WP2, [7,12,30]. The assembly flow process starts from storing and supplying subcomponents and progresses to the final delivery of the fully assembled, designated product type onto the cell conveyor unit, which functions as a transport line. The workpiece to be assembled consists of five components: a base pallet, a body, a top, and two inserted cylinders, which may be either plastic or metal. To introduce flexibility in manufacturing, two workpieces can be handled in the robotic cell depending on the operator's selection. The top part is different, it can be either with square edges (WP1) or with a rounded shape (WP2). The assembly process involves RM performing repetitive tasks of picking, handling, and transporting the subcomponents to the specified assembly area within the cell, task planning for the assembly process of WP1 or WP2, is shown in Figure 8. First, the base pallet is placed on the conveyor FC2, then the product is partially assembled on a suspended bracket called Assembly Tray FC1. The body, top, and cylinders are fitted in this location, then the structure is placed on the base pallet. Once the assembly is complete, the assembled workpiece is transported via the conveyor FC2 to the right side of the cell, the delivery point for assembled products, serving as a starting point for another manufacturing process.

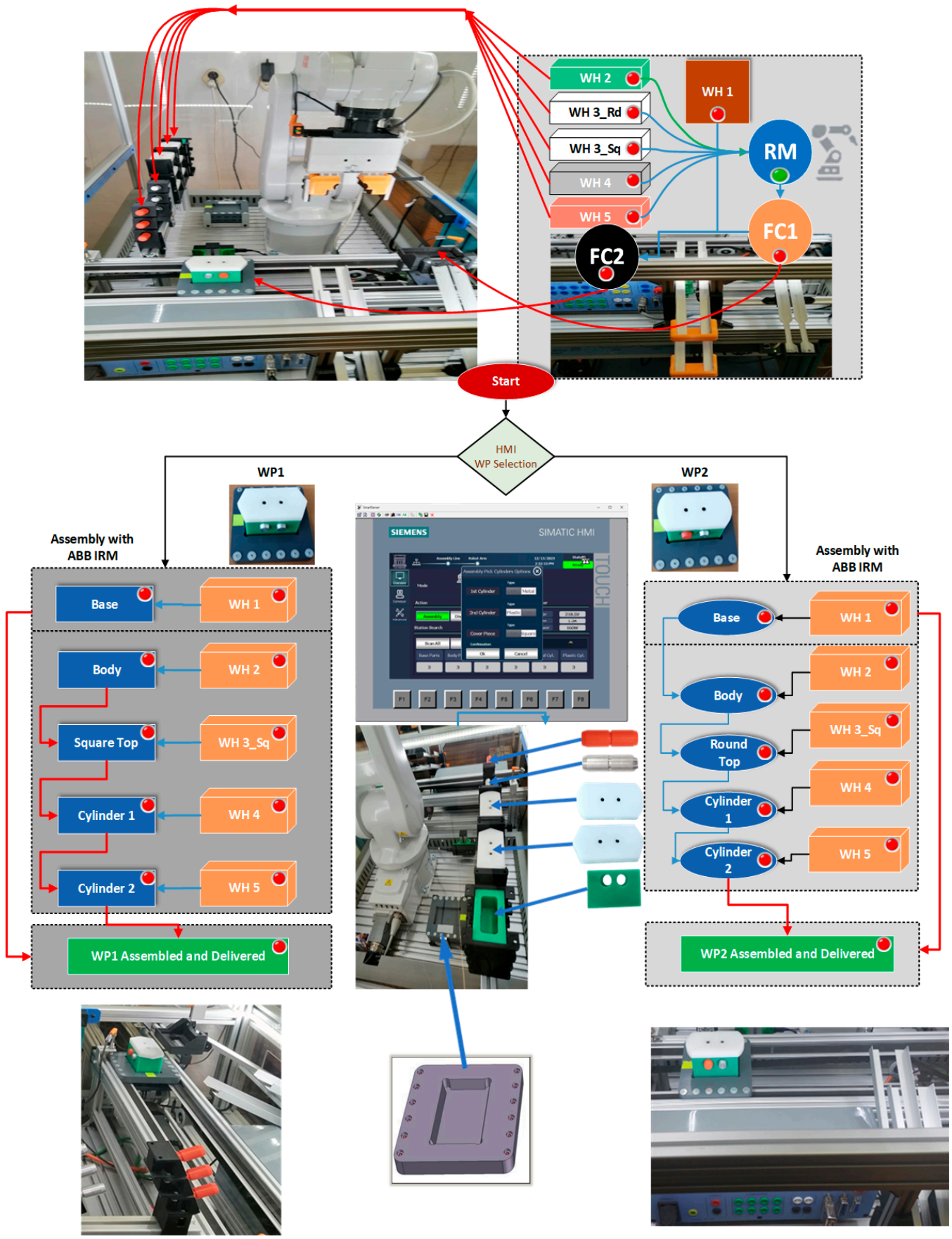


Figure 8. Node-RED task planning for the A/DRC assembly operation of WP1 or WP2.

3.2.2. STPN Model for Assembly

STPN model for A/DRC assembly operation starts by selecting one of the workpieces WP1 or WP2 to be integrated per component by the manufacturing process into the final product. The assembly process involves RM picking the subcomponents from the storage units, handling and assembling them on the Assembly Tray FC1, and transporting the assembled workpiece on the conveyor FC2 to the right-side exit line of the cell, all this with the corresponding time durations. Synchronizing signals for assembly processes are required, for time-align and monitoring checks issues of the PLC program. Therefore, a Ready signal from previous operations like assembly or disassembly is a condition for starting the current cell task. The STPN model, simulated in Sirphyco [31] is shown in Figure 9.

The STPN is defined as follows:

$$\mathrm{STPN_A} = \left\{\mathrm{TPN_A},\quad \mathrm{E},\quad \mathrm{Sync}\right\}, \tag{44}$$

where $\mathrm{TPN_A}$ is the Timed Petri Net model and corresponds to the assembly function.

$$\mathrm{TPN_A} = \left\{\mathrm{P_A},\ \mathrm{T_A}, \mathrm{Pre_A}, \mathrm{Post_A}, \mathrm{m_{0_A}}, \mathrm{tempo_A}\right\} \tag{45}$$

$\mathbf{P_A}$ is the partitioned set of assembly $\mathrm{TPN_A}$ model places:

$$\mathrm{P_A} = \left\{\mathrm{P_{control_A}}, \mathrm{P_{assembly_A}}, \mathrm{P_{supervision_A}}\right\} \tag{46}$$

$\mathbf{P_{control_A}}$ (Orange color places) represent the states of the PLC program functions control tasks:

$$\mathrm{P_{control_A}} = \left\{\mathrm{P_{i_A}}\right\}_{\mathrm{i_A}=\overline{1,8}} \tag{47}$$

$\mathbf{P_{assembly_A}}$ (Olive color places) are the discrete states of the flexible assembly tasks:

$$\mathrm{P_{assembly_A}} = \left\{\mathrm{P_{j_A}}\right\}_{\mathrm{j_A}=\overline{10,23}} \tag{48}$$

$\mathbf{P_{supervision_A}}$ (Blue color places) are the states generated for cell monitoring and supervision of the important tasks during the flexible assembly process:

$$\mathrm{P_{supervision\ A}} = \left\{\mathrm{P_{j_A}}\right\}_{\mathrm{j_A}=\overline{30,37}} \tag{49}$$

$\mathbf{T_A}$ is the partitioned transition set of assembly for $\mathrm{TPN_A}$:

$$\mathrm{T_A} = \left\{\mathrm{T_{CondForAssembly_A}}, \mathrm{T_{ReadyForAssembly_A}}, \mathrm{T_{assembly_A}}, \mathrm{T_{assembled_A}}\right\} \tag{50}$$

- $\mathbf{T_{CondForAssembly_A}}$ are the discrete transitions acknowledging that no other operation is running (assembly or disassembly);
- $\mathbf{T_{ReadyForAssembly_A}}$, transition for assembly conditions: SCADA assembly command is ON, and cell is Ready (no other operation is running);
- $\mathbf{T_{assembly_A}}$ are the discrete transitions of the flexible assembly tasks;
- $\mathbf{T_{assembled_A}}$ marks the final transition when the assembled workpiece is conveyed by FC2 to the right-side exit line of the robotic cell.

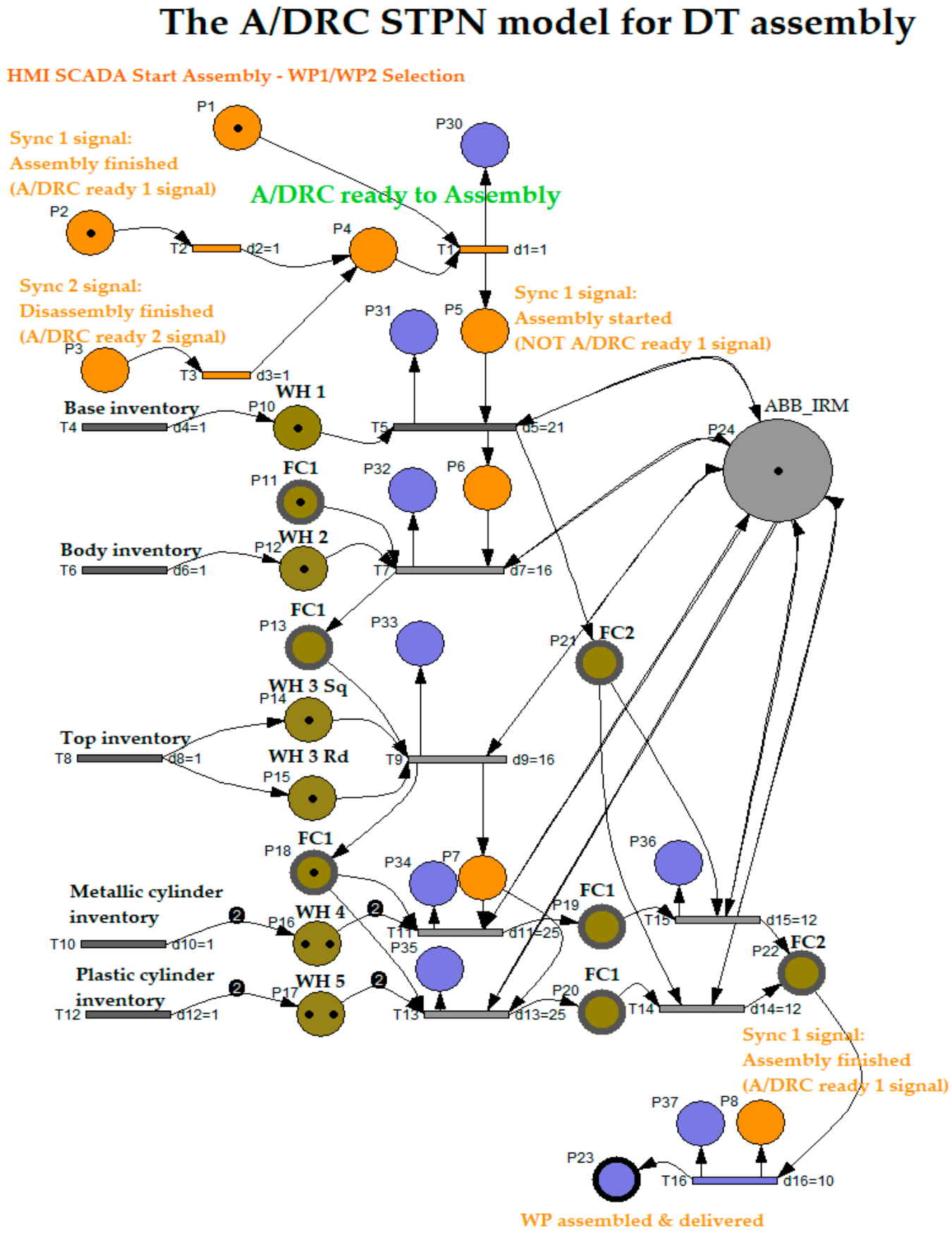


Figure 9. STPN model for the A/DRC assembly operation of WP1 or WP2.

$$T_{CondForAssembly_A} = \{T_2, T_3\} \quad (51)$$

$$T_{ReadyForAssembly_A} = \{T_1\} \quad (52)$$

$$T_{assembly_A} = \{T_{i_A}\}_{j_A=\{\overline{4,16}\}} \quad (53)$$

$$T_{assembled_A} = \{T_{16_A}\} \quad (54)$$

For assembly on the robotic cell assisted by ABB 120 IRM, the supervision places (Figure 10) in the set (49) delegated to monitor the transitions set (50) are as follows:

- P30, T1 supervision transition is executed when the assembly task starts, and conditions are met, meaning the SCADA assembly command is ON, and A/DRC is Ready (no other operation is running);
- P31, transition T5 is executed when Base plate is placed on the conveyor FC2;
- P32, T7 supervision transition is performed when the Body part is placed on the Assembly Tray FC1;
- P33, T9 transition is carried out when assembly actions for choosing Top part with round or square edges corresponding to selected WP1 or WP2 to be manufactured;
- P34, transition T11 is executed when choosing the Metal cylinder for assembly;
- P35, transition T13 is executed when choosing the Plastic cylinder for assembly;
- P36, T15 transition is running when the assembled workpiece on the conveyor FC2 is transported to the exit line of the robotic cell;
- P37, T16 supervision transition is executed when the Assembly task is finalized in the cell and the Ready signal is set to ON.

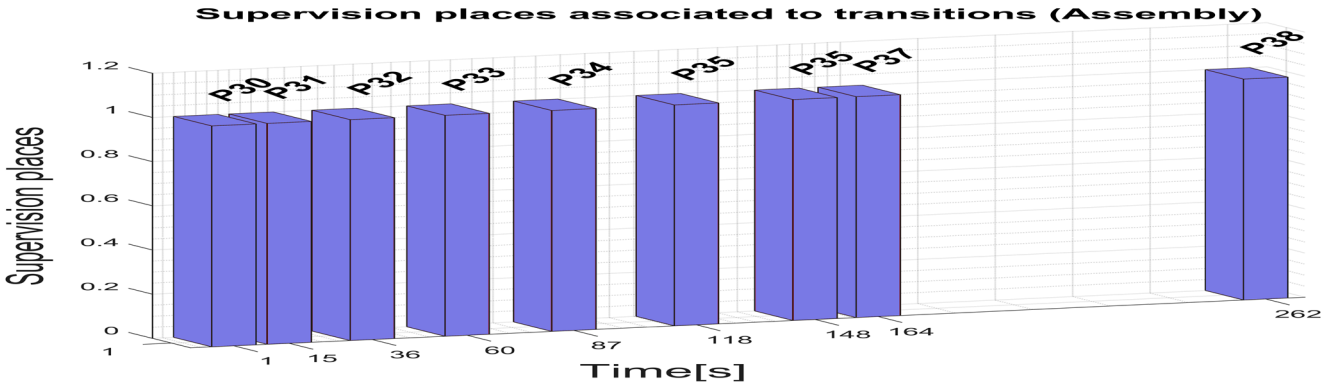


Figure 10. Sirphyco simulation of STPN model for the A/DRC assembly operation of WP1 or WP2.

