Development of an Interactive Twinverse System : Metaverse Platform Integrating Digital Twin and AI Agent

Yongho Lee[1], Huichan Park[2], Seongbin Choi[3] and Sang Won Lee[4]

[1,2,3] *Department of Mechanical Engineering, Sungkyunkwan University, Suwon-si, Gyeonggi-do, 16419, Republic of Korea*
hope685@g.skku.edu
parkhc97@g.skku.edu
cosogi@g.skku.edu

[4] *School of Mechanical Engineering, Sungkyunkwan University, Suwon-si, Gyeonggi-do, 16419, Republic of Korea*
sangwonl@skku.edu

ABSTRACT

Digital twin (DT) platforms are increasingly used for PHM, yet most systems still lack real-time, bi-directional control between physical assets and virtual models, and a unified semantic layer that grounds natural-language commands in plant constraints. To address this gap, this research present Twinverse, an interactive metaverse environment that integrates a ROS/Kafka-based bi-directional DT, a knowledge-graph (KG) semantic backbone, and an LLM-powered agent. The KG encodes structural/operational constraints (e.g., kinematics, limits) and is serialized into a vector store to support RAG-based intent interpretation, while a constraint-aware execution pipeline verifies workspace, joint limits, and speed bounds prior to dispatch. Implemented on an industrial robot cell in Unity, the system provides real-time synchronization and multi-user operation within a single immersive interface. In evaluation, the platform maintained tight virtual–physical tracking and stable latency under increasing user load, and it enabled PHM-oriented functions such as anomaly interrogation and explainable, context-aware action generation. Our contribution is a cohesive DT–KG–LLM architecture that (1) grounds language-to-action in machine-readable plant constraints, (2) closes the loop from natural-language intent to verified execution, and (3) operationalizes PHM analytics inside an immersive DT environment. This work demonstrates a practical path toward interactive, explainable, and real-time PHM decision support.

1. INTRODUCTION

1.1. Background and Motivation

Cyber-Physical Systems (CPS) and Digital Twin (DT) technologies form a core foundation of Industry 4.0, enabling real-time synchronization between physical assets and virtual counterparts. Recent advancements in Digital Twin (DT) technology have transformed it from static 3D representations into dynamic, interoperable platforms that integrate real-time sensor data streams, cloud-based computation, and standardized frameworks such as the Asset Administration Shell (AAS). Visualization through Virtual and Augmented Reality (VR/AR) further enhances spatial awareness and situational understanding, expanding applications in remote maintenance, training, and collaborative operations. Liu et al. reviewed DT applications in condition monitoring, fault diagnosis, and predictive analysis [1], while Qi et al. highlighted IoT, simulation, and analytics as essential DT enablers [2]. Immersive interfaces, such as VR/AR, enhance human–machine interaction; Yun and Jun demonstrated VR-based manufacturing support through the Immersive and Interactive CPS (I2CPS) framework [4], and Fuchs et al. validated VR-assisted design for improved accuracy [3]. Concurrently, AI agents have evolved beyond rule-based control, leveraging large language models (LLMs) to interpret complex industrial instructions and translate them into executable control commands. The integration of Retrieval-Augmented Generation (RAG) enables these agents to combine knowledge graph (KG) semantics with live operational data, ensuring domain relevance and operational safety in generated instructions. These developments collectively provide a robust foundation for intelligent CPS applications, including predictive maintenance, process optimization, and adaptive decision-making within DT–metaverse environments. Mon-Williams et al. showed that embodied

---

LLMs execute complex robotic tasks in unstructured environments [5], while Ismail et al. proposed NARRATE to convert natural language instructions into robot trajectories [6]. These developments suggest that combining LLM-driven AI agents with DT–metaverse platforms can facilitate advanced applications such as predictive maintenance, control optimization, and adaptive process planning.

