

## Article

# Cyberphysical System Modeled with Complex Networks and Hybrid Automata to Diagnose Multiple and Concurrent Faults in Manufacturing Systems

Alejandro Velazquez <sup>1,\*</sup>, Fernando Martell <sup>1</sup>, Irma Y. Sanchez <sup>2</sup> and Carlos A. Paredes <sup>1</sup>

<sup>1</sup> Technology Development Department Centro de Investigaciones en Óptica, A.C., Aguascalientes 20200, Mexico; fmartell@cio.mx (F.M.); cparedes@cio.mx (C.A.P.)

<sup>2</sup> Research and Development Department Ingeniería Mecatrónica, SA de CV., Aguascalientes 20235, Mexico; isanchez@ingmt.com

\* Correspondence: alejandrovj@cio.mx

**Abstract:** Cyber-physical systems use digital twins to provide advanced monitoring and control functions, including self-diagnosis. The digital twin is often conceptualized as a 3D model, but mathematical models implemented in numerical simulations are required to reproduce the dynamical and functional characteristics of physical systems. In this work, a cyber-physical system scheme is proposed to monitor and diagnose failures. The virtual system, embedded at the supervisory control level, combines concepts from complex networks and hybrid automata to detect failures in the hardware components and in the execution of the sequential logic control. An automated storage and retrieval system is presented as a case study to show the applicability of the proposed scheme. The functional test and the obtained results validate the implemented system that is shown to be capable of fault diagnosis and location in real time. The online execution of the digital twin present several advantages for diagnosing multiple concurrent failures in sensors, actuators, and the control unit. This approach can be incorporate into diverse manufacturing systems.

**Keywords:** cyber-physical systems; fault diagnosis; hybrid automata; complex networks



**Citation:** Velazquez, A.; Martell, F.; Sanchez, I.Y.; Paredes, C.A.

Cyberphysical System Modeled with Complex Networks and Hybrid Automata to Diagnose Multiple and Concurrent Faults in Manufacturing Systems. *Appl. Sci.* **2023**, *13*, 10603. <https://doi.org/10.3390/app131910603>

Academic Editor: Maurizio Faccio

Received: 27 July 2023

Revised: 26 August 2023

Accepted: 1 September 2023

Published: 22 September 2023



**Copyright:** © 2023 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

Over the years, the industry has had changes in terms of the technology implemented to improve and produce more with fewer human resources; we can see this in the so-called industrial revolutions. The last revolution is Industry 4.0, which emphasizes the use of digital technology developed in the third industrial revolution, but takes it to a more advanced level with the help of interconnectivity through the Internet of Things (IoT), access to data in real time and the introduction of cyber-physical systems. An existing problem in these systems is that, due to their complexity in the occurrence of faults, their diagnosis and location are very complex, which motivates this research work.

Cyber-physical systems are integrated by two subsystems: a physical system composed of sensors, actuators, Programmable Logic Controllers (PLC), and in some cases, a human-machine Interface (HMI) [1–3], and a digital system that is considered a part of the systems that develop different tasks such as the monitoring, optimization and diagnosis of the physical system [4–6]. The digital system in the project comprises a supervisory controller, fault diagnosis component and digital twin. Additionally, a communication protocol to send/receive information in both subsystems is needed. In cyberphysical systems, a mathematical tool called hybrid automata is used that provides the possibility of modeling digital and analog variables and allows the use of time delays that are important for the system to communicate with its digital twin [7–9]. The cyber-physical systems (CPS) must be in real-time communication to be able to verify information about the process.

Among the different possibilities for the operation of the digital Twin inside a CPS, which is a widely accepted approach, the Digital Twin provides a means for the remote and simulated commissioning of manufacturing systems, thus allowing for the rapid reconfiguration of the automated manufacturing system. In contrast, the presented CPS approach is oriented for fault detection, which requires the Digital Twin to run in parallel with the physical systems. The run-time execution and comparison may allow multiple and concurrent fault detection. In the application of Industry 4.0, there exist possibilities to propose, design and implement new concepts that may improve production systems, which is of interest to both academia and the industry.

We found in [10–12] that the combination of hybrid automata and complex network theory has been applied for system analysis but not for diagnosing system failures. A complex network can reproduce the evolution of a networked system if the system is controlled by a hybrid automata that changes from one state to another [13]. The hybrid control system activates or deactivates actuators in response to sensor values. Adjacency matrices are constructed for the analysis of complex networks, which places correspondences to the connected nodes [14–16]. From the reading of sensors and the activation and deactivation of actuators of the physical system, information is saved in the adjacency matrix using arbitrary weights, corresponding to the space assigned by each sensor or actuator [17].

In the present work, a functional and practical approach to a cyber–physical system is presented to diagnose multiple concurrent failures in an automatic storage and retrieval system. The virtual twin comprises a hybrid automata and a numerical simulation of the physical system. Sensors, actuators, and control units, as connected devices, are modeled as complex networks to take advantage of some concepts and analysis tools derived from complex networks theory. The objective of the proposed cyber–physical system approach is to monitor and diagnose failures in the sensors and actuators and in the execution of the sequential logic control. The digital twin is embedded in the supervisory controller and is executed simultaneously with the physical process. The application of the computational algorithms is to detect failures in a timely manner and thus reduce delays in manufacturing production processes, and improve productivity. This paper is organized as follows: Section 2 recalls some concepts of cyberphysical systems, hybrid automata, and complex networks. In Section 3, the material and methods, the proposed scheme and the results are presented. Section 4 discusses the results of the implemented CPS scheme and the challenges in its implementation. Section 5 is the conclusion and future work.

## 2. Background

This section describes the theoretical background required for the conceptualization of a cyber–physical system. This includes definitions of cyber–physical systems, hybrid systems, hybrid automata, complex networks, and analysis tools.

### 2.1. Cyber–Physical Systems

A cyber–physical system (CPS) refers to a system with combined computational and physical capabilities, as it interacts with users through new modalities. The ability to communicate and expand the operations of the physical world through computer technology, interaction, and control are aspects of high importance in technological development. Autonomous cars, hybrid or fully electric vehicles, the design and development of aircrafts, and remotely controlled manufacturing systems represent research challenges and opportunities. In the control area, progress has been made in the implementation of engineering methods and tools, such as the linear control, discrete control, optimization, prediction, filtering, time and frequency domain, and state space analysis. In the area of computer systems, progress has been made in technological advances such as information security and system failure detection, reliability innovation, integrated software and new computer system architectures, visualization procedures, and real-time simulation techniques, based on models and new simulation languages. The development of cyber–physical systems aims to integrate the behavior of a physical system in the computing and engineering areas

(mechatronics, software, human interaction, artificial intelligence, as well as chemical and material sciences and other engineering disciplines) to develop new systems' CPS [18].

### 2.2. Hybrid Systems

Hybrid systems are characterized by continuous and discrete dynamics, since in these systems, logical decision making and regulatory control actions are combined with continuous physical processes. Mathematical models should combine the continuous and discrete dynamics of the system. Such mathematical models consist of differential or difference equations of continuous dynamics and automata or other discrete event models for discrete dynamics. The combination of modeling techniques is the basis of hybrid systems theory, which plays an important role in the multidisciplinary design of many cyber–physical systems [19].

### 2.3. Hybrid Automata

Hybrid automata provide formal models for hybrid dynamical systems and can be observed as an extension of finite automata by adding continuous dynamics into each of its discrete states (also called modes). Each mode is associated with constraints, within which the continuous dynamics evolve. Edges between modes are annotated with guards that specify the conditions for the mode transition to be triggered; each edge is also associated with a reset map indicating how the continuous variables are being updated after the discrete transition. Hybrid automata are defined below [19].