$\mathbf{Pre_A}$ is the input incidence function.

$$\mathbf{Pre_A} : P_A \times T_A \rightarrow Q_{+_{Apre}} \quad (55)$$

$\boldsymbol{Post_A}$ is the output incidence function.

$$\mathbf{Post_A} : P_A \times T_A \rightarrow Q_{+_{Apost}} \quad (56)$$

m_{0_A} is the initial marking set of tokens corresponding to Assembly places of the STPN modeled process.

$\mathbf{tempo_A}$ is a function that defines the timings associated with the transitions.

$$\mathbf{tempo_A} : T_A \rightarrow Q_{+_{AT}} \cup \{0\} \quad (57)$$

E is the set of external events.

$$\mathbf{E} = \left\{ \mathrm{Ev}^1, \mathrm{Ev}^2 \right\} \cup \{\mathrm{e}\}, \tag{58}$$

Sync application in definition (44) is a function from the set of discrete transitions corresponding to the assembly of A/DRC external events joined with the neutral element e,

$$\mathbf{Sync} : \{\mathrm{T}_2,\ \mathrm{T}_3\} \rightarrow \left\{ \mathrm{Ev}^1,\ \mathrm{Ev}^2 \right\} \cup \{\mathrm{e}\} \tag{59}$$

The synchronization signal that generates the Ready signal when no Assembly task is running in the robotic cell:

$$\mathbf{Sync\ 1_A} :\ \mathrm{T}_2 \rightarrow \left\{ \mathrm{Ev}^1 \right\} \tag{60}$$

$$\mathrm{Ev}^1 = \mathbf{Sync\ 1_A}\ \text{signal} \tag{61}$$

The synchronization signal that generates the Ready signal when no disassembly task is running in the robotic cell:

$$\mathbf{Sync\ 2_A} : \mathrm{T}_3 \rightarrow \left\{ \mathrm{Ev}^2 \right\} \tag{62}$$

$$\mathrm{Ev}^2 = \mathbf{Sync2_A}\ \text{signal} \tag{63}$$

3.3. Disassembly Tasks Planning and STPN Mode

3.3.1. Planning Disassembly Tasks

Workpieces ready for the disassembly process are delivered on conveyor FC2, on the right side of the robotic cell, serving as a "check-in" location point for products ready for disassembly. Further, the workpiece is transported by conveyor FC2 to the dismantling location where RM will do the rest of the disassembly process through repetitive tasks of picking and placing subcomponents in the reverse order as the assembly operation was performed. Cylinders, top, and body parts will be extracted and placed in the dedicated slide compartments on the cell, front bottom panel. The last task performed by RM is picking the base pallet from FC2 and placing it into the designated storage place W1, inside the cell. Task planning for the disassembly process is shown in Figure 11. The local control system-HMI is shown as well, for selection and starting the disassembly process.

3.3.2. STPN Model for Disassembly

The disassembly process involves first that WP1 or WP2 workpieces will be delivered for disassembly process on the right side of the station and transported on the FC2 conveyor. RM picked and placed the structure in the Assembly Tray FC1 where all the subcomponents will be dismantled, then finally the last part is handled, and the base pallet will be stored on the dedicated storage unit. All these disassembly operations and transitions are executed with the corresponding time durations. The same two synchronizing signals, as for the assembly process are required, for time-align and monitoring checks issues of the PLC program. Therefore, Ready signals from previous operations like assembly or disassembly will be the starting condition for the disassembly task. The STPN model, simulated in Sirphyco [31], is drafted in Figure 12.

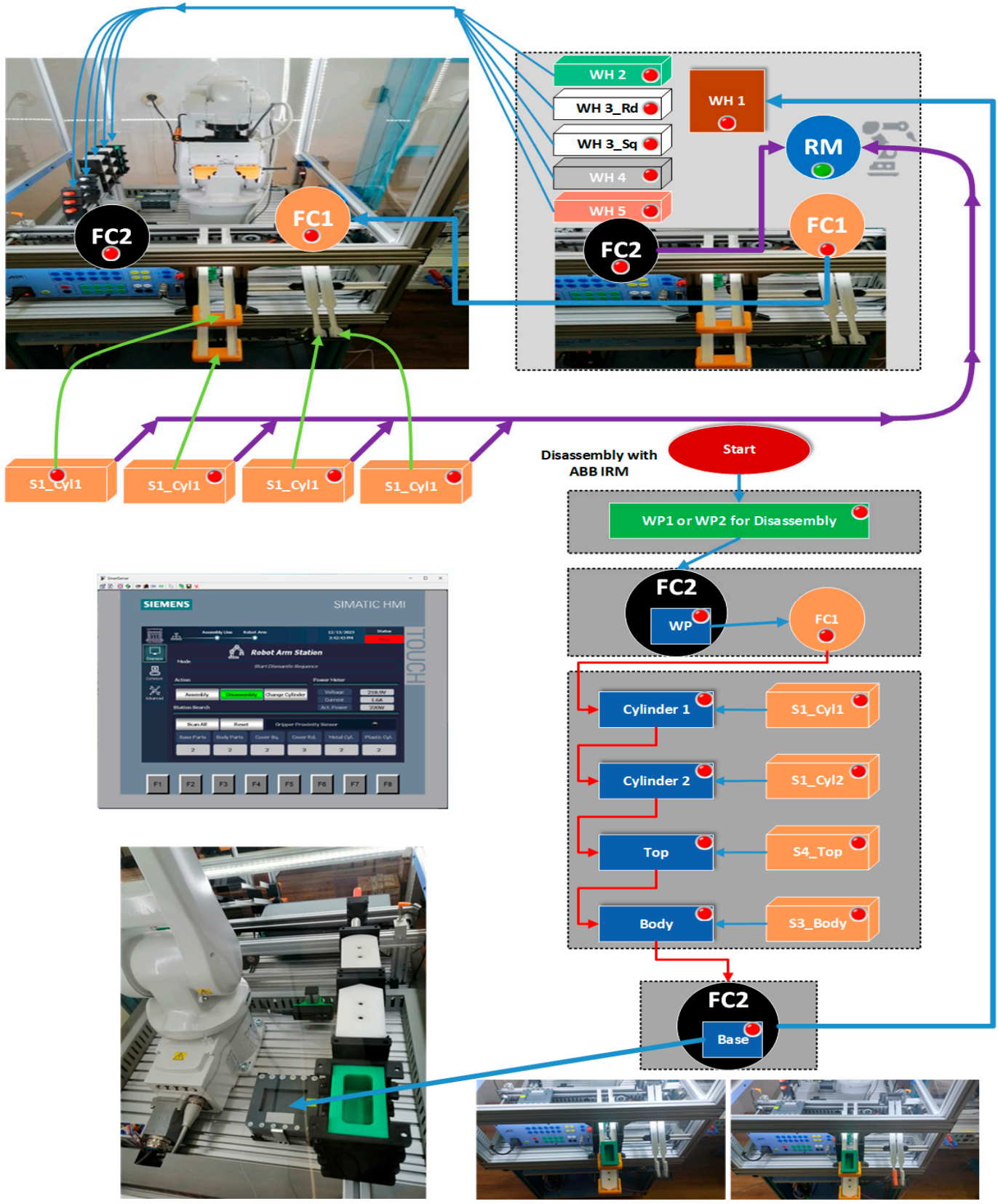


Figure 11. Node-RED task planning for the A/DRC disassembly operation of WP1 or WP2.

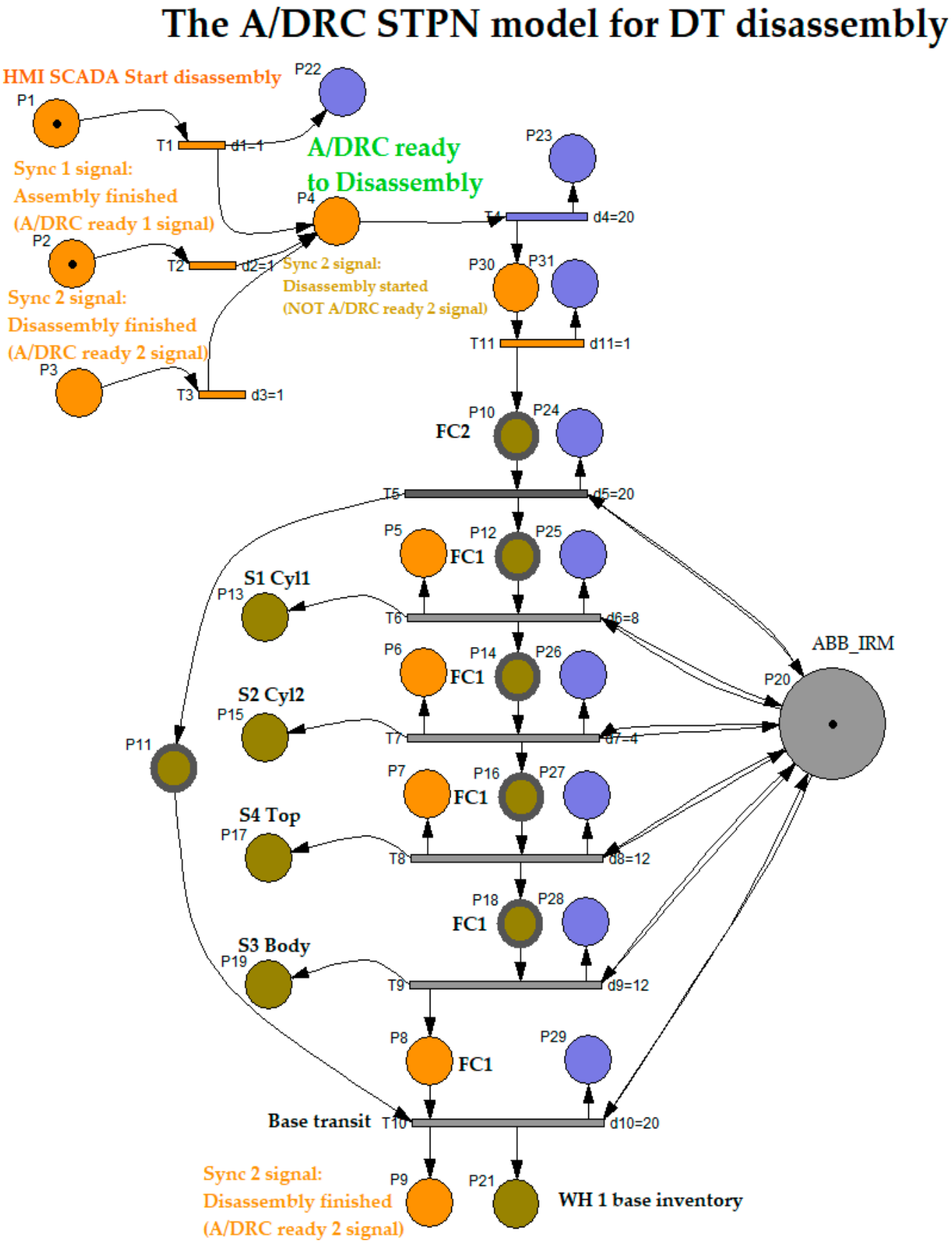


Figure 12. STPN model for the A/DRC disassembly operation of WP1 or WP2.

The STPN model for the disassembly process is defined by the following set:

$$\mathrm{STPN_D} = \left\{\mathrm{TPN_D},\quad \mathrm{E},\quad \mathrm{Sync}\right\}, \tag{64}$$

where TPN is the Timed Petri Net model, *E* is the external events array of transitions, and Sync is an application signals set corresponding to a set of transitions—conditions to be met, defined as external events involved in the disassembly process.

The TPN is a sextuplet:

$$\mathrm{TPN_D} = \left\{\mathrm{P_D},\ \mathrm{T_D},\ \mathrm{Pre_D},\ \mathrm{Post_D},\ \mathrm{m_{0_D}},\ \mathrm{tempo_D}\right\} \tag{65}$$

The elements of the Timed Petri Net model, $\mathrm{TPN_D}$, corresponding to the disassembly function are as follows:

$\mathbf{P_D}$ is the partitioned set of disassembly $\mathrm{TPN_D}$, model places:

$$\mathrm{P_D} = \left\{\mathrm{P_{control_D}}, \mathrm{P_{disassembly_D}}, \mathrm{P_{supervision\ D}}\right\} \tag{66}$$

$\mathbf{P_{control_D}}$ (Orange color places) represents the states of the PLC program disassembly functions control tasks:

$$\mathrm{P_{control_D}} = \left\{\mathrm{P_{i_D}}\right\}_{\mathrm{i_D}=\overline{1,9}} \tag{67}$$

$\mathbf{P_{disassembly_D}}$ (Olive color places) represents the discrete set of states, associated with the functions of the flexible disassembly tasks:

$$\mathrm{P_{disassembly_D}} = \left\{\mathrm{P_{j_D}}\right\}_{\mathrm{j}=\overline{10,21}} \tag{68}$$

$\mathbf{P_{supervision_D}}$ (Blue color places) are the states generated for robotic cell monitoring and supervision of the main tasks during flexible disassembly process for WP1 or WP2:

$$\mathrm{P_{supervision\ D}} = \left\{\mathrm{P_{j_D}}\right\}_{\mathrm{j}=\overline{22,29}} \tag{69}$$

$\mathbf{T_D}$ is the partitioned transition set of assembly for $\mathbf{TPN_D}$**:**

$$\mathrm{T_D} = \left\{\mathrm{T_{CondForDisassembly_D}}, \mathrm{T_{ReadyForDisassembly_D}}, \mathrm{T_{disassembly_D}}, \mathrm{T_{disassembled_D}}\right\} \tag{70}$$

- $\mathbf{T_{CondForDisassembly_D}}$ are the discrete transitions acknowledging that no other operation is running (Assembly or Disassembly).
- $\mathbf{T_{ReadyForDisassembly_D}}$, is the transition for disassembly conditions: SCADA disassembly command is ON, and the cell is Ready (no other operation is running).
- $\mathbf{T_{disassembly_D}}$ are the discrete transitions of flexible assembly tasks.
- $\mathbf{T_{disassembled_D}}$ is the final transition when workpiece subcomponents are dismantled, transported, placed, and stored in their storage units.

$$\mathrm{T}_{CondForDisassembly_D} = \{\mathrm{T_2},\ \mathrm{T_3}\} \tag{71}$$

$$\mathrm{T}_{ReadyForDisassembly_D} = \{\mathrm{T_4}\} \tag{72}$$

$$\mathrm{T_{disassembly_D}} = \left\{\mathrm{T}_{j_D}\right\}_{\mathrm{j_D}=\{\ 1;\overline{5,10}\ \}} \tag{73}$$

$$T_{disassembled_D} = \left\{T_{10_A}\right\} \tag{74}$$

For the robotic cell, WP1 or WP2 disassembly operations, assisted by ABB 120 IRM, the supervision places in the set (69) are delegated to monitor the transitions in the set (70).

