Digital Twin Framework for Humanoid Robotics in Construction: A Conceptual Approach for Scene and Task Simulation

Xuling Ye, Liu Liu and Markus König

Computing in Engineering, Faculty of Civil and Environmental Engineering, Ruhr University Bochum, Germany
xuling.ye@rub.de, liu.liu-m6r@rub.de, m.koenig@rub.de

Abstract –

Humanoid robots offer a promising solution for construction and building operation tasks because they can operate in environments designed for humans without requiring major changes to infrastructure or workflows. However, deploying humanoids in real construction and building environments remains challenging due to safety risks, complex interactions with the built environment, and limited task reliability. To address these challenges, this paper introduces and structures a conceptual Digital Twin for Construction Humanoids (DTCH) framework that integrates Building Information Modeling (BIM), physics-based simulation, and humanoid task definition. Using the Unitree G1 humanoid as a representative platform, the framework formalizes a workflow to convert BIM models into geometric MuJoCo simulation scenes and semantic information, allowing humanoid tasks to be defined, systematically decomposed into inner actions, and tested in a virtual environment before deployment. Through a simulation-based scenario demonstration, the framework illustrates a reproducible method to explore humanoid action behavior and task execution within a controlled digital environment. The proposed approach demonstrates how the digital twin methodology can serve as a structured foundation for early-stage research and development of humanoids for construction and post-construction tasks, reducing development risks and supporting more reliable task execution prior to real-world deployment.

Keywords –

Construction Humanoid; Robotics; Digital Twin; Simulation

1 Introduction

The construction industry continues to face long-standing challenges related to labor shortages, limited productivity growth, and the need for safer and more efficient working practices. Despite the increasing adoption of automation technologies, many activities across both active construction sites and building operation phases remain heavily dependent on manual labor, particularly those requiring mobility in human-designed environments such as corridors, staircases, and confined indoor spaces, including inspection, facility management, and safety-related tasks that are repetitive and time-consuming for human workers [1–3].

While recent construction robotics research has predominantly focused on mobile collaborative robots (e.g., wheeled and quadruped platforms), humanoid robots have begun to emerge as a potential complementary solution. Due to their human-like morphology and ability to operate within existing built environments without major infrastructure modification, they present an opportunity for future integration, with prior studies indicating their potential to support inspection, documentation, logistics, and monitoring tasks across different building lifecycle stages [2,3]. However, deploying humanoid robots directly in real construction or operational environments remains challenging due to uncertainties in locomotion stability, perception reliability, and task execution under varying environmental conditions [2].

To address these challenges, simulation-based development has become an essential step for evaluating humanoid behavior before physical deployment. Physics-based simulators enable controlled testing of robot motion, sensing, and interaction with the environment while reducing safety risks and development costs [4,5]. At the same time, the construction domain increasingly relies on Building Information Modeling (BIM), which provides standardized digital representations of the building geometry and structured information throughout the lifecycle [6]. Combining humanoid simulation with building-scale digital models offers an opportunity to evaluate robot performance within realistic spatial contexts that reflect actual buildings rather than abstract test environments.

However, to effectively support humanoid deployment, such simulations must also integrate task logic and maintain a consistent relationship between virtual evaluation and physical execution, motivating the adoption of the digital twin concept. In general, a digital twin refers to a virtual representation of a physical system that is linked to its real-world counterpart through models and data exchange [6]. Therefore, this paper proposes and conceptually defines a Digital Twin for Construction Humanoids (DTCH) framework, where the digital twin is instantiated as a coupled representation of the building environment, humanoid robot, and task execution logic for simulation-based evaluation and transfer to physical deployment. While prior studies have integrated BIM, robotics, and task planning, this work specifically extends these approaches to humanoid robots and formalizes their interaction within a unified digital twin structure tailored to human-centric built environments. The proposed framework enables systematic exploration of construction- and operation-related activities using humanoids, providing a coherent methodology for studying humanoid applicability and task feasibility across building scenarios.