1.2. Limitations of Existing Approaches

Despite advances, digital twin and immersive CPS implementations face limitations. Protic et al. [7] achieved bi-directional DT control in academic settings but were limited to predefined scenarios, reducing scalability. Many DTs still lack robust virtual-to-physical control and seamless integration of heterogeneous data. Zillner et al. [8] advocated the Asset Administration Shell (AAS) for interoperability, yet practical use often fails to unify CAD, sensor, and simulation data into a coherent semantic framework. Interaction methods remain constrained; Wang et al. [9] applied vision–language models for natural language queries but required fixed templates, limiting adaptability. Furthermore, PHM integration is underdeveloped—Latsou et al. [10] proposed multi-agent anomaly detection without immersive visualization, while Yu et al. [11] applied Bayesian networks for health monitoring in non-immersive environments.

1.3. Research Objective

To address these challenges, this study proposes an Interactive Twinverse System, a metaverse-based platform that tightly integrates bi-directional digital twins with an AI agent powered by large language models and Retrieval-Augmented Generation (RAG). Building upon the architectural foundations demonstrated in earlier works such as Protic et al. [7], this system introduces a real-time synchronization framework between Unity-based immersive environments and physical assets via ROS and Kafka middleware, enabling seamless two-way interaction. A central feature of the proposed approach is the development of a knowledge graph–based semantic backbone, designed to unify structural, behavioral, and operational data across heterogeneous sources. Unlike conventional AAS or isolated database, this knowledge graph establishes explicit relationships among physical components, control logic, and diagnostic data, thereby enabling more efficient data retrieval and interoperability. The system further incorporates an LLM-based AI agent capable of translating high-level, natural language user commands into executable actions grounded in the operational context of the target asset. By leveraging RAG, as demonstrated in related AI-agent integration works such as Ismail et al. [6], the agent can retrieve domain-specific knowledge from the semantic backbone, synthesize it with real-time sensor inputs, and generate responses or control strategies that are both context-aware and explainable. This integration directly supports PHM-oriented applications, enabling operators to detect anomalies, identify probable root causes, and simulate maintenance procedures within the immersive Twinverse environment. In doing so, the proposed system not only overcomes the unidirectional and fragmented nature of existing digital twin solutions but also establishes an extensible framework for intelligent, human-centric CPS in industrial domains, addressing the limitations identified in Latsou et al. [10] and Yu et al. [11].

2. TWINVERSE ENVIRONMENT

This study introduces a bi-directional digital twin system, termed the "Twinverse," that integrates a real-time industrial digital twin into an immersive metaverse environment. The Twinverse enables users to both monitor and control physical equipment from within a virtual space, supporting human-machine collaboration for advanced manufacturing tasks.

2.1. Industrial Digital Twin Implementation

To realize the Twinverse, a real-world industrial robot (Yaskawa GP25, 6-DOF) was replicated as a digital twin using Unity3D. This twin reflects real-time states of the robot, including joint positions, motor torques, and temperatures. ROS (Robot Operating System) serves as the communication middleware, enabling control of the robot's joints and live data streaming. Users can observe and manipulate the digital twin as if interacting with the physical equipment.

2.2. Metaverse Environment Design

As shown in Figure 1, The physical workspace—including the robot, worktable, and surrounding equipment—was digitally reconstructed using CAD and implemented into Unity3D. This allows users to immerse themselves in an environment that mirrors the real manufacturing setup. Within this virtual environment, an intuitive UI was designed to support real-time interaction with the robot, making control tasks seamless and accessible to non-expert users.

(a) Physical Workspace **(b) Digital Reconstruction**

Figure 1. Digital transformation of physical industrial spaces into the metaverse

2.3. Bi-Directional Communication Architecture

A Kafka-based system architecture was developed to enable robust, real-time communication between the virtual and physical domains. Figure 2 shows that the system architecture consists of:

Layer 1 (Data Collection): Interfaces with robots, sensors, and vision systems using ROS-bridge, TCP/IP, OpenCV, and serial protocols.

Layer 2 (Middleware): Manages asynchronous and heterogeneous data streams across devices and users.

Layer 3 (Application Layer): Facilitates user interactions, data management, and decision-making processes.

This architecture ensures low-latency bidirectional interaction between digital twins and physical assets, enabling real-time monitoring and control via the metaverse.