Supervision transitions modeled with STPN in Sirphyco, Figure 13, for the disassembly process of WP1 or WP2, are as follows:

- P22, T1 supervision transition is executed when the SCADA disassembly command is triggered;
- P23, T4 supervision transition is executed when the disassembly task starts and conditions are met, meaning the SCADA disassembly command is ON, and the cell is Ready (no other operation is running);
- P24, T5 supervision transition is executed when the workpiece to be dismantled is delivered on transport conveyor FC2, at a specific position to be handled by the ABB 120 IRM;
- P25, T6 supervision transition is executed when the upper part of the workpiece is placed on the Assembly Tray FC1, and RM is extracting the first cylinder;
- P26, T7 transition is carried out when RM is extracting the second cylinder;
- P27, T8 supervision transition is performed when the Top part is picked and placed by RM on the corresponding storage unit;
- P28, T9 supervision transition is executed when Body part is picked and placed by RM on the corresponding storage unit;
- P29, T10 supervision transition is executed when the Base part (Pallet) is transported by the conveyor FC2 to the right-side exit line of the cell, to be externally stored, the disassembly process is finalized, and the Ready signal is set to ON.

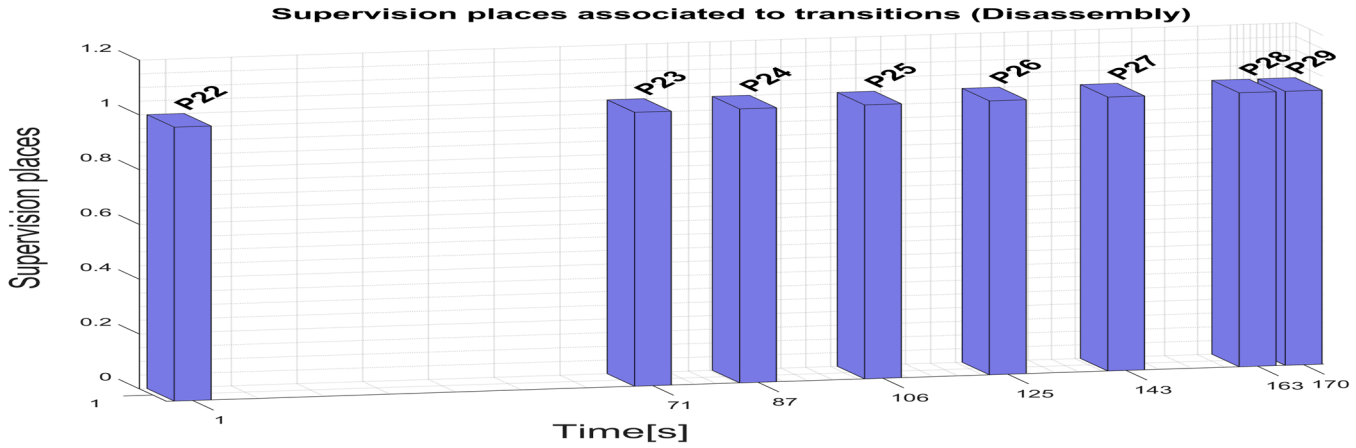


Figure 13. Sirphyco simulation of STPN model for the A/DRC disassembly operation.

$\mathbf{Pre_D}$ is the input incidence function.

$$\mathbf{Pre_D} : P_D \times T_D \rightarrow Q_{+_{Dpre}} \tag{75}$$

$\mathbf{Post_D}$ is the output incidence function.

$$\mathbf{Post_D} : P_D \times T_D \rightarrow Q_{+_{Dpost}} \tag{76}$$

$\mathbf{m_{0_D}}$ is the initial marking set of tokens corresponding to the Disassembly places of the STPN modeled process.

$\mathbf{tempo_D}$ is a function that defines the timings associated with the transitions.

$$\mathbf{tempo_D} : T_D \rightarrow Q_{+_{DT}} \cup \{0\} \tag{77}$$

Same as in the Assembly process, the **Sync** application in definition (64) is a function from the set of discrete transitions corresponding to assembly and disassembly external events:

$$\textbf{Sync} : \{T_2,\ T_3\} \rightarrow \left\{Ev^1,\ Ev^2\right\} \cup \{e\} \tag{78}$$

The synchronization signal that generates a Ready signal when no assembly task is running in the robotic cell:

$$\textbf{Sync 1_D} :\ T_2 \rightarrow \left\{Ev^1\right\} \tag{79}$$

$$Ev^1 = \textbf{Sync 1_D}\ \text{signal} \tag{80}$$

The synchronization signal that generates an A/DRC Ready signal when no disassembly task is running in the robotic cell:

$$\textbf{Sync 2_D} :\ T_3 \rightarrow \left\{Ev^2\right\} \tag{81}$$

$$Ev^2 = \textbf{Sync2_D}\ \text{signal} \tag{82}$$

3.4. Three-Dimensional Virtual Model Implementation Using Siemens NX MCD

In this paper, 3D modeling and dynamics of the A/DRC including robot kinematic motion is developed in Siemens NX MCD [13], starting with the ABB 120 IRM model, developed in CATIA CAD Tool (Figure 14). The rest of the integrated mechatronic equipment is also digitally modeled like the conveyor belt, gripper, storage units for each of the final product components, and the housing of the complete setup. In this 3D model project, we use specifications and models from manufacturer catalogs to accurately replicate and mirror the physical layer. This approach ensures a consistent representation of the machines in the factory, aligning their dimensions, physical properties, and kinematic behavior with the actual mechatronics equipment [32,33]. Three-dimensional model representation, as an important part of the Digital Twin (DT) technology, must be converted to a proper simulation model that interacts with the twins, using specialized software tools, by integrating mechanical and software components into a unified simulation environment [1,2].

These include product lifecycle management (PLM) platforms, such as Siemens NX used in this research, which integrates Siemens Digital Industries Software products, or other 3D factory simulation software solutions for manufacturing and robot offline programming (OLP). MCD is an application inside Siemens NX that allows the simulation and design of kinematic models using external signals based on CAD files. The physics-based 3D model, utilizing Digital Twin (DT) technology, can be controlled and simulated in real-time with the actual PLC through Hardware-in-the-Loop (HIL). This approach enables early verification of the code and identification of potential issues for engineering process control program validation. As shown in Figure 14, the 3D model in NX MCD design starts with creating Solids and, subsequently, a Rigid Body feature is assigned to each part. For that, physical characteristics are assigned as well like weight, center of mass, and inertia, which influence the body dynamics in the simulation. Further Collision Body proprieties are assigned to motion graphical elements, to be able to have physical interaction with the other objects. Machine motions and dynamic interactions between components are applied for each of the 6 robot axes $(\theta_1, \theta_2, \theta_3,\ \theta_4, \theta_5, \theta_6)$. Therefore, NX MCD joint connection implementation is performed, to enable virtual kinematic movements [32], according to linked signals interfaced and calculated by the virtual controller. Furthermore, controls are defined for the physics simulation of the entire machine in the digital space, providing an interface for simulating kinematic model behaviors in response to PLC inputs. NX MCD allows communication protocols with external programs; in this project, the standard OPC UA interface is used for broader connectivity. The 3D virtual interface, NX MCD establishes

communication over OPC UA with PLC. A separate MCD signal should be created for each IO Field variable, which enables the mapping of sensor and actuator signals to virtual components for precise testing.

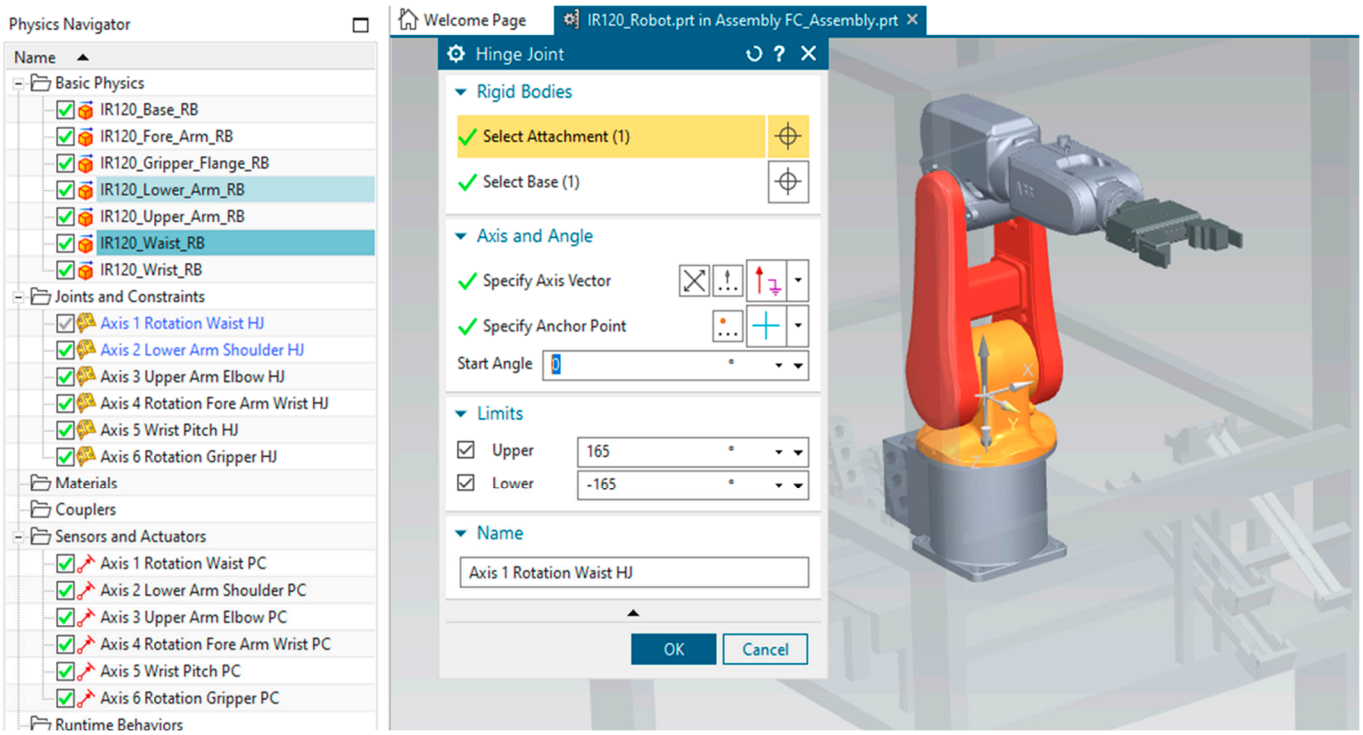


Figure 14. DT 3D virtual model design in NX Mechatronics Concept Designer. Mapping the properties and behavior of virtual components to external signals.

4. Digital Twin Technology

4.1. Multilevel Architecture

The multilevel control structure, features both distributed and centralized topology, all shown in Figure 15, consisting of the following equipment layers:

- Remote operation level, SCADA, and local HMI, for remote control of the A/D devices, for handling operations, and for real-time monitoring and data acquisition [14];
- Communication level handles data transfer and A/D tasks synchronization. SIMATIC IOT2050 has an important role in this hardware structure concept, as a bidirectional data transfer gateway for all edge devices with the Cloud platform [9];
- Control level with PLC S7-1200 which acts as a Central System that controls and synchronizes the overall A/D operations. Moreover, the PLC handles communication and program execution for all devices like ABB IRC5C IRB120 controller, with the Intelligent Siemens Servomotor Drives SINAMICS V90 and NVIDIA Jetson Nano—a small, powerful computer used for embedded applications, especially for Image-system processing solutions. The components detection task is performed by the OVP 800 video processing unit (VPU) with OpenCV-Python 4.4.0.44 open-source library [16];
- Process level, the robotic cell devices like: ABB 120 IRM with electric gripper, conveyor motors, storage, electrical and pneumatic actuators, all IO Field sensors, and HMI KTP 700 as A/DRC local visualization and control system.

