"AI-Driven Industrial Automation: Optimizing Vertical Logistics and Storage with Robotic Arms for Enhanced Efficiency"

DENGUIR Oualid, AZZOUZI Majid, SHAO Pingyi
Electrical Engineering Department
National Institute of Applied Sciences (INSA)
Strasbourg, France

Abstract -- This project aims to develop an industrial automation system leveraging artificial intelligence (AI) and robotic arms to enhance efficiency in vertical logistics and storage in an industrial application. The primary goal is to implement through AI robots with real-time decision-making capabilities, optimizing handling, sorting, and storage operations. The proposed solution is meant to increase productivity, reduce errors, and optimize the utilization of available space.
The available materials include Siemens ET200SP PLC, Siemens TP700 Comfort Human-Machine Interface (HMI), Niryo NED 2 robotic arm, a conveyor system, Jetson Nano with a camera, and a small-scale model for vertical storage.

Keywords: industrial automation, artificial intelligence, robotic arms, logistics, vertical storage, efficiency, optimization, productivity, error reduction, space utilization, technology.

I. INTRODUCTION

SEt in the field of industrial innovation, this project embarks on the development of an advanced industrial automation system driven by Artificial Intelligence (AI) and harnessing the capabilities of robotic arms. Our primary objective is to elevate the efficiency of vertical logistics and storage within industrial settings through the incorporation of intelligent decision-making processes facilitated by AI-driven robotic systems.

The foundational work accomplished by FIP GE5 students (Laurent URBAN – Benoit KUHN) during the academic year 2022/2023, successfully crafted a prototype for vertical storage. The present project focuses on the augmentation and expansion of the existing framework. Specifically, the project entails the integration of a dual Niryo NED2 robotic arm system and a conveyor to facilitate the seamless transportation and storage of designated items on the pre-established storage model.

This introduction lays the groundwork for an exploration into the development, implementation, and potential impact of the proposed industrial automation system. By building upon the achievements of the previous academic year's students and adapting the system for a dual-robot configuration, our aim is to enhance productivity, minimize errors, and optimize spatial utilization in industrial processes. The figure bellow (Fig 1) shows the system.

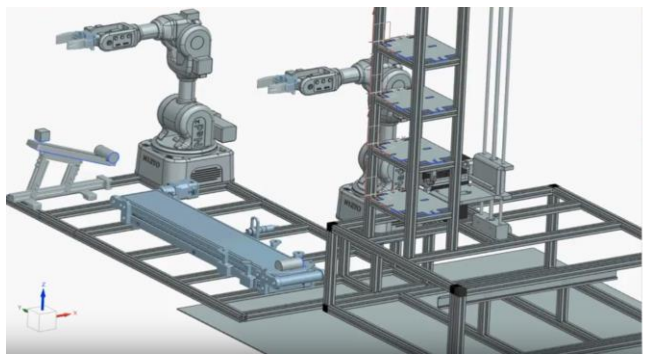


Figure 1: Presentation of the system

The remainder of the document is organized as follows. Section II lists some related works. New features and improvements are shown in section III. Section IV shows the final project. The description of the experiments and their results are provided in section V. Finally, the conclusion and future work are shown in section VI.

II. WORK PLAN

A comprehensive assessment of potential features for our project is presented in this section. The integration of multiple Siemens software components can be achieved through various methods [1].

To facilitate the integration of all sensors, we employ an IO-Link interface, streamlining the implementation through libraries in the Programmable Logic Controller (PLC) software (TIA Portal for Siemens) [2].

Beyond Niryo NED 2 robotic arm [3], external interfaces can be programmed to replicate certain native functionalities of the robotic arm, particularly useful when utilizing alternate hardware such as a PLC.

To expedite processes, the automation of database generation within a 3D Computer Aided Design (CAD) system proves beneficial, minimizing transitional steps [4]. Recognizing the PLC as a cornerstone in industrial automation, software

optimization becomes crucial for maximizing system performance [5].
Various communication protocols, including Modbus TCP, RTU, Ethernet, Profibus, and Profinet, find application in Distributed Control Systems (DCS) [6]. The PLC, often employed to eliminate human interaction for enhanced productivity and safety [7], can benefit from wireless control interfaces. In our case, a MODBUS TCP/IP protocol is employed, alongside ultrasound sensors on the conveyor for object presence detection and contact sensors strategically placed throughout the system.
Drawing inspiration from related works, our project seeks to elevate the efficiency of vertical logistics and storage within industrial settings through the incorporation of intelligent decision-making processes facilitated by AI-driven robotic systems and optimize the code of the last project for smoother, more comprehensible functionality while ensuring swift execution.
Additionally, the numerical twin of the system, acting as a digital counterpart, will be refined to align with the physical system, mirroring its movements on the computer. This holistic approach, encompassing software, hardware, and mechanical improvements, aims to elevate the overall performance and efficiency of the industrial automation system.

III. METHODOLOGY

A. *System Design:*

Develop a detailed system architecture outlining the integration of each hardware component, with a particular focus on interactions among the Siemens ET200SP PLC, Siemens TP700 Comfort HMI, Niryo NED 2 robotic arm, conveyor, Jetson Nano with the thermal camera, and the vertical storage prototype.

B. *Siemens PLC Programming:*

Utilize TIA Portal software to program the Siemens ET200SP PLC, ensuring code optimization for maximizing system performance and implement the MODBUS TCP/IP protocol to enable efficient communication between the PLC, robotic arms, and other system components.

C. *Siemens HMI Development:*

Design an intuitive Human-Machine Interface (HMI) using Simatic WinCC software for real-time monitoring and control of the automated system.

D. *Robotic Arms Integration:*

Program the Niryo NED 2 robotic arms to perform specific tasks such as handling, sorting, and storing products using artificial intelligence algorithms and implement real-time decision-making based on sensor data to optimize the operations of the robotic arms.

E. *Sensor Integration:*

Install ultrasound sensors on the conveyor to detect object presence.

Strategically position contact sensors throughout the system for precise monitoring.

Calibrate sensors to ensure accurate and reliable data acquisition.