2 Related Work

2.1 Humanoid Robotics for Construction

The construction sector, including both active construction and post-construction stages throughout the building lifecycle, continues to face persistent challenges related to low productivity growth, labor constraints, and the need for reliable inspection and documentation practices [1]. Compared to manufacturing, construction sites are characterized by unstructured, dynamic environments and human-centered tools and workflows, which limit the effectiveness of conventional automation solutions [2]. Beyond the activities at construction sites, a significant portion of work occurs after structural completion, such as quality assessment, facility inspection, and lifecycle monitoring of buildings. These tasks are often repetitive, labor-intensive, and time-consuming, relying heavily on manual work and human judgment in the stable built environment. Humanoid robots offer a potential solution that is distinct from other robotic platforms due to their anthropomorphic structure [2,3,7]. While platforms such as quadruped robots have demonstrated strong capabilities in traversing uneven terrain and carrying sensors, humanoids are uniquely suited to operate within human-centered environments using human-oriented affordances (e.g., stairs, tools, and workspaces) without requiring adaptation [2,3]. In addition, their anthropomorphic form may enable more intuitive human–robot interaction, which is particularly relevant for collaborative and supervision-intensive tasks [2]. Prior studies indicate that humanoid deployment can improve site safety, reduce injuries associated with repetitive or physically demanding tasks, and support more consistent task execution, contributing to shorter project duration and improved work quality [3]. However, existing literature on humanoid deployment in construction remains limited, and empirical validation is still emerging.

Research has identified a wide range of construction activities where humanoid robots may provide value, including progress tracking, auto-documentation and inspection and surveillance of tasks [3]. These tasks require mobility across stairs, uneven floors, and partially completed structures, as well as the ability to carry sensors and perform observations from human-accessible viewpoints. Compared to fixed, wheeled, or quadruped robots, humanoids provide a different set of advantages aligned with human-like reach, manipulation, and viewpoint positioning, rather than purely mobility performance [2]. In addition, autonomous data collection by humanoid robots can reduce the reliance on manual site walks, while enabling more frequent and objective monitoring of construction progress and quality [1]. As a result, certain humanoid-assisted work, such as inspection, has been shown to reduce manual effort, improve data consistency, and support more accurate decision-making [3].

Despite their potential, the practical deployment of humanoid robots in construction remains at an early stage due to unresolved challenges in locomotion robustness, perception reliability, energy efficiency, and task adaptability in dynamic environments [2]. Given the technical and environmental uncertainties associated with deploying humanoid robots in real buildings, sim-to-real strategies, mostly based on simulation-driven development, play a critical role as prerequisites for both evaluation and deployment [2,4]. Simulation enables systematic testing of humanoid behaviors under controlled yet diverse conditions, allowing risks, failure modes, and performance limits to be identified before physical trials [7]. Within this context, digital twin frameworks provide a structured sim-to-real methodology , in which a digital counterpart replicates construction scenes and simulates task executions, enabling the assessment of humanoid performance, stability, and sensing strategies under repeatable scenarios [6]. Cost-effective and Artificial Intelligence (AI)-oriented humanoids such as Unitree H1 or G1, offer an opportunity to explore this approach, supporting early-stage experimentation and conceptual validation of humanoid-assisted construction workflows [4]. To realize such digital twin–based evaluation, a physics-based simulation environment capable of accurately modeling humanoid dynamics and environment interaction is required.

2.2 Humanoid Simulation in Built Environment

Physics-based simulation is a critical enabler for humanoid robotics, particularly in domains such as construction where real-world testing is costly, disruptive, and potentially unsafe. Among available simulation tools, MuJoCo is widely adopted for humanoid research due to its accurate modeling of contact-rich dynamics, joint actuation, and whole-body balance [5,8]. Building on these capabilities, several recent works use MuJoCo as the primary development environment for Unitree humanoid platforms, showing successful sim-to-real transfer for locomotion, stair climbing, teleoperation, whole-body control, and multi-contact behaviors under varying ground conditions [4,9–11]. These capabilities are particularly important for construction applications, where humanoid robots may routinely walk on unfinished floors, navigate stairs, and maintain balance on uneven surfaces during construction activities and post-construction inspection tasks. MuJoCo therefore provides a physically grounded environment to evaluate whether a humanoid robot can safely operate within building-like conditions before site deployment.