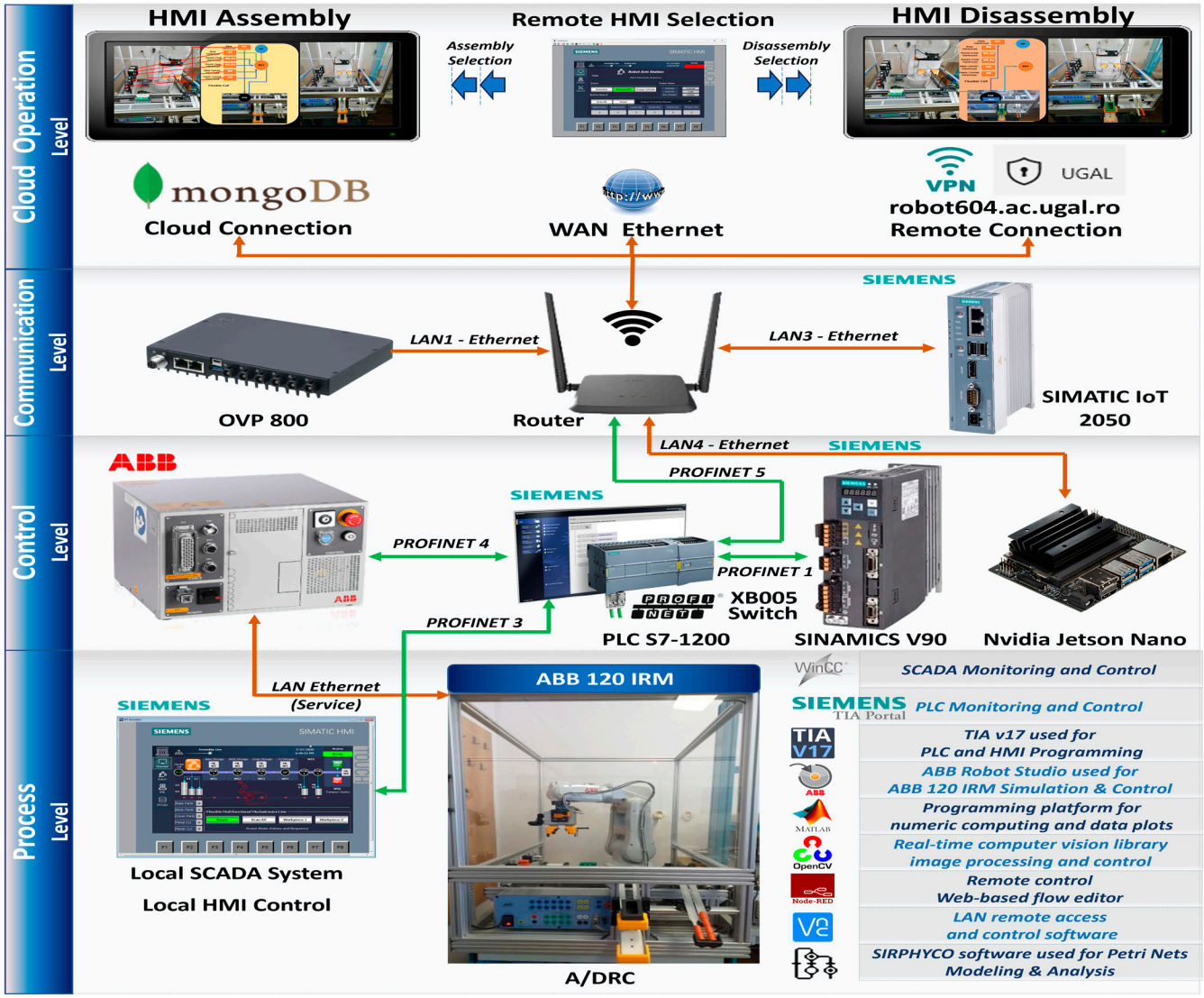


Figure 15. Cloud and VPN-based Control Multilevel Architecture enables hierarchical coordination of system components and organizing processes into multiple layers.

4.2. PLC Software Design

Starting from the main idea of building a virtual replica of the physical process and behavior, a PLC project is designed in the Siemens TIA Portal to simulate, monitor, and optimize its performance before software deployment and onsite commissioning [34]. Moreover, additional features can be added, like integration with IoT devices or Cloud Storage data and History Cloud DT playback for tracking and diagnosing potential events or problems that can arise during normal operation [9]. Therefore, for developing an accurate virtual model, PLC software is mirrored, keeping the same logic and behavior as the physical system, but programmed to communicate via OPC UA with the twins—NX MCD 3D virtual model [13]. OPC UA is one of the most straightforward and efficient protocols for enabling communication between multiple Industry 4.0/5.0 and IoT applications, minimizing the potential for communication issues, in this project, PLC acts as an OPC Server. Additional software tools are integrated for model validation, simulating various scenarios and control strategies and detecting potential issues or software bugs. The physical system I/O handler providing robotic cell status from sensors, actuators, and ABB 120 IRM positions and axis angles are replicated within a separate PLC function block—referred to as a Virtual I/O handler, by evaluating the same control logic and sensor

behaviors. When DT is active, the Virtual I/O handler is the middleware for I/O data sent and received by the virtual model, as seen in Figure 16.

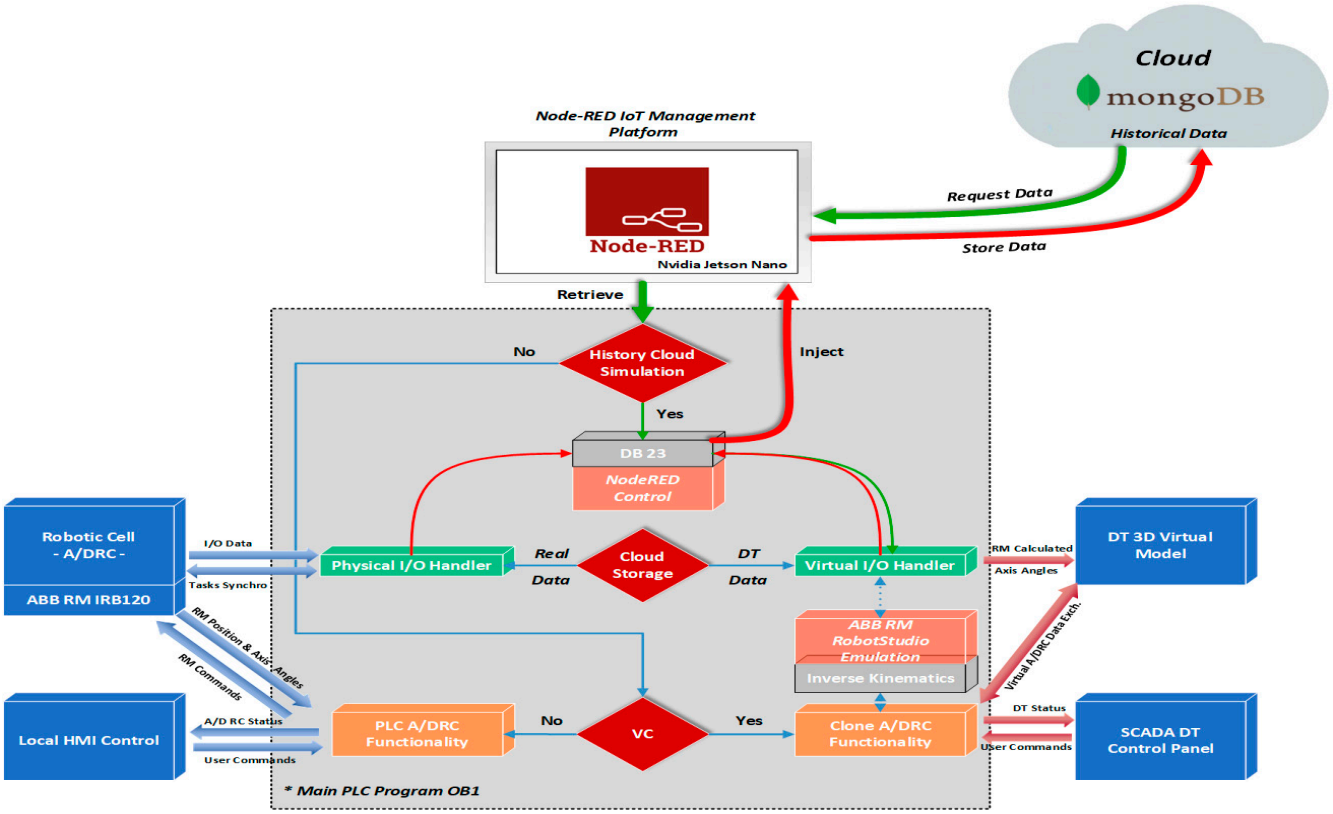


Figure 16. Siemens PLC Main program OB1 functionality chart and structure.

ABB RobotStudio functions [19], dealing with operations of A/DRC are also emulated in the main PLC, generating the designated motion angles and positions through inverse kinematics implementation and sent further to Virtual I/O Handler. The core function that implements the analytical method algorithm for computing the joint values $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ of the ABB 120 IRM is the inverse kinematics function block. Function blocks for inverse kinematics calculation, motion planning, and path generation, along with the software integration, communication with Virtual I/O Handler, and operation within the DT model, are developed in Siemens TIA Portal V17 [34]. The design aims to create a reusable framework applicable to other PLCs. While modern PLCs feature more efficient processors and expanded memory, implementing and designing a 6-DOF robot inverse kinematics calculation remains an uncommon task in PLC programming and engineering. This is because analytical methods for determining joint solutions based on the robot end effector's position require intensive mathematical computations, particularly higher-order matrix operations.

These demands challenges related to PLC-specific issues, such as execution cycle time limitations and expertise level for code optimization. The designed program calculates and returns a single joint angle solution based on the robot arm's position, provided as input from the ABB RobotStudio emulation software module, keeping the standard ABB *robtarget* data type format for the position. Based on the SCADA user input selection, whether it involves visualizing the robot's live parameters or real-time process data, opting to model and simulate various A/D operations with different cycle times, analyzing and comparing performance, or identifying potential issues, the PLC consistently receives the end effector's position as an input. Based on the provided linear speed parameters, a separate function block handles path planning and motion control by segmenting the path between robot

targets and interpolating the positions. The function block responsible for path planning and controlling linear motion (using the MoveL command) from the initial robot pose to the final goal position and orientation is shown in Figure 17. It employs a smooth function, known as Cubic Polynomials, when the desired velocities at intermediate points are non-zero and is an efficient way to interpolate between two points in space, ensuring that both position and velocity constraints are met.

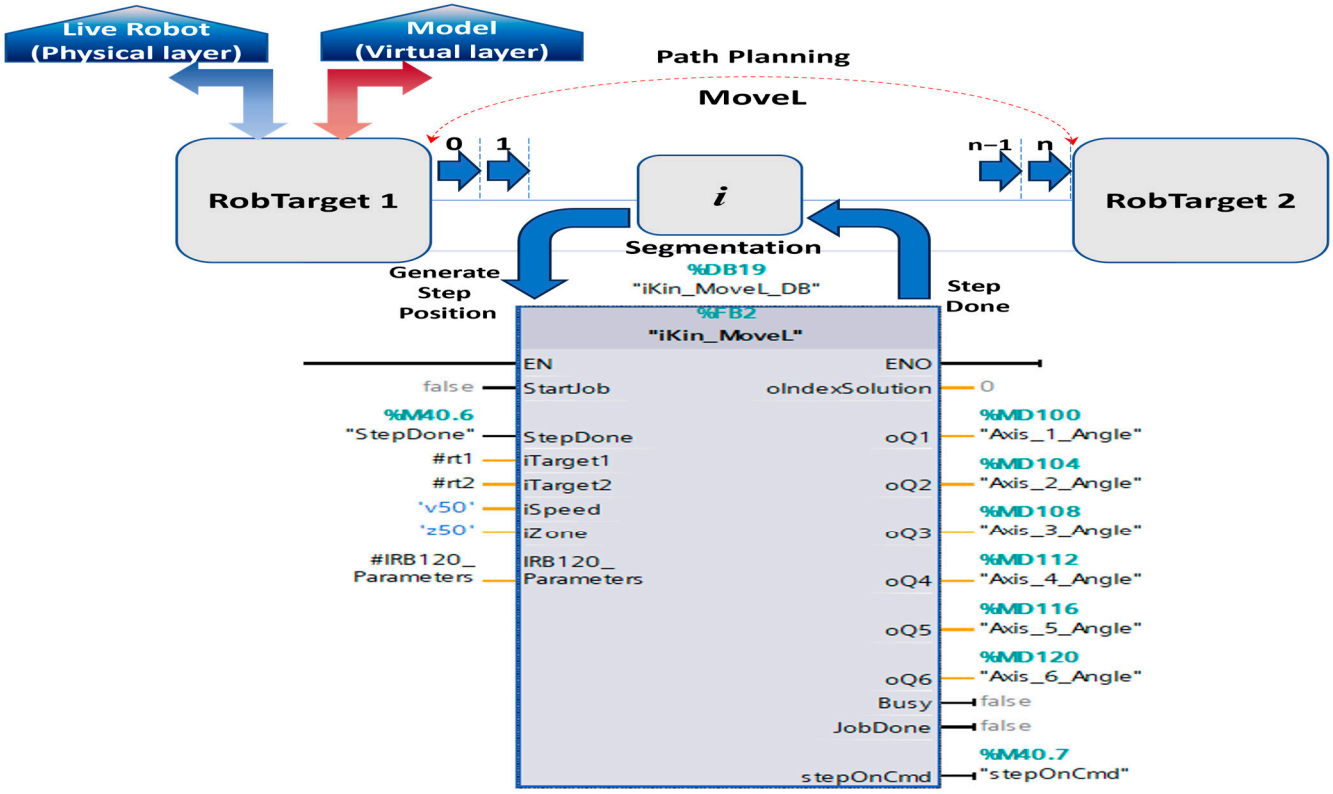
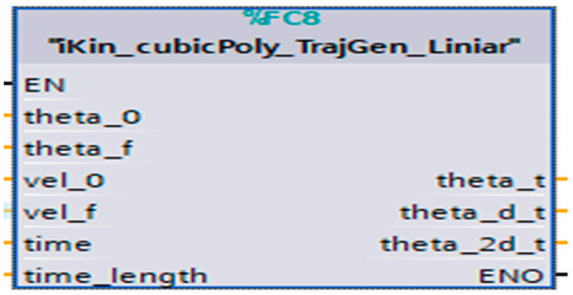


Figure 17. PLC function block design algorithm for MoveL linear motion solution through path segmentation.

Cubic polynomials is a commonly used method in robotics for smooth path planning, particularly in applications requiring continuous motion and velocity profiles (Figure 18). This method ensures efficient and precise control of robotic systems, balancing simplicity with effective performance in many practical scenarios. A separate robot position is generated for each timestamp. i out of n time segments, which are then used as input for each calculation step of the robot joints through the inverse kinematics algorithm. A watchdog was implemented to ensure smooth graphical movement at each timestamp during the robot's motion between two targets.



$$\theta_0 = a_0;\ \dot{\theta}_0 = a_1$$

$$\theta_f = a_0 + a_1 * t_f + a_2 * {t_f}^2 + a_3 * {t_f}^3$$

$$\dot{\theta}_f = a_1 t_f + 2 * a_2 * t_f + 3 * a_3 * {t_f}^2$$

$$a_2 = \frac{3}{{t_f}^2}(\theta_f - \theta_0) - \frac{1}{t_f}(2 * \dot{\theta}_0 - \dot{\theta}_f)$$

$$a_3 = -\frac{2}{{t_f}^3}(\theta_f - \theta_0) + \frac{1}{{t_f}^2}(\dot{\theta}_0 + \dot{\theta}_f)$$

Figure 18. Siemens PLC function block for Trajectory path generation implementing Cubic Polynomials smooth function.