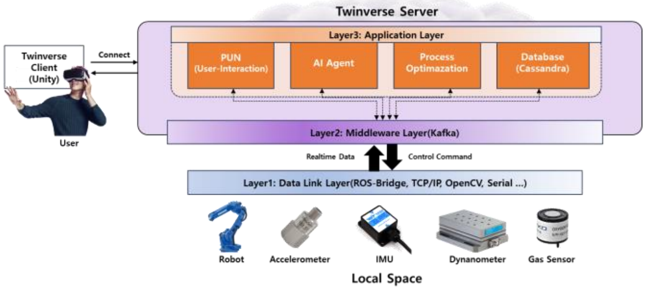


Figure 2. Bi-directional Communication Architecture

2.4. System Performance and Multi-User Support

The implemented Twinverse environment was evaluated in a simulated industrial scenario. The positional tracking error between the virtual and physical robots was maintained under 5%, validating synchronization accuracy. Furthermore, as shown in Figure 3, Kafka's architecture demonstrated stable latency even under increasing multi-user connections, confirming the system's scalability and responsiveness in collaborative settings.

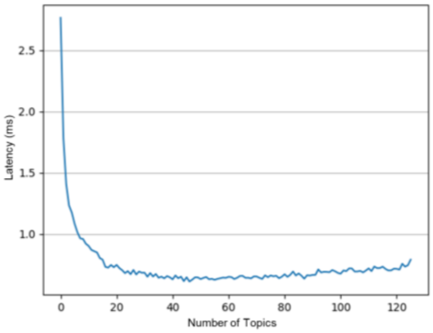


Figure 3. Performance of Kafka System during Multi-user Connections

2.5. System Scalability and Multiverse Integration

The proposed Twinverse system, while initially designed for a single digital twin environment, inherently supports scalability toward a multi-Twinverse architecture. In this extended framework, multiple Twinverse environments—each encapsulating its own set of digital twins, assets, and domain-specific knowledge—can be seamlessly connected to a centralized Twinverse Hub—Multiverse System. Figure 4 shows overview of Multiverse System architecture. Through a plug-and-play mechanism, newly instantiated Twinverse environments can register with the Multiverse without requiring significant reconfiguration. This is achieved by employing standardized data exchange protocols, such as OPC UA and Asset Administration Shell (AAS), and a common semantic layer maintained within the knowledge graph infrastructure. Once connected, assets within each Twinverse become globally discoverable and can be utilized across domains. From a user perspective, this architecture enables intuitive navigation between different Twinverse instances, allowing operators to move fluidly between environments and interact with heterogeneous assets as if they resided in a single unified space. For example, a maintenance engineer could transition from a Twinverse representing a robotic assembly line to another representing a CNC machining cell, while retaining access to AI-driven PHM functions, shared analytics tools, and historical operational data. This modular and interconnected approach significantly reduces the overhead of expanding the Multiverse system, enabling incremental adoption across manufacturing plants, supply chain nodes, or even cross-industry collaborations. Furthermore, by leveraging the existing LLM-based AI agent framework, the system can perform cross-Twinverse reasoning—aggregating information from multiple environments to generate richer diagnostic insights, predictive maintenance schedules, and optimization strategies.

