



**HAL**  
open science

# **A Proof-of-Concept Digital Twin for Real-Time Simulation: Leveraging a Model-Based Systems Engineering Approach**

Bulent Soykan, Grégoire Blanc, Ghaith Rabadi

## ► **To cite this version:**

Bulent Soykan, Grégoire Blanc, Ghaith Rabadi. A Proof-of-Concept Digital Twin for Real-Time Simulation: Leveraging a Model-Based Systems Engineering Approach. *IEEE Access*, 2025, 13, pp.58899-58912. <10.1109/ACCESS.2025.3557367>. <hal-05028584>

**HAL Id: hal-05028584**

**<https://imt-mines-ales.hal.science/hal-05028584v1>**

Submitted on 10 Apr 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Received 28 February 2025, accepted 24 March 2025, date of publication 2 April 2025, date of current version 9 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3557367

## RESEARCH ARTICLE

# A Proof-of-Concept Digital Twin for Real-Time Simulation: Leveraging a Model-Based Systems Engineering Approach

BULENT SOYKAN<sup>1</sup>, GRÉGOIRE BLANC<sup>2</sup>, AND GHAITH RABADI<sup>3</sup>

<sup>1</sup>Institute for Simulation and Training, University of Central Florida, Orlando, FL 32826, USA

<sup>2</sup>IMT Mines Alès, 30100 Alès, France

<sup>3</sup>School of Modeling, Simulation, and Training, University of Central Florida, Orlando, FL 32826, USA

Corresponding author: Bulent Soykan (Bulent.Soykan@ucf.edu)

**ABSTRACT** This paper presents a proof-of-concept digital twin designed to enhance industrial process monitoring and maintenance through real-time and bidirectional communication. The integrated approach combines a physical twin—constructed using LEGO<sup>®</sup> Spike Prime kits and multiple Raspberry Pi 4 microcontrollers—with a digital twin modeled in FlexSim discrete-event simulation software package. Data is continuously synchronized via the MQTT protocol, allowing the digital twin to accurately mirror key operational parameters such as object sorting status, conveyor speed, and splitter position. The system leverages Model-Based Systems Engineering (MBSE) techniques, implemented with Capella and the Arcadia methodology, to support a robust design and integration of physical and digital components. Experimental scenarios, including normal operation and simulated malfunctions, demonstrate the system's ability to trigger corrective actions and provide valuable insights for predictive maintenance. The results confirm that such a modular digital twin approach can improve real-time simulation, fault detection, and operational efficiency in industrial applications.

**INDEX TERMS** Digital twin, real-time communication, model-based systems engineering, industrial automation, cyber-physical systems.

## I. INTRODUCTION

Digital twins are dynamic virtual representations of physical assets, processes or systems [1]. Digital twins are constantly refreshed with real-time data from sensors and Internet of Things (IoT) devices, allowing the digital model to accurately reflect the current state of its physical counterpart [2]. This digital replication enables simulation, monitoring, and analysis of operational behaviors under various conditions. In industrial manufacturing, digital twins are important because they support real-time and proactive decision-making. They allow users and decision-makers to test control strategies, predict maintenance needs, and optimize performance without interrupting the actual production process. The bidirectional communication between the physical

system and its digital twin facilitates rapid response to anomalies and enhances overall system reliability [3]. Digital twins also create a risk-free environment for experimenting with process changes, leading to reduced downtime, improved quality, and cost-effective operations [2]. Their ability to replicate complex manufacturing processes in a virtual space is critical for achieving operational efficiency and fostering innovation. According to a recent Gartner report, the digital twin industry is anticipated to expand substantially, with worldwide earnings forecasted to rise from \$35 billion in 2024 to \$379 billion by 2034 [4]. This significant growth projection underscores the strategic importance of digital twins in revolutionizing industrial manufacturing systems.

As promising as these advantages are, many organizations still face significant challenges in transitioning digital twin prototypes into fully operational systems [5]. Despite the potential to revolutionize manufacturing processes,

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez<sup>1b</sup>.

successful implementation is often hindered by a lack of rigorous development frameworks and an underdeveloped organizational culture [6]. Without robust testing, validation, and active user involvement from the outset, digital twins may fail to meet performance expectations, limiting their impact on live operations.

To overcome these challenges, it is essential to adopt a structured systems engineering approach, such as MBSE, that integrates established architectural principles with a clear mission and vision [7]. Specifically, MBSE provides tangible benefits like enhanced requirements traceability from concept to implementation, early identification of design flaws through model simulation and analysis, and improved communication across engineering disciplines, crucial for managing the complexity of digital twin development. This approach involves formulating an envisioned future for digital twin implementations, centered on the people, process, and technology triad. By incorporating user feedback early and adhering to proven methodologies and standards, organizations can ensure that digital twins are carefully designed, extensively validated, and fully aligned with operational requirements. Such rigor not only prepares digital twin systems for scalability but also ensures that they deliver tangible benefits from the very beginning.

Ultimately, applying systems engineering principles bridges the gap between prototype and practice, enabling digital twins to operate reliably in increasingly complex industrial environments. This disciplined methodology supports proactive decision-making, enhances real-time monitoring, and underpins predictive maintenance strategies by providing continuous data-driven insights. As digital twins transition from concept to indispensable operational tools, the integration of disciplined engineering practices ensures that they not only facilitate immediate process improvements but also foster long-term innovation and efficiency.

In modern industrial environments, systems are becoming increasingly complex and interconnected [8]. This complexity demands continuous, real-time monitoring to detect anomalies as soon as they occur. Improved system monitoring provides the data needed to understand operational status at all times, thereby reducing unforeseen failures and unplanned downtime. Real-time data feeds also support better decision-making [9]. With accurate, up-to-the-minute information, operators and managers can quickly identify issues and adjust control parameters on the fly. This dynamic approach enables more efficient responses to emerging problems and contributes to optimized process control. Also, predictive maintenance relies on ongoing data collection and analysis to anticipate equipment failures before they happen [10]. By identifying trends and subtle shifts in system behavior, maintenance activities can be scheduled proactively rather than reactively. This not only minimizes the cost of repairs and downtime but also extends the lifetime of critical industrial components [11].

The main goal of this research is to develop a real-time synchronized proof-of-concept digital twin that pioneers a low-cost, modular framework using consumer-grade hardware (LEGO<sup>®</sup>, Raspberry Pi) and standard protocols (MQTT), validated through an MBSE approach, to enhance industrial process monitoring and maintenance. This involves designing a physical object-sorting mechanism and integrating it with a digital twin that mirrors its behavior in real time. The objective is to demonstrate that such a system can be used to enhance process monitoring, drive dynamic decision-making, and support predictive maintenance in industrial applications. The contributions of this work are multifaceted and serve to advance digital twin technology for industrial applications. One key contribution is the integration of Raspberry Pi microcontrollers to manage the physical twin, ensuring real-time data collection and control. In addition, the utilization of LEGO<sup>®</sup> Spike kits enables the construction of a modular object-sorting system that can be rapidly prototyped and adapted, providing significant flexibility in system design. The development of a digital twin using FlexSim further enhances the approach by offering 3D modeling and simulation of the physical system, which facilitates real-time visualization and analysis. The implementation of the MQTT protocol enables robust, bidirectional communication between the physical and digital twins, ensuring continuous synchronization and immediate feedback loops. Together, these integrated elements collectively demonstrate a novel approach for enhancing system monitoring, decision-making, and predictive maintenance in industrial processes.

The remainder of the paper is organized as follows: Section II reviews the relevant literature and background, setting the stage for the research by outlining existing digital twin technologies and systems engineering methodologies. Section III presents the detailed system design and methodology, describing the architecture of the modular digital twin, its integration with the physical system, and the communication framework employed. Section IV outlines the experimental results, highlighting the performance metrics observed under both normal conditions and simulated malfunctions. Section V provides a discussion of the findings, addressing the benefits, limitations, and practical implications of the proposed approach. Finally, Section VI concludes the paper by summarizing the key contributions and proposing directions for future research.

## II. LITERATURE REVIEW AND RELATED WORK

This section reviews the state-of-the-art research in digital twins, Model-Based Systems Engineering (MBSE), and communication protocols in industrial IoT, highlighting key trends and challenges that underpin our research approach.