This guarantees stable, real-time communication, and rapid data reception, preventing error handling and unintended manipulator movement [6,29]. Out of the eight generated solutions for the inverse kinematics, a single solution is selected by computing the best fit that complies with the mechanical limitations and physical characteristics of the ABB 120 IRM. This solution is chosen based on the target configuration quadrant for axes 1, 4, and 6, along with the ABB *robtarget* data type.

Three conditional execution controls are implemented in the main PLC program OB1, as an option that determines if the specific instructions or module will be executed, according to operator choice, as seen in Figure 16: History Cloud Simulation, Cloud Storage Source, and Virtual Commissioning. In normal situations, with no user intervention in the cell functionality, the A/D operations are carried out/executed on the physical system. The DT is turned off, and essential robotic cell structured data, such as axis angles, positions, sensor statuses, actuator states, and various process stages, are captured and stored in a JSON-like format in the Cloud. Cloud is used to collect, process, and store data for real-time analysis and long-term storage [9].

In this project setup, the MongoDB Cloud platform is selected for its flexibility, scalability, and ease of use, making it an ideal solution for virtually mapping the entire physical process. The calculated virtual data can be stored in the Cloud, instead of real process data, when the DT system is running, this option is selected from the SCADA DT Panel. By reading the Siemens PLC DB23 process live data, Node-RED is used to store data in MongoDB offering a powerful, flexible, and scalable solution for handling data from both sources [15].

Virtual Commissioning option is also integrated into the system by testing Clone PLC functionality against the real function block programs. Sensors from the physical system are integrated into the digital model allowing real-time data to be used for the simulation. The real physical I/O handler sensor data can be fed into the virtual system, enabling the simulation to reflect real-world sensor behavior. Therefore, the readings from the conveyor and ABB 120 IRM can be compared to the calculated values data in the DT to ensure consistency and accuracy. History Cloud Simulation is used to replay and simulate past scenarios and robotic cell system operations based on real-world data collected over time, often using data stored in the Cloud. The process data, along with the robot's state, axis angles, and positions are logged and can be monitored at each moment during the motion path segmentation. This simulation can be used for various purposes, such as performance analysis, predictive modeling, system optimization, and event failures to better understand process behavior and make more informed decisions for the future.

4.3. DT Communication, Synchronization and Control

For building the 3D virtual interface, to be able to connect the PLC in an HIL configuration, NX MCD establishes communication over OPC UA [13], therefore a Signal Mapping table is required for cross-reference connection with external signals, configured in the OPC UA Server from PLC. The OPC interface supports mapping with the IEC61131-3 language-PLC-specific protocol and offers a wide range of use and connectivity. Moreover, OPC UA is more and more preferred as a standardized protocol to transfer data between different levels of automation. Communication is performed in real-time, and information can be transferred between client and server and from server to server, the minimum update time for OPC is approx. 100 ms. When using a virtual controller in a Hardware-in-the-Loop configuration seen in Figure 19, a physical PLC is needed to synchronize and run the simulation with the same internal clock speed as the rest of the system, avoiding communications latency between the PLC and emulation software [35]. The primary objective of this study is to develop a Digital Twin application by implementing a simulated inverse kinematics

model using MATLAB/Simulink. Subsequently, the data will be manually transferred from the MATLAB workspace to the A/DRC controller (PLC) for further processing and back from the virtual model to MATLAB. For future work, we plan to establish a communication link to adjust model parameters in real-time, transmit various A/DRC state information for processing, and read the updated information back into the system to enhance the simulation's accuracy and adaptability.

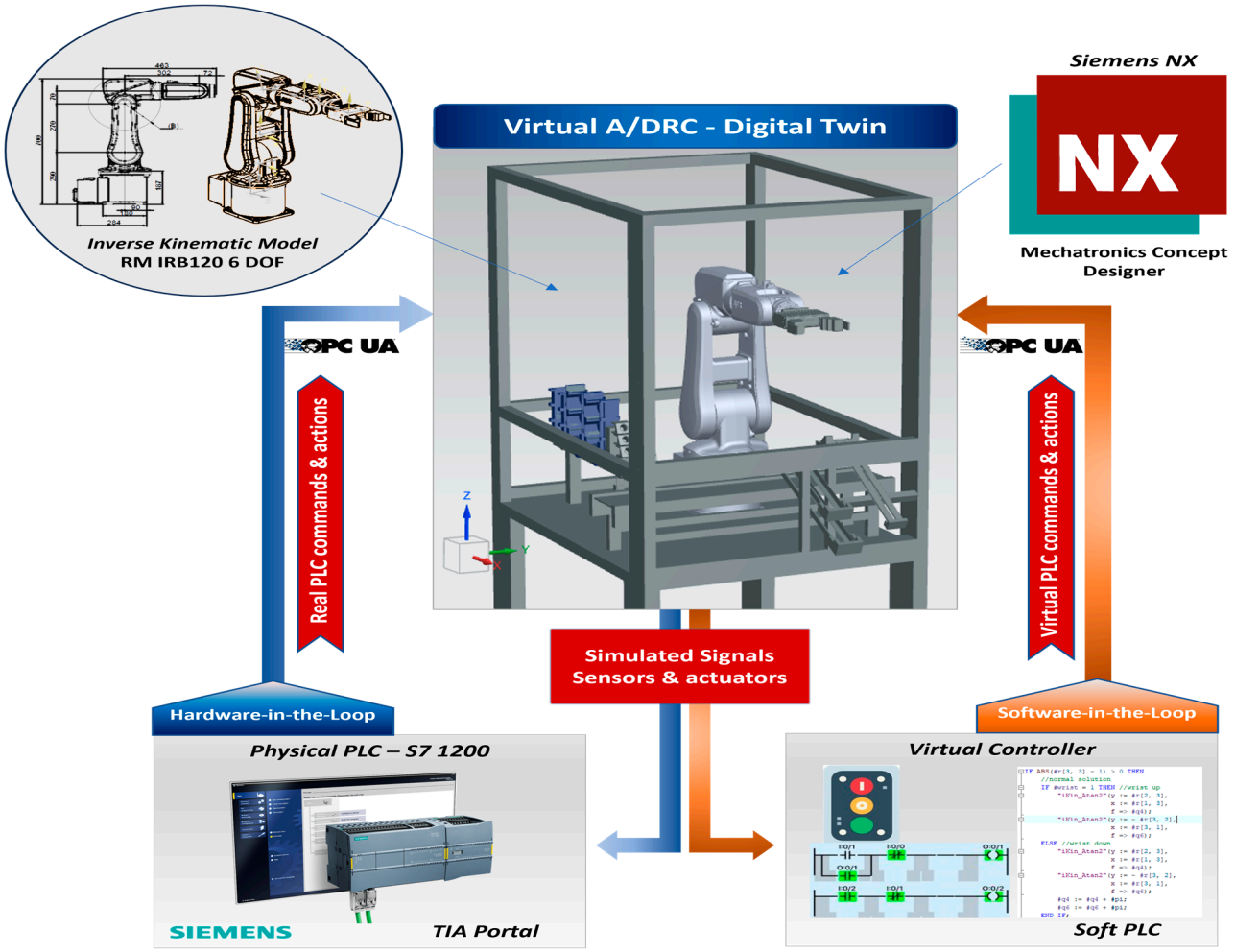


Figure 19. Hardware-in-the-Loop (HIL) Virtual Commissioning Concept.

Further, PLC software design also provides a new approach, besides the logic for the robotic cell and RM automation control—known as the physical layer, it provides communication interface links through PROFINET IO to SCADA and HMI [14]. For linking PLC with the DT, an extra gateway is used, SIMATIC IOT2050, providing IoT support for data to be transmitted to the Cloud and can exchange data between physical and virtual models, providing data transmission and data aggregation, in case is needed. It is well used for converting a range of communication protocols, due to specific features like its capability of Remote accessing I/O's (Digital, Analog) and expanding with different types of smart sensors or Edge Devices [9,36].

4.4. Virtual Commissioning Concept; Hardware-in-the-Loop (HIL) Simulation

Aligning with the current trends in manufacturing, the integration of twins in the engineering process is accelerating mainly due to Industry 4.0/5.0, bringing these "smart" manufacturing systems capabilities into the digital world. As part of DT technology, the VC tool concept and technique is used, offering the potential to minimize the considerable de-

lays and expenses typically involved in the challenging process of classical commissioning, system integration, and operation with (PLC).

The presented manufacturing technology allows the integration of the VC concept by taking advantage of the integrated bi-directional data exchange flow between the dynamic digital representation and the physical system. To connect the A/DRC hardware controllers with the NX MCD 3D model virtual plant, mainly the HIL simulation strategy is adopted, implemented, and performed as a Soft Commissioning approach, by testing the designed PLC programs against the physical PLC (Figure 19). By using the real physical controller, an HIL configuration offers the advantage of delivering the same performance as it would run in actual production by testing the hardware in all situations to ensure it can succeed in all scenarios as programmed. The same happens when there is a big software or hardware change; it ensures that everything still works the way it is intended. As mentioned, a key requirement for testing VC strategy setup is that the virtual factory model must behave exactly as the real model.

In what concern is HIL implementation for virtual plant models to work in tandem with PLC, since simulation has a specific, separate internal clock, this must be synchronized with the real, onsite PLC, for tasks to operate and for time-dependent functions to remain valid. For the implemented DT, specific functions are designed for PLC to be manipulated, synchronized, and run at the same speed as the rest of the system without affecting the test performance. HIL focuses on the PLC software that operates these systems.

Its primary benefit is the ability to identify errors at the initial stages of the factory implementation process, thereby reducing the need for extensive debugging during the physical commissioning phase. A set of PLC function blocks are implemented for the kinematic computation, from MATLAB simulation to PLC code, for digital representation of the complete engineering process.

5. Remote Monitoring and Control

5.1. Digital Twin Control Panel—SCADA

In this research, for VC implementation of the A/DRC physical process, beyond the 3D digital model representation, there is a need for a graphical user interface (GUI) tool. This visualization tool, serving as a DT operator control panel, is essential for interacting with the DT, it refers to a seamless two-way, bidirectional automated data exchange interface between physical and digital objects. It facilitates data monitoring, analysis, and comparison, as well as signaling and processing various events. Additionally, it supports simulation time reporting, user interaction, and setup control. SCADA also allows the users to manage data and control different scenarios or operations within the DT platform, including custom mimics—interactive graphical displays, specially designed for testing the calculation of the inverse kinematics, DT interaction, and process representation based on live and history Cloud data. These mimics enable the user to execute and test various functions in the DT virtual model for centralized control and management of different hypotheses and scenarios. This enables testing and diagnosing potential operational lags or calculation errors and identifying performance issues or slowdowns in the engineering process. As seen in Figure 16, by using the computation resources for PLC code execution (as described in Section 4.2) users can take advantage of the visualized interactive data regarding the inverse kinematics calculation by entering the end effector robot position given as an input, all the eight possible solutions are displayed on the screen. The rotation matrix is visualized in SCADA. Based on the operator's selection, clicking the Simulate button allows the robot's movement calculated set of angles to be visualized on the 3D model within NX MCD. The user can monitor and analyze real-time DT readings, at each step of the robot's movement, starting from the given position to the final target, including linear

translation path segmentation and inverse kinematics calculations (Figure 20). Robotic cell performance parameters can be monitored on this mimic, by setting different linear motion velocities or robot axis speeds, allowing various operations to be tested, including using Live Robot data as part of the Digital Shadow concept within DT technology. If a test scenario meets the required performance criteria, the validated speed parameters can be downloaded by the operator into the PLC. Historical Cloud Data are also presented, and upon request a target event is made available and is entered and sent further for processing in Node-RED for building the necessary Cloud interrogation queries [37,38]. Furthermore, the retrieved data are plotted on the screen. The major timeline spikes are visualized from the 180 seconds History interrogation query time and can be sent for playback in the DT 3D Model representation in NX MCD visualization.