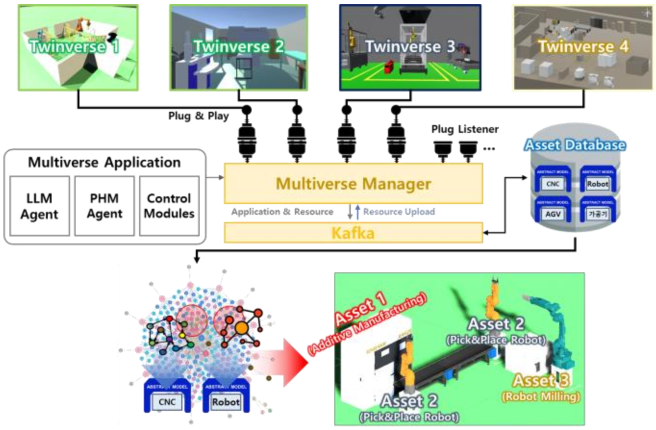


Figure 4. Multiverse System based on Plug & Play

3. AI AGENT INTEGRATION WITH KNOWLEDGE GRAPH

3.1. Knowledge Graph as the Semantic Backbone of the Twinverse

In the proposed Twinverse architecture, the knowledge graph (KG) functions as a semantic backbone that unifies heterogeneous data sources, structural hierarchies, and operational constraints into a single coherent model. While the bi-directional digital twin described in Chapter 2 enables real-time synchronization between the physical and virtual domains, the KG provides the interpretive layer that allows the AI agent to reason about the system's configuration, capabilities, and current state. Unlike conventional relational databases that primarily store static attributes, as shown in Figure 5 and Figure 6, the KG encodes both structural relationships (e.g., kinematic chains, sensor-actuator links) and behavioral dependencies (e.g., motion constraints, thermal limits, process parameters) in a graph-based formalism. Each physical or virtual asset in the Twinverse—such as robot links, joints, sensors, and task modules—is represented as a node, with edges describing their functional interdependencies. Properties attached to nodes and edges capture quantitative parameters such as mass, inertia tensors, motion limits, and calibration transforms. By maintaining a consistent semantic schema across all Twinverse assets, the KG serves as an interoperability layer that bridges disparate data standards, middleware protocols, and domain-specific models. This ensures that both AI-driven reasoning and system-level control share a single "source of truth," thereby reducing redundancy and enhancing reusability across different assets and operational scenarios.

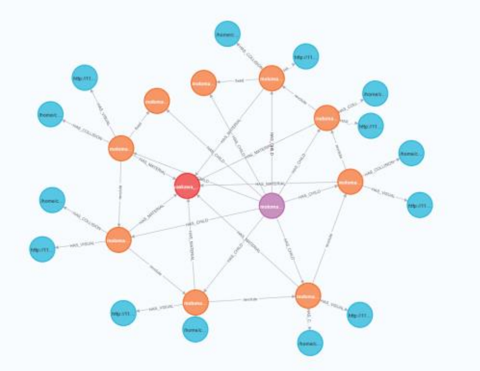
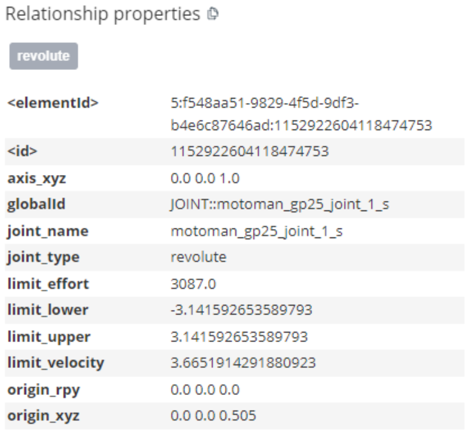


Figure 5. Asset Representation using Knowledge Graph



Relationship properties

revolute

<elementId>	5:f548aa51-9829-4f5d-9df3-b4e6c87646ad:1152922604118474753
<id>	1152922604118474753
axis_xyz	0.0 0.0 1.0
globalId	JOINT::motoman_gp25_joint_1_s
joint_name	motoman_gp25_joint_1_s
joint_type	revolute
limit_effort	3087.0
limit_lower	-3.141592653589793
limit_upper	3.141592653589793
limit_velocity	3.6651914291880923
origin_rpy	0.0 0.0 0.0
origin_xyz	0.0 0.0 0.505

Figure 6. Attributes stored in the edge

3.2. LLM-based AI Agent for Semantic Interaction

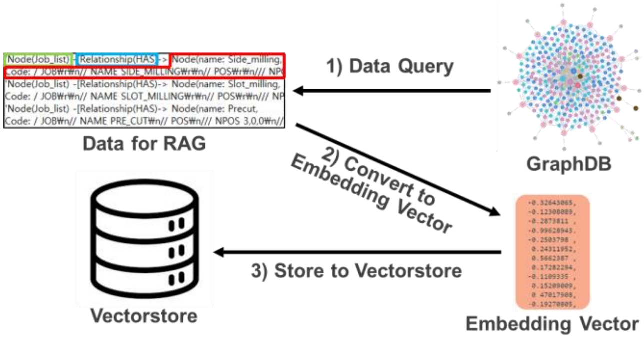
This section describes how the AI agent in the Twinverse operates as an intelligent semantic interface, transforming high-level human instructions into executable actions grounded in the structural and operational context of the system. The process extends beyond simple command–response matching, incorporating semantic reasoning over the knowledge graph (KG) to ensure safe and contextually valid execution. The overall workflow proceeds as follows.