F. *Continuous Optimization:*

Conduct comprehensive system testing, emphasizing coordination between components, precision of robotic arm movements, and responsiveness of sensors and identify and address any performance, efficiency, or interaction issues between different elements of the system.

G. *Digital Twin:*

A critical aspect of our methodology involves the development of a digital twin using NX software. This digital twin serves as a virtual replica of the physical system, enabling advanced simulation and predictive analysis. This numerical representation plays a pivotal role in refining system performance by allowing precise adjustments before implementation on the actual hardware. Simulations conducted within the digital twin aid in anticipating operational scenarios, contributing to more informed decision-making during the optimization phase.

IV. FINAL PROJECT

A. *Functional analysis*

Our system is composed of a PLC, 2 Niryo ned2 arm robots, an IO-link device, an HMI, a conveyer, and many sensors. (Fig2)

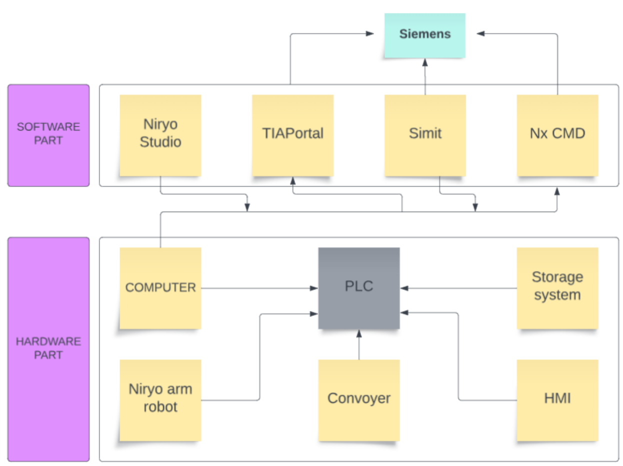


Figure 2: Functional analysis of the system

The diagram presented above (Fig 2) delineates the various components of our system. The software segment comprises the TIA Portal, employed for programming both the PLC and the HMI. Additionally, we utilize NiryoStudio, a software package that comes with the Niryo NED2. While not directly integrated into our system, NiryoStudio serves as a valuable tool for program development and monitoring, assisting in the creation and observation of our programming endeavors.
The hardware components of our system include a computer hosting the software packages, the PLC responsible for overall

system control, the Niryo Robot along with its conveyor, and several sensors. Additionally, pumps are employed for the manipulation of items within the system. Lastly, our setup encompasses the storage system and the entire mechanical structure supporting the seamless functioning of the automation system. (Fig3)

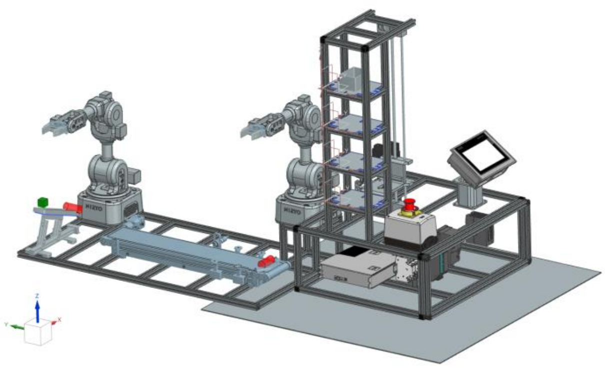


Figure 3: Structure of the system

The figure above shows the hardware part of the system. We can see the mechanical structure, the Robots, the vertical storage prototype, and the conveyer.

B. The PLC

A PLC, an acronym for "Programmable Logic Controller," serves as the central processing unit within the system architecture. Functioning as a specialized type of computing device known as an "automaton," the PLC is characterized by its robust and responsive nature. This technology is intricately linked to all input data within a given system, exercising precise control over its outputs.

The PLC's unique capability lies in its aptitude to monitor the real-time state of the system and exert control over it by influencing the outputs. The programming paradigm employed for the PLC is sequential logic, wherein the execution of the program occurs step by step, contingent upon the prevailing state of the inputs. Preceding the operational phase, the program must undergo compilation and be loaded onto the PLC to ensure its coherent functionality.

In the context of this project, we utilize the "ET200SP" PLC, a product of SIEMENS. Engineered with simplicity, compactness, and efficiency in mind, this automaton is well-suited for the intricacies of the project. To enable the operation and execution of the project, additional modules are required for our PLC (see Fig 4).

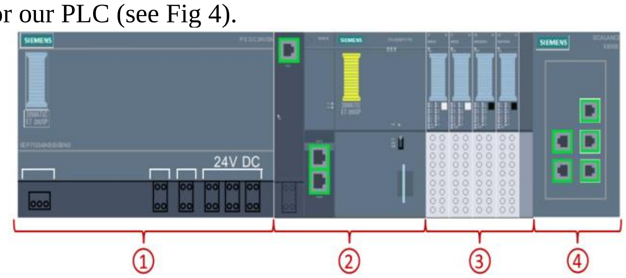


Figure 4: Presentation of the PLC

1. AC/DC Power Conversion (230V to 24V):

To align with the operational requirements of the PLC, which functions on a 24V DC system, an AC/DC converter with a voltage rating of 230V to 24V is employed. This conversion is essential to provide the requisite power supply for the PLC.

2. PLC Model - ET200S:

Our automation system utilizes the ET200S PLC model, a robust and versatile programmable logic controller. Manufactured by SIEMENS, the ET200S PLC is designed to meet the specific demands of industrial automation, ensuring seamless integration and efficient control.

3. Input and Output Modules:

Input and output (I/O) modules play a pivotal role in the overall functionality of the PLC. Output modules are responsible for delivering signals at either 0V or 24V, while input modules can interpret signals ranging from -30V to +5V, representing "0," and 11V to +30V, representing "1." Each module is equipped with 8 input or output pins. In our project configuration, we employ two of each module, resulting in a total of 16 inputs and 16 outputs.

4. Ethernet Switch:

Facilitating communication between various components, an Ethernet switch is incorporated into the system. The Ethernet switch, utilizing RJ45 connections, establishes a network linking multiple devices to our PLC. This crucial module ensures seamless communication and data exchange among interconnected machines, enhancing the overall connectivity and interoperability of the system.