### A. DIGITAL TWINS

Digital twin serves as a dynamic virtual counterpart of a physical asset, process or system, operating in real-time or near real-time, using data and specific characteristics

from the physical system [12]. Digital twins have become a popular concept in both industry and academia. They promise enhanced monitoring, prediction, and optimization capabilities [13]. However, much of the academic literature concentrates on isolated, application-specific implementations rather than on approaches that can be broadly applied. This narrow focus means that while digital twins can be highly effective within a given context, the methods developed are rarely transferable to different domains.

One frequently observed limitation is that many studies describe digital twins by targeting a single component of a system. Such work tends to neglect the complex coupling and interdependencies inherent in real-world applications. For example, a digital twin designed for predictive maintenance in manufacturing might concentrate solely on monitoring a specific machine part. In doing so, it fails to consider interactions with other subsystems, such as the influence of environmental factors or interconnected processes. Similarly, in the aerospace sector, digital twin models might simulate engine performance in isolation, ignoring their coupling with thermal management or structural integrity systems. These single-component models provide only a partial view of system behavior, reducing their overall utility in complex settings [14].

Another challenge is that many of the existing digital twin implementations require considerable technical expertise and substantial deployment resources [15]. The specialized knowledge needed to develop these systems often confines their utility to well-funded research laboratories or large industrial players. This requirement limits broader adoption and slows the development of more generalized methodologies. Without clear, accessible technical descriptions that bridge the gap between narrow, application-specific solutions [12], [13] and comprehensive, system-level models, the promise of digital twins remains under-realized. Our approach addresses this by emphasizing a generalizable, MBSE-driven methodology using accessible tools, aiming to fill the gap in scalable and easily reproducible frameworks often missing in highly specialized or resource-intensive implementations.

The current approach to digital twins often involves bespoke, one-off solutions that fall short in addressing the broader, systemic challenges of complex industrial systems. A rigorous, general, and unified mathematical and computational foundation is essential to transition from these isolated cases to digital twins that are fully functional, robust, and accessible at scale [16]. Such a foundation would incorporate standard definitions, modeling paradigms, and analytical techniques that can be uniformly applied across various domains. In current industrial applications, companies implement digital twins by integrating sensor networks, IoT devices, and advanced analytics to mirror and monitor the behavior of machines and operations. These digital replicas enable predictive maintenance, process optimization, and performance monitoring by simulating various operational scenarios.

Many industrial digital twin systems adopt a layered architecture. At the basic level, physical assets are instrumented with sensors and IoT devices that collect real-time operational data. This data flows through a communication layer often built on protocols like MQTT, RESTful APIs, or other messaging systems that ensure reliable data transfer. The core of a digital twin system is the digital representation, which is developed using simulation models, data analytics, and machine learning algorithms. These models run on cloud platforms, edge computing nodes, or hybrid systems to enable real-time processing and decision support. Also, modular architectures based on microservices and service-oriented designs provide flexibility and scalability, allowing different functional components—from data ingestion to advanced analytics and visualization—to be updated independently. Together, these technologies and architectural choices form the backbone of modern digital twin implementations in industry. However, technical issues such as data synchronization, sensor accuracy, and network latency challenge the fidelity of the simulations. Also, a lack of standardized frameworks and interoperability guidelines hampers efforts to scale digital twins across varied industrial settings, limiting their overall robustness and accessibility.

## B. MODEL-BASED SYSTEM ENGINEERING (MBSE)

MBSE is an engineering approach that centers on creating and using detailed models to support the design, analysis, and validation of complex systems [17]. Rather than relying solely on extensive documents, MBSE replaces or supplements conventional documentation with rigorous, formalized representations—often using languages like SysML—to capture requirements, system architecture, behavior, and interactions [18]. Traditionally, MBSE has been applied to complex system design by guiding the entire development process, from conceptualization and requirements analysis to design, integration, and testing. In practice, engineering teams use these models to explore design alternatives, perform trade-off analyses, and verify system behavior under various scenarios [19]. This model-centric approach improves communication among multidisciplinary teams, enhances traceability from requirements to implementation, and helps manage the inherent complexity of large-scale systems, making it a favored method in sectors such as aerospace, defense, and automotive engineering.

Frameworks like Capella and methodologies like Arcadia play a significant role in system development by providing structured, model-based approaches that manage complex system architectures [20]. Capella offers an integrated, open-source environment that supports the entire systems engineering lifecycle, from initial requirements analysis to detailed design and validation [21]. It embodies the Arcadia methodology, which provides a process-driven framework emphasizing clear system decomposition, traceability, and iterative refinement [22]. Together, they enable engineers to capture diverse viewpoints, articulate key system

behaviors, and systematically explore design alternatives. This structured approach improves communication among interdisciplinary teams, reduces the likelihood of design inconsistencies, and facilitates early detection of integration issues. In essence, Capella and Arcadia contribute to more robust and scalable system development by ensuring that every aspect of a complex system is methodically modeled, analyzed, and validated.

Integrating MBSE with digital twin projects offers significant advantages [23], [24]. MBSE provides a structured framework for capturing system requirements and behavior in detailed models. These models enhance traceability and support early validation of design choices. When combined with digital twins, which mirror real-time data and operational conditions, the result is a more cohesive representation of a system from design to performance. This integration can lead to improved consistency between the virtual models and actual system behavior, enabling better decision-making and predictive maintenance. However, challenges also exist. Many MBSE tools are built for static or design-phase models and may struggle with the dynamic, real-time data flows characterizing digital twins. There can be difficulties in ensuring that the formal MBSE models remain synchronized with evolving physical systems, and bridging the gap between model abstraction and operational detail demands considerable expertise. Standardization and interoperability between MBSE platforms and digital twin technologies are still evolving, further complicating their seamless integration in complex system development.

### C. COMMUNICATION PROTOCOLS IN INDUSTRIAL IOT

In industrial IoT applications, selecting the appropriate communication protocol is crucial for supporting high-speed, reliable, and low-latency data exchanges that form the backbone of digital twin interactions. In this context, MQTT is favored for IoT communication in digital twin applications because it is a lightweight protocol designed for efficient data exchange [25]. Its publish/subscribe mechanism makes it well-suited for environments where devices need to share real-time data without the overhead of constant point-to-point connections [26]. This design allows digital twin systems to receive timely updates from sensors and other IoT devices while managing network traffic effectively. Also, MQTT supports varying levels of Quality of Service (QoS), enabling applications to balance between speed and reliability as needed. MQTT's minimal resource requirements also make it ideal for devices with limited processing power, which are common in industrial settings. These features, combined with its widespread adoption and ease of integration, make MQTT a robust choice for maintaining the dynamic communication loops essential to digital twin applications.

MQTT's lightweight design makes it efficient, but it also introduces several limitations when it comes to maintaining stable and reliable connections. One issue is that MQTT primarily relies on TCP for transport, which can be problematic

in networks with high latency or intermittent connectivity. While MQTT offers three levels of QoS, the lower levels (e.g., QoS 0) do not guarantee message delivery, resulting in potential data loss during unstable connections. Using higher QoS levels like QoS 1 or QoS 2 improves reliability by ensuring messages are acknowledged, but this success comes at the cost of increased overhead, which can lead to delays and duplicate messages if not managed carefully. Also, the protocol's keep-alive mechanism, which is meant to detect and manage dropped connections, can prematurely disconnect devices in environments with fluctuating network conditions. These factors make MQTT less suited for scenarios demanding high reliability without additional infrastructure to monitor and manage connectivity issues.

Communication protocols serve as the bridges that connect physical systems with their digital counterparts, ensuring that data from sensors and devices is transferred in real time. These protocols define the rules and methods for data exchange, allowing digital systems to receive frequent updates about the state of physical assets. For example, protocols like MQTT or AMQP enable a publish-and-subscribe model where sensors publish data and digital twins subscribe to these updates, ensuring minimal delay. Real-time synchronization is supported by mechanisms such as QoS levels, which help manage delivery reliability and order while combating network latency issues. Also, custom keep-alive messages and error-checking functions in these protocols ensure that any loss or corruption of data is quickly detected and corrected. In this way, communication protocols maintain a steady and continuous link between physical and digital systems, enabling accurate, up-to-date representations essential for decision-making and process control.