However, most humanoid simulation studies focus on robot performance in isolation, typically assume idealized environments and short-term task execution [4,5,8–11]. They do not provide a systematic methodology for integrating robot simulation with building-scale representations and construction-oriented tasks, and offer limited integration of building semantics, task context, or lifecycle considerations. For applications related to construction sites or post-construction buildings, humanoids must operate within sustained stability, repeatable locomotion, and reliable interaction with architectural features [2,3]. This exposes a gap between task-driven humanoid simulation and environment-driven simulation, where the physical context of buildings must be explicitly modeled rather than abstracted away. Taken together, the reviewed works suggest a clear need for a digital twin framework that integrates humanoid simulation, digital representations of building environments, and task definitions into a unified environment. In contrast to existing approaches, this study explicitly targets humanoid-specific constraints and opportunities, particularly the interaction between anthropomorphic embodiment, building semantics, and task execution logic within a sim-to-real pipeline.

3 Framework

To systematically investigate the applicability of humanoid robots for taking over the human manual work in construction contexts, a Digital Twin for Construction Humanoids (DTCH) framework is developed and shown in Figure 1. The framework explicitly distinguishes

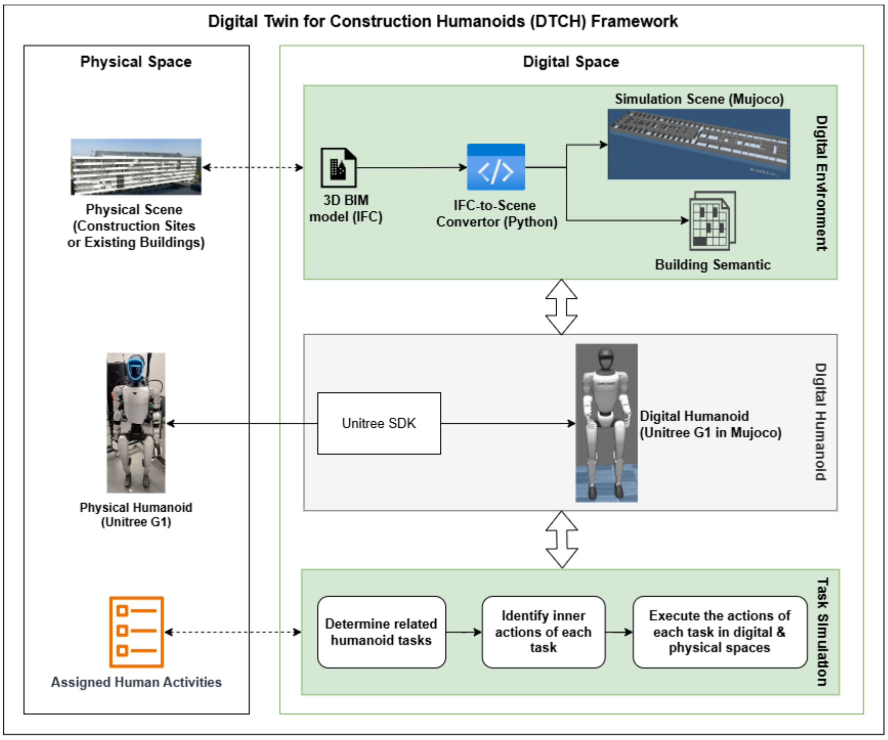

Figure 1 Overview of the proposed DTCH framework

between physical and digital spaces, specifying the physical internal components and their associated digital counterparts. In the physical space, the framework consists of the physical scene (construction sites or existing buildings), the physical humanoid robot (Unitree G1), and assigned human activities for the humanoid to perform. In the digital space, it includes three tightly coupled components: a digital environment representing the built scene, a digital humanoid that mirrors the physical humanoid, and a task simulation that defines and executes humanoid behaviors.

The digital environment provides spatial geometry and building-related information, which constrains and informs humanoid's perception and movement. The digital humanoid encapsulates the its kinematic structure, sensing configuration, and control interfaces, enabling interactions with both the digital environment and the task simulation. The task simulation operates on the digital humanoid, specifying what the robot should do and how it should act within the given environment.