C. The Niryo Robot ned 2:

The Niryo Ned 2 Robot (Fig 5) is a robotic arm for training, research and education. However, the Dobot offers industrial precision and robustness.

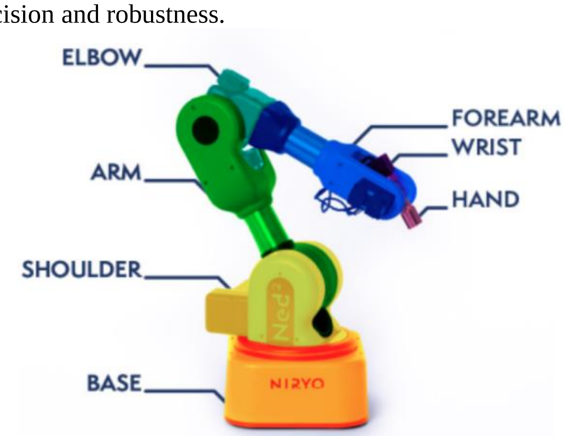


Figure 5: The Niryo Ned 2 Robot

Two Niryo NED robotic arms, each featuring six axes, are employed in the automation system. The first robotic arm is dedicated to the precision task of picking and placing items onto the conveyor. This process involves not only accurate manipulation but also the capability to discern the nature and color of the items. The first robot is equipped with a sophisticated sensory apparatus that facilitates real-time detection and analysis, contributing to a comprehensive understanding of the items being handled.

Simultaneously, the second Niryo NED robotic arm takes on the responsibility of picking items from the conveyor and strategically placing them within the vertical storage systems. This secondary robotic arm is designed to optimize the storage process, ensuring efficient organization and utilization of available space. Together, these two robotic arms form a cohesive and dynamic system, seamlessly coordinating to execute distinct yet interconnected tasks in the automation project.

D. ModBus TCP/iP:

The Robots are under the control of the logic controller through Modbus TCP/IP. To establish communication with the robot, its proprietary communication protocol is employed. Each instruction within this protocol is comprised of a frame, and for each movement and action, there exists a unique frame. The frame consists of a header packet that delineates the initiation point of the instruction. The length ID specifies the length of the subsequent parameters. The instruction ID serves as an identifier for the function; actions are categorized into groups of functions, and the ID facilitates the identification of these groups. The control byte designates whether the action pertains to a write or read operation. Function parameters are variables capable of modifying the chosen operation.

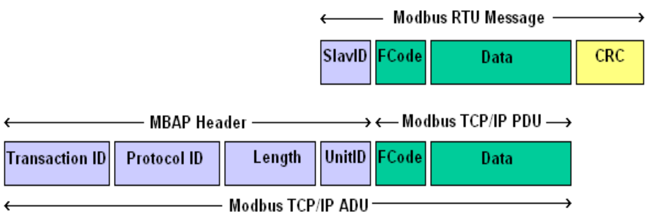


Figure 6: General frame of the ModBus Protocole

And we recommend visiting the website of the constructor to obtain more information.

https://docs.niryo.com/product/ned2/v1.0.0/en/index.html

E. The Jetson Nano Board:

The first robotic arm, dedicated to artificial intelligence applications, is controlled by a Jetson Nano board. This compact yet powerful computing platform utilizes its GPU capabilities to execute AI algorithms and processes in real-time. Advanced image recognition, object detection, and decision-making capabilities are facilitated by the Jetson Nano, enhancing the robot's ability to interact intelligently with its environment. The integration of the Jetson Nano into the control system highlights the application of cutting-edge technology for sophisticated AI-driven functionalities.

Furthermore, it is noteworthy that the same communication protocol, Modbus TCP/IP, is employed for seamless communication between the Jetson Nano board and the PLC. This standardized communication protocol ensures a robust and reliable exchange of information between the AI-driven robotic arm and the central logic controller, contributing to the overall efficiency and coordination of the automation system. The utilization of a consistent communication protocol fosters integration, allowing different components of the system to work cohesively toward achieving the project's objectives.

Figure 7: Jetson Nano Board

V. Programming Part

Utilizing a SIEMENS Programmable Logic Controller (PLC), the programming process will employ the Totally Integrated Automation Portal (TIA Portal), an integrated software package developed by SIEMENS specifically for programming the PLC using SIMATIC STEP 7. Furthermore, TIA Portal facilitates the administration of the visualization component through SIMATIC WINCC, contributing to the creation of the Human Machine Interface (HMI).

The program will initially undergo architectural development, followed by the subsequent implementation of the Human Machine Interface (HMI).

➢ Architecture of the program:

- An initialization of the Niyro Robot
- Connect the first robot with the PLC.
- Connect the second robot with the jetson nano board.
- A bloc for the principal program for the Robot.
- Motor control.
- Servomotor control.
- Packing Data.
- The program for HMI.

Each element is programmed within a designated unit referred to as a "block" in the software. These blocks are coded using the conventional automation language known as Ladder Diagram (LADDER). Consequently, individual blocks are allocated for each specific feature, complemented by additional blocks designed to enhance overall system functionality.

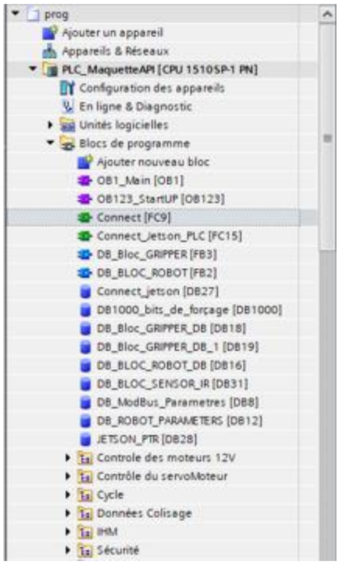


Figure 8: Architecture of the program

The initial block serves as a start-up block and is utilized only once during the initialization phase. Within this block, the connection between the PLC and the secondary niyro robot is established using the Modbus TCP/IP protocol as the communication bus, as previously mentioned.