## III. SYSTEM DESIGN AND METHODOLOGY

In this section, we provide an overview of our system design and methodology, detailing the overall architecture of our proof-of-concept digital twin system, the integration of its hardware and software components, and the implementation details that enable real-time synchronization between the physical and digital environments.

### A. OVERALL SYSTEM ARCHITECTURE

The physical twin is an object-sorting machine built with LEGO<sup>®</sup> components. It uses LEGO<sup>®</sup> Spike Prime kits to create a modular structure capable of handling moving parts like conveyors and sorters. This design allows the system to sort objects based on attributes such as color. The system uses sensors embedded in the LEGO<sup>®</sup> setup to detect whether an object is conforming or defective. To drive its functionalities—such as controlling motors, reading sensor inputs, and coordinating actions—the physical twin integrates multiple Raspberry Pi microcontrollers. Each Raspberry Pi is tasked with managing a specific part of the system; for example, one unit may control the main conveyor belt while others manage the sorting mechanisms for conforming and defective products. This distributed

control architecture not only enhances the reliability of the physical system but also simplifies maintenance and future expansions. The microcontrollers communicate with each other and with higher-level control systems through a lightweight interface provided by the Raspberry Pi Build HAT, ensuring that data from the physical processes are captured and transmitted in real time.

The simulation model is implemented using FlexSim [27], a discrete-event simulation software that creates a system simulation model of the complex physical system. FlexSim offers real-time visualization, mirroring the physical twin's operations and state through continuous data updates. The simulation reflects key system elements—such as conveyor behavior, sensor readings, and sorting mechanisms—to ensure an accurate representation of the physical process. This digital model not only depicts normal operational dynamics but also allows for testing under simulated malfunction conditions, like conveyor blockages, to assess system responses. By integrating simulation with live data, the digital twin supports dynamic decision-making, predictive maintenance, and the optimization of control strategies for improved industrial process performance.

The integration process leverages MBSE to ensure that the physical and digital components work as a single system. Using MBSE, the entire system—from requirement definition to detailed design—can be thoroughly modeled and analyzed, reducing integration risks and facilitating early identification of potential issues [28].

In this project, the MBSE approach was implemented using Capella, a system engineering tool that supports the Arcadia methodology. Arcadia provides a structured framework for developing complex systems by guiding the design through several stages: from defining high-level operational concepts to specifying technical details. Capella was used to capture the functional requirements, allocate responsibilities between the physical system (the LEGO<sup>®</sup>-based system with Raspberry Pi microcontrollers) and the digital system (the FlexSim-based simulation), and define the interfaces and communication protocols (such as MQTT) that connect them.

This integrated process in Capella allowed us to create a detailed model that exhibits the interactions between the physical and digital twins. Through iterative refinement and simulation, potential mismatches—such as timing issues, data transfer delays, or control discrepancies—were identified and resolved early on. As a result, the MBSE approach not only provided clear documentation and traceability of requirements but also enabled effective validation of system behavior under both normal and fault conditions, ensuring robust system performance before actual implementation.

The operational analysis diagram in Figure 1 illustrates how the physical and digital parts of a proof-of-concept digital twin system interact in real time. It shows the hardware elements, such as sensors and mechanical components, alongside the software modules that model, simulate, and update the system's behavior. The operator is depicted at the

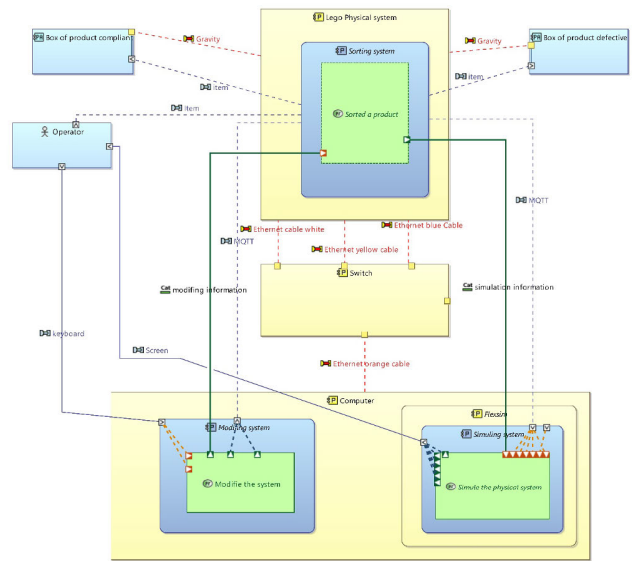


FIGURE 1. Operational analysis.

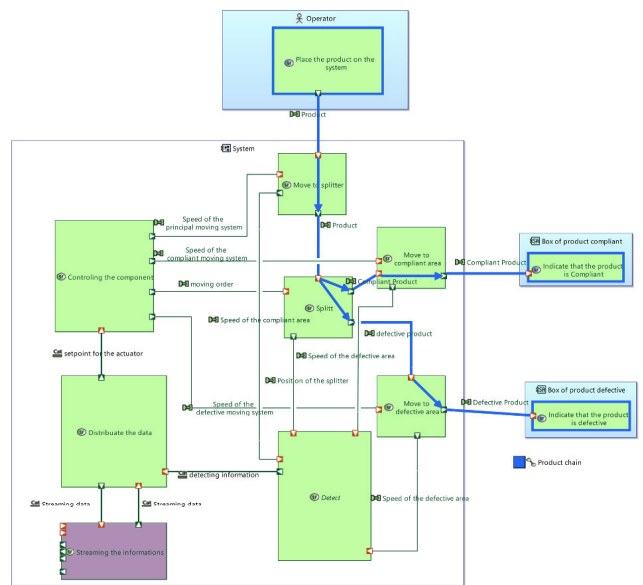


FIGURE 2. Functional architecture.

physical layer, sending and receiving signals through various channels. The digital twin side processes incoming data, runs simulations, and sends commands back to the physical system. This continuous loop ensures synchronized behavior and accurate mirroring between the real environment and its virtual counterpart.

The functional architecture of the system in Figure 2 depicts the key components and the processes they perform. It organizes the system into blocks that each represent a specific function or process, showing the flow of data or energy between these functions. This figure emphasizes the relationships between various system elements, illustrating how the system's functionality is distributed across different

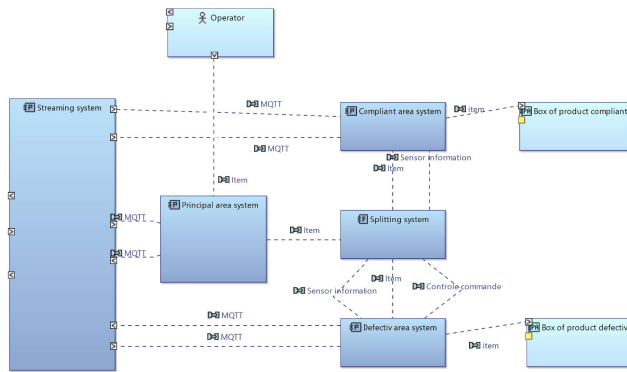


FIGURE 3. Logical architecture.

components. Each function or block has a input-output relationship, which makes it easy to understand how the system as a whole behaves by analyzing the interactions and connections of these functional elements. This figure also shows how different functional actions coordinate to move and track products, detect defective items, and share data with an operator. It begins with the operator placing a product on the system, which then triggers tasks such as controlling the conveyor, distributing data, and streaming sensor information. Each green box represents a function, while arrows show the flow of inputs and outputs between them. As the product moves through the process, signals indicate whether it is complete or defective, and the system relays that status to the operator.

The logical structure of the system in Figure 3 visualizes the key components and their relationships. Each block represents a major system component or module, such as the physical system, data processing elements, or interfaces. This figure emphasizes how these blocks are interconnected, typically through data flows, control signals, and interactions. By defining the responsibilities and interactions of each component, it provides a high-level overview of how the digital twin system is organized from a logical perspective, without delving into the detailed physical or functional behavior. This figure also shows how different blocks—such as the operator, streaming system, principal area system, splitting system, compliant area system, and defective area system—communicate with each other. Each dotted line denotes an exchange of MQTT messages, sensor information, or physical items. The operator is positioned at the top, sending and receiving data through the streaming system and principal area system. The splitting system determines where each item should go, routing it to the compliant or defective area as needed. Together, these connections highlight how the system’s logical components cooperate to manage product flow and status updates.