A hybrid automaton  $H$  is a collection:

$$H = (Q, X, f, Init, Inv, E, G, R)$$

where:

- $Q = (q_1, q_2, \dots)$  is a finite set of discrete states;
- $X \subseteq R^n$  represents the state space of the continuous state variables;
- $f : Q \times X \rightarrow R^n$  assigns to each discrete state  $q \in Q$  an analytic vector field  $f(q, \cdot)$ ;
- $Init \subseteq Q \times X$  is the set of initial states;
- $Inv : Q \rightarrow 2^X$  assigns to each state  $q \in Q$  a set  $Inv(q) \subseteq X$  called the invariant set;
- $E \subseteq Q \times Q$  is the set of discrete transitions;
- $G : E \rightarrow 2^X$  assigns to each discrete transition  $(q, q') \in E$  a guard set  $G(q, q') \subset X$ ;
- $R : E \times X \rightarrow 2^X$  is a reset map.

### 2.4. Complex Networks

Complex networks are known as sets of interconnected nodes. The nodes of a network can be called vertices or elements and are represented by the symbols  $v_1, v_2, \dots, v_N$ , where  $N$  is the total number of nodes in the network. If a node  $v_i$  is connected to another node  $v_j$ , this connection is represented by an ordered pair  $(v_i, v_j)$ . The mathematical definition of a network (also called a graph) is as follows: Formally, a network  $R$  consists of a set of nodes  $V = v_1, v_2, \dots, v_N$  and a set of ordered pairs  $E = (v_i, v_j) \subset V \times V$ . Each ordered pair  $(v_i, v_j)$  is a directed connection from node  $v_i$  to node  $v_j$ . A non-directed type network consists of that for each pair  $(v_i, v_j) \in \epsilon$  there also exists the pair  $(v_j, v_i) \in \epsilon$ , as shown in Figure 1a. If the above is not fulfilled, the network directed, as Figure 1b. Illustrates. Nodes that are directly connected to a  $v_i$  node are called  $v_i$ 's neighbors [20].

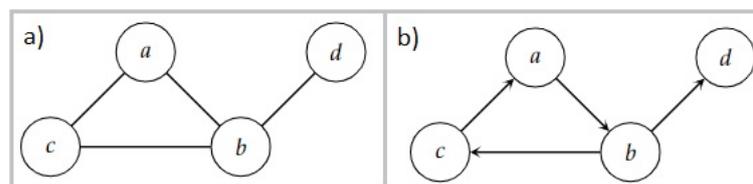


Figure 1. (a) Undirected network. (b) Directed network.

### 3. Matherial and Methods

Most of the proposed approaches for fault diagnostic using DT and CPS are more oriented for commissioning manufacturing systems for reconfiguration and optimization purposes. The proposed approach differs from that found in the literature, since the DT is required to run in parallel to be able to perform an online fault Diagnostics. In this section, a basic conceptualization of a cyber–physical system incorporating a complex network is presented.

#### 3.1. Cyber–Physical System Components

The cyber–physical system proposed here is composed by a communication structure between two systems, physical and digital systems. The physical system comprises a programmable logic controller (PLC), and a human–machine interface (HMI) interconnected to the supervisory controller through the PLC. The digital system comprises a supervisory control and a digital twin embedded in the digital system. The complete proposed CPS scheme is shown in the diagram of Figure 2.

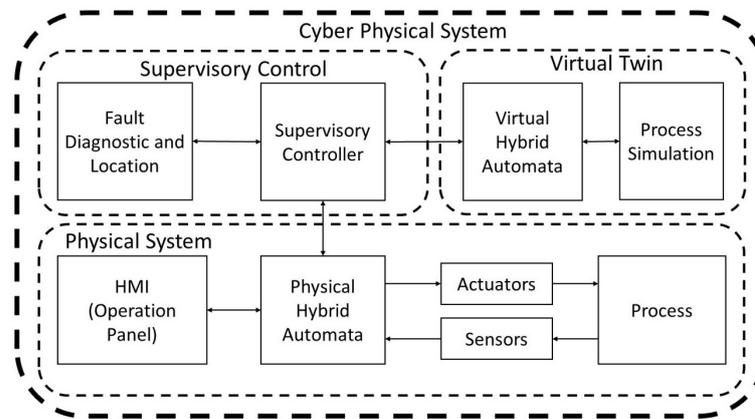


Figure 2. Proposed scheme of the cyber–physical system.

In the proposed cyber–physical system, there are two automata:  $H_P$ , which is the physical hybrid automata, and  $H_V$ , that is, the hybrid virtual automata, which is executed in a virtual system.  $T_S$  is the synchronization time and  $Adj_M$  is the Adjacency Matrix.

$$CPS = \{H_P, H_V, T_S, Adj_M\}$$

The hybrid automatras  $H_F$  and  $H_V$  share the same sets of states, functions and variables; however, the  $H_V$  variables are from the virtual system and the  $H_P$  variables from the physical system. The graphic representation of the automata is presented in Figures 3 and 4.

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$$

$$dx = R \text{ denotes the range of the current based in the outputs activated } x = \alpha * y_{n-1} + (1 - \alpha)y_n$$

$$Init = \{q_0\}x\{1\} \text{ or } \{q_{10}\}x\{1\}$$

$$f(q_0, x_1) = 10 \text{ and } f(q_1, x_2) = 10 \text{ and } f(q_2, x_3, x_4, x_5) = 5, 5, 10 \text{ and}$$

$$f(q_3, x_6, x_7) = 2, 10 \text{ and } f(q_4, x_8, x_9, x_{10}, x_{11}) = 2, 2, 10, 10 \text{ and } f(q_5, x_{12}, x_{13}) = 2, 10, \text{ and}$$

$$f(q_6, x_{14}, x_{15}) = 2, 10, \text{ and } f(q_7, x_{16}, x_{17}, x_{18}) = 2, 2, 10 \text{ and } f(q_8, x_{19}, x_{20}) = 2, 10, \text{ and}$$

$$f(q_9, x_{21}) = 10 \text{ and } f(q_{10}, x_{22}) = 10 \text{ and } (q_{11}, x_{23}) = 10, \text{ and } f(q_{12}, x_{24}) = 10 \text{ and}$$

$$f(q_{13}, x_{25}) = 10, \text{ and } f(q_{14}, x_{26}) = 2, \text{ and } f(q_{15}, x_{27}, X_{28}) = 2, 10, \text{ and}$$

$$f(q_{16}, x_{29}, x_{30}) = 2, 10, \text{ and } f(q_{17}, x_{31}, x_{32}) = 2, 10 \text{ and}$$

$$f(q_{18}, x_{33}, x_{34}, x_{35}, x_{36}, x_{37}) = 5, 5, 2, 2, 10 \text{ and } f(q_{19}, x_{38}, x_{39}, x_{40}) = 5, 5, 10 \text{ and}$$

$$f(q_{20}, x_{41}, x_{42}, x_{43}) = 5, 5, 10 \text{ and } f(q_{21}, x_{44}, x_{45}, x_{46}) = 5, 5, 10$$

$$Inv(q_0) = \{x \in R : x \geq 10\} \text{ and } Inv(q_1) = \{x \in R : x \geq 10\} \text{ and}$$

$Inv(q2) = \{x \in R : x \geq 20\}$  and  $Inv(q3) = \{x \in R : x \geq 12\}$  and  
 $Inv(q4) = \{x \in R : x \geq 24\}$  and  $Inv(q5) = \{x \in R : x \geq 12\}$  and  
 $Inv(q6) = \{x \in R : x \geq 12\}$  and  $Inv(q7) = \{x \in R : x \geq 14\}$  and  
 $Inv(q8) = \{x \in R : x \geq 12\}$  and  $Inv(q9) = \{x \in R : x \geq 10\}$  and  
 $Inv(q10) = \{x \in R : x \geq 10\}$  and  $Inv(q11) = \{x \in R : x \geq 10\}$  and  
 $Inv(q12) = \{x \in R : x \geq 10\}$  and  $Inv(q13) = \{x \in R : x \geq 10\}$  and  
 $Inv(q14) = \{x \in R : x \geq 2\}$  and  $Inv(q15) = \{x \in R : x \geq 12\}$  and  
 $Inv(q16) = \{x \in R : x \geq 12\}$  and  $Inv(q17) = \{x \in R : x \geq 12\}$  and  
 $Inv(q18) = \{x \in R : x \geq 24\}$  and  $Inv(q19) = \{x \in R : x \geq 20\}$   
 $Inv(q20) = \{x \in R : x \geq 20\}$  and  $Inv(q21) = \{x \in R : x \geq 20\}$