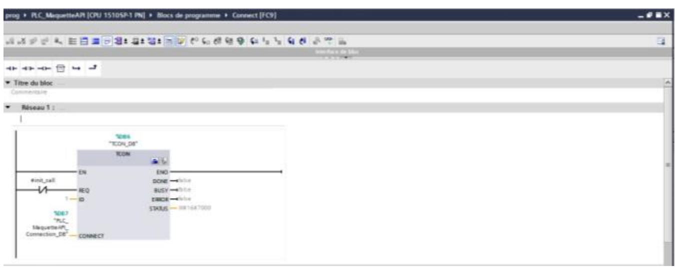


Figure 9 : Connect the first robot with the PLC

The latter section (Fig 10) of this block is employed for preparing the frame, enabling the transmission or retrieval of data. In the context of this project, it is specifically designed to send and receive the positions of each joint of the robot continuously, ensuring seamless functionality and establish the calibration of the Niryo robot if needed.

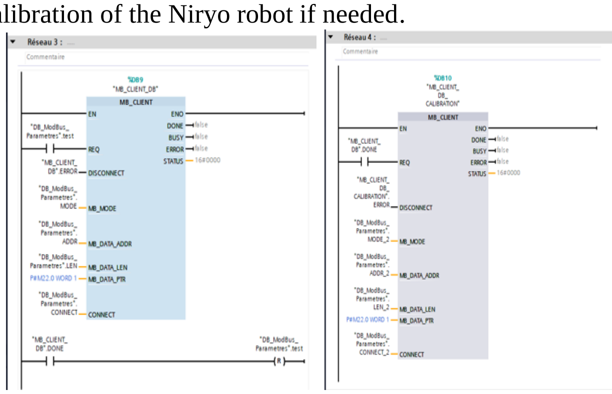


Figure 10: Calibration launch

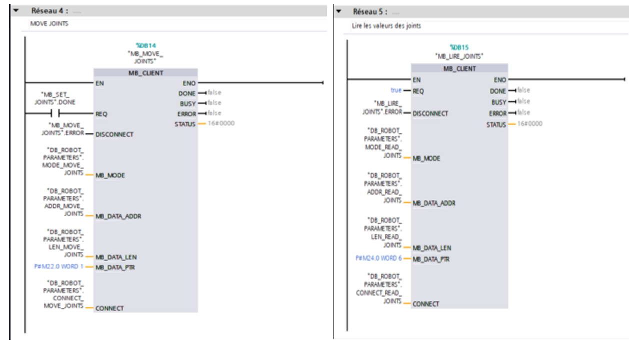


Figure 11: send and receive the positions of each joint.

Structured Control Language (SCL) is a programming language employed in the principal robot's program to manipulate information within the previously mentioned blocks. SCL is a high-level programming language designed for industrial automation and control systems. It provides a structured and intuitive syntax for expressing complex control logic, mathematical operations, and data manipulation within the context of programmable logic controllers (PLCs) or similar automation devices. SCL enhances the efficiency of program development by offering a clear and organized approach to writing control algorithms.

Figure 10: Example of SCL program to send the positions

In the aforementioned section, a program is implemented to gather data from the sensor, specifically to determine the quantity of items on the conveyor. This involves the utilization of a counter to keep track of the item count, enabling efficient monitoring and control of the conveyor system.

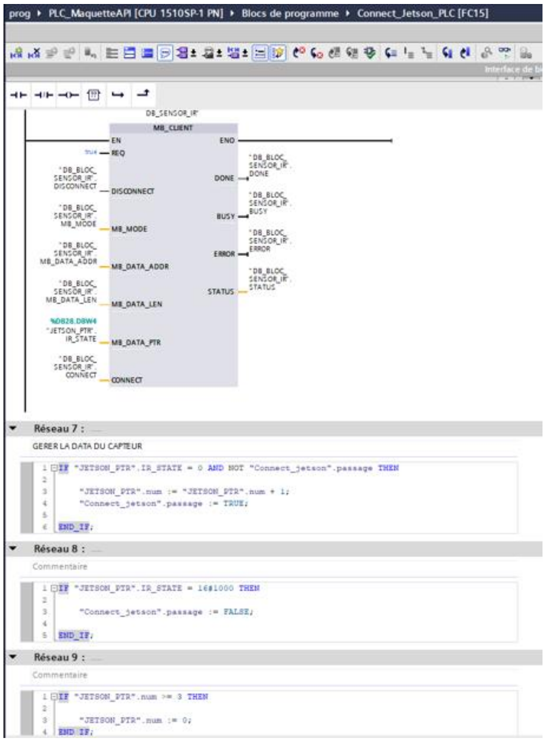


Figure 11: Sensor Program

For the integration of the Jetson Nano and the PLC, we have established a connection through the same communication bus, leveraging the Human Machine Interface (HMI). In the TIA Portal, we have configured the communication settings to facilitate seamless interaction between the Jetson Nano and the PLC. The subsequent discussion will elaborate on the methodology employed in other aspects of this integration.

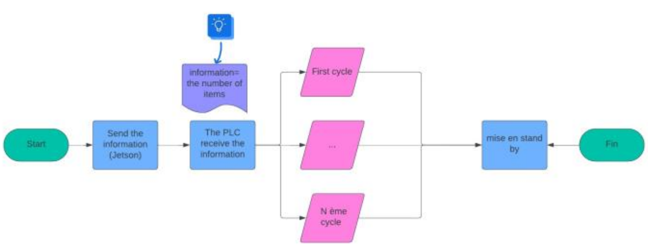


Figure 12: The operating cycle

We will transmit information from the Human Machine Interface (HMI), specifically the color of the object, so that the first robot can determine the color to be collected. Subsequently, it will convey objects of the identified color onto a conveyor belt, allowing the second robot to store them in the dedicated storage model.

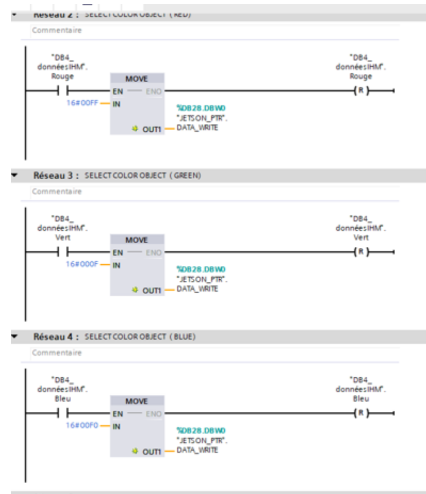


Figure 13: Sending the color information

The automation system integrates various components, including Siemens PLCs, Human Machine Interface (HMI) through the TIA Portal, and the utilization of the Modbus TCP/IP communication protocol. The programming is conducted in Structured Control Language (SCL), facilitating efficient manipulation of information within designated blocks.