3.1 Digital Environment

Within the digital environment, the workflow starts with importing a BIM model in IFC format that represents the physical scene. This model is converted by an IFC-to-Scene converter into a physics-based simulation scene and an associated building semantic file. The converter acts as an interface between BIM-based building representations and humanoid simulation, translating building information into a form suitable for contact-rich locomotion and task evaluation. Architectural elements that directly affect humanoid movement and perception, such as floors, walls, corridors, and staircases, are extracted and converted into static geometries defined in the MuJoCo Modeling XML File (MJCF) with collision properties, while preserving spatial relationships and scale.

The algorithm of the IFC-to-Scene converter is illustrated in Figure 2 with two parts. In Part 1, geometric IFC elements are tessellated into mesh assets and stored as OBJ files. In Part 2, basic building descriptors for each IFC element, including element type, spatial role, and the defined semantic label, are extracted and stored as semantic information. For example, as shown in Figure 3, an `IfcDoor` element in the IFC model is converted into a static `<geom>` in the MJCF scene containing an "`IFCDOOR_1z3daxTJHCo9bu8wlnMvKH.obj`" mesh file (Part 1), while its identifier and type (i.e., `IfcDoor`) are stored in a lightweight semantic XML file (Part 2). The resulting MJCF scene supports robot–environment interaction, and the semantic file enables environment-aware task execution within the proposed digital twin framework.

Algorithm 1 IFC-to-Scene Converter with Geometry and Semantics

```
Input  : M_ifc: IFC building model
Output: F_geo: MuJoCo modeling XML file (MJCF);
        F_sem: Building semantic XML file
Load M_ifc
/* Prepare lists for meshes and semantics                        */
Initialize geometry and semantic lists L_geo and L_sem
foreach IFC element e ∈ M_ifc do
    Extract IFC GlobalId id_e
    /* ----- Part 1:  Geometry generation from IFC model -----   */
    if e has geometric representation then
        /* Convert geometry to mesh using ifcopenshell            */
        Tessellate e into triangle mesh
        Export mesh as OBJ file
        Register mesh with reference id_e in L_geo
    else
        /* No geometry available                                  */
        Skip e
    end
    /* ----- Part 2:  Semantic generation from IFC model -----   */
    if e has IFC type information then
        /* Map IFC class to semantic label                        */
        Assign semantic label based on IFC class
        if e is structural (e.g., wall, door) then
            /* Structural elements block navigation               */
            Mark as obstacle or openable object and store with reference id_e in
             L_sem
        else if e is navigable (e.g., floor, stair) then
            /* Areas where agents can move                        */
            Mark as walkable region and store with reference id_e in L_sem
        else
            /* Other objects, e.g., furniture                     */
            Mark as general object and store with reference id_e in L_sem
        end
    else
        /* Unknown type fallback                                  */
        Assign default semantic label and store with reference id_e in L_sem
    end
end
/* Combine all meshes into MJCF scene                            */
Generate a MJCF scene file F_geo linked with the mesh references from L_geo
/* Link semantic labels to elements                              */
Generate a semantic XML file F_sem combining all entries from L_sem
/* Return the scene and semantic files                           */
return F_geo, F_sem
```

Figure 2. Algorithm of geometry (part 1) and semantic (part 2) generation from an IFC model

3.2 Digital Humanoid

As for the digital humanoid, it replicates the physical robot's kinematic structure and joint configuration, while dynamic behaviors are approximated through the physics engine of the simulation environment. Communication between the physical and digital humanoids is enabled through the officially provided Unitree SDK [12], which supports bidirectional data exchange of robot states and control commands, thereby forming a closed interaction loop between simulation and the real system as shown in Figure 1.

The Unitree SDK exposes interfaces for accessing low-level robot state information, including joint positions, velocities, and inertial measurements, as well as high-level motion and service commands. These include predefined locomotion and posture services such as `Sit()` and velocity-based motion commands `MoveTo(float x, float y, float z, float yaw)`, along with action-level interfaces for upper-body and arm execution, such as `ExecuteAction(int32_t`