$E1 = \{(q0, q12)(q0, q1), (q1, q2)(q2, q3)(q3, q4)(q4, q5)$   
 $(q5, q6)(q6, q7)(q7, q8)(q8, q9)(q9, q10)(q10, q11)\}$

$E2 = \{(q12, q0)(q12, q13), (q13, q14)(q14, q15)(q15, q16)(q16, q17)$   
 $(q17, q18)(q18, q19)(q19, q20)(q20, q21)(q21, q12)\}$

$G(q0, q1) = \{x \in R : x1 \geq 9.8\}$  and  $G(q1, q2) = \{x \in R : x2 \geq 9.8\}$  and  
 $G(q2, q3) = \{x \in R : x3 \geq 19.8\}$  and  $G(q3, q4) = \{x \in R : x4 \geq 11.8\}$  and  
 $G(q4, q5) = \{x \in R : x5 \leq 23.8\}$  and  $G(q5, q6) = \{x \in R : x6 \leq 11.8\}$  and  
 $G(q6, q7) = \{x \in R : x7 \leq 11.8\}$  and  $G(q7, q8) = \{x \in R : x8 \geq 13.8\}$  and  
 $G(q8, q9) = \{x \in R : x9 \geq 11.8\}$  and  $G(q9, q10) = \{x \in R : x1 \geq 9.8\}$  and  
 $G(q10, q11) = \{x \in R : x1 \geq 9.8\}$  and  $G(q11, q0) = \{x \in R : x1 \geq 9.8\}$

$G(q0, q12) = \{x \in R : x1 \geq 9.8\}$

$G(q12, q13) = \{x \in R : x2 \geq 9.8\}$  and  $G(q13, q14) = \{x \in R : x3 \geq 9.8\}$  and  
 $G(q14, q15) = \{x \in R : x4 \geq 1.8\}$  and  $G(q15, q16) = \{x \in R : x5 \geq 11.8\}$  and  
 $G(q16, q17) = \{x \in R : x6 \geq 11.8\}$  and  $G(q17, q18) = \{x \in R : x7 \geq 11.8\}$  and  
 $G(q18, q19) = \{x \in R : x8 \geq 23.8\}$  and  $G(q19, q20) = \{x \in R : x9 \geq 19.8\}$   
 $G(q20, q21) = \{x \in R : x8 \geq 19.8\}$  and  $G(q21, q12) = \{x \in R : x9 \geq 19.8\}$

$G(q12, q0) = \{x \in R : x8 \geq 9.8\}$

$R((q0, q1), x) = \{x\}$  and  $R((q1, q2), x) = \{x\}$  and  $R((q2, q3), x) = \{x\}$  and  
 $R((q3, q4), x) = \{x\}$  and  $R((q4, q5), x) = \{x\}$  and  $R((q5, q6), x) = \{x\}$  and  
 $R((q6, q7), x) = \{x\}$  and  $R((q7, q8), x) = \{x\}$  and  $R((q8, q9), x) = \{x\}$  and  
 $R((q9, q0), x) = \{x\}$  and  $R((q10, q11), x) = \{x\}$  and  $R((q11, q12), x) = \{x\}$  and  
 $R((q12, q13), x) = \{x\}$  and  $R((q13, q14), x) = \{x\}$  and  $R((q14, q15), x) = \{x\}$  and  
 $R((q15, q16), x) = \{x\}$  and  $R((q16, q17), x) = \{x\}$  and  $R((q17, q18), x) = \{x\}$  and

$R((q18, q19), x) = \{x\}$  and  $R((q19, q10), x) = \{x\}$

$AdjMInputs = \{V_{11}, V_{12}, V_{13}, V_{14}, V_{15}, V_{16}, V_{17}, V_{18}, V_{19},$   
 $V_{110}, V_{111}, V_{112}, V_{113}, V_{114}, V_{115}, V_{116}, V_{117}\}$

$AdjMOutputs = \{V_{01}, V_{02}, V_{03}, V_{04}, V_{05}, V_{06}, V_{07}, V_{08}, V_{09},$   
 $V_{010}, V_{011}, V_{012}, V_{013}, V_{014}, V_{015}, V_{016}, V_{017}\}$

$T_S$  = The synchronization time will depend on the supervisory control. For the proposed system, a time delay of 100 ms is considered for each state change to compare the input and output vectors.

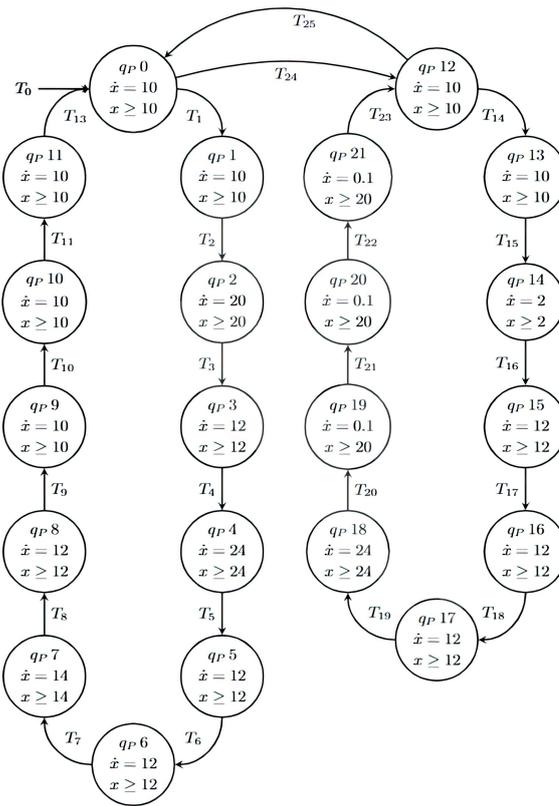


Figure 3. Physic hybrid automaton.

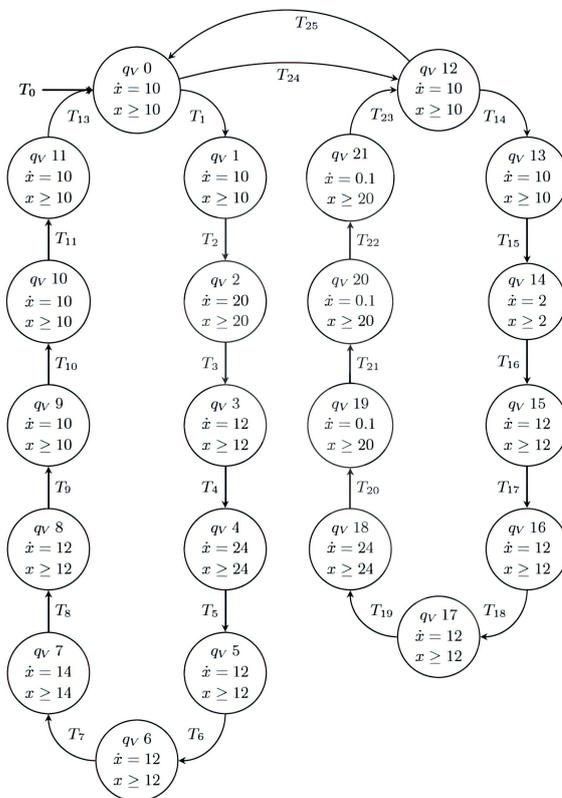
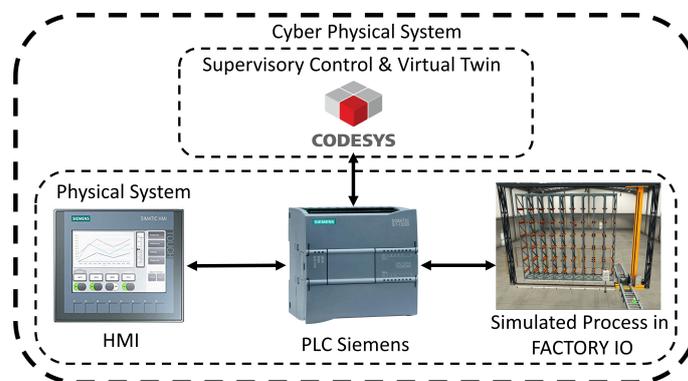


Figure 4. Virtual hybrid automaton.