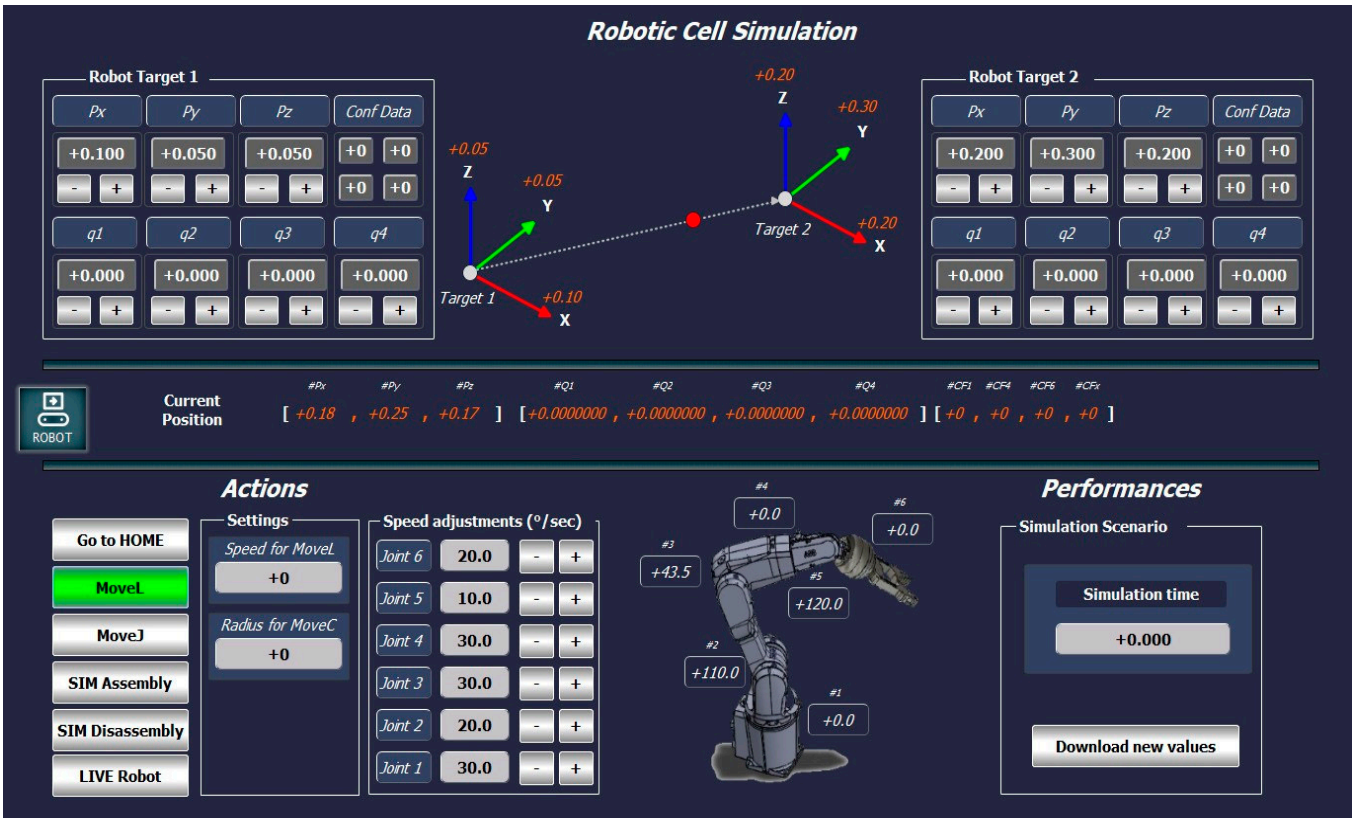


Figure 20. SCADA DT control panel of the A/DRC operations and RM motion control.

5.2. IoT-Cloud-Based Control and Data Acquisition

The purpose of the presented architecture features the IoT Cloud connection for collecting all IO Field sensor data and enabling database storage and analytics for integrating the processed data into the DT application, for virtual historical and real-time process presentation [38]. Using an IoT-Cloud-based DT online monitoring system brings important features for the end user, regarding process information and online remote control of the A/DRC tasks using cutting-edge technology to monitor and detect abnormal behavior and equipment failure using model-driven HIL simulations. Moreover, Cloud storage providers offer robust security features, including encryption, access controls, and compliance with industry standards, especially ISO 27001 [39], regarding information security management systems [1,9,36].

The proposed framework for remote monitoring and control of the A/DRC via the model-driven DT using IoT consists of the following components (Figure 21):

- IoT sensing device;

- Embedded computer NVIDIA Jetson Nano. Acts as a Node-RED IoT interface and Docker Engine client-server application. Remote connection is also available through dedicated VNC-Compatible Free Remote Desktop Software [37];
- ABB IRC5C controller. Executes the user-configured program for the industrial robotic manipulator of the robotic cell and communication to PLC;
- OVP 800 Video Processing Unit. For handling the station HD Cameras, through which the scanning process is performed and the configuration of the workpieces to be assembled is identified [16,17];
- Siemens IOT2050. An IoT gateway device, which enables the bidirectional communication of the PLC with the IoT Cloud platform over a wireless connection;
- Siemens PLC 1214C type. Acts as the main control unit, sends and receives data from all the robotic cell devices including LAN-connected servo drives, and process sensor data, and is responsible for synchronization of the A/DRC tasks. Data are further available for SCADA systems, Cloud storage (MongoDB), and DT applications.

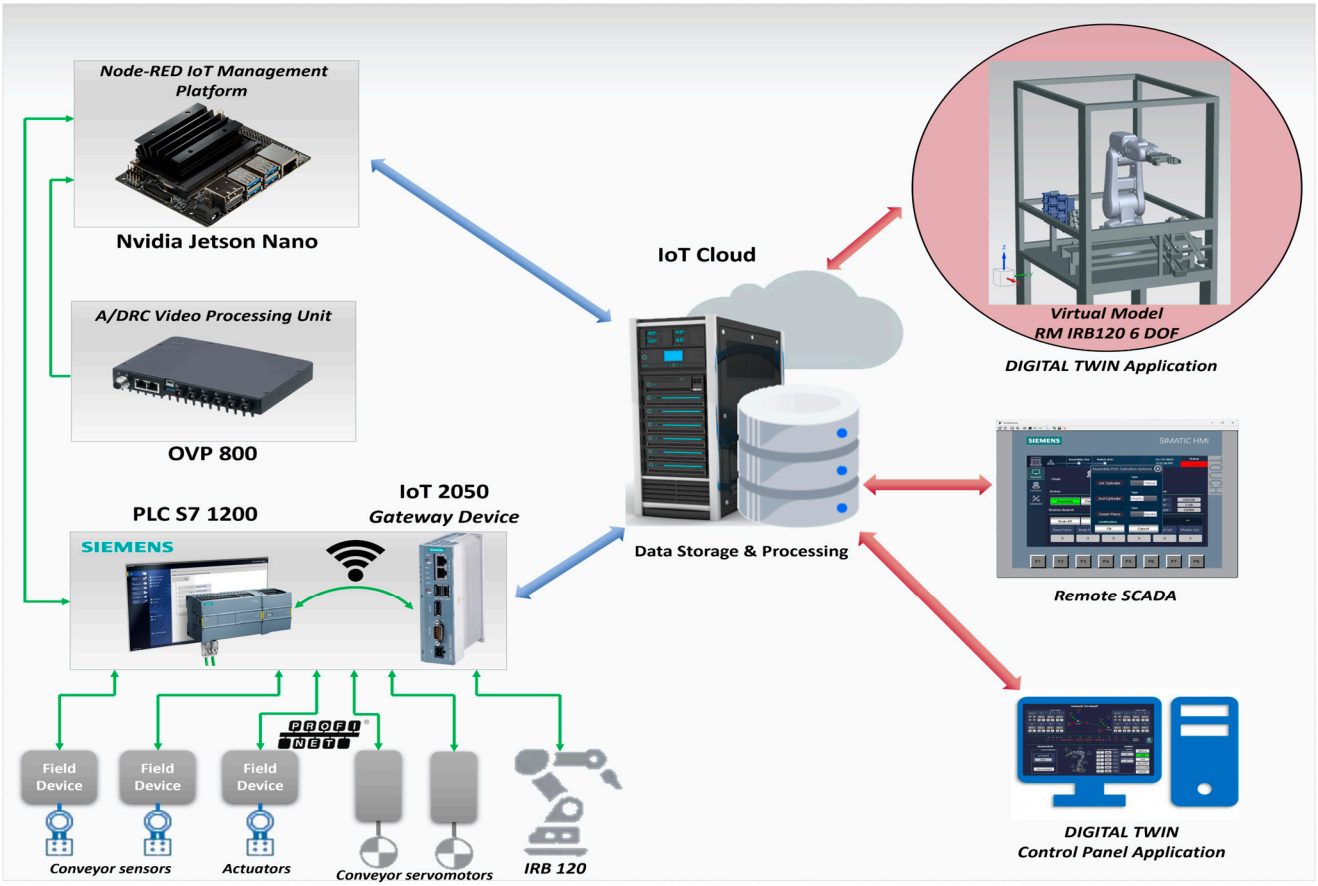


Figure 21. Overview of the general system setup for A/DRC model-driven digital twin (DT) framework using IoT Cloud-based Control and Data acquisition.

Locally, at the operation level, by using the station HMI or on the Remote SCADA application, the user can control and simulate the process A/D operations, as well as monitor all the signals, sensor values, calculated data, and RM motion positions and angles transmitted by PLC on the screens [14]. Additionally, by using the onsite ABB touch pendant, the operator can execute the actions configured by users in the ABB IRC5C controller and program in RobotStudio [19].

The video processing unit, OVP 800, by running ROS2 Humbles camera management, controls the three IFM Cameras video streams through the IFM3D_ROS2 library. It facili-

tates data exchange with the PLC and integration with Node-RED hosted on Embedded computer NVIDIA Jetson Nano which acts as an IoT gateway and Cloud middleware device [15,37] (Figure 22). By integrating the video processing unit connection with the Jetson Nano, OVP 800 acts as an IoT sensing device. The Jetson Nano-embedded computer can send all the image processing data, robotic cell events, and sensor data to the IoT Cloud platform over a wireless Internet network connection for storage and for further vision data processing [16,17]. Siemens IOT2050 is a cutting-edge platform ideal for developing IoT smart applications and automating various industrial processes, designed for efficient IoT data management and communication [36]. In this paper, DT allows the integration with cell devices and sensors, sending/receiving data to/from PLC, enabling real-time data collection, processing, and control via the Cloud platform with enhanced connectivity features (Figure 20).

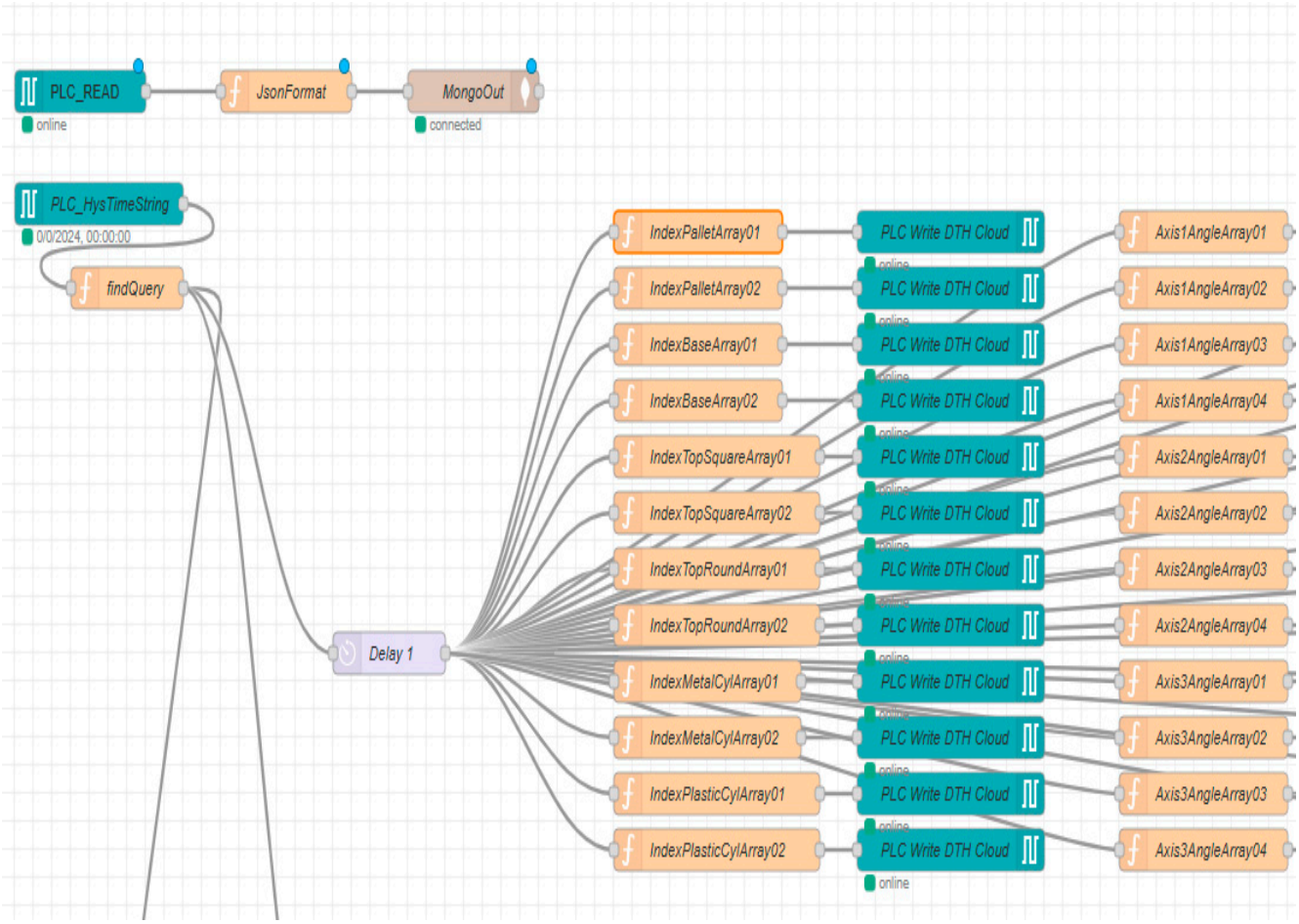


Figure 22. Diagram of the Node-RED and Cloud-PLC flow interaction.

Data collected on the Cloud platform (MongoDB) can be further manipulated for advanced processing, prepared for analysis and presentation, for building different IoT Cloud applications such as remote SCADA for monitoring and Data analysis Server and, moreover, for receiving an earlier robotic cell timeframe data set for playback in DT application [9]. Cloud storage (MongoDB) receives real-time data from A/DRC via Node-RED, capturing cell status parameters essential for retrieving historical event data upon request from the DT. The following live A/DRC parameters are continuously stored: RM joint axis angles, RM current position, storage status (Pallet, Base, Top, Cylinders), RM status, conveyor status, A/DRC status, and emergency stops.

Network connection reliability for Wi-Fi and the Internet has a critical role; therefore, in the event of a Wi-Fi connection failure, IoT capability and functionality for Cloud-based Control and Data acquisition is affected, no data transmission is accomplished, and only DT warning messages and Cloud reporting services will be pushed in the application message viewer.