The first part of the program initiates with a start-up block, establishing communication with a secondary robot via Modbus TCP/IP. This connection enables the exchange of joint positions, vital for continuous operation. The subsequent program section focuses on data collection from sensors using a counter to monitor the quantity of items on the conveyor.

The integration extends to the collaboration between the Jetson Nano and the PLC, achieved through the same communication bus and configured within the TIA Portal. The HMI serves as a central control point, allowing the transmission of object color information. This information guides the actions of the first robot, which selectively collects and transfers objects of the specified color to a conveyor belt. The second robot then stores these objects in a dedicated storage model.

This comprehensive approach ensures a synchronized and automated operation, emphasizing the seamless interaction between hardware components, programming languages, and communication protocols to achieve the intended automation objectives.

The human machine Interface:

The HMI is the interface between the system and the user; the user will be able to choose the items which we want to store.

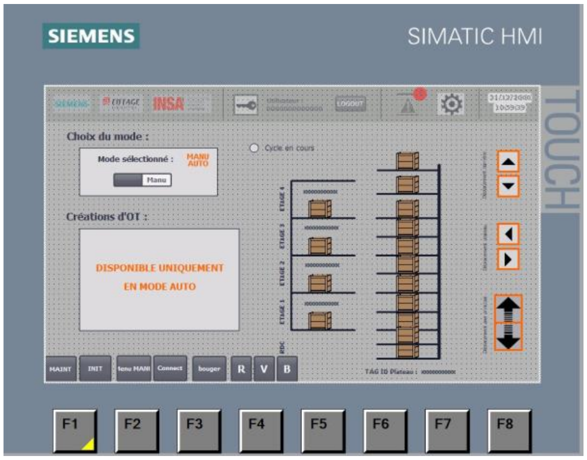


Figure 13: The human machine interface

Connect: establish the connection between the robot and the PLC
R: chose the red color
V: chose the red color
B: chose the blue color
Bouger: Move the robot manually.
INT: Initialize the hall system.

In the Human Machine Interface (HMI), we have incorporated a system to visually represent the functioning of the overall automation system, akin to a digital twin. This visual representation serves as a graphical depiction of the system's operation, providing a real-time and dynamic overview of the processes and interactions between various components. The digital twin in the HMI enhances operator understanding and facilitates effective monitoring and control of the entire automation system.

VI. JETSON NANO'S PROGRAMMING PART

In this section, we integrate the Jetson Nano with a PLC and the Niryo Ned2 robotic arm. Upon receiving commands from the user through the HMI, the Jetson Nano processes images captured by the robotic arm's camera. Utilizing AI algorithms, it identifies the positions of specific blocks corresponding to the commands and then directs the robotic arm to perform pick-and-place tasks on these blocks.

A. Required Installation Packages

The following is a list of key installation packages required for this section:

1) Pyniryo

'pyniryo' is a Python library specifically designed for interacting with Niryo robots, including the Niryo Ned2. The Niryo Ned2 is a more advanced version of the Niryo One, a six-axis robotic arm widely used in education, research, and light industrial applications. The pyniryo package provides a user-friendly interface to control and automate the Niryo Ned2 directly from Python scripts, making it highly accessible for both beginners and advanced users in the field of robotics.

2) OpenCV

OpenCV (Open-Source Computer Vision Library) is a highly popular open-source computer vision and machine learning software library. The OpenCV library is widely used in various applications such as real-time image processing, video capture and analysis, face recognition, motion tracking, and much more.

3) Pymodbus

'pymodbus' is a full-featured and easy-to-use Python library for communicating with devices (like PLCs, VFDs, and other industrial equipment) using the Modbus protocol. Modbus is a widely used communication protocol in industrial environments, allowing for communication among many devices connected to the same network.

```python
import numpy as np
from pyniryo import *
import math
from utils import *
import cv2
import time
from pymodbus.server.asynchronous import StartTcpServer
from pymodbus.datastore import ModbusSequentialDataBlock
from pymodbus.datastore import ModbusSlaveContext, ModbusServerContext
from pymodbus.transaction import ModbusSocketFramer
import threading
import sys
```

Figure 14: All installed packages

B. Robotic Arm Control

In this section, we detail the process of controlling a Niryo Ned2 robotic arm using a Jetson Nano, facilitated by the 'pyniryo' package. The initial step involves establishing a WiFi connection between the Jetson Nano and the robotic arm, followed by executing the 'calibrate' command. This setup not only enables the synchronization of the devices but also aids in identifying the tool ID on the robotic arm, such as the pump attached to it. Concurrently, we activate the conveyor belt to transport small blocks towards a secondary robotic arm.

```python
def connect(robot_ip_address, observation_pose):
    # Connecting to robot
    client = NiryoRobot(robot_ip_address)
    client.calibrate(CalibrateMode.AUTO)
    client.update_tool()
    tool_id = client.get_current_tool_id()
    if tool_id == 11 or tool_id == 12 or tool_id == 13:  # If it is a gripper
        client.close_gripper()  # close gripper so that workspace is more visible
    client.move_pose(*observation_pose.to_list())
    return client
```

Figure 15: Code for connexion

Subsequently, the robotic arm is maneuvered to a vantage point, optimizing its field of view for enhanced image capture of the workspace. This strategic positioning is crucial for the arm's camera to acquire real-time images of the workspace, which are pivotal for subsequent AI-driven computations.