The physical architecture in Figure 4 illustrates system’s physical hardware components and their relationships within the system. It highlights key elements such as sensors, conveyor systems, cameras, and the operator interface,

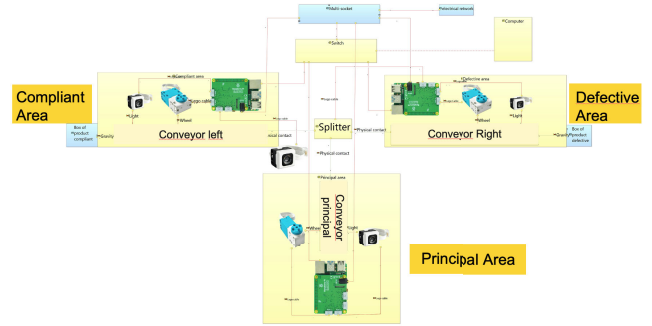
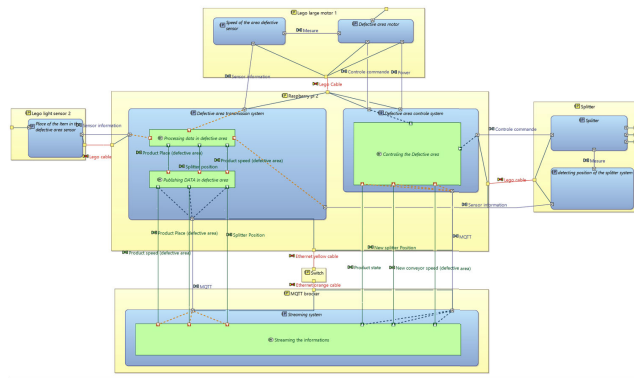


FIGURE 4. Physical architecture.

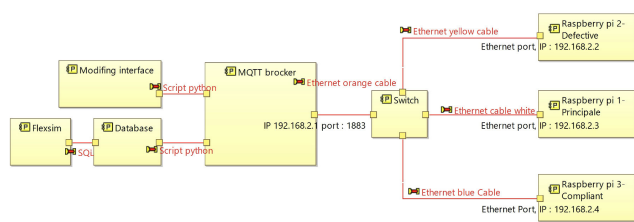
showing how each component is connected and interacts. The various blocks represent distinct physical subsystems, with lines denoting physical connections like wires, power supplies, or network links. This physical architecture forms the foundation for the digital twin system, where physical elements are paired with their respective counterparts in the logical and functional systems. This figure emphasizes the tangible components that enable the real-time operations and product tracking. This figure also shows how the system’s hardware components—such as the principal conveyor, the left and right conveyors, and the splitter—are laid out and connected. Each area (principal, compliant, and defective) is shown with its corresponding conveyor and control board. Lines indicate electrical or network connections, including the switch that links the computer to the conveyors. This arrangement enables real-time product flow, inspection, and routing, ensuring each item is directed to the appropriate area based on its status.

The physical layout of the Raspberry Pi system in Figure 5 illustrates the Raspberry Pi system in relation to peripheral hardware components. At the core, the Raspberry Pi is shown interconnected with several modules such as sensors, actuators, and attached systems. Lines represent physical connections like GPIO pins or USB ports used for communication. Specific components like cameras, motor controllers, and other sensors are linked to the Raspberry Pi, enabling it to collect data, process it, and control various devices in real time. This physical configuration ensures that the Raspberry Pi functions as the central hub for managing system operations and interfacing with the surrounding hardware. This figure also illustrates how the Raspberry Pi connects to sensors, motors, and other hardware elements in the system. Arrows represent cables for data, power, and control signals. The Raspberry Pi receives signals from sensors, processes the information, and then sends commands to actuators like the splitter or motor. Each block in the diagram shows a distinct hardware component, and their connections highlight the flow of information and control. This setup ensures real-time interaction between the Raspberry Pi and the physical parts of the system.

The system’s communication architecture in Figure 6 depicts various communication components and their



**FIGURE 5. Representative Raspberry Pi physical architecture showing key connections for a single controller node. Arrows indicate data/control flow via MQTT (network), sensor/actuator signals (GPIO/Build HAT interface), and power. This configuration is replicated for each of the three specialized Raspberry Pi controllers (Main Conveyor, Defective Sorter, Conforming Sorter).**

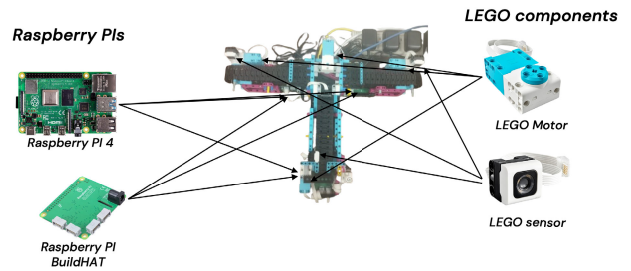


**FIGURE 6. Architecture - Communication.**

relationships within the system. It highlights how different blocks, such as devices, controllers, and networks, interact using communication protocols. Lines and arrows show the flow of data and messages between devices, sensors, and systems, which are necessary for tasks such as product tracking, data exchange, or status updates. Each block represents a separate communication element, and the diagram emphasizes the channels and mechanisms that allow these elements to work together efficiently. This layout illustrates the overall connectivity needed to synchronize information and ensure smooth system functionality. This figure also illustrates how the MQTT broker, switch, and various Raspberry Pi boards exchange data over an Ethernet network. Each Pi has its own IP address and is linked to the broker through distinct cables and ports. The broker coordinates messages from the Pi devices and also connects to external services, such as a database or a Python script interface. This setup ensures real-time data sharing and centralized control for the entire system.

**B. HARDWARE COMPONENTS**

The LEGO® Spike Prime kits include a variety of pre-engineered hardware elements that are well-suited for rapid prototyping and scalable system development. The main hardware components of our physical system are shown in Figure 7. The LEGO® motors are responsible for driving the moving parts of the system, such as the conveyor belt and



**FIGURE 7. Main hardware components.**

object-sorting mechanisms. They offer controlled, consistent motion that is essential for positioning and operation during the sorting process. These sensors are vital for the system’s ability to distinguish between different objects. By detecting specific color patterns—such as green for conforming items and red for defective ones—the sensors enable the system to identify and segregate objects according to predefined criteria. The LEGO® elements allow for the construction of robust and flexible conveyor belts. These belts facilitate the smooth transport of objects through various stages of the sorting process. Their modular nature also permits adjustments to the design, accommodating changes in the sorting strategy or scale of operation.

Together, the motors, sensors, and conveyor mechanisms provided by the LEGO® Spike Prime kits form a coherent and adaptable hardware foundation. This foundation is critical for the physical twin, ensuring that the system can reliably execute object-sorting tasks while also supporting future modifications and integration with digital control elements. The three Raspberry Pi 4 microcontrollers are configured to divide the system’s workload into manageable, dedicated tasks. Each microcontroller is assigned to a specific subsystem, ensuring that the overall operations run smoothly and responsively. One Raspberry Pi, designated the ‘Main Conveyor Controller’, is used to control the main conveyor belt. It manages the continuous movement of objects through the sorting process. This unit handles sensor data input from the LEGO®-built structure and ensures the belt operates at the desired speed. It also relays real-time status updates to the digital twin. The second Raspberry Pi, the ‘Defective Sorter Controller’, supervises the sorting line for defective products. It processes signals from color sensors to identify objects marked as defective. When a red object is detected, this microcontroller activates the mechanism that diverts the item to the defective line. It is finely tuned to react quickly to sensor inputs, ensuring that defective items are efficiently sorted out. The third Raspberry Pi, the ‘Conforming Sorter Controller’, oversees the conforming product sorting line. It monitors sensor outputs for conforming objects, typically recognized by a green color. Upon detecting a conforming object, it controls the positioning of the sorting mechanism to channel the item correctly. This unit also assists in maintaining the optimal flow of objects on the conveyor

belt. All three microcontrollers communicate with each other, using lightweight interfaces provided by the Raspberry Pi Build HAT. This configuration allows them to synchronize their operations and exchange real-time data. Their use not only divides the computational workload but also enhances the system's overall reliability and responsiveness.