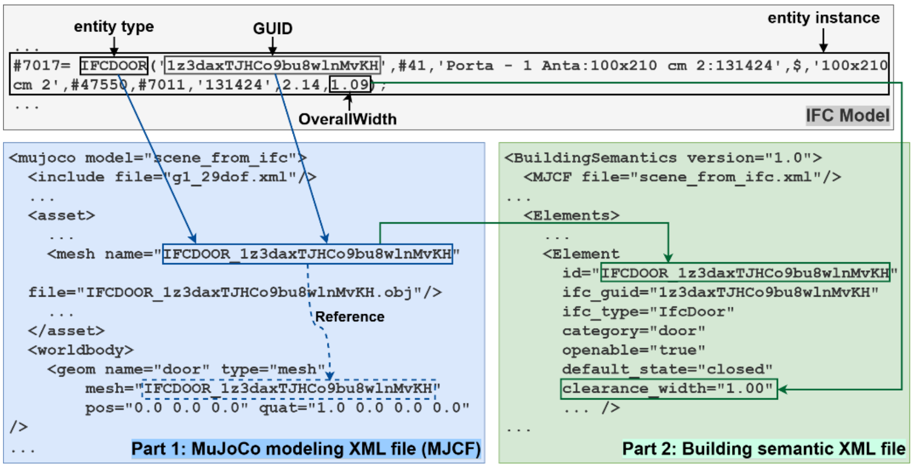

Figure 3. Conversion example from an IfcDoor instance into its scene (Part 1) and a semantic element (Part 2)

`action_id`). Through these interfaces, motion policies and task logic developed and validated in the digital environment can be systematically deployed to the physical humanoid.

3.3 Task Simulation

The task simulation provides a structured mechanism to translate assigned human activities into humanoid-executable actions. First, each assigned human activity is used to determine its related humanoid tasks according to its execution requirements. Second, each task is decomposed into inner actions with specified types and parameters. Finally, these actions are executed and evaluated in the digital space, enabling iterative refinement. This mechanism follows a rule-based and modular design philosophy, which consists of: (1) Task classification, adapted from established construction activity typologies reported in prior literature [2], and selected to represent a broad spectrum of typical construction and post-construction activities (e.g., handling, assembly, inspection, and hazardous operations), acknowledging that a single construction task may consist of multiple humanoid actions, as illustrated in Table 1. (2) Workflow specification, defining the logical execution order of tasks and their inner actions. This rule-based formulation is intentionally adopted in this study as an initial abstraction to enable clear mapping between human activities and humanoid actions. However, it does not capture adaptive or learning-based behaviors, which remain as future extensions.

To specify task classification, Table 1 presents a hierarchical decomposition of construction and post-construction human activities into humanoid tasks and function-level inner actions. Each human activity category (adapted from prior construction task classifications [2] and generalized to ensure coverage of representative lifecycle activities rather than exhaustiveness) contains a set of scoped humanoid tasks, which are further decomposed into atomic inner actions defined as callable functions with explicit input and output parameters. Each task is defined as a function or a module, specifying: The task goal (e.g., inspection navigation, object inspection), called action functions (e.g., plan inspection route, navigate route, detect inspection target, capture data), actions' sequence, and the involved humanoid subsystems (e.g., legs for walking, arms for manipulation, onboard cameras for perception). These tasks are composed into higher-level workflows (i.e., human activities), allowing different tasks to be chained together.

Table 1. Hierarchical decomposition from human activity category into humanoid actions

Human Activity Category	Humanoid Task	Inner Actions
Material Handling and Transport	Material pickup	Navigate to material; Detect material; Align for grasp; Grasp material
	Material transport	Plan transport path; Walk with load; Monitor load stability
	Material placement	Align for placement; Place material; Release grasp
Assembly and Installation	Component positioning	Navigate to assembly site; Detect assembly target; Align end-effector
	Component installation	Fine pose adjustment; Execute installation; Verify installation
Inspection and Quality Control	Inspection navigation	Plan inspection route; Navigate route
	Object inspection	Detect inspection target; Capture data

	Inspection reporting	Log inspection data; Report result
Demolition and Hazardous Work	Hazard approach	Navigate to hazard zone; Adopt safety posture
	Hazard interaction	Detect hazard target; Execute manipulation
	Safe retreat	Retreat to safe zone; Reset posture

The inner actions of each task, with structure defined in Figure 4, represent the fundamental operational units of the humanoid system and are designed for direct implementation using the Unitree SDK. Each action is defined by an identifier, an action type, certain input and output variables, and the called SDK functions. In addition, the action types are defined and shown in Table 2, indicating the level of autonomy and external assistance required during execution. For example, based on the action type definition (Table 2), in the "Material pickup" humanoid task (Table 1), the "Navigate to material" is an environment-aware action, the "Detect material" is an AI-assisted action, the "Align for grasp" is a general action, and the "Grasp material" is a hardware-assisted action which relies on end-effector hardware and sensing.