The final phase involves the AI module analyzing these images to pinpoint the center of the small blocks. Utilizing these calculated coordinates, the robotic arm is then instructed to precisely pick up the blocks, identified by their color, from the workspace. The selected blocks are subsequently placed onto the moving conveyor belt, demonstrating a seamless integration of robotics and AI in an automated system.

```python
def pick_place(obj_pose, drop_pose):
    client.move_pose(obj_pose)
    time.sleep(0.5)
    client.grasp_with_tool()
    time.sleep(0.5)
    client.move_joints(-0.03, 0.49, -0.49, -0.20, 0.1, 0.01)
    time.sleep(0.5)
    client.move_pose(*drop_pose.to_list())
    time.sleep(0.5)
    client.release_with_tool()
    time.sleep(0.5)
```

Figure 15: Code for pick-&-place

C. AI-Driven Color Block Analysis in Images with OpenCV

In this section, we delineate the process initiated by instructions received from a Programmable Logic Controller (PLC), complemented by previously captured images of the workspace. The procedure commences with the conversion of the original BGR image to a threshold image, tailored to a specified color using the threshold_hsv command. Subsequent to this transformation, the findContours function from OpenCV is employed to detect the contours of small blocks within the image.

```python
def draw_center_of_color(image_workspace, color):

    image_threshold, nom = HSV_color(int(color))
    image_threshold_init, nb_objet = mask_obj_contour(image_threshold)

    contours, _=cv2.findContours(image_threshold_init, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    for cnt in contours:
        x, y, w, h = cv2.boundingRect(cnt)
        cx = int(x + w / 2)
        cy = int(y + h / 2)

        if color == '1':
            couleur = (0,0,255)
        elif color == '2':
            couleur = (255,0,0)
        else:
            couleur = (0,255,0)
        cv2.circle(image_workspace, (cx, cy), 5, couleur, -1)
    return cx, cy
```

Figure 16: Code for getting pixel coordinates

Each identified contour is then meticulously analyzed using the boundingRect method to ascertain the boundaries of the respective contour. This analysis facilitates the calculation of the central coordinates of each bounding box in pixel units. Following this, the relative_pose_from_pixels function is utilized to transform these pixel coordinates into relative positions. Concurrently, the orientation of each block is ascertained using the get_contour_angle function.

Finally, the amassed data is converted into relative movement parameters for the robotic arm through the get_target_pose_from_rel function. This conversion enables the robotic arm to accurately position itself and retrieve the designated blocks, as dictated by the PLC-based instructions.

```python
cx_rel, cy_rel = relative_pos_from_pixels(img_workspace, cx, cy)
angle = get_contour_angle(contour)
obj_pose = client.get_target_pose_from_rel(workspace, height_offset=-0.01, x_rel = cx_rel - 0.05, y_rel = cy_rel, yaw_rel = angle)
```

Figure 17: Code for getting relative movements parameters

D. Intergrating Jetson Nano with PLC via Ethernet Using Pymodbus

In this section, we define a function named 'run_server' with the objective of initiating a Modbus TCP server within a Python environment. Utilizing the ModbusSlaveContext, we have established an object comprising four distinct types of data blocks: Discrete Inputs, Coils, Holding Registers, and Input Registers. This configuration enables a Programmable Logic Controller (PLC) to write command-specific data into the Holding Registers. Subsequently, the Jetson Nano can retrieve these commands via the getValues method. Additionally, the Nano is programmed to populate the Input Registers with the quantified count of blocks using the setValues method, thereby facilitating the PLC's access to this information.

```python
def run_server():
    global global_context


    store = ModbusSlaveContext(
        di = ModbusSequentialDataBlock(0, [0]*10),
        co = ModbusSequentialDataBlock(0, [0]*10),
        hr = ModbusSequentialDataBlock(0, [0]*10),
        ir = ModbusSequentialDataBlock(0, [0]*10))

    global_context = ModbusServerContext(slaves=store, single = True)


    StartTcpServer(global_context, address=("192.168.0.100", 503))
```

Figure 18: Code for creating Modbus TCP server.

VII- The digital Twin

A digital twin serves as a virtual counterpart to a physical object or system, meticulously crafted to emulate its real-world attributes and functionalities. Going beyond conventional computer-aided design (CAD) models, this concept integrates real-time data and dynamic updates, resulting in a dynamic and responsive digital replica of the physical entity.

Widely adopted across diverse sectors such as manufacturing, healthcare, smart cities, and the Internet of Things (IoT), digital twins play a pivotal role. The fundamental objective is to elevate comprehension, surveillance, and control of the physical entity by harnessing the abundant data produced across its lifecycle.

A. Modeling the industrial automation mock-up on NX.

The initial phase of modeling on NX commenced with utilizing the digital model of the existing vertical storage mock-up. This process involved the precise importation of the dimensions and features of the mock-up to ensure seamless integration with the new project elements. The benefits of this approach lie in the consistency between the existing design and the planned additions, enabling effective coordination of components.

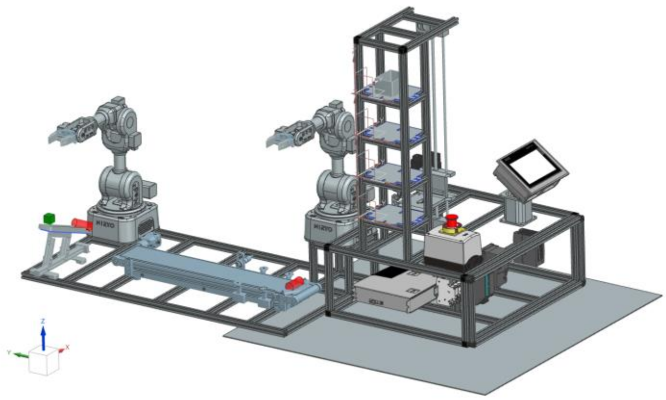


Figure 17: the Model of the system

B. 3D model of the vertical storage mock-up.

The initial phase of modeling on NX began by leveraging the digital model of the existing vertical storage prototype. This process involved the precise importation of the prototype's dimensions and characteristics to ensure optimal integration with the new project elements. The advantages of this approach lie in the consistency between the existing design and the planned additions, facilitating an effective coordination of components.