### 3.2. Cyber–Physical System Implementation

Within the industry, engineering-based system design has been characterized by separating the control system from the construction features of hardware and software. Once the control of the system is completed, a complete simulation is usually performed to verify the optimal operation and correct uncertainty in the model, as well as external disturbances not considered in the model. An important characteristic of industrial systems is their increasing complexity due to the use of advanced technologies for instrumentation, communication and multilevel information processing, posing an important challenge to design and build future generation industrial manufacturing systems. In the design and construction of cyber–physical systems, a new methodology is needed that leads to a reliable and optimal integration of the system elements that were designed independently. The proposed system can be implemented for testing with a programmable logic controller (PLC) and a human–machine interface (HMI). To simulate the process, a graphic environment is used to reproduce the operation of the automatic storage system, Figure 5. The digital system is implemented on a codesys soft plc that incorporates the digital twin and the online diagnostic functions based upon the complex network.



**Figure 5.** Automatic warehouse.

### 3.3. Physical Hybrid Model of Warehouse Operation

The operation of an automatic warehouse is modeled as an hybrid automaton. The states of the physical system are divided into two modes of operation: loading and filling the warehouse and unloading and emptying the warehouse. In addition, the variable that will be measured is the current consumed by the actuators activated in each state. The loading mode of the elevator has the following programming, the  $q_{P0}$  status activates the indicator lights of the buttons and the light column, state  $q_{P1}$  activates two belt motors that transport the product for storage, state  $q_{P2}$  activates the movement of the forks to the left, state  $q_{P3}$  activates the movement of the forks upwards, state  $q_{P4}$  activates the movement of the forks retracting to the left, state  $q_{P5}$  activates an elevator that carries the package to a specific position, state  $q_{P6}$  activates the movement of the forks to the right, state  $q_{P7}$  activates the movement of the forks retracting from the top, state  $q_{P8}$  activates the movement of the forks retracting to the right, state  $q_{P9}$  is the elevator sent to its home position, state  $q_{P10}$  is the elevator in the home position, and state  $q_{P11}$  is the delay time for the system in Figure 6B,C. The hardware configuration of the system can be observed.

The unloading mode has the following programming, as shown in Figure 6A,C: State  $q_{P12}$  activates the indicator lights of the buttons and the light column, state  $q_{P13}$  activates the elevator that goes to the unloading position, state  $q_{P14}$  activates the movement of the forks to the right, state  $q_{P15}$  activates the movement of the forks upwards, state  $q_{P16}$  activates the movement of the forks to the right retractor, state  $q_{P17}$  is the elevator sent to its home position, state  $q_{P18}$  is the movement of the forks to the right being activated, state  $q_{P19}$  is the movement of the forks retracting downwards being activated, state  $q_{P20}$  is the movement of the forks retracting to the right being activated, and state  $q_{P21}$  is the elevator being sent home, as well as the conveyor belts that remove the object from the warehouse

being activated. Figure 3 shows the automaton and Table 1 shows the states and outputs activated in each one.

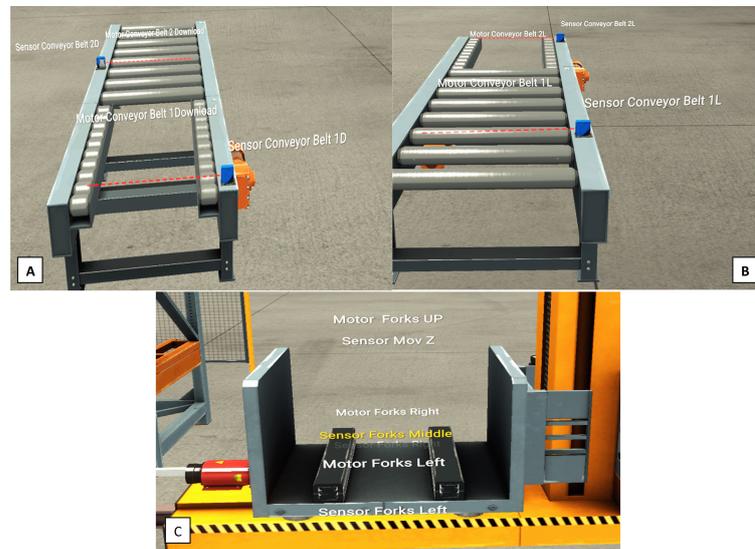


Figure 6. (A) Sensors and actuators in conveyor belt download. (B) Sensors and actuators in conveyor belt load. (C) Sensors and actuators in elevator base.

Table 1. State and Output Activation.

State	Output	State	Output	State	Output	State	Output
$q_{P0}$	Leds <sup>1</sup>	$q_{P5}$	Elevator motors (QW30)	$q_{P10}$	Leds <sup>1</sup>	$q_{P15}$	Elevator motors
$q_{P1}$	Conveyor Motors (Q10.6, Q10.7)	$q_{P6}$	Motor forks (Q11.0 Right)	$q_{P11}$	Elevator motors	$q_{P16}$	Motor forks
$q_{P2}$	Motor forks (Q11.1 Left)	$q_{P7}$	Motor forks (Q11.2 Up)	$q_{P12}$	Motor forks	$q_{P17}$	Motor forks
$q_{P3}$	Motor forks (Q11.2 Up)	$q_{P8}$	Motor forks (Q11.0 Right)	$q_{P13}$	Motor forks	$q_{P18}$	Motor forks
$q_{P4}$	Motor forks (Q11.1 LeftR)	$q_{P9}$	Elevator motors (QW30)	$q_{P14}$	Motor forks	$q_{P19}$	Conveyor motor

<sup>1</sup> Led Outputs = Q10.0, Q10.1, Q11.3, Q10.2, Q10.3, Q11.4, Q10.3.

### 3.4. Digital Hybrid Warehouse Operation Model

Cyber-physical systems are characterized by having a knowledge base in a virtual environment; therefore, to verify that the system is working in its optimal mode, the automata shown in the previous section are modeled with virtual variables, as shown in Figure 4.

### 3.5. Complex Network of Automated Warehouse

A complex network from an automatic warehouse model is presented. The elements of the storage system that are considered to perform the construction of the complex network are transducers and actuators such as buttons, contactors, motors, and PLCs, and are considered as nodes (Figure 7). The links are interpreted as the relationships between the elements and represent the electrical connection between the nodes of the network (Table 2 shows assignments of input and output ports).

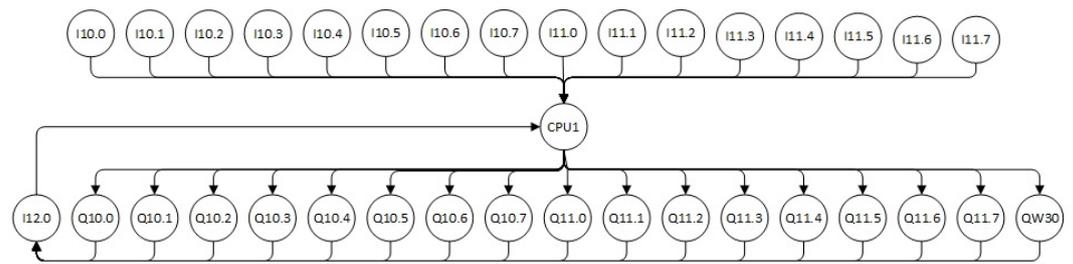


Figure 7. Complex network of the physical system.