3.2.1. User Command Processing and Subgraph Extraction

When a user issues a command—via text, speech, or VR-based interaction—the AI agent first parses it into an intent representation. This representation is then translated into a KG query that identifies and filters only the nodes and relationships relevant to the requested action. For example, if the command involves controlling a specific joint of a robot, the agent extracts a subgraph containing the joint node, its connected links, actuation limits, tool frames, and current sensor readings. This subgraph serves as the contextual boundary for subsequent reasoning steps, ensuring that downstream processing remains tightly coupled to the actual state and capabilities of the target asset.

3.2.2. Subgraph Serialization and Vector Store Construction

As shown in Figure 7, Once the relevant subgraph is identified, it is serialized into a descriptive text that captures both static physical parameters (e.g., link lengths, motion ranges, tool offsets) and dynamic operational data (e.g., live sensor readings, error states). This serialized description is then embedded into high-dimensional vectors using a domain-adapted language model. The resulting embeddings are stored in a vector database (vectorstore), enabling semantic retrieval even when the user's phrasing diverges from the original asset descriptions. This step effectively transforms structured KG knowledge into a searchable, context-rich memory layer.



Node(Job_list) -Relationship(HAS)-> Node(name: Side_milling,
Code: / JOB₩r₩n// NAME SIDE_MILLING₩r₩n// POS₩r₩n/// NP
'Node(Job_list) -[Relationship(HAS)-> Node(name: Slot_milling,
Code: / JOB₩r₩n// NAME SLOT_MILLING₩r₩n// POS₩r₩n/// NP
'Node(Job_list) -[Relationship(HAS)-> Node(name: Precut,
Code: / JOB₩n// NAME PRE_CUT₩n// POS₩n/// NPOS 3,0,0₩n//

Figure 7. Creating Vectorstore based on KG

3.2.3. RAG-Based Language-to-Action Mapping

The user's intent vector is compared against the vector store to retrieve the most relevant operational knowledge. This retrieved context is injected directly into the prompt (Figure 8) of a large language model (LLM), constraining its generative process so that all proposed actions remain consistent with the real-time constraints and capabilities of the Twinverse. For instance, if a requested movement exceeds a joint's mechanical limits or conflicts with an ongoing operation, the LLM can revise the execution plan before generating actionable instructions. In the final stage of reasoning, the AI agent merges the retrieved KG-derived context with the original user prompt, producing a coherent and executable set of instructions that directly reflect both the user's intent and the system's operational state.

```
prompt = f"""
You are to give coordinates and speed of a 6DOF robot movement.
Identify the following items from the text:
- final coordinates of the robot after movement
- speed in which the robot moves to the final coordinates

The text is delimited with triple backticks. \
Format your response as a JSON object with \
"x" , "y" , "z" , and "speed" as the keys.
If I give the command as forward, right, up, that means forward : +x direction, right : +y direction, Up : +z direction
Otherwise, If I give the command as backward, left, down, that means Backward : -x direction, Left : -y direction, Down : -z direction
If the information isn't present, use 0 \
as the value.
Make your response as short as possible.

Current position: {cur_pos}
Review text: '''{command}'''
"""
```

Figure 8. Prompt for RAG

3.2.4. Execution and Feedback Loop

The generated instructions are visualized within the Twinverse for user validation. Upon approval, as shown in Figure 9, they are transmitted through ROS or Kafka-based middleware to the physical system. Sensor feedback from the execution is continuously integrated back into the KG, closing the loop between perception, reasoning, and actuation. This closed-loop architecture ensures that the AI agent's decision-making process is explainable, traceable, and adaptive to evolving industrial conditions.