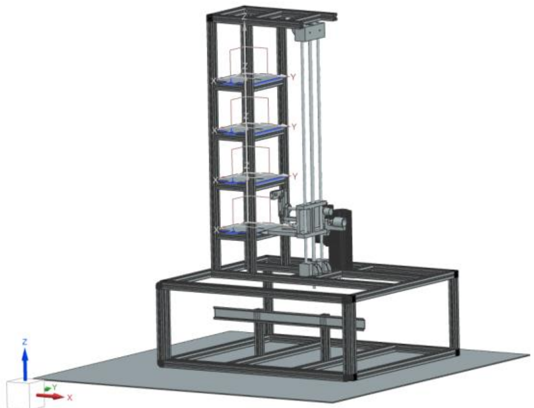


Figure 18: Model of the vertical storage prototype.

C. Integration of the vertical storage mock-up and the robotic arm.

The vertical storage mock-up has been carefully positioned relative to the robotic arm. This spatial arrangement was determined to mimic realistic conditions in an automated factory, ensuring smooth interaction between the two elements.

The precise alignment between the vertical storage mock-up and the robotic arm has been achieved to accurately replicate pick-and-place operations and arm movements within an automated environment.

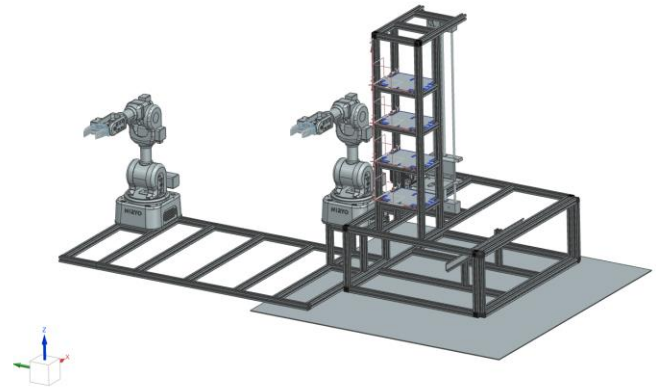


Figure 19: Prototype featuring vertical storage and both robotic arms.

Synchronization of Movements: The movements of the robotic arm have been synchronized with the operations of the vertical storage prototype, ensuring seamless coordination in the movement and manipulation of products.

<u>*Visualization of Interactions:*</u>

Simulation of Operations: By visually integrating the vertical storage prototype and the robotic arm into the overall assembly, we obtain a comprehensive visual representation of interactions between these two key elements. This provides an essential overview for assessing the functionality of the automated system.

<u>*Validation and Adjustments:*</u>

This phase of adding the vertical storage prototype and the robotic arm undergoes thorough validations where the consistency of movements, collision management, and operation synchronization are rigorously assessed. Adjustments are made to optimize performance and ensure a realistic simulation.

<u>*Challenges and Outlook:*</u>

The successful integration of the vertical storage prototype and the robotic arm marks a significant milestone, laying the foundation for a functional and adaptable factory of the future. This integration sets the stage for the next steps of the project, including the design of the profile structure and the addition of final components for a comprehensive and operational digital representation of our automated system.

D. Addition of the 3D models of the conveyor, ramp, and cube.

We selected a 3D model of a conveyor tailored to our requirements, considering the nature and dimensions of the goods to be transported. The addition of the conveyor to the NX prototype was carried out with careful attention to precise integration with the existing structure. Dimensions and placement were adjusted to ensure a continuous and efficient flow of products.

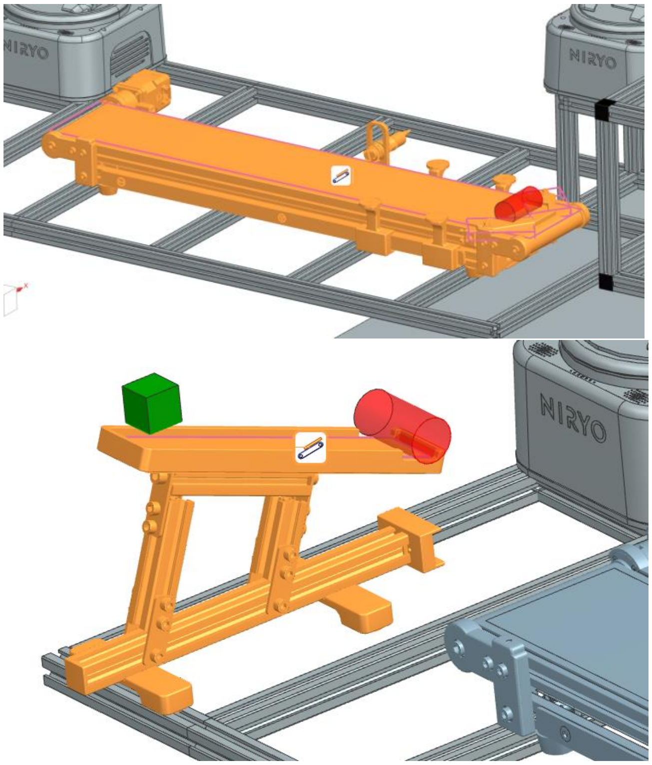
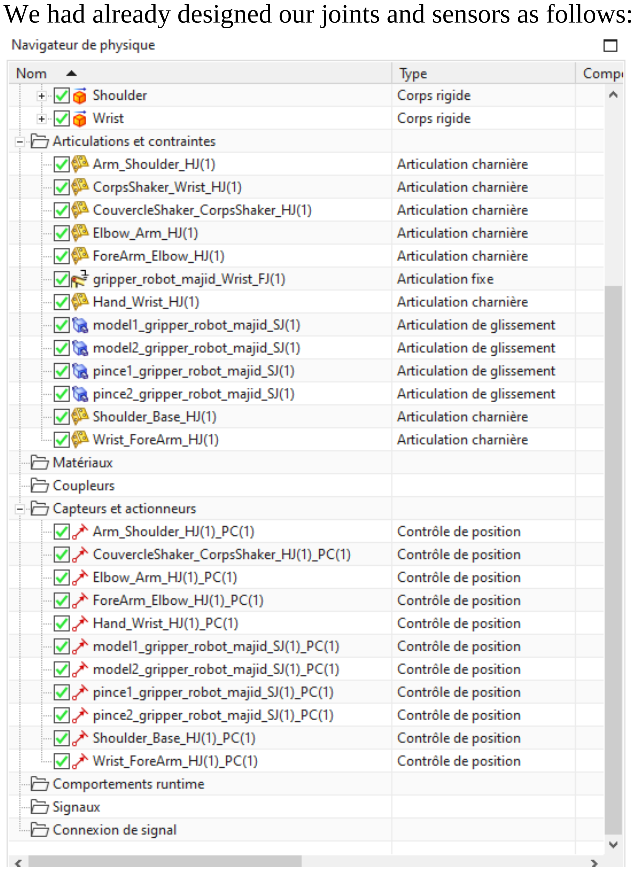