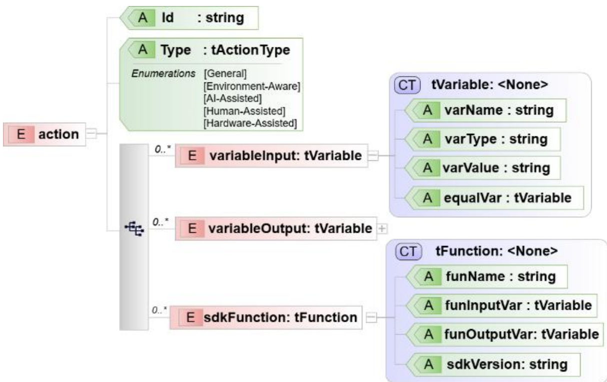

Figure 4. Defined humanoid action architecture

Table 2. Humanoid action type definition

Action-Type	Meaning (Humanoid Perspective)
General	Can be executed without additional intelligence or assistance
Environment-Aware	Requires perception of surroundings
AI-Assisted	Requires learning-based perception or reasoning
Human-Assisted	Requires supervision or teleoperation
Hardware-Assisted	Requires specialized tools or sensors

As for workflow specification, it specifies that the task execution follows a structured workflow. For each task, a goal is first defined, followed by selection of corresponding action functions. Input parameters such as target location, sensing range, or motion speed are specified. The actions are then executed sequentially in simulation, where outcomes such as balance stability, collision events, and perception coverage are evaluated. After successful validation, the same action sequence is executed on the physical humanoid using identical task definitions, ensuring consistency between simulation and real-world operation. It should be noted that, although AI-assisted actions are defined in the framework, their implementation in the current study remains limited, and most actions are executed using predefined logic rather than adaptive learning-based models.

4 Scenario Demonstration

Although the digital twin concept, which focuses on virtual simulation and evaluation, is intended to be applicable across multiple lifecycle phases, its initial demonstration is best conducted in controlled and comparatively stable environments. Commercial building operation scenarios, particularly those related to facility management, provide an ideal entry point, as these environments are typically unaffected by weather, well-documented through BIM models, and subject to fewer rapid structural changes than active construction sites. Furthermore, many facility management activities, as summarized in Table 3, are still commonly carried out by human workers, particularly those that require mobility, visual assessment, or situational judgment, even when parts of these activities are repetitive or follow standardized procedures. The examples of frequent human activities serve as concrete instances to explore the proposed DTCH framework.

To demonstrate the DTCH framework, the human activity "Escape route accessibility checks" (highlighted in Table 3) is selected as a representative human activity performed on one floor of a large office building. The digital environment is established by using the IFC-to-Scene converter. The geometry and semantic context are derived from an IFC-based floor model and represented in both MJCF and XML files. This digital environment provides the required geometrical information for environment-aware navigation, visualized as the "Floor map" in Figure 5.

Table 3. Examples of human activities in facility management

Category (Table 1)	**Examples for Frequent Human Activities**	**Key Standards**

Inspection and Quality Control	Technical inspection: Visual equipment checks; Gauge reading; Valve state verification	ISO 41001 [13]
	Safety & Emergency inspection: Fire alarm system inspection; **Escape route accessibility checks**	ISO 45001 [14]
Assembly and Installation	Maintenance: Building components replacement or resetting	ISO 55000 [15]

After setting up the digital environment, the selected human activity is first decomposed into humanoid tasks and then further into inner actions (as shown in Table 4). This human activity requires continuous locomotion through predefined routes and situational inspection when obstacles or abnormalities are encountered, which can be determined with two humanoid tasks: 1) Route-based walking, which focuses on navigation and mobility, and 2) Obstacle inspection, which focuses on localized perception and documentation. Following the workflow introduced in Section 3.3, each humanoid task is further decomposed into atomic inner actions, and each inner action is associated with an action type reflecting the required level of autonomy, perception, or assistance. For example, both "Load predefined inspection route" and "Walking along route" are general actions, the "Detect route obstacle" is an environment-aware action, and the "Capture visual data" is a hardware-assisted action.