Table 2. Input and Output Denomination.

Input	Denomination	Output	Denomination	Input	Denomination	Output	Denomination
I10.0	Start button L	Q10.0	Start (Light)	I11.0	Sensor mov X	Q11.0	Forks right
I10.1	Stop Button L	Q10.1	Start L (Light)	I11.1	Sensor mov Z	Q11.1	Forks left
I10.2	Reset button L	Q10.2	Stop L (Light)	I11.2	Emergency stop	Q11.2	Forks Up
I10.3	Sensor conveyor 1L	Q10.3	Stop (Light)	I11.3	Start button D	Q11.3	Start D (Light)
I10.4	Sensor conveyor 2L	Q10.4	Reset L (Light)	I11.4	Stop button D	Q11.4	Stop D (Light)
I10.5	Sensor forks right	Q10.5	Reset (Light)	I11.5	Reset button D	Q11.5	Reset D (Light)
I10.6	Sensor forks middle	Q10.6	Conveyor 1L	I11.6	Sensor conveyor 1D	Q11.6	Conveyor 1D
I10.7	Sensor forks left	Q10.7	Conveyor 2L	I11.7	Sensor conveyor 2D	Q11.7	Conveyor 2D
				I12.0	Sensor current	QW30	Target position

Construction of Adjacency Matrix

The network constructed from the physical system (Figure 7) is characterized because it is directed, and the adjacency matrix can be constructed. In this matrix, a number “1” is placed for each node that has a link directed to another node. Because there are nodes that are not connected to each other, only the rows of the adjacency matrix that have information on the connections are shown; the interpretation of node/row “I2.0” is that for each activated output (yellow row), the current consumed will be measured and based on the current value, and the state of the hybrid automata creates an equivalence of the activated outputs and is sent to the plc/row (blue) in Figure 8.

	Inputs																	PLC	Output																
	I0.0	I0.1	I0.2	I0.3	I0.4	I0.5	I0.6	I0.7	I1.0	I1.2	I1.3	I1.4	I1.5	I1.6	I1.7	I2.0	Q0.0		Q0.1	Q0.2	Q0.3	Q0.4	Q0.5	Q0.6	Q0.7	Q1.0	Q1.1	Q1.2	Q1.3	Q1.4	Q1.5	Q1.6	Q1.7	QW30	
I2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
PLC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Figure 8. Adjacency Matrix.

From the adjacency matrix (see complete matrix in Appendix A), it is observed that, because the network is of the directed type, only the rows of I2.0 and PLC will change, providing there is information to determine the optimal functioning of the system.

3.6. Data Models and Equations

3.6.1. Digital Data Model

The model of the system is based on the monitoring and comparison of the data obtained from the behavior of the physical system and the data obtained from the ideal behavior of the virtual system. The adjacency matrix is composed of rows of data representing information from the sensors and actuators; therefore, the same information can be represented with a set of vectors. For physical inputs, an array of physical binary inputs  $DI_{PS_k}$  is defined, where each place represents the value of different sensors  $S_p$  with corresponding physical PLC addresses. The virtual environment simulates the behavior of

the physical system, and there is a corresponding array  $DI_{VS_k}$  of the virtual binary with inputs for simulated sensors  $Sv$ , as shown below:

$$DI_{PS_k} = [Sp_{I0.0}, Sp_{I0.1}, Sp_{I0.2}, Sp_{I0.3}, Sp_{I0.4}, Sp_{I0.5}, Sp_{I0.6}, Sp_{I0.7}, Sp_{I1.0}, Sp_{I1.1}, Sp_{I1.2}, Sp_{I1.3}, Sp_{I1.4}, Sp_{I1.5}, Sp_{I1.6}, Sp_{I1.7}] \quad (1)$$

$$DI_{VS_k} = [Sv_{I0.0}, Sv_{I0.1}, Sv_{I0.2}, Sv_{I0.3}, Sv_{I0.4}, Sv_{I0.5}, Sv_{I0.6}, Sv_{I0.7}, Sv_{I1.0}, Sv_{I1.1}, Sv_{I1.2}, Sv_{I1.3}, Sv_{I1.4}, Sv_{I1.5}, Sv_{I1.6}, Sv_{I1.7}] \quad (2)$$

As can be observed in the adjacency matrix (Appendix A), only one column (strong blue) provides information on the actuators that have been activated by showing a value of “1” when the controller has activated them. The algorithm holds the information of activated outputs by means of a row arrangement (gray row). This arrangement of output elements can be represented as a vector. For the physical output, an array of physical binary outputs  $DO_{PS_k}$  is defined, where each place represents the value of different output  $Op$  with corresponding physical PLC addresses. For the virtual system, there is a corresponding array  $DO_{VS_k}$  of virtual binary outputs for simulated outputs  $Ov$ , as follows:

$$DO_{PS_k} = [Op_{Q0.0}, Op_{Q0.1}, Op_{Q0.2}, Op_{Q0.3}, Op_{Q0.4}, Op_{Q0.5}, Op_{Q0.6}, Op_{Q0.7}, Op_{Q1.0}, Op_{Q1.1}, Op_{Q1.2}, Op_{Q1.3}, Op_{Q1.4}, Op_{Q1.5}, Op_{Q1.6}, Op_{Q1.7}, Op_{Qw0}] \quad (3)$$

$$DO_{VS_k} = [Ov_{Q0.0}, Ov_{Q0.1}, Ov_{Q0.2}, Ov_{Q0.3}, Ov_{Q0.4}, Ov_{Q0.5}, Ov_{Q0.6}, Ov_{Q0.7}, Ov_{Q1.0}, Ov_{Q1.1}, Ov_{Q1.2}, Ov_{Q1.3}, Ov_{Q1.4}, Ov_{Q1.5}, Ov_{Q1.6}, Ov_{Q1.7}, Ov_{Qw0}] \quad (4)$$

### 3.6.2. Analog Data Model

The model also considers vectors for holding data from analog input and outputs connected to physical sensors and actuators. The value of the current consumed by each sensor or actuator is shown. Therefore, the vectors  $AI_{PS_k}$  and  $AI_{VS_k}$  represent the values of analog physical inputs  $AI_P$  and analog virtual inputs  $AI_V$ , as follows:

$$AI_{PS_k} = [AIp_{I0.0}, AIp_{I0.1}, AIp_{I0.2}, AIp_{I0.3}, AIp_{I0.4}, AIp_{I0.5}, AIp_{I0.6}, AIp_{I0.7}, AIp_{I2.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I1.0}, AIp_{I2.0}] \quad (5)$$

$$AI_{VS_k} = [AIv_{I0.0}, AIv_{I0.1}, AIv_{I0.2}, AIv_{I0.3}, AIv_{I0.4}, AIv_{I0.5}, AIv_{I0.6}, AIv_{I0.7}, AIv_{I2.0}, AIv_{I1.0}, AIv_{I1.0}, AIv_{I1.0}, AIv_{I1.0}, AIv_{I1.0}, AIv_{I1.0}, AIv_{I1.0}, AIv_{I2.0}] \quad (6)$$

The function of the sensor I2.0 is to provide a measurement of the current consumed by all the activated actuators.