Figure 19: Conveyer and Ramp

E. Configuration du jumeau numérique avec SIMIT.

We had already designed our joints and sensors as follows:



Navigateur de physique

Nom	Type	Comp
Shoulder	Corps rigide	
Wrist	Corps rigide	
Articulations et contraintes		
Arm_Shoulder_HJ(1)	Articulation charnière	
CorpsShaker_Wrist_HJ(1)	Articulation charnière	
CouvercleShaker_CorpsShaker_HJ(1)	Articulation charnière	
Elbow_Arm_HJ(1)	Articulation charnière	
ForeArm_Elbow_HJ(1)	Articulation charnière	
gripper_robot_majid_Wrist_FJ(1)	Articulation fixe	
Hand_Wrist_HJ(1)	Articulation charnière	
model1_gripper_robot_majid_SJ(1)	Articulation de glissement	
model2_gripper_robot_majid_SJ(1)	Articulation de glissement	
pince1_gripper_robot_majid_SJ(1)	Articulation de glissement	
pince2_gripper_robot_majid_SJ(1)	Articulation de glissement	
Shoulder_Base_HJ(1)	Articulation charnière	
Wrist_ForeArm_HJ(1)	Articulation charnière	
Matériaux		
Coupleurs		
Capteurs et actionneurs		
Arm_Shoulder_HJ(1)_PC(1)	Contrôle de position	
CouvercleShaker_CorpsShaker_HJ(1)_PC(1)	Contrôle de position	
Elbow_Arm_HJ(1)_PC(1)	Contrôle de position	
ForeArm_Elbow_HJ(1)_PC(1)	Contrôle de position	
Hand_Wrist_HJ(1)_PC(1)	Contrôle de position	
model1_gripper_robot_majid_SJ(1)_PC(1)	Contrôle de position	
model2_gripper_robot_majid_SJ(1)_PC(1)	Contrôle de position	
pince1_gripper_robot_majid_SJ(1)_PC(1)	Contrôle de position	
pince2_gripper_robot_majid_SJ(1)_PC(1)	Contrôle de position	
Shoulder_Base_HJ(1)_PC(1)	Contrôle de position	
Wrist_ForeArm_HJ(1)_PC(1)	Contrôle de position	
Comportements runtime		
Signaux		
Connexion de signal		

And we exported all our sensors to the runtime inspector. From there, we were able to create our Simit model. We started by importing our signals from MCD. Next, we developed our interface for managing the gripper. Finally, we created our interface for managing the movements of the Niryo robot.

Figure 21: System Management on SIMIT

VIII-CONCLUSION:

This innovative initiative represents a significant step forward in the realm of industrial automation, harnessing the power of artificial intelligence to revolutionize traditional logistics and storage practices. The incorporation of robotic arms equipped with real-time decision-making capabilities promises not only heightened operational efficiency but also a reduction in errors, leading to an overall enhancement of productivity.

Furthermore, the project's focus on optimizing space utilization reflects a commitment to sustainability and resource efficiency. The anticipated benefits extend beyond mere technological advancement, offering a transformative solution to address the evolving demands of modern industrial processes.

As the system takes shape, it becomes evident that this endeavor has the potential to redefine industry standards, setting a precedent for intelligent and adaptive automation. The successful integration of AI-driven robotics not only aligns with current industry trends but also positions the project at the forefront of innovative solutions that have the capacity to shape the future of industrial automation.

IX- REFERENCES.

[1] Husam Salih, Hammam Abdelwahab, Areej Abdallah. Automation design for a syrup production line using Siemens PLC S7-1200 and TIA Portal software. Presented at 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE). [Online]. Available: https://ieeexplore.ieee.org/document/7866702

[2] Sadik Tamboli, Mallikarjun Rawale, Rupesh Thorajet, Sudhir Agashe. Implementation of Modbus RTU and Modbus TCP communication using Siemens S7-1200 PLC for batch process. Presented at 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM). [Online]. Available:
https://ieeexplore-ieee-org.ezproxy.insastrasbourg.fr/document/7225424

[3] Barz Cristian, Oprea Constantin, Erdei Zoltan, Pop Vadean Adina, Petrovan Florica. The control of an industrial process with PLC. Presented at 2014 International Conference on Applied and Theoretical Electricity (ICATE). [Online]. Available:
https://ieeexplore-ieee-org.ezproxy.insa-strasbourg.fr/document/6972663

[4] Aashish Mehra. What is a digital twin. Published in 2022 on IBM Website. [Online]. Available: https://www.ibm.com/topics/what-is-a-digital-twin

[5] Construction of a Digital Twin Framework Using Free and Open-Source Software Programs.
The conception of a digital twin of a enclosed heat system, using free softwares. Written by : Karan Shah , T. V. Prabhakar , Sarweshkumar C. R., Abhishek S. V., and Vasanth Kumar T,Indian Institute of Science, Bangalore, 560012, India
Available : https://ieeexplore.ieee.org/abstract/document/9325003

[6] Kou Zuxing, Zhang Wi, Zhang Libao. Design of automatically converting materials control system based on TIA Portal configuration software. Presented at 34th Youth Academic Annual Conference of Chinese Association of Automation(YAC).[Online].Available:
https://ieeexplore.ieee.org/document/8787638

[7] Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application.
Available:
https://www.sciencedirect.com/science/article/pii/S0168169921006037?via%3Dihub

[8] Development of an obstacle avoiding autonomous vehicle by using stereo depth estimation and artificial intelligence based semantic segmentation
Available:
https://www.sciencedirect.com/science/article/pii/S0952197623009922

[9] Enabling intelligent onboard guidance, navigation, and control using reinforcement learning on near-term flight hardware
https://www.sciencedirect.com/science/article/pii/S0094576522003502