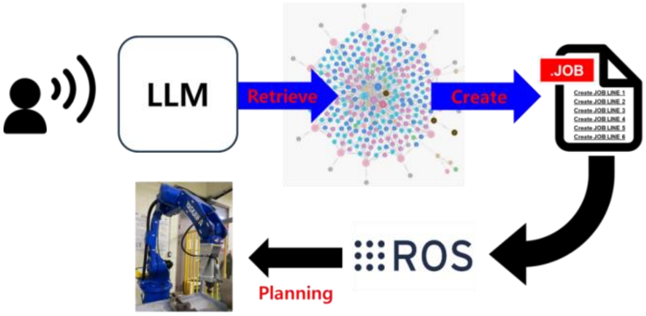


Figure 9. Command to the Physical Execution

This architecture extends the concept of semantic interoperability into the realm of human–machine interaction by enabling explainable, context-aware control in immersive metaverse environments. The KG not only constrains and validates the agent's decisions but also provides a transparent reasoning trail that can be inspected for verification or debugging purposes.

3.3. Role in PHM-Oriented Applications

Within the PHM (Prognostics and Health Management) context, the integration of the KG and LLM-based agent allows the Twinverse to evolve from a monitoring platform into a decision-support system capable of diagnosing faults, explaining their probable causes, and recommending predictive maintenance actions. Because operational data streams from sensors, controllers, and simulation modules are continuously incorporated into the KG, the AI agent can adapt its reasoning to reflect the current health state of the system. This capability addresses a key limitation of traditional PHM tools, which often operate in isolation from control and interaction systems. By embedding PHM reasoning directly into the Twinverse's semantic and interaction layers, maintenance personnel can not only detect anomalies but also simulate corrective actions, assess their impact in the virtual environment, and then deploy them to the physical system with minimal risk. From a broader research standpoint, this integration demonstrates how knowledge-driven generative CPS architectures can unify physical system modeling, semantic reasoning, and natural language interaction. In doing so, it provides a foundation for scalable, multi-domain Twinverse deployments where AI agents serve as both interpreters and orchestrators of complex industrial processes.

4. CONCLUSION

This study presented the development of an Interactive Twinverse System, a metaverse-based platform that integrates bi-directional digital twins with an LLM-powered AI agent for industrial Prognostics and Health Management (PHM). The proposed framework enables real-time synchronization between physical assets and their virtual counterparts through Unity, ROS, and Kafka, while offering immersive interaction capabilities for operators and engineers. A knowledge graph–based semantic backbone was implemented to unify heterogeneous data sources, facilitating efficient information retrieval and interoperability. The LLM-based AI agent, enhanced with Retrieval-Augmented Generation (RAG), demonstrated the ability to interpret user intent, retrieve relevant domain knowledge, and execute context-aware actions. Integrated within the Twinverse environment, the system supports PHM-oriented functionalities such as anomaly detection, fault diagnosis, and predictive maintenance in an interactive and explainable manner. The results of this work indicate that combining immersive metaverse environments with intelligent agents and semantic data integration can significantly enhance the operational applicability of digital twins. The Twinverse approach addresses several limitations of existing DT–CPS systems, including unidirectional data flow, fragmented data management, and constrained human–machine interaction modalities. While the present work primarily establishes the foundational architecture, it also clarifies how the proposed framework conceptually differs from existing DT or NLP-based systems. Although the study does not provide extensive quantitative comparisons with other digital twin or natural language interface frameworks, its key contribution lies in defining an integrated and extensible architecture that unifies these domains. The Twinverse demonstrates how LLM-based agents, knowledge graphs, and real-time digital twins can operate coherently within a single metaverse environment—an integration not achieved by conventional DT or NLP-based control systems. This architectural convergence enables future researchers to embed diverse PHM algorithms, benchmark them within a shared semantic framework, and quantitatively evaluate their performance under consistent experimental conditions. Rather than competing solely on numerical performance metrics, this work provides a foundational infrastructure for scalable, explainable, and interoperable PHM research across heterogeneous industrial systems.