The  $AO_{PS_k}$  and  $AO_{VS_k}$  represent the vectors of analog physical outputs  $AO_P$  and analog virtual outputs  $AO_V$ , as follows:

$$AO_{PS_k} = [AOp_{Q0.0}, AOp_{Q0.1}, AOp_{Q0.2}, AOp_{Q0.3}, AOp_{Q0.4}, AOp_{Q0.5}, AOp_{Q0.6}, AOp_{Q0.7}, AOp_{Q1.0}, AOp_{Q1.1}, AOp_{Q1.2}, AOp_{Q1.3}, AOp_{Q1.4}, AOp_{Q1.5}, AOp_{Q1.6}, AOp_{Q1.7}, AOp_{Qw30}] \quad (7)$$

$$AO_{VS_k} = [AOv_{Q0.0}, AOv_{Q0.1}, AOv_{Q0.2}, AOv_{Q0.3}, AOv_{Q0.4}, AOv_{Q0.5}, AOv_{Q0.6}, AOv_{Q0.7}, AOv_{Q1.0}, AOv_{Q1.1}, AOv_{Q1.2}, AOv_{Q1.3}, AOv_{Q1.4}, AOv_{Q1.5}, AOv_{Q1.6}, AOv_{Q1.7}, AOv_{Qw30}] \quad (8)$$

### 3.7. Development of the System

In the cyber–physical system, the proposed supervisory controller will make decisions based on the comparison of the operation of the physical and the virtual systems (Figure 9).

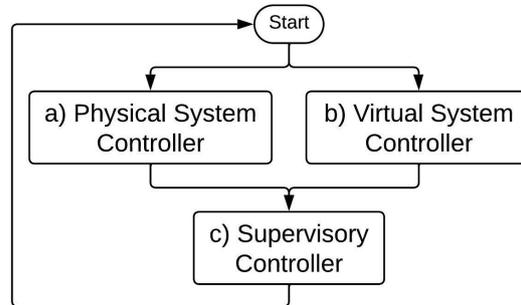


Figure 9. (a) Physical System Controller; (b) Virtual System Controller; (c) Supervisory Controller.

#### 3.7.1. Physical System Development

This section explains, with a mathematical algorithm, the operation of the physical system (Figure 9a) with its digital variables,  $DI_{PS_k}$  and  $DO_{PS_k}$ , and analog variables,  $AI_{PS_k}$  and  $AO_{PS_k}$ .

The operation of Algorithm 1, for the physical system, assigns values to the digital and analog input vectors (Equations (9) and (13)) as well as to the output vectors (Equations (11) and (14)). If there is a change of state  $St_{PS_k}$  in the physical system, the inputs and outputs are measured and assigned correspondingly to the digital input–output and analog input–output vectors. Within the digital analysis, the difference between the current state and the previous state is calculated with Equation (9) for digital inputs  $DId_{PS_k}$  and Equation (11) for digital outputs  $DOd_{PS_k}$ . The input and output differences are stored in their respective memory equations,  $DIm_{PS_k}$  (Equation (10)) or  $DOM_{PS_k}$  (Equation (12)). Regarding the analog analysis, it consists of filtering  $(1 - \alpha)$  the input and output currents and storing their values in an analog input memory,  $AIm_{PS_k}$ , given by (Equation (13)), and an analog output memory,  $AOm_{PS_k}$ , according to (Equation (14)), respectively.

---

**Algorithm 1:** Development of the digital and analog physical system:

---

**Data:**  $DI_{PS_k}, DO_{PS_k}, AI_{PS_k}, AO_{PS_k}$

**Result:**  $DIm_{PS_k}, DOM_{PS_k}, AIm_{PS_k}, AOm_{PS_k}$

**if**

$$(St_{PS_k} \neq St_{PS_{k-1}})$$

**then**

(Digital Physical System):

$$DId_{PS_k} = |DI_{PS_k} - DI_{PS_{k-1}}| \tag{9}$$

$$DIm_{PS_k} = DIm_{PS_{k-1}} + DId_{PS_k} \tag{10}$$

$$DOd_{PS_k} = |DO_{PS_k} - DO_{PS_{k-1}}| \tag{11}$$

$$DOM_{PS_k} = DOM_{PS_{k-1}} + DOd_{PS_k} \tag{12}$$

(Analog Physical System):

$$AIm_{PS_k} = \alpha AIm_{PS_{k-1}} - (1 - \alpha) AI_{PS_k} \tag{13}$$

$$AOm_{PS_k} = \alpha AOm_{PS_{k-1}} - (1 - \alpha) AO_{PS_k} \tag{14}$$

**end**

---

### 3.7.2. Virtual System Development

This section explains, with a mathematical algorithm, the operation of the virtual system, with its digital variables  $DI_{VS_k}$  and  $DO_{VS_k}$ , and analog variables  $AI_{VS_k}$  and  $AO_{VS_k}$  representing the function (Figure 9b) in the following:

Algorithm 2, for the virtual system, is similar to the algorithm of the physical system, except that, in this case, the ideal numerical simulation of the system is carried out, which will help to corroborate the optimal functioning of the physical system. Values are assigned to the vectors of the digital virtual input and digital virtual output  $DI_{VS_k}$  and  $DO_{VS_k}$ . In turn, the virtual input and analog virtual output vectors are assigned  $AI_{VS_k}$  and  $AO_{VS_k}$ . Unlike Algorithm 1, the signal that was activated in the physical system is expected to start the ideal virtual model, by  $ST_{PS_k}$ , to replicate the event in the virtual system, based on the virtual hybrid automaton (Figure 4). The virtual digital and analog values from inputs and outputs are added. An analysis of the previous virtual state and the current digital virtual state is performed with Equation (15) for  $DId_{VS_k}$  and Equation (17) for  $DOd_{VS_k}$ . These input and output differences are assigned to  $DIm_{VS_k}$  by (16) and  $DOm_{VS_k}$  by (18), respectively. The analogous ideal simulation produces the input and output vectors  $AIm_{VS_k}$  and  $AOm_{VS_k}$ , which are filtered by  $\alpha$  and  $(1 - \alpha)$  to produce mean values stored in (19) and (20), respectively, for the analysis of the supervisory controller.

---

**Algorithm 2:** Development of the digital and analog virtual system:

---

**Data:**  $DI_{VS_k}, DO_{VS_k}, AI_{VS_k}, AO_{VS_k}$

**Result:**  $DIm_{VS_k}, DOm_{VS_k}, AIm_{VS_k}, AOm_{VS_k}$

---

if

$$(St_{VS_k} \neq St_{VS_{k-1}})$$

then

(Digital Virtual System):

$$DId_{VS_k} = |DI_{VS_k} - DI_{VS_{k-1}}| \tag{15}$$

$$DIm_{VS_k} = DIm_{VS_{k-1}} + DId_{VS_k} \tag{16}$$

$$DOd_{VS_k} = |DO_{VS_k} - DO_{VS_{k-1}}| \tag{17}$$

$$DOm_{VS_k} = DOm_{VS_{k-1}} + DOd_{VS_k} \tag{18}$$

(Analog Virtual System):

$$AIm_{VS_k} = \alpha AIm_{VS_{k-1}} - (1 - \alpha) AI_{VS_k} \tag{19}$$

$$AOm_{VS_k} = \alpha AOm_{VS_{k-1}} - (1 - \alpha) AO_{VS_k} \tag{20}$$

end

---

### 3.7.3. Digital Control Operation and Fault Diagnosis

To diagnose faults, the physical input and output memory vectors and the virtual input and output vectors are continuously compared (Figure 9c, but with a synchronization time, considering the change of state in both the physical automata and the virtual automata, and it is implemented Algorithm 3.