### C. SOFTWARE COMPONENTS

FlexSim is used to create a 3D simulation model of the physical system and to support real-time data visualization. In this simulation setup, the physical components—such as conveyors, sensors, and sorting mechanisms—are modeled in a virtual environment. The software replicates the spatial layout and dynamics of the actual system so that its behavior can be observed and analyzed in real time.

Data from the physical twin is continuously sent to FlexSim, allowing the simulation to update the 3D model with the current state of the system. This synchronization is crucial for maintaining an accurate digital representation. One can observe the movement of parts, such as the conveyor belts and the diversion of objects, as if they were monitoring the physical system directly. Additionally, real-time performance metrics—such as speeds, sensor activations, and processing delays—are visualized within FlexSim. This feature provides immediate feedback that is beneficial for diagnosing issues and adjusting operational parameters.

The simulation setup not only creates a virtual mirror of the physical process but also serves as a test-bed. It enables designers to simulate different scenarios (e.g., normal operation versus system malfunctions) without disrupting the actual hardware. This dual functionality makes FlexSim a powerful tool in bridging the gap between physical operation and digital control, enhancing both system understanding and decision-making processes.

A snapshot of the FlexSim Model representing the system's simulation setup is shown in Figure 8 which visualizes the flow of operations within the system using a FlexSim simulation model. It shows various entities such as products moving along a conveyor, passing through inspection points, and reaching different areas (e.g., compliant or defective). The colored blocks represent different simulation modules, which model elements like the product flow, decision points, and actions taken based on system conditions. Arrows indicate how items are routed across different stages, allowing the simulation to track and optimize performance in a controlled, virtual environment. This approach helps visualize the system's behavior before physical implementation. This figure shows a top-down view of a T-shaped conveyor setup, with a source feeding products from the bottom conveyor to a central detector. Depending on the product's status, the system directs items left to a "Compliant Box" or right to a "Defective Box." The colored boxes at each end represent collection points for compliant or defective items. The center "State Detector" checks each item's condition and helps route it correctly. This layout demonstrates how FlexSim

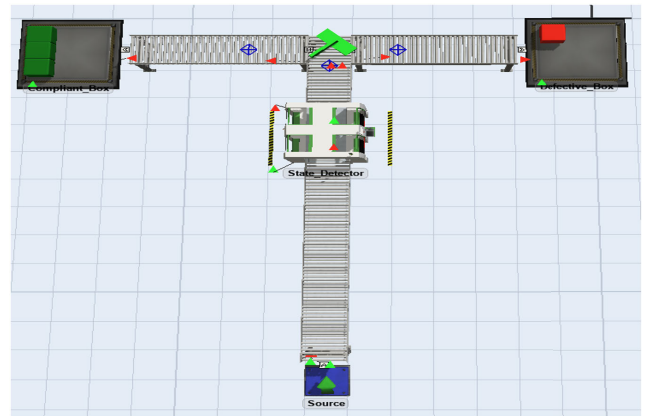


FIGURE 8. FlexSim model.

models real-time sorting and routing processes in a simulated environment.

The MQTT protocol is used to create a continuous, bidirectional communication channel between the physical and digital twins. In this setup, the physical twin—comprising LEGO<sup>®</sup> components and Raspberry Pi microcontrollers—collects data such as sensor readings, conveyor speeds, and sorting status. This data is then published as messages to an MQTT broker under specific topics. The digital twin, built using FlexSim, subscribes to these topics, receiving real-time updates to accurately mirror the state and behavior of the physical system.

Similarly, the digital twin sends control commands and operational adjustments back to the physical twin. These commands are published to other predefined MQTT topics, which the Raspberry Pi microcontrollers monitor. This bidirectional flow of information ensures that any changes or corrective actions determined by the digital twin can be promptly executed by the physical system.

By using a publish/subscribe architecture, MQTT decouples the sender and receiver. Such decentralization increases reliability and system flexibility while reducing network overhead. The protocol's lightweight design is particularly well suited for IoT applications, ensuring that even with limited bandwidth and low latency requirements, communication remains robust. MQTT plays a crucial role in synchronizing the digital and physical domains, allowing them to function as a cohesive, real-time system.

### D. IMPLEMENTATION DETAILS

The real-time synchronization mechanism between the digital and physical systems is built on continuous data exchange and immediate feedback loops. The process works as follows:

- The physical system continuously gathers data from sensors (such as color detection and position measurements) and monitors actuator status. Raspberry Pi microcontrollers collect this data from the LEGO<sup>®</sup> components.

- The collected data is published immediately to an MQTT broker using predefined topics. This lightweight protocol minimizes delay and ensures data integrity even on low-bandwidth networks.
- The digital twin, developed in FlexSim, subscribes to these MQTT topics. Once it receives the sensor data, the 3D simulation and system state are updated in near-real-time, mirroring the physical processes accurately.
- In the reverse direction, if the digital twin identifies the need for changes (like adjusting conveyor speed or repositioning sorting mechanisms), it publishes control commands back to the MQTT broker on specific topics.
- The Raspberry Pi microcontrollers, already configured to listen to these command topics, retrieve and execute the incoming instructions. This closes the control loop, ensuring that operational adjustments are promptly reflected in the physical system.

This bidirectional synchronization, enabled by MQTT's publish/subscribe architecture, guarantees continuous, low-latency communication between the physical and digital components, thereby fostering real-time monitoring, rapid feedback, and adaptive system control.

Feedback loops and control commands play a central role in maintaining the system's responsiveness and stability. In this system, sensors on the physical twin continuously capture data—such as the current speed of the conveyor or the position of the splitter. This sensor data is sent to the digital twin in near-real-time, where it is analyzed and compared with desired performance benchmarks.

If the digital twin detects any deviations or issues, it computes necessary adjustments. For instance, if the conveyor speed is below the target, the digital twin generates a control command to increase the speed. Similarly, if the splitter is not positioned correctly to divert defective items, a command is issued to adjust its position. These commands are transmitted back to the physical twin via the MQTT protocol.

In the physical system, the Raspberry Pi microcontrollers are already configured to listen for these commands. When a command—such as adjusting conveyor speed or repositioning the splitter—is received, the microcontrollers respond by sending precise signals to the motors and actuators. This immediate response corrects the observed deviations. By constantly cycling through sensing, analyzing, and adjusting, this feedback loop ensures that both the physical and digital twins remain synchronized. The digital twin not only reflects current performance but also acts as a controller, initiating corrective actions to improve accuracy, efficiency, and overall system performance.

Key implementation parameters were established to ensure robust operation: MQTT communication was configured using QoS level 1, providing a balance between guaranteed message delivery and network overhead. Data from the physical twin's sensors was published at a frequency of approximately 10 Hz (100 ms interval). The FlexSim

simulation model was set with a corresponding simulation time step of 0.1 seconds to process incoming data promptly. During testing, the Raspberry Pi 4 controllers exhibited CPU utilization consistently below 40%, indicating sufficient processing capacity for the implemented tasks.

#### IV. EXPERIMENTAL RESULTS

In this section, we detail our evaluation by describing the experimental setup, outlining the performance metrics employed, and providing a thorough analysis of the results obtained.

##### A. EXPERIMENT SETUP

In the experimental setup, two primary scenarios were designed to evaluate system performance and robustness: one under normal operating conditions and another under a simulated system malfunction.

*Normal Operation:* In the normal operation scenario, the physical and digital twins work in sync in an error-free environment. The object-sorting machine operates continuously, with LEGO<sup>®</sup>-based components moving objects along a conveyor. Color sensors accurately identify objects—distinguishing between conforming and defective items—while the Raspberry Pi microcontrollers orchestrate motor commands and data collection. The data is transmitted in real time via MQTT to the FlexSim-based digital twin, which updates the 3D model accordingly. This scenario verifies that sensor readings, actuator responses, and real-time data synchronization occur as expected under standard operating conditions. This scenario was run continuously for a 60-minute period, during which approximately 500 LEGO<sup>®</sup> objects were sorted to gather sufficient performance data.