5.3. Remote HMI VPN Control

The proposed system uses two SCADA applications, one locally and one remote for monitoring and control of the A/D operations on a robotic cell process, equipped with an ABB 120 IRM. The structure and functionality of the entire system are based on a multilevel, decentralized architecture, integrating IoT, smart manufacturing, and intelligent digital technologies, like the VC technique in the context of Industry 4.0/5.0. Therefore, authorized users can safely monitor and control the cell process from anywhere, providing context that will help operators comprehend and predict potential unwanted events by delivering effective situational awareness. Moreover, by implementing and programming the robotic cell 3D virtual model A/D operations and task planning, including the ABB 120 IRM motions, remote SCADA acts as a DT operator panel for process simulation, data collection presentations, and analysis, with remote operation capabilities via Cloud or VPN [9,35,40,41]. The DT platform allows the complete management and control of the A/DRC processes and devices, with the possibility of customizing the simulations according to specific needs. In this paper, an IoT Cloud-based control and remote access application is implemented. Cloud Remote access allows operators to intervene on the robotic cell in a fast and secure way, reducing onsite interventions, and saving time and resources. A third-party Cloud software platform is required, Mongo DB, to interconnect IoT devices, Cloud computing, and storage to dynamically access remotely, for visualization and monitoring in SCADA, for analysis or prediction. MongoDB also provides additional features, like integrating with other Cloud applications or Reporting Services for data visualization [38]. Node-RED is a programming tool for dealing with IoT devices typically used for connecting different nodes, APIs, and online services to build a powerful smart IoT integrated automation system. The Node-RED server running locally on the Jetson Nano IoT device provides Secure Remote access and visualizes the robotic cell stored data on the Cloud. A simple and intuitive operational interface is developed for remote monitoring of A/D operations and controlling all IoT devices, for security configuration, reports, trends, and alarm acknowledgment. Accessing the Node-RED dashboard over the Internet through VPN connections provides an enhanced security environment by using Industry-Standard Encryption like Secure Socket Layer (SSL) and Transport Layer Security (TLS) [36]

6. Robotic Cell Real-Time Results and DT Virtual Model Validation

This research presents the design of an assembly/disassembly robotic cell application integrated with a Digital Twin for remote monitoring and control. It operates within an HIL simulator system, enabling testing, real-time process mirroring and monitoring, predictive maintenance, and remote operation. The system supports various tasks requiring the integration of intelligent digital technologies and Industry 4.0/5.0 concepts, such as Virtual Commissioning (VC), the Internet of Things (IoT), and Cloud computing.

Furthermore, the proposed Digital Twin (DT) serves as a crucial link, a bridge between the real and virtual worlds, enabling seamless interaction and data exchange, allowing monitoring, controlling, and optimizing of the physical processes in real time.

The control structure in the real world, serving as the counterpart to the Digital Twin, features a hierarchical yet multilevel and decentralized architecture. It is designed for supervising, monitoring, and controlling processes, executing operations, and planning

assembly/disassembly (A/D) tasks. By continuously synchronizing with its physical counterpart through sensors and data analytics, allows the optimization and flexibility of A/D operations, predictability, and process efficiency.

In this paperwork, STPN modeling and simulation of A/D operations are integrated into a real-time application through SCADA and DT HIL platform. This is achieved by interfacing them with synchronized signals collected from the real process through PLCs and sensors. To evaluate and validate the DT model environment against the A/DRC real mechatronic process assisted by ABB 120 IRM, several tests and analyses were performed.

After implementation, the results of real-time control are presented and compared with the monitoring signals obtained from the DT IRM, which emulates movements specific to assembly and disassembly operations (Figure 23).

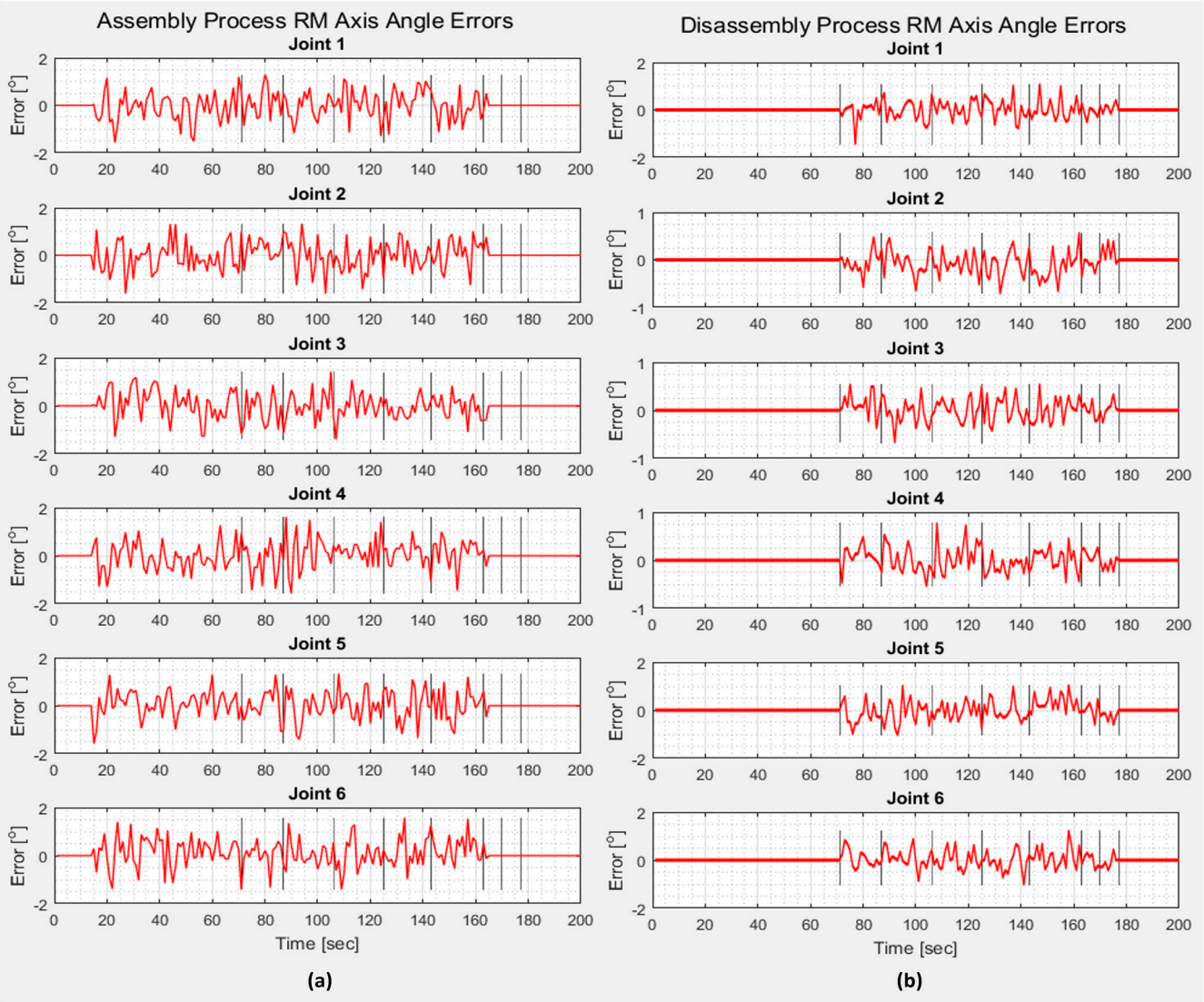


Figure 23. RM axis angles errors comparison for A/DRC process tasks according to STPN model transitions for physical system and DT virtual model: (**a**) A/DRC assembly operation; (**b**) A/DRC disassembly operation.

A/DRC assembly/disassembly operations involve repetitive tasks of IRM picking and placing subcomponents from the storage units, for handling and further transporting the workpiece on the A/DRC conveyor, all this with the corresponding time durations. For each IRM task, a MoveL motion command is performed, a linear movement from an initial position to a desired final position, a command specific to the ABB RobotStudio hardware

controller. A function that emulates linear movements is also designed, implemented, and tested for DT IRM. By comparing the results for a single MoveL motion command from home to a desired position, the proposed virtual model emulated motion demonstrates high accuracy in path following, closely matching the real world. It also provides a reliable approximation for robot tracking, positioning, and joint axis angles.

These movements are simulated as well in Sirphyco, reflecting the discrete states of the corresponding STPN models. Comparing the results from the Sirphyco simulation of STPN models, slight differences were identified between the simulated data and the monitoring signals provided by PLC, which correspond to the real-time execution of control programs. Thus, the virtual digital component of the robotic cell manufacturing technology is validated in real time.

Following digital model development and running simulations, real-time implementations are integrated within the digital environment for validation and testing, where the results from the physical system are cross-referenced with simulation outputs (Figure 23). DT model slight differences were found, originating from PLC inverse kinematics algorithm calculations, especially from rounding the huge expressions of the rotation matrices and conversions, as well as some communication delays generated from data communication links, but these errors tend slowly to zero and do not affect the overall setup performance or bring behavior difference (Figure 23).

The robot's virtual model and behavior were able to accurately replicate and follow with a high degree of reliability the A/DRC real mechatronic process, assisted by the ABB 120 IRM, to be as close as possible to the real industrial world.

7. Discussion

This research paper was undertaken to build an HIL DT framework with Cloud solutions and Remote VPN capabilities for a robotic cell assisted by ABB 120 IRM, based on a PLC implementation for an RM inverse kinematic virtual model.

The Digital Twin model is used to virtually replicate the physical robotic cell structure and represents an approximation of the real behavior of the A/DRC manufacturing process, enabling real-time monitoring, simulation, and optimization of processes for improved efficiency and performance.

Although the A/DRC functionality is designed for a specific manufacturing process, handling multiple assembly and disassembly operations for a predefined five-component workpiece, the PLC-based hardware and software DT design architecture is flexible and can be easily adapted to more complex manufacturing processes. As previously mentioned, the fundamental design concept of the hardware and software HIL DT design architecture integrates three major subsystems, which can operate, communicate, and synchronize with each other: PLC, SCADA, and DT 3D virtualization of the process using Siemens NX MCD.

The S7-1200 PLC is programmed using the Siemens TIA Portal to control both the physical and virtual A/DRC models within an HIL structure. If, at the real-time operation level, the automation control of the A/DRC manufacturing process is designed for a specific set of tasks—modeled and simulated using STPN models—the NX MCD virtual plant model implementation remains easily adaptable to a different robot manipulator or even an entirely different assembly/disassembly robotic cell process. The PLC program is designed for the kinematics of a six-axis industrial robot manipulator, with analytical methods for inverse kinematics calculation and implementation structured as modular PLC functions. These functions are generally applicable to any six-DOF robot manipulator by specifying its product details and characteristics. In this research, within the mechatronic laboratory robotic cell, the A/D tasks are executed by the ABB 120 IRM, with its specifications assigned as internal PLC variables.

Remote and local SCADA systems serve as the operator control panel for interacting with the Twins, integrating key functions such as real-time monitoring, control, and visualization of A/D tasks, as well as managing data and controlling different scenarios or operations for the DT with the HIL platform. An intuitive and interactive operational interface is developed in SCADA, primarily used for testing the calculation of the six-DOF robot manipulator's inverse kinematics. The remaining graphical displays are dedicated to DT interaction, providing a live representation of A/DRC runtime data. These displays provide also support for Cloud data collection, presentation, and analysis through Remote VPN, integrating IoT for dynamic remote access, analysis, or prediction. The SCADA environment used in this project as the DT Operator Panel is highly flexible and can be configured to suit a variety of industrial applications and manufacturing processes. Additionally, it supports a wide range of communication protocols such as PROFINET IO, OPC UA, and MQTT, enabling connections with various third-party hardware and software. While it is currently implemented for a single robotic cell with a six-DOF RM, it can be easily deployed, modified, and adapted for large-scale operations involving multiple robotic cells in a variety of industrial applications.

The DT environment uses Node-RED, a powerful and highly adaptable open-source, flow-based development tool for smart IoT-integrated automation systems. It runs on the Jetson Nano IoT device, enabling secure remote access and visualization of the robotic cell's stored data in the Cloud. Moreover, Cloud provides global scalability, allowing access to data from any location worldwide.

It is important to underline that developing and employing a DT application brings also limitations to enhancing operational efficiency. We will generally address the DT limitations, but we will primarily focus on the potential limitations of the proposed approach:

- **DT virtual model accuracy and fidelity.** A high level of accuracy in the DT model is crucial for organizations aiming to optimize productivity and operational performance. Virtual models need some more extensively validated dynamics and response validation and task planning algorithms to accurately represent the physical counterpart. The proposed A/DRC DT virtual model implementation is based on the RM inverse kinematics model and not on the internal robot's dynamics, which adds a limitation to the level of RM model accuracy;
- **HIL DT extensive development time.** It is important to underline the fact that DT implementation, interaction, and 3D virtual model design during product development require significant time, resulting in longer development timelines. Therefore, DT system implementation can have an expensive initial cost in designing and deploying into the real engineering process;
- **RM inverse kinematics analytical solution is limited to simple robotic cell designs.** This paper proposes an analytical solution to the inverse kinematics problem of a 6-DOF RM using a geometric approach that utilizes trigonometric relationships and vector algebra to derive joint angles. This method is highly efficient and accurate, providing precise joint configurations without approximation, making it suitable for real-time applications where calculation speed is crucial. However, it has some limitations, as it does not apply to more complex structures and configurations and can be sensitive to singularities;
- **Reliability and data quality.** Integrating the IoT Cloud-based data for DT technology in a laboratory system faces challenges and concerns about the quality of the data. Robotic cells generate vast amounts of real-time data; therefore, streams from IoT devices require advanced storage and processing capabilities. Reliability and data quality for faster automation processes and DT communication interfaces are vital, which restricts their performance and prediction accuracy in real-world applications.

Low-latency communication between the physical robotic cell and its twin is crucial for real-time applications, and Cloud connection speed rate and interruptions can lead to inconsistent data, especially when using IoT and History Cloud. Data security policies, data verification, checksums, and encryption techniques are not a concern in the A/DRC laboratory setup. However, these procedures should be implemented in real-world manufacturing processes, where data collection and transmission depend on robust IoT ecosystems and face real challenges related to data complexity and cybersecurity threats.

New directions for future research and development have arisen, to overcome the above-mentioned limitations. Several key areas require further development in future research:

- **Integrating Intelligent Digital Technologies**, such as Artificial Intelligence and Machine Learning, along with enhanced integration with IoT ecosystems;
- **Advancements in system optimization and digital twin efficiency**. The development of more efficient Digital Twin implementations and algorithms for virtual model dynamics and response validation enhances accuracy, real-time performance, and system reliability in industrial applications;
- **Addressing the challenges in Industry 4.0/5.0**, such as the integration of collaborative robots into manufacturing systems, particularly within robotic cells, and their connection with DT technology. This is an area that requires further attention, designing robotic cells to ensure safe collaboration with human workers requires additional safety protocols, sensors, and programming;
- **Enhancing/Increasing complexity in robotic cell Integration with Digital Twins**. Adding layers of complexity in terms of data collection, processing, and response of the robotic cell requires advanced design, careful integration, and more sophisticated control systems to ensure the accuracy of DT models, efficiency, and reliability. Moreover, the complexity of a robotic cell in its design, operation, and integration with DT should be addressed, starting from adding multiple robots or adding extra tasks into a single robotic cell integrating more sophisticated control algorithms, and machine learning approaches, and combining with the implementation of VC techniques in this setup.