**Algorithm 3:** Development of the digital control

---

**Data:**  $Dim_{PS_k}, Aim_{PS_k}, Dim_{VS_k}, Aim_{VS_k}, Dom_{PS_k}, AOm_{PS_k}, Dom_{VS_k}, AOm_{PS_k}$   
**Result:**  $DIP_{S_{Diagnosis}}, DOPS_{Diagnosis}$

```

while ( $St_{PS_k} \& St_{VS_k}$ ) = TRUE do
  if ( $Dim_{PS_k} \neq Dim_{VS_k}$ ) or ( $Aim_{PS_k} \neq Aim_{VS_k}$ ) then
     $DIP_{S_{DF}}$  ; /* DF:Diagnosis of Digital Input Fault */
     $AIP_{S_{DF}}$  ; /* DF:Diagnosis of Analog Input Fault */
    Stop the System
  else
    if ( $Dim_{PS_k} = Dim_{VS_k}$ ) & ( $Aim_{PS_k} = Aim_{VS_k}$ ) then
       $DIP_{S_{WC}}$  ; /* WC:Digital Input Working Correct */
       $AIP_{S_{WC}}$  ; /* WC:Digital Input Working Correct */
      Delay
    end
  end
  if  $Dom_{PS_k} \neq Dom_{VS_k}$  or ( $AOm_{PS_k} \neq AOm_{VS_k}$ ) then
     $DOPS_{Diagnosis}$  ; /* DF:Diagnosis of Digital Input Fault */
     $AOPS_{Diagnosis}$  ; /* DF:Diagnosis of Digital Input Fault */
    Stop the System
  else
    if  $Dom_{PS_k} = Dom_{VS_k}$  & ( $Aim_{PS_k} = Aim_{VS_k}$ ) then
       $DOPS_{WC}$  ; /* WC:Digital Output Working Correct */
       $AOPS_{WC}$  ; /* WC:Analog Output Working Correct */
      Delay
    end
  end
end
end

```

---

**4. Results and Discussion**

The presented approach of a cyber–physical system at a supervisory control level is intended for the monitoring and diagnosis of hardware and software failures in an automated storage and retrieval system, as shown in Figure 5. The virtual system is implemented by a real-time numerical simulation to model the behavior of the physical system. Hybrid automata are used to design the control functions of the physical controller and to ease the implementation of the virtual controller. The sensors and actuators were defined as complex network nodes, and their asynchronous activation was defined as events affecting the weight links. The complex network and the hybrid automata for both the digital twin and the physical system were coordinated. Every time that the physical controller activates an actuator or reads a sensor, data vectors save the information and are compared with the corresponding data vectors in the virtual controller. To detect failures in the hardware components, adjacency matrices are compared, and to detect failure in the execution of the sequential logic, the current states are compared. The simulation and implementation of the system present a problem of synchronization between the physical and virtual systems. This problem was solved by considering a time of delay between the data transfer and the change in the physical state and virtual state.

Figure 9 shows the execution of the physical system and the virtual system. It can be observed that the virtual and physical hybrid automata change at the same time with an specific time of synchronization. In Figure 10, input and output changes can also be observed, besides changes between 0 and 1. The system updates and saves the values from the activation and deactivation events, and the algorithms detect discrepancies of data values of the physical controller with respect to data values of the virtual system. In this way, it is possible to detect failures in sensors, actuators, and in the execution of the control logic.

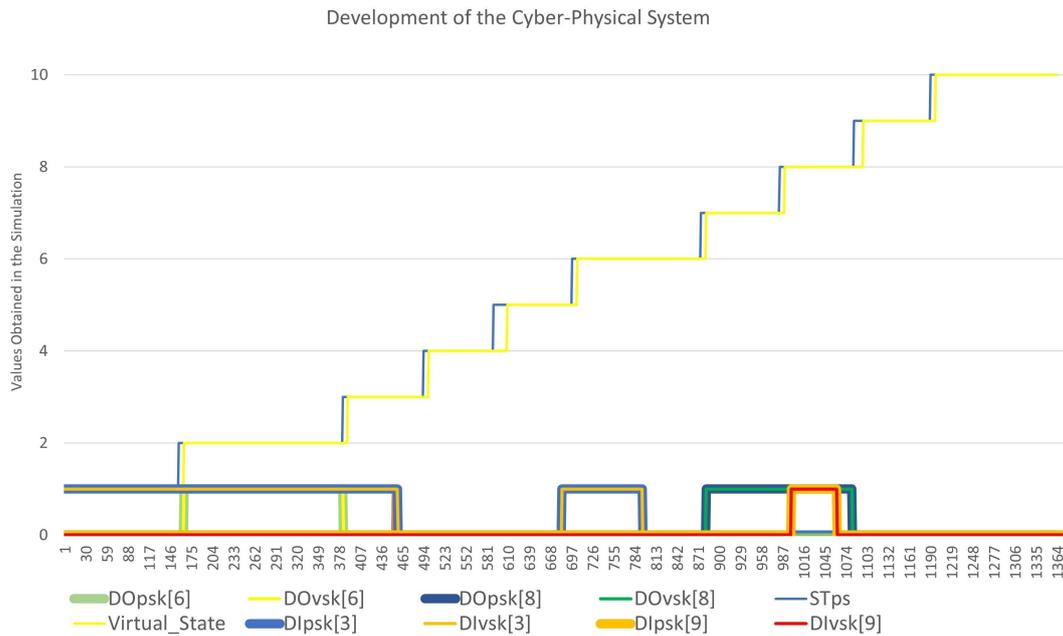


Figure 10. Development of the cyber-physical system.

To corroborate the operation of the algorithm, the stored information presented in the Figure 9 was examined. In the case where there are no failures, the data vectors are identical, and in the case that the data are not equal, there are failures. Figure 11 shows the difference in the data vectors with red rectangles to indicate failures and the green rectangles to indicate that the elements are working correctly.

DOmpsk1	DOmvsk1	DImpsk1	DImvsk1
DOmpsk1[0] 0	DOmvsk1[0] 0	DImpsk1[0] 1	DImvsk1[0] 1
DOmpsk1[1] 0	DOmvsk1[1] 0	DImpsk1[1] 2	DImvsk1[1] 2
DOmpsk1[2] 0	DOmvsk1[2] 0	DImpsk1[2] 0	DImvsk1[2] 0
DOmpsk1[3] 0	DOmvsk1[3] 0	DImpsk1[3] 0	DImvsk1[3] 2
DOmpsk1[4] 0	DOmvsk1[4] 0	DImpsk1[4] 0	DImvsk1[4] 1
DOmpsk1[5] 0	DOmvsk1[5] 0	DImpsk1[5] 0	DImvsk1[5] 0
DOmpsk1[6] 0	DOmvsk1[6] 1	DImpsk1[6] 0	DImvsk1[6] 0
DOmpsk1[7] 0	DOmvsk1[7] 1	DImpsk1[7] 0	DImvsk1[7] 0
DOmpsk1[8] 0	DOmvsk1[8] 0	DImpsk1[8] 0	DImvsk1[8] 0
DOmpsk1[9] 0	DOmvsk1[9] 1	DImpsk1[9] 0	DImvsk1[9] 0
DOmpsk1[10] 0	DOmvsk1[10] 1	DImpsk1[10] 0	DImvsk1[10] 0
DOmpsk1[11] 0	DOmvsk1[11] 0	DImpsk1[11] 0	DImvsk1[11] 0
DOmpsk1[12] 0	DOmvsk1[12] 0	DImpsk1[12] 1	DImvsk1[12] 1
DOmpsk1[13] 0	DOmvsk1[13] 0	DImpsk1[13] 0	DImvsk1[13] 0
DOmpsk1[14] 0	DOmvsk1[14] 0	DImpsk1[14] 0	DImvsk1[14] 0
DOmpsk1[15] 0	DOmvsk1[15] 0	DImpsk1[15] 0	DImvsk1[15] 0
DOmpsk1[16] 1	DOmvsk1[16] 1	DImpsk1[16] 0	DImvsk1[16] 0

Figure 11. Diagnosis of faults.

Finally, Figure 12 shows an operator panel with indicators to display the elements that present a failure.

In order to evaluate the efficiency of the system, four different cases (row represents the number of simulations) of failure diagnosis were evaluated in different stages of the process, and the tests were carried out both in the loading and unloading configuration

of the warehouse. It is observed that the system is capable of diagnosing and locating the failures induced in inputs and outputs selected in Table 3; in the third column, a specific number of induced failures is selected, and it is observed that the system in the eighth column can satisfactorily diagnose and locate all the induced failures.