*Simulated System Malfunction:* In the simulated system malfunction scenario, a deliberate fault such as a conveyor blockage is introduced to assess the system's diagnostic and recovery behaviors. In this case, an obstruction is introduced along the conveyor path, causing deviations in expected sensor readings, altered conveyor speed, and irregular sorting information. The digital twin, receiving these anomalous signals via MQTT, immediately reflects the malfunction in its real-time visualization. It then initiates appropriate feedback loops and control commands—such as adjusting the conveyor speed or repositioning the sorter—to test the system's response strategies. This scenario demonstrates the digital twin's role in early fault detection and enables operators to evaluate the effectiveness of predictive maintenance and remedial control measures during abnormal conditions. The malfunction scenario was similarly tested over a 60-minute period with around 500 objects to allow for direct comparison with normal operation.

##### B. PERFORMANCE METRICS

The performance metrics focused on evaluating how well the digital twin mirrored the physical system and how quickly and accurately control commands were executed. Key metrics include:

### Synchronization Accuracy

- **Object Sorting Status:** This metric measures how correctly the digital twin represents whether objects are sorted as conforming or defective.
- **Conveyor Speed:** The consistency between the actual speed of the physical conveyor belt and the speed reported by the digital twin is tracked.
- **Splitter Position:** The precise alignment of the splitter in both the physical system and the digital model is verified to ensure accurate operation.

### Command-Response Times

- **Response Time:** This metric measures the time interval between the digital twin issuing a control command (e.g., adjusting conveyor speed or repositioning the splitter) and the physical system executing that command.
- **Latency Analysis:** The delay in data transmission from sensors to the digital twin and back from the control command to physical actuators is closely monitored.

By analyzing these metrics, the system's performance is assessed in terms of how accurately and rapidly the physical and digital twins remain in sync, ensuring effective real-time monitoring and control.

## C. RESULTS ANALYSIS

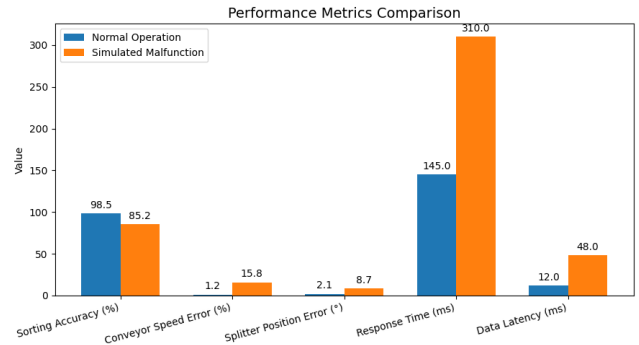
Under normal operating conditions, the digital twin mirrors the physical system. Sensor data—including object color, conveyor speed, and splitter position—is continuously transmitted via MQTT and instantaneously updated within the FlexSim model. This real-time synchronization is reflected in our quantitative results: as shown in Table 1, the system achieves a high Sorting Accuracy of 98.5% (based on 10 independent trials, 95% CI [97.8%, 99.2%]), a low Conveyor Speed Error of 1.2% ( $\pm 0.3\%$  standard deviation), and minimal Splitter Position Error of 2.1° ( $\pm 0.5^\circ$  standard deviation). Furthermore, the Response Time remains swift at 145 ms with a Data Latency of only 12 ms. These figures collectively confirm that the digital twin provides an accurate and reliable representation of the physical system during steady-state operation.

In contrast, when a simulated malfunction such as a conveyor blockage is introduced, the digital twin rapidly registers significant deviations from normal performance. The table illustrates this degradation: Sorting Accuracy falls to 85.2%, while errors in Conveyor Speed and Splitter Position increase to 15.8% and 8.7°, respectively. Both Response Time and Data Latency are substantially elevated, reaching 310 ms and 48 ms. These numerical differences underscore the system's sensitivity to faults and its ability to detect anomalies in real time.

To complement the numerical data, Figure 9 presents a grouped bar chart that visually compares the performance metrics between the two scenarios. In the chart, the x-axis lists each metric—Sorting Accuracy, Conveyor Speed Error, Splitter Position Error, Response Time, and Data Latency—with adjacent bars representing normal operation

**TABLE 1. Performance metrics comparison: Normal Operation (NO) vs. Simulated Malfunction (SM).**

Metric	NO	SM
Sorting Accuracy (%)	98.5	85.2
Conveyor Speed Error (%)	1.2	15.8
Splitter Position Error (°)	2.1	8.7
Response Time (ms)	145	310
Data Latency (ms)	12	48



**FIGURE 9. Experimental results.**

and simulated malfunction conditions. The clear separation and corresponding annotations on each bar make it evident that the system's performance significantly degrades under malfunction conditions. The visual representation provided by the figure reinforces the conclusion drawn from the table: under normal conditions, the digital twin mirrors the physical system with high precision, while in the presence of faults, it immediately reflects the resulting anomalies.

These results demonstrate that the digital twin plays a dual role. Not only does it continuously mirror the physical system to ensure accurate real-time monitoring, but it also serves as a dynamic control center that initiates corrective actions when anomalies are detected. This capability is crucial for predictive maintenance; continuous monitoring allows operators to recognize early signs of potential component wear or failure. For instance, a consistent deviation in conveyor speed may prompt preemptive maintenance to address mechanical issues before they escalate. Simultaneously, the rapid corrective responses—evident from immediate adjustments in response times and control commands—minimize disruptions to the object-sorting process. In summary, both the precise numerical evidence in Table 1 and the visual clarity of Figure 9 validate the system's ability to maintain synchronization during normal operations as well as its dynamic responsiveness under fault conditions.

## V. DISCUSSION

In this section, we examine the key findings and system advantages, address the challenges and limitations encountered, and explore the broader implications for industrial applications.

### A. KEY FINDINGS

Bidirectional, real-time communication and synchronization offer several advantages for industrial systems. First, this approach enables constant monitoring. Data from sensors is transmitted continuously, allowing the digital twin to reflect the state of the physical system accurately at any given moment. This transparency is critical for detecting deviations or faults early.

Second, the immediate exchange of information facilitates rapid corrective action. When the digital twin identifies anomalies—such as unexpected changes in conveyor speed or misalignment in the splitter—it can quickly issue control commands to adjust system parameters. This minimizes downtime and enhances system reliability.

Third, the integration of real-time data with control commands supports a robust feedback loop. The system is able to dynamically adjust operations in response to sensed conditions. This not only improves operational efficiency but also contributes to effective predictive maintenance by flagging potential issues before they lead to failures.

Overall, the bidirectional, real-time communication architecture enhances system responsiveness, optimizes performance, and ensures greater consistency between the digital and physical twins, making it a valuable strategy in industrial applications.

### B. SYSTEM ADVANTAGES

The integrated system offers several notable advantages:

- *Improved System Monitoring:* The real-time synchronization between the physical twin and the digital twin, enabled by continuous data exchange via MQTT, creates a transparent monitoring system. Operators can observe detailed, real-time behaviors—from sensor outputs to actuator responses—allowing them to promptly identify deviations or anomalies. This constant visibility not only facilitates immediate responses to emerging issues but also builds a comprehensive operational history for long-term analysis.
- *Enhanced Operational Efficiency:* By continuously relaying live data and control commands, the system maintains a seamless feedback loop. The digital twin reflects even the subtle changes within the physical process, enabling rapid adjustments such as recalibrating conveyor speed or repositioning sorting mechanisms. This dynamic adjustment minimizes downtime, reduces manual intervention, and optimizes overall throughput. The modularity provided by the LEGO®-based structure, combined with the distributed management of tasks via Raspberry Pi microcontrollers, further reinforces the system's adaptability and operational robustness.
- *Support for Predictive Maintenance:* Continuous monitoring ensures that performance metrics—such as sensor readings, actuator speeds, and sorting accuracy—are constantly evaluated against expected benchmarks. Early detection of gradual drifts or sudden anomalies

allows maintenance teams to identify potential component failures before they result in significant disruptions. This predictive capability helps in scheduling proactive maintenance, thereby reducing unexpected breakdowns, minimizing repair costs, and extending the lifespan of critical hardware components.