8. Conclusions

In this paper, we propose an HIL DT application of A/DRC, assisted by the ABB 120 IRM, with the following key contributions:

- **Multilevel architecture, hardware, and software setup;**
- **Addressing potential limitations of the HIL DT laboratory setup.** These limitations are related to the accuracy and fidelity of the virtual model, which impacts DT efficiency and production performance but can be improved through process optimization and predictive maintenance;
- **A/DRC STPN simulation for validation of model and control strategy based on task planning integration and synchronization**. This hybrid modeling approach ensures real-time adaptability and flexibility in production workflows, facilitating task synchronization and enhancing the system's responsiveness to production requests and actions;
- **3D modeling using NX MCD and virtual model PLC implementation based on an inverse kinematics analytical approach**. In robotic manufacturing, model kinematic equations must be solved at a high rate for PLC execution; therefore, a closed-form solution is preferred over a numerical one;

- **SCADA DT design and functionality**. SCADA enhances DT by enabling monitoring, visualization, and Cloud Storage. It serves as a DT operator panel for simulation, data analysis, and remote operation. Additionally, it diagnoses and prevents problems, alerts operators, and improves A/DRC system reliability while reducing production costs and safety risks.
- **A/DRC Virtual Commissioning platform implementation**. Developing and simulating a DT model for VC enhances engineering, optimizes design process, and enables smarter testing of physical systems. Combined with HIL, IoT, and Cloud solutions, DT improves manufacturing by enabling precise control, remote monitoring, and prediction of physical processes;
- **Configuration and integration of protocols for data synchronization and validation**. Reliable DT communication is crucial for performance and accuracy. To address these challenges, clear protocols such as PROFINET IO, Ethernet, and WAN-Ethernet with OPC-UA are implemented to ensure stable, secure communication between local and Cloud systems, enabling efficient remote control and monitoring;
- **IoT-Cloud-based Control and data acquisition**. The IoT-Cloud platform enables real-time monitoring, remote access, data analytics, and cloud storage for A/DRC. Additionally, Cloud and Edge computing should be used to process data closer to the source, reducing latency, enhancing decision-making, and ensuring data security. Implementing these strategies improves DT applications and data quality, enhancing reliability, and automation.

This research work aims to evaluate whether intelligent digital technologies integrated into manufacturing applications, especially in Industrial robotics systems, like Virtual Commissioning, should be included in their suite of tools for testing the programs under various scenarios. Additionally, integrating a digital model with Cloud storage solutions can easily accommodate growing data requirements, enabling real-time data collection and monitoring of robotic cells equipped with IoT sensors. This facilitates faster decision making, quicker troubleshooting of potential issues, and a significant reduction in downtime associated with maintenance tasks. This includes optimizing operational tasks, addressing delays, enabling predictive maintenance, and reducing costs related to on-site commissioning.

The research presented is still ongoing, with the main objective of adapting the A/DRC assisted by ABB 120 IRM technology, currently implemented on a laboratory system, to align with the rapidly evolving developments in Industry 4.0 and 5.0. Through this study, we extended the degree of automation and efficiency by introducing smart manufacturing technologies, such as IoT and digitalization, into the manufacturing process. As a result, this approach leads to improved accuracy, predictability, robustness, and overall performance.

Although robotic cell Digital Twin technology with the HIL approach has been used and tested at the level of a laboratory, it can be extended to ensure its applicability to real-world industrial systems and various applications, ultimately enhancing operational efficiency and boosting productivity. The hardware and software configuration for the robotic cell Digital Twin approach with an HIL structure enhances robustness against defects and uncertainties across all subsystems, ensuring adaptability and flexibility-key characteristics of Industries 4.0 and 5.0.

Author Contributions: Conceptualization, all authors; methodology: A.F. (Adrian Filipescu), D.I., A.F. (Adriana Filipescu), and G.S.; software: G.S. and D.I.; validation: A.F. (Adrian Filipescu); formal analysis: G.S., D.I., and A.F. (Adriana Filipescu); writing—original draft preparation: A.F. (Adrian Filipescu) and A.F. (Adriana Filipescu); writing—review and editing: A.F. (Adrian Filipescu) and A.F. (Adrian Filipescu); project administration: A.F. (Adrian Filipescu); funding acquisition, A.F. (Adrian Filipescu). All authors have read and agreed to the published version of the manuscript.

Funding: This article (APC) will be supported by "Dunarea de Jos" University of Galati.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ma, X.; Tao, F.; Zhang, M.; Wang, T.; Zuo, Y. Digital twin enhanced human-machine interaction in product lifecycle. *Procedia CIRP* **2019**, *83*, 789–793. [CrossRef]
2. Botín-Sanabria, D.M.; Mihaita, A.-S.; Peimbert-García, R.E.; Ramírez-Moreno, M.A.; Ramírez-Mendoza, R.A.; Lozoya-Santos, J.d.J. Digital Twin Technology Challenges and Applications: A Comprehensive Review. *Remote Sens.* **2022**, *14*, 1335. [CrossRef]
3. Florescu, A. Digital Twin for Flexible Manufacturing Systems and Optimization Through Simulation: A Case Study. *Machines* **2024**, *12*, 785. [CrossRef]
4. Werner, K.; Matthias, K.; Georg, T.; Jan, H.; Wilfried, S. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-Pap.* **2018**, *51*, 1016–1022. [CrossRef]
5. Elisa, N.; Luca, F.; Marco, M. A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manuf.* **2017**, *11*, 939–948. [CrossRef]
6. David, G.-Z.; Alex, B.; Gilberto, E.; Arthur, S. Digital Twin in a Manufacturing Integrated System: Siemens TIA and PLM Case Study. In Proceedings of the ASME 2019 International Mechanical Engineering Congress and Exposition, Salt Lake City, UT, USA, 11–14 November 2019. [CrossRef]
7. Mincă, E.; Filipescu, A.; Cernega, D.; Șolea, R.; Filipescu, A.; Ionescu, D.; Simion, G. Digital Twin for a Multifunctional Technology of Flexible Assembly on a Mechatronics Line with Integrated Robotic Systems and Mobile Visual Sensor—Challenges towards Industry 5.0. *Sensors* **2022**, *22*, 8153. [CrossRef] [PubMed]
8. Guerrero, L.V.; López, V.V.; Mejía, J.E. Virtual Commissioning with Process Simulation (Tecnomatix). *Comput. Aided Des. Appl.* **2014**, *11* (Suppl. S1), S11–S19. [CrossRef]
9. Saha, O.; Dasgupta, P. A Comprehensive Survey of Recent Trends in Cloud Robotics Architectures and Applications. *Robotics* **2018**, *7*, 47. [CrossRef]
10. Filipescu, A.; Simion, G.; Ionescu, D.; Filipescu, A. IoT-Cloud, VPN, and Digital Twin-Based Remote Monitoring and Control of a Multifunctional Robotic Cell in the Context of AI, Industry, and Education 4.0 and 5.0. *Sensors* **2024**, *24*, 7451. [CrossRef]
11. Konstantinov, S.; Assad, F.; Ahmad, B.; Vera, D.A.; Harrison, R. Virtual Engineering and Commissioning to Support the Lifecycle of a Manufacturing Assembly System. *Machines* **2022**, *10*, 939. [CrossRef]
12. Filipescu, A.; Ionescu, D.; Filipescu, A.; Mincă, E.; Simion, G. Multifunctional Technology of Flexible Manufacturing on a Mechatronics Line with IRM and CAS, Ready for Industry 4.0. *Processes* **2021**, *9*, 864. [CrossRef]
13. Siemens NX Mechatronics Concept Designer (MCD). Available online: https://plm.sw.siemens.com/en-US/nx/cad-online/ (accessed on 20 January 2025).
14. SIMATIC SCADA Systems. Available online: https://www.siemens.com/global/en/products/automation/industry-software/automation-software/scada.html (accessed on 20 January 2025).
15. Node-RED Web-Based Flow Editor for Remote Control, Node-RED 4.0 Released—News—Node-RED Forum (nodered.org). Available online: https://discourse.nodered.org/t/node-red-4-0-released/88815 (accessed on 20 January 2025).
16. OpenCV. Available online: https://opencv.org (accessed on 20 January 2025).
17. Microsoft Visual Studio. Available online: https://www.visualstudio.com/vs/cplusplus (accessed on 20 January 2025).
18. Hadi Barhaghtalab, M.; Meigoli, V.; Golbahar Haghighi, M.R.; Nayeri, S.A.; Ebrahimi, A. Dynamic analysis, simulation, and control of a 6-DOF IRB-120 robot manipulator using sliding mode control and boundary layer method. *J. Cent. S. Univ.* **2018**, *25*, 2219–2244. [CrossRef]
19. ABB RobotStudio Desktop Robotics. Available online: https://new.abb.com/products/robotics/software-and-digital/robotstudio/robotstudio-desktop (accessed on 20 January 2025).
20. Manocha, D.; Canny, J.F. Efficient inverse kinematics for general 6R manipulators. *IEEE Trans. Robot. Autom.* **1994**, *10*, 648–657. [CrossRef]
21. Lee, C.S.G.; Ziegler, M. Geometric approach in solving inverse kinematics of PUMA robots. *IEEE Trans. Aerosp. Electron. Syst.* **1984**, *AES-20*, 6. [CrossRef]
22. MathWorks. Available online: https://www.mathworks.com (accessed on 20 January 2025).
23. Chapelle, F.; Bidaud, P. Closed form solutions for inverse kinematics approximation of general 6R manipulators. *Mech. Mach. Theory* **2004**, *39*, 323–338. [CrossRef]

24. Ionescu, D.; Filipescu, A.; Simion, G.; Șolea, R.; Filipescu, A.; Șerbencu, A. PLC Inverse Kinematics Model-Driven Digital Twin Focused on HIL for a Flexible Robotic Cell. In Proceedings of the 18th International Conference on Control, Automation, Robotics and Vision (ICARCV), Dubai, United Arab Emirates, 12–15 December 2024; pp. 1195–1200. [CrossRef]
25. Hartenberg, R.S.; Denavit, J. A kinematic notation for lower pair mechanisms based on matrices. *J. Appl. Mech.* **1955**, *77*, 215–221. [CrossRef]
26. Rocha, C.; Tonetto, C.; Dias, A. A comparison between the Denavit-Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. *Robot. Comput. Integr. Manuf.* **2011**, *27*, 723–728. [CrossRef]
27. Seven, O.F.; Ankarali, A. Inverse kinematics analysis of IRB120 robot arm. *SETSCI Proc. ISAS 19* **2019**, *4*, 383–390.
28. Nicolescu, A.F.; Ilie, F.M.; Alexandru, T.G. Forward and inverse kinematics study of industrial robots taking into account constructive and functional parameters modeling. *Proc. Manuf. Syst.* **2015**, *10*, 157–164.
29. Almaged, M. Forward and Inverse Kinematic Analysis and Validation of the ABB IRB 140 Industrial Robot. *Int. J. Electron. Mech. Mechatron. Eng.* **2017**, *7*, 1383–1401. [CrossRef]
30. Ionescu, D.; Filipescu, A.; Simion, G.; Mincă, E.; Cernega, D.; Șolea, R.; Filipescu, A. Communication and Control of an Assembly, Disassembly and Repair Flexible Manufacturing Technology on a Mechatronics Line Assisted by an Autonomous Robotic System. *Inventions* **2022**, *7*, 43. [CrossRef]
31. Sirphyco Simulateur de Réseaux de Petri, Sirphyco-Simulateur-de-Reseaux-de-Petri. Available online: https://www.toucharger.com (accessed on 20 January 2025).
32. Abderrahim, B.; Moulay El, E.-C.; Hicham, A.; Hassan, S. The Inverse Kinematics Evaluation of 6-DOF Robots in Cooperative Tasks Using Virtual Modeling Design and Artificial Intelligence Tools. *Int. J. Robot. Res.* **2023**, *12*, 121–130. [CrossRef]
33. Schluse, M.; Rossmann, J. From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems. In Proceedings of the 2016 IEEE International Symposium on Systems Engineering (ISSE), Edinburgh, UK, 3–5 October 2016; pp. 1–6. [CrossRef]
34. Totally Integrated Automation Portal, TIA Portal V17. Available online: https://www.siemens.com/tia-portal (accessed on 20 January 2025).
35. Pacheco-Gutierrez, S.; Niu, H.; Caliskanelli, I.; Skilton, R. A Multiple Level-of-Detail 3D Data Transmission Approach for Low-Latency Remote Visualisation in Teleoperation Tasks. *Robotics* **2021**, *10*, 89. [CrossRef]
36. Roy, S.; Vo, T.; Hernandez, S.; Lehrmann, A.; Ali, A.; Kalafatis, S. IoT Security and Computation Management on a Multi-Robot System for Rescue Operations Based on a Cloud Framework. *Sensors* **2022**, *22*, 5569. [CrossRef] [PubMed]
37. Nițulescu, I.-V.; Korodi, A. Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions. *IoT* **2020**, *1*, 76–91. [CrossRef]
38. Omidi, S.A.; Baig, M.J.A.; Iqbal, M.T. Design and Implementation of Node-Red Based Open-Source SCADA Architecture for a Hybrid Power System. *Energies* **2023**, *16*, 2092. [CrossRef]
39. ISO/IEC. "ISO/IEC 27001:2022". Available online: https://www.iso.org/standard/27001 (accessed on 20 January 2025).
40. Arenas-Rosales, F.; Martell-Chavez, F.; Sanchez-Chavez, I.Y.; Paredes-Orta, C.A. Virtual UR5 Robot for Online Learning of Inverse Kinematics and Independent Joint Control Validated with FSM Position Control. *Robotics* **2023**, *12*, 23. [CrossRef]
41. VNC (Virtual Network Computing). RealVNC® Remote Access Software for Desktop and Mobile | RealVNC. Available online: https://www.realvnc.com (accessed on 20 January 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.