5. FUTURE WORK

Although this work primarily focuses on platform development rather than extensive experimental validation, the Twinverse establishes an extensible foundation for future PHM research. By modularizing the AI agent and knowledge-graph architecture, the platform allows diverse PHM models—ranging from data-driven prognostic algorithms to hybrid diagnostic frameworks—to be integrated as plug-in tools within the same interactive environment. This scalability ensures that future studies can quantitatively evaluate and benchmark different PHM techniques directly through the Twinverse, transforming it from a proof-of-concept system into a practical research and deployment infrastructure for intelligent, explainable, and real-time industrial health management. Also, this research will focus on expanding the scope of the Twinverse to accommodate multi-asset and multi-domain scenarios, enabling scalability across diverse industrial sectors, not only robotic systems but any other industrial system such as additive manufacturing, machining and so on. Additional research will explore the integration of reinforcement learning and adaptive control strategies to further enhance autonomous decision-making capabilities. Moreover, the incorporation of standardized interoperability frameworks, such as extended Asset Administration Shell models, will be investigated to facilitate cross-platform collaboration. Finally, user studies will be conducted to evaluate the system's effectiveness in improving PHM decision-making, usability, and overall operator engagement in real-world industrial settings.

ACKNOWLEDGEMENT

This research was supported through the Industry Technology Alchemist Project (20025702, "Development of smart manufacturing multiverse platform based on multisensory fusion avatar and interactive AI") funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

REFERENCES

Liu, M., et al. (2021). Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems, 58, 346–361.

Qi, Q., et al. (2021). Enabling technologies and tools for digital twin. Journal of Manufacturing Systems, 58, 3–21.

Fuchs, M., et al. (2022). A collaborative knowledge-based method for the interactive development of cabin systems in virtual reality. Computers in Industry, 136, 103590.

Yun, H., & Jun, M. B. G. (2022). Immersive and interactive cyber-physical system (I2CPS) and virtual reality interface for human involved robotic manufacturing. Journal of Manufacturing Systems, 62, 234–248.

Mon-Williams, R., et al. (2025). Embodied large language models enable robots to complete complex tasks in unpredictable environments. Nature Machine Intelligence.

Ismail, S., et al. (2024). NARRATE: Versatile Language Architecture for Optimal Control in Robotics. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Protic, A., et al. (2020). Implementation of a bi-directional digital twin for industry 4 labs in academia: A solution based on OPC UA. Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 979–983.

Zillner, A., et al. (2024). The Asset Administration Shell: Key enabler for interoperability in the Industry 4.0. Software and Systems Modeling, 23(1), 211–229.

Wang, T., et al. (2021). Digital twin improved via visual question answering for vision-language interactive mode in human–machine collaboration. Journal of Manufacturing Systems, 58, 261–269.

Latsou, C., et al. (2023). Digital twin-enabled automated anomaly detection and bottleneck identification in complex manufacturing systems using a multi-agent approach. Journal of Manufacturing Systems, 67, 242–264.

Yu, J., et al. (2021). A digital twin approach based on nonparametric Bayesian network for complex system health monitoring. Journal of Manufacturing Systems, 58, 293–304.

BIOGRAPHIES

Yongho Lee is now Ph.D. candidate in the Sustainable Design and Manufacturing Laboratory, Department of Mechanical Engineering, Sungkyunkwan University. He obtained his bachelor's degree (Mechanical Engineering) from Sungkyunkwan University in 2020. His research interest is Digital Twin for manufacturing environment and AI-agent based on Large Language Model

Huichan Park is now master & Ph.D combined course student in the Sustainable Design and Manufacturing Laboratory, Department of Mechanical Engineering, Sungkyunkwan University. He obtained his bachelor's degree (Mechanical Engineering) from Sungkyunkwan University in 2024. His research interest is Digital Twin for manufacturing environment and AI-agent based on Large Language Model

Seongbin Choi is now master student in the Sustainable Design and Manufacturing Laboratory, Department of Mechanical Engineering, Sungkyunkwan University. She obtained her bachelor's degree (Mechanical Engineering) from Sungkyunkwan University in 2024. Her research interest is AI-based prognostics and health management for smart manufacturing.

Sang Won Lee is now professor in the school of Mechanical Engineering, Sungkyunkwan University. He obtained his bachelor's degree in 1995 and master’s degree in 1997 (Mechanical Design and Production Engineering) from Seoul National University. He obtained his Ph.D. degree (Mechanical Engineering) from University of Michigan in 2004. His research interest includes prognostics and health management (PHM), cyber-physical system (CPS), additive manufacturing, and data-driven design.