### C. CHALLENGES AND LIMITATIONS

Hardware integration presented several challenges during the implementation phase. For instance, the LEGO®-based system and its sensors, actuators, and Raspberry Pi microcontrollers must fit within a compact physical space. This limited space creates constraints on how components can be arranged, making it difficult to manage wiring and connectivity. The available ports on the Raspberry Pi microcontrollers are limited, and organizing the connections without interference requires careful planning. Additionally, the short cable lengths provided by the LEGO® Spike kits can restrict flexibility. This requires precise placement of sensors and motors to accommodate the necessary connections without compromising performance. These hardware integration hurdles emphasize the need for meticulous design and layout planning to ensure that all components work together reliably within the available physical space.

MQTT is designed to be lightweight and efficient, but unstable network conditions can expose some vulnerabilities. For example, when network connectivity is sporadic or the bandwidth is low, message delivery can be interrupted, leading to potential loss of data. This loss affects the continuous real-time communication between the physical and digital twins. In unstable conditions, MQTT's reliance on short-lived connections may result in frequent disconnections. If the session is not managed properly, important messages could be delayed or dropped during reconnection periods. Using appropriate QoS levels can help mitigate some of these risks, but even then, persistent network instability might result in duplicate messages or message ordering issues.

Also, the protocol's lightweight nature means it lacks robust error-checking and recovery mechanisms found in heavier protocols. This makes it more susceptible to issues such as synchronization errors between the physical and digital systems when network disruptions occur. While MQTT is well-suited for IoT applications, its performance can degrade under unstable network conditions. For example, simulations introducing 20% packet loss indicated a temporary drop in synchronization fidelity, impacting sorting accuracy by up to 5% until connectivity stabilized. Such vulnerabilities highlight the need to carefully design the communication layer. Specific mitigations could include implementing a local data buffer on each Raspberry Pi to cache critical messages during brief network outages and configuring MQTT's Last Will and Testament (LWT) feature for faster disconnection detection. Continuously monitoring network performance remains crucial for ensuring reliable real-time operation.

#### D. IMPLICATIONS FOR INDUSTRIAL APPLICATIONS

The system's modular nature offers significant advantages for scaling and adapting to more complex scenarios.

The physical twin's modular hardware design enables quick reconfiguration and expansion. Additional sensors, motors, or actuators can be integrated with minimal changes. New modules or even entire system segments can be incorporated to monitor or control more complex operations. On the digital side, the FlexSim-based simulation provides a flexible virtual environment. Digital models can be updated to simulate complex workflows and larger systems. The simulation framework can be expanded to represent additional machinery, or multiple production lines. Integrating more variables and sophisticated process dynamics becomes straightforward through the simulation software, which can accommodate detailed modeling of new industrial elements.

The use of the MQTT protocol for communication between the physical and digital twins ensures that scaling does not compromise real-time feedback. MQTT's lightweight, publish/subscribe architecture allows additional data channels to be created as more sensors and units are added. Although unstable network conditions require attention, robust design practices—such as employing buffering strategies and redundant network paths—can address these challenges as the system grows.

The overall integration process, supported by MBSE with Capella and the Arcadia methodology, further strengthens the system's scalability. These approaches provide a structured framework that can document requirements and system interactions clearly, making it easier to integrate new modules and maintain system cohesion.

#### VI. CONCLUSION AND FUTURE WORK

The proof-of-concept demonstrated the potential of real-time digital twin integration. The project built a dual-component system in which the physical twin—a LEGO®-based system driven by Raspberry Pi microcontrollers—and the digital twin—modeled in FlexSim—communicated continuously and bidirectionally. This real-time synchronization allowed the digital twin to accurately mirror the physical system's state, resulting in precise tracking of object sorting, conveyor speeds, and splitter positions.

The implementation showcased robust feedback loops. Control commands generated in the digital twin were rapidly transmitted to the physical machine, enabling immediate corrective actions when anomalies were detected. This integration improved system monitoring and laid the groundwork for effective predictive maintenance by flagging potential issues before they escalated.

Overall, the proof-of-concept confirmed that the digital twin could serve both as an accurate representation of the physical process and as an active controller. The unique contribution lies in demonstrating the integration of MBSE principles (via Capella/Arcadia) with low-cost hardware and standard protocols to create an accessible and

scalable real-time digital twin framework, offering a practical alternative to more complex or proprietary solutions. The effective use of MQTT for data exchange, combined with a structured MBSE approach using Capella and the Arcadia methodology, underscored the project's success in advancing digital twin technology toward practical applications.

The project's contributions extend to enhancing process monitoring and maintenance. By integrating the physical twin with the digital twin in a real-time, bidirectional communication framework, the system offers continuous visibility into key performance metrics. This constant monitoring allows for early detection of anomalies, which in turn supports proactive, predictive maintenance strategies and helps prevent unexpected downtime.

Also, the feedback loops and immediate corrective actions enable the digital twin not only to mirror the physical system accurately but also to actively regulate it. This active control improves operational efficiency, as adjustments—such as modifying conveyor speed or repositioning sorting mechanisms—are executed promptly based on real-time data. The use of MQTT for data exchange and the structured integration approach using MBSE further ensure that these improvements scale well, making the system adaptable to more complex industrial environments.

For future work, the communication protocols can be refined to enhance overall reliability. One approach is to optimize the MQTT QoS settings. An adaptive QoS mechanism could allow the system to automatically switch between levels based on current network conditions. This would help ensure that message delivery is maintained even when network performance varies. Another improvement is the introduction of redundant communication paths. For example, a dual-broker system or additional backup channels could be implemented to take over if the primary connection fails. This redundancy would reduce the risk of message loss during network disruptions. Future validation will also include additional fault scenarios, such as simulated sensor failure or intermittent communication loss, to further assess system resilience.

It is also worth exploring the integration of supplementary protocols alongside MQTT. Protocols such as OPC UA or AMQP offer different strengths in reliability and error handling. A hybrid solution that leverages the best aspects of each protocol could provide more robust communication for critical industrial applications. Enhancing error-checking mechanisms is another area for improvement. Implementing stronger data integrity checks and automatic retransmission algorithms can help recover or flag missing and corrupted messages quickly. This, in combination with real-time network monitoring, would allow the system to detect and mitigate issues sooner. Finally, incorporating edge computing can further improve reliability. By buffering and processing data locally before sending it through the network, the system can reduce latency and protect critical operations from temporary connectivity issues.

To handle more complex and larger industrial systems, future work could focus on a few key areas:

- **Scalability of the Physical and Digital Models:** The current approach, demonstrated on a small-scale system, can be scaled up to manage entire production lines or even multiple interconnected processes. Expanding the digital twin to model a more complex physical environment with a higher number of sensors, actuators, and control nodes would be the first step. This might involve integrating additional data sources and extending the FlexSim model to monitor various interdependent sub-systems.
- **Advanced Distributed Control Architectures:** For larger industrial settings, a centralized control strategy might not suffice. Deploying a distributed control system—potentially leveraging edge computing—can help in managing localized decisions while maintaining overall coordination. This would ensure that high-frequency, critical control tasks occur near the data source, while higher-level management actions are coordinated across the facility.
- **Enhanced Data Management and Analysis:** Increasing system complexity will generate vast amounts of real-time data. Future work could explore the use of advanced data analytics and machine learning algorithms integrated into the digital twin. These approaches would help in not only predicting maintenance needs more accurately but also in optimizing production flows across a more extensive system.
- **Robust Communication Infrastructure:** As the number of connected devices increases, existing communication protocols may require optimization. Integrating multiple communication standards or adopting a hybrid protocol architecture could help manage network traffic, reduce latency, and provide higher reliability in more demanding industrial environments.
- **Modular and Interoperable System Design:** Designing the system with modularity in mind makes it easier to add or reconfigure components as requirements change. By ensuring that both the physical and digital elements are easily reconfigurable and interoperable, future implementations can quickly adapt to new processes, integrate additional machinery, or scale up operations without a complete redesign.

By addressing these areas, the current approach can evolve into a comprehensive solution capable of handling the complexity and scale of large industrial systems, while maintaining the benefits of real-time monitoring, predictive maintenance, and efficient process control.