In the "Load predefined inspection route" action, a predefined route is set within this action function using Python code. In this route, the humanoid starts from position S, walks straight, turns from W1 to W2, passes through a doorway at W3, and in the end stops at position E (as shown in "Floor map" of Figure 5). Following route initialization, the humanoid preforms the "Walking along route" action, and representative simulation results are shown in Figure 5. Upon reaching waypoint W3, the humanoid triggers the "Detect route obstacle" action, as the door is detected to be closed and cannot be opened within the robot's available strength limits. In this case, the door is flagged as an unexpected obstacle for escape route, and the humanoid stops at the location with reporting an "obstacle detected" message containing the obstacle position (the "Capture visual data (image/video)" action is not executed due to sensor limitations in the current simulation setup).

Table 4. Example of decomposing from a human activity into humanoid actions

Human Activity	**Humanoid Task**	**Inner actions**
Escape route accessibility checks	1) Route-based walking; 2) Obstacle inspection	1) Load predefined inspection route; Walking along route; 2) Detect route obstacle; Capture visual data

Although the demonstration is based on a single and a simple human activity, it initially simulates how a human activity can preform by humanoids to cover the fundamental steps of demonstrating the framework. The demonstration shows how the conceptual framework can be implemented in practice by translating environment-aware human activities into executable humanoid behaviors, with the simulation of a simple "Escape route accessibility checks" human activity. However, the current study does not include quantitative evaluation metrics such as simulation accuracy, task completion time, or computational cost, as the focus is on conceptual validation rather than performance benchmarking.

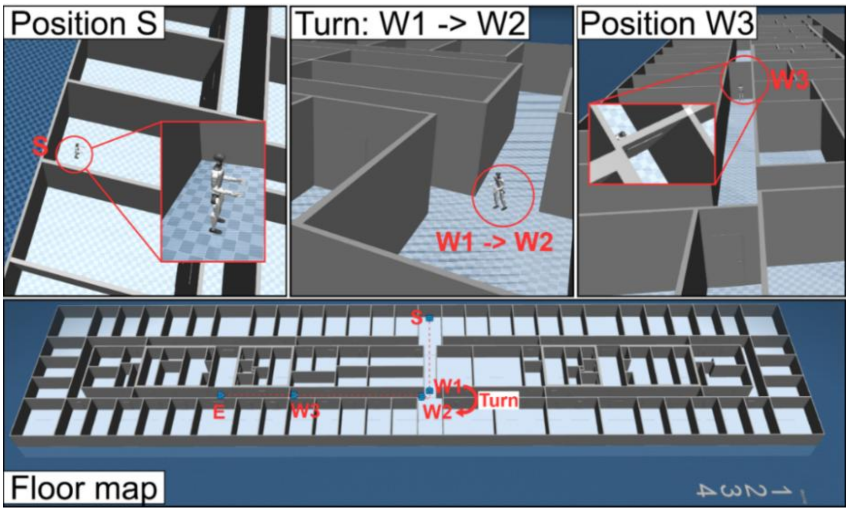

Figure 5. Initial MuJoCo-based simulation result of the humanoid performing "Escape route accessibility checks" human activity in the BIM-generated scene

5 Conclusion

This paper proposed a digital twin framework for humanoid robotics in construction and building operation contexts, with a particular focus on scene representation and task simulation. By integrating BIM-based building models, physics-based humanoid simulation in MuJoCo, and a structured task simulation module, the framework provides a systematic approach to translate human activities into executable humanoid actions. Using a facility management inspection scenario as an initial demonstration, the study illustrated how environment-aware humanoid tasks, such as route-based walking, can be defined and simulated within a virtual environment before deployment. The framework demonstrates how digital twins can support safe, repeatable, and scalable evaluation of humanoid robot capabilities in built environments.