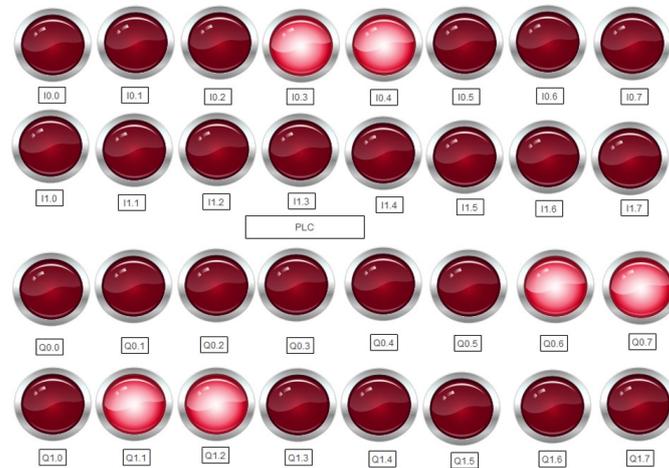


Figure 12. Location of faults.

Table 3. Input and Output Denomination.

No. Simulation	State	Input/Output Induce Fault	Input Physical Vector	Input Virtual Vector	Output Physical Vector	Output Virtual Vector	Fault Detection
1	$q_{P3}$	I10.4, Q11.4	I10.4 = 0	I10.4 = 1	Q11.4 = 0	Q11.4 = 1	I10.4, Q11.4—Correct
2	$q_{P9}$	I10.5, Q11.0	I10.5 = 0	I10.5 = 1	Q11.0 = 0	Q11.0 = 1	I10.5, Q11.0—Correct
3	$q_{P14}$	I11.0, I11.1, Q11.4	I11.0 = 0, I11.1 = 0	I11.0 = 1, I11.1 = 1	Q11.4 = 0	Q11.4 = 1	I11.0, I11.1, Q11.4—Correct
4	$q_{P19}$	I10.5, Q11.6, Q11.7	I10.5 = 0	I10.5 = 1	Q11.6 = 0, Q11.7 = 0	Q11.6 = 1, Q11.7 = 1	I10.5, Q11.6, Q11.7—Correct

The proposed approach for a basic cyber–physical system is composed of a virtual twin simulating the process of the physical system. The virtual twin executed simultaneously with the operation of the physical systems allows for a comparison in real time. When the behavior of both do not match, a fault is diagnosed. The presented approach can be extended to diverse manufacturing systems.

### 5. Conclusions

This work proposed a novel scheme of a basic cyber–physical system that incorporates a virtual system at the supervisory control level. The virtual system is executed in a real-time simulation, providing means to monitor and diagnose failures in sensors, actuators and in the execution of the sequential logic control. In this approach, the digital twin is mainly a numerical simulation of the process and its controller, which is made up of algorithms that combine a hybrid automata as a virtual controller, along with the sensors and actuators modeled in a complex network. The implemented system acts as a supervisory control system capable of diagnosing and locating failures in physical systems in real time. The algorithms were validated in an automated storage and retrieval system. The implementation was required to solve some synchronization problems presented in the coordination of the digital twin with the physical system. One of the main tests showed



## References

1. TcaciucGherasim, S.A. A Solution for an Industrial Automation and SCADA System. In Proceedings of the 2022 International Conference and Exposition on Electrical And Power Engineering (EPE), Iași, Romania, 20–22 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 297–300.
2. Khadra, A.; Rammal, R. SCADA System for Solar Backup Power System Automation. In Proceedings of the 2022 International Conference on Smart Systems and Power Management (IC2SPM), Beirut, Lebanon, 10–12 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 75–79.
3. Nuhel, A.K.; Sazid, M.M.; Ahmed, K.; Bhuiyan, M.N.M.; Hassan, M.Y.B. A PI Controller-based Water Supplying and Priority Based SCADA System for Industrial Automation using PLC-HMI Scheme. In Proceedings of the 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kinabalu, Malaysia, 13–15 September 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
4. Hamzah, M.; Islam, M.M.; Hassan, S.; Akhtar, M.N.; Ferdous, M.J.; Jasser, M.B.; Mohamed, A.W. Distributed Control of Cyber Physical System on Various Domains: A Critical Review. *Systems* **2023**, *11*, 208. [[CrossRef](#)]
5. Zhang, K.; Shi, Y.; Karnouskos, S.; Sauter, T.; Fang, H.; Colombo, A.W. Advancements in industrial cyber-physical systems: An overview and perspectives. *IEEE Trans. Ind. Inform.* **2022**, *19*, 716–729. [[CrossRef](#)]
6. Villalonga, A.; Beruvides, G.; Castano, F.; Haber, R.E. Cloud-based industrial cyber–physical system for data-driven reasoning: A review and use case on an industry 4.0 pilot line. *IEEE Trans. Ind. Inform.* **2020**, *16*, 5975–5984. [[CrossRef](#)]
7. Alshalalfah, A.L.; Mohamed, O.A.; Ouchani, S. A framework for modeling and analyzing cyber-physical systems using statistical model checking. *Internet Things* **2023**, *22*, 100732. [[CrossRef](#)]
8. Li, P.; Zhang, F.; Yang, Y.; Ma, X.; Yao, S.; Yang, P.; Zhao, Z.; Lai, C.S.; Lai, L.L. The integrated modeling of microgrid cyber physical system based on hybrid automaton. *Front. Energy Res.* **2022**, *10*, 748828. [[CrossRef](#)]
9. Staroletov, S. Automatic proving of stability of the cyber-physical systems in the sense of Lyapunov with KeYmaera. In Proceedings of the 2021 28th Conference of Open Innovations Association (FRUCT), Moscow, Russia, 25–29 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 431–438.
10. Ravasio, D.; Tuissi, L.; Spinelli, S.; Ballarino, A. A Compressed Air Network Energy-Efficient Hierarchical Unit Commitment and Control. In Proceedings of the 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Sydney, Australia, 3–5 March 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 469–473.
11. Guo, Z.; Zhang, Y.; Zhao, X.; Song, X. CPS-based self-adaptive collaborative control for smart production-logistics systems. *IEEE Trans. Cybern.* **2020**, *51*, 188–198. [[CrossRef](#)] [[PubMed](#)]
12. Sun, T.; Xia, W.; Zhao, X.; Sun, X.M. A Novel Mathematical Characterization for Switched Linear Systems Based on Automata and Its Stabilizability Analysis. *IEEE Trans. Control. Netw. Syst.* **2023**. [[CrossRef](#)]
13. Alonso, M.; Turanzas, J.; Amaris, H.; Ledo, A.T. Cyber-physical vulnerability assessment in smart grids based on multilayer complex networks. *Sensors* **2021**, *21*, 5826. [[CrossRef](#)] [[PubMed](#)]
14. Zhao, Z.; Xu, Y.; Li, Y.; Zhao, Y.; Wang, B.; Wen, G. Sparse actuator attack detection and identification: A data-driven approach. *IEEE Trans. Cybern.* **2023**, *53*, 4054–4064. [[CrossRef](#)] [[PubMed](#)]
15. Pósfai, M.; Barabasi, A.L. *Network Science*; Citeseer: London, UK, 2016.
16. Giudici, R. *Introducción a la Teoría de Grafos*; Equinoccio: Baruta, Miranda, Venezuela, 1997.
17. Mu, D.; Yue, X.; Ren, H. Robustness of Cyber-Physical Supply Networks in Cascading Failures. *Entropy* **2021**, *23*, 769. [[CrossRef](#)] [[PubMed](#)]
18. Platzer, A. *Logical Foundations of Cyber-Physical Systems*; Springer: Cham, Switzerland, 2018; Volume 662.
19. Lin, H.; Antsaklis, P.J. *Hybrid Dynamical Systems*; Foundations and Trends® in Systems and Control: Delft, The Netherlands, 2015.
20. Meyn, S. *Control Techniques for Complex Networks*; Cambridge University Press: Cambridge, UK, 2008.

**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.