## REFERENCES

- [1] M. W. Grieves, "Digital twins: Past, present, and future," in *The Digital Twin*. Cham, Switzerland: Springer, 2023, pp. 97–121.
- [2] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, and A. Calinescu, "Digital twins: State of the art theory and practice, challenges, and open research questions," *J. Ind. Inf. Integr.*, vol. 30, Nov. 2022, Art. no. 100383.
- [3] B. Soykan and G. Rabadi, "Optimizing humanitarian logistics with deep reinforcement learning and digital twins," in *Proc. Annu. Modeling Simulation Conf. (ANNSIM)*, May 2024, pp. 1–13.
- [4] P. E. Brown and A. Velosa. (2024). *Emerging Tech: Revenue Opportunity Projection of Simulation Digital Twins*. Accessed: Dec. 22, 2024. [Online]. Available: <https://www.gartner.com/en/documents/5451563>
- [5] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. D. J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sens.*, vol. 14, no. 6, p. 1335, Mar. 2022.
- [6] H. Zhu, B.-G. Hwang, Y. Z. Tan, and F. Wei, "Building on digital twin: Overcoming barriers and unlocking success in the construction industry," *J. Construct. Eng. Manage.*, vol. 150, no. 10, Oct. 2024, Art. no. 04024142.
- [7] L. Zhang, L. Zhou, and B. K. P. Horn, "Building a right digital twin with model engineering," *J. Manuf. Syst.*, vol. 59, pp. 151–164, Apr. 2021.
- [8] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial cyberphysical systems: A backbone of the fourth industrial revolution," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 6–16, Mar. 2017.
- [9] R. Santos, J. Basto, S. G. Alcalá, E. Frazzon, and A. Azevedo, "Industrial IoT integrated with simulation—a digital twin approach to support real-time decision making," in *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, 2019, pp. 23–26.
- [10] R. K. Mobley, *An Introduction to Predictive Maintenance*. Amsterdam, The Netherlands: Elsevier, 2002.
- [11] J. Lee, J. Ni, J. Singh, B. Jiang, M. Azamfar, and J. Feng, "Intelligent maintenance systems and predictive manufacturing," *J. Manuf. Sci. Eng.*, vol. 142, no. 11, Nov. 2020, Art. no. 110805.
- [12] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020.
- [13] S. Mihai, M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, B. Barn, D. Shetve, R. V. Prasad, H. Venkataraman, R. Trestian, and H. X. Nguyen, "Digital twins: A survey on enabling technologies, challenges, trends and future prospects," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2255–2291, 4th Quart., 2022.
- [14] M. G. Juarez, V. Botti, and A. Giret, "Digital twins: Review and challenges," *J. Comput. Inf. Sci. Eng.*, vol. 21, no. 3, Feb. 2021, Art. no. 030802.
- [15] D. Sargiotis, "Harnessing digital twins in construction: A comprehensive review of current practices, benefits, and future prospects," *Benefits, Future Prospects*, vol. 2024, pp. 1–46, Jul. 2024.
- [16] National Academies of Engineering, Engineering National Academies of Sciences, and Medicine, *Foundational Research Gaps and Future Directions for Digital Twins*. Washington, DC, USA: Academies Press, 2024. [Online]. Available: <https://nap.nationalacademies.org/catalog/26894/foundational-research-gaps-and-future-directions-for-digital-twins>
- [17] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," *Syst. Eng.*, vol. 21, no. 3, pp. 172–190, May 2018.
- [18] B. P. Zeigler, S. Mittal, and M. K. Traore, "MBSE with/out simulation: State of the art and way forward," *Systems*, vol. 6, no. 4, p. 40, Nov. 2018.
- [19] G. S. Parnell, N. J. Shallcross, E. Specking, and M. R. Phillips, "MBSE enabled trade-off analyses," in *Proc. INCOSE Int. Symp.*, Jul. 2021, vol. 31, no. 1, pp. 1406–1416.
- [20] P. Roques, "MBSE with the Arcadia method and the capella tool," in *Proc. 8th Eur. Congr. Embedded Real Time Softw. Syst. (ERTS)*, Jan. 2016, pp. 1–12.
- [21] N. Badache and P. Roques, "Capella to SysML bridge: A toolled-up methodology for MBSE interoperability," in *Proc. 9th Eur. Congr. Embedded Real Time Softw. Syst. (ERTS)*, Jan. 2018, pp. 1–13.
- [22] S. Bonnet, J.-L. Voirin, D. Exertier, and V. Normand, "Not (strictly) relying on SysML for MBSE: Language, tooling and development perspectives: The Arcadia/Capella rationale," in *Proc. Annu. IEEE Syst. Conf. (SysCon)*, Apr. 2016, pp. 1–6.
- [23] R. Honcak and A. Wooley, "An MBSE approach for virtual verification & validation of systems with digital twins," in *Proc. ACM/IEEE 27th Int. Conf. Model Driven Eng. Lang. Syst.*, Sep. 2024, pp. 390–400.
- [24] A. M. Madni, C. C. Madni, and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, Jan. 2019.

- [25] E. A. Thijssen, M. van de Molengraft, I. Adan, Q. Dang, and N. Singh, "MQTT based communication framework for agvs in a digital twin," *Manuf. Syst.*, vol. 2021, pp. 1–99, May 2021.
- [26] M. Amoretti, R. Pecori, Y. Protskaya, L. Veltri, and F. Zanichelli, "A scalable and secure publish/subscribe-based framework for industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 3815–3825, Jun. 2021.
- [27] Nordgren, "Flexsim simulation environment," in *Proc. Winter Simulation Conf.*, vol. 1, Dec. 2003, pp. 197–200.
- [28] J. A. Estefan, "Survey of model-based systems engineering (MBSE) methodologies," *In cose MBSE Focus Group*, vol. 25, no. 8, pp. 1–12, 2007.

**BULENT SOYKAN** received the Ph.D. degree from Old Dominion University.

He is currently a Research Scientist with the Institute for Simulation and Training, University of Central Florida. His research interests include mathematical optimization, machine learning, optimal control to develop data-driven computational tools for decision-making in dynamic and uncertain environments, improving sequential decision making under uncertainty, building embodied agents for decision making, and addressing the technical challenges of large-scale data-driven systems. He focuses on creating adaptive models that leverage real-time data to improve efficiency, robustness, and predictive capabilities. By combining theory with practical applications, his work advances solutions for complex challenges in areas, such as digital twins and autonomous systems. He is particularly passionate about simulation-based inference, geometric learning, and developing hybrid models that combine the strengths of deep learning with traditional mathematical approaches.

**GRÉGOIRE BLANC** received the Engineering degree from IMT Mines Alès, France. He is currently pursuing the Ph.D. degree with Université Paris-Saclay, conducting research in knowledge engineering.

**GHAITH RABADI** received the B.S. degree in industrial engineering from Jordan University, Amman, Jordan, and the M.S. and Ph.D. degrees in industrial engineering from the University of Central Florida, Orlando, FL, USA, in 1996 and 1999, respectively.

From 2000 to 2002, he was a Postdoctoral Researcher and a Visiting Assistant Professor with the Department of Industrial Engineering and Management Systems, University of Central Florida. He was an Assistant Professor, an Associate Professor, and a Full Professor with the Department of Engineering Management and Systems Engineering, Old Dominion University, Norfolk, VA, USA, from 2002 to 2008, from 2008 to 2016, and from 2016 to 2022, respectively. From 2018 to 2020, he was the Vice President of Research and Development with Princess Sumaya University for Technology, Amman. Since 2022, he has been a Professor and the Interim Director of the School of Modeling, Simulation, and Training, University of Central Florida. He is the author of two books and more than 100 peer-reviewed journal articles, conference papers, and book chapters. His research and teaching interests include planning and scheduling, operations research, simulation modeling and digital twins, artificial intelligence, and supply chain management and logistics. Application areas include production systems, disaster response, and transportation system optimization.

Dr. Rabadi was a recipient of the NASA Faculty Fellowship, the NASA Software Invention Award, the NASA Board Action Invention Award, and the Fulbright Specialist Program Award. His research has been funded by U.S. Department of Education, NASA, NATO, Department of Homeland Security (DHS), Virginia Port Authority (VPA), MITRE Corporation, and Qatar Foundation. He is also a Co-Founder and the Chief Editor of *International Journal of Planning and Scheduling*.

• • •