Despite its contributions, this study has several important limitations that should be acknowledged. First, the framework is validated only through a simplified and controlled scenario, which does not capture the full complexity of real construction environments, such as dynamic changes, multi-agent interactions, or uncertain conditions. Second, the current implementation relies on rule-based task definitions and predefined action sequences, which limits adaptability in highly dynamic or uncertain environments. Third, the simulation setup simplifies perception, sensing, and actuation constraints, and does not fully represent real-world noise, sensing errors, or hardware limitations. In addition, no quantitative evaluation (e.g., accuracy, efficiency, or cost) is conducted, as the focus of this work is on conceptual framework development rather than performance validation.

Building upon these limitations, future work should incorporate learning-based methods, particularly reinforcement learning, to enable humanoids to acquire robust locomotion, perception, and decision-making policies through simulation-based training. Furthermore, while the framework conceptually supports environment-aware tasks, deeper integration of perception, object detection, and semantic reasoning remains an open challenge. Extending the framework to support richer scene interaction, such as adaptive obstacle handling, inspection reasoning, and human–robot collaboration, will be critical for real-world adoption. Finally, future task definition methods should evolve toward more standardized and reusable task templates or techniques (such as Goal Oriented Action Planning), facilitating collaboration between engineers, domain experts, and operators, and enabling humanoid robots to more directly and reliably take over routine human activities in construction and building operation workflows.

References

[1] J.M. Davila Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, and H. Owolabi. Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. *Journal of Building Engineering*, 26: 100868, 2019.

[2] T. Uthai, H. You, M. Wang, K. Smith, E. Spackman, Z. Ryan, S. Li, and J. Du. Opportunities challenges and roadmap for humanoid robots in construction, *Scientific Reports*, 2025.

[3] M. Ikuabe, C. Aigbavboa, and E. Kissi. Potential applications and benefits of humanoids in the construction industry: a South African perspective. *International Journal of Building Pathology and Adaptation*, 41: 254–268, 2023.

[4] B. Feng, J. Liu, R. Li, H. Liu, X. Zhao, B. Li, and Y. Zhang. Real-Time Remote Control Framework for Humanoid Robots Based on Motion Capture. In: *IEEE 44th Chinese Control Conference*, pages 4840–4845, Chongqing, China, 2025.

[5] L. Li, Q. Zhang, B. Xing, H. Chai, Y. Yin, and B. Liu. Reinforcement Learning-Based Whole Body Skill Behavior Generation and Control for Humanoid Robots. In: *IEEE 15th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems*, pages 462–467, Shanghai, China, 2025.

[6] Z. Li, D.K.H. Chua, Y. Liang, and S. Zhang. Automated generation of assembly-oriented template models for digital twin implementation in DfMA construction. *Automation in Construction*, 177:106314, 2025.

[7] T. Bock, T. Linner, and W. Ike. Exoskeleton and Humanoid Robotic Technology in Construction and Built Environment. In: *The Future of Humanoid Robots - Research and Applications*, 2012.

[8] M. Crotti, L. Rossini, B.K. Hodossy, A. Pace, G. Grioli, A. Bicchi, and M.G. Catalano. Soft Adaptive Feet for Legged Robots: An Open-Source Model for Locomotion Simulation. *IEEE Access*, 13: 164818–164828, 2025.

[9] G.C.R. Bethala, H. Huang, N. Pudasaini, A.M. Ali, S. Yuan, C. Wen, A. Tzes, and Y. Fang. H2 -COMPACT: Human-Humanoid Co-Manipulation via Adaptive Contact Trajectory Policies. In: *IEEE-RAS 24th International Conference on Humanoid Robots*, pages 1004–1011, Seoul, South Korea, 2025.

[10] C.-T. Vu, C.-C. Lai, and Y.-C. Liu. Learning Virtual Passive Dynamic Walking Using the Kneed Walker Model for Guiding Policies. In: *IEEE-RAS 24th International Conference on Humanoid Robots*, pages 1094–1100, Seoul, South Korea, 2025.

[11] J. Yoon, S. Lee, and Y.S. Ihn. Enhancing Humanoid Robot Dynamics: An Optimization Framework for Shoulder Base Angle Adjustment. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1108–1115, Hangzhou, China, 2025.

[12] Unitree. G1 SDK Development Guide. Online: https://support.unitree.com/home/en/G1_developer, Accessed: 15/01/2025

[13] ISO 41001, Facility management, 2018.

[14] ISO 45001, Occupational health and safety management systems, 2018.

[15] ISO 55000, Asset management, 2024.