

Unifying Design and Execution: A Hybrid Digital Twin for Robotic Fabrication in Construction

Zi Jie Tan¹, Hongjie Cai¹, Rongbo Hu¹, Sounggho Chae¹, Keiji Tanaka¹

¹Kajima Technical Research Institute Singapore, Kajima Corporation, Singapore
zj.tan@kajima.com.sg, h.cai@kajima.com.sg, r.hu@kajima.com.sg, s.chae@kajima.com.sg,
k.tanaka@kajima.com.sg

Abstract –

Robotic fabrication holds significant potential for improving productivity in construction, yet its adoption remains limited by the task variability, high-mix low-volume production, and the tight coupling requirement between design and production. Furthermore, translating design intent into physical production often relies on fragmented software ecosystems operated by different specialists, increasing coordination effort and limiting flexibility, thus, highlighting a need for a more integrated approach. This paper presents a hybrid digital twin framework that unifies a design-centric environment with a control-centric environment into a single platform for flexible, customized robotic production. The design layer is implemented in Rhinoceros and Grasshopper, leveraging their widespread adoption in AEC for parametric modelling and interoperability, while the control layer is built on HAL Robotics and ROS 2, providing an agnostic and extensible backend for motion planning, hardware abstraction, and real-time execution. Bidirectional data flow between design, simulation, and physical robots enables users to import project data, prototype task logic, and operate multiple robots within a cohesive platform. This unification lowers technical barriers and demonstrates a scalable pathway toward democratizing robotics in construction.

Keywords –

Automation; Digital Twin; Simulation; Motion Planning; Collision Avoidance; Digital Fabrication

1 Introduction

The construction industry remains one of the most labor-intensive sectors, with numerous processes that could benefit from automation. Among these, fabrication tasks present a significant opportunity for improvement through robotic technologies [1]. While industrial robots have demonstrated strong potential for automating fabrication, their application in construction faces persistent challenges due to highly customized nature of components specific to projects and predominance of high-mixed, low-volume production. This stands in

contrasts with conventional robotic deployments, which are optimized for repetitive motions and low-mix, high-volume production, and typically require skilled engineers to program, calibrate, and operate [2]. Another key challenge for robotic fabrication lies in the tight coupling between design and production. Unlike industrial manufacturing, where geometry is largely fixed prior to fabrication, construction workflows require continuous geometric authoring and validation in response to material behaviour, fabrication constraints, and assembly logic. Effective robotic deployment in this context thus depends not only on motion planning, but also on robust integration with design environments capable of preparing production-ready geometric instruction [3,4]. To realize the full potential of robotic fabrication in construction, robotic systems must therefore be intuitive to program, operate, and manage, even by non-specialist users.

Digital twins offer a promising means to address this by providing a virtual representation of the physical environment in which project data can be integrated, robotic tasks planned, and validated workflows transferred to physical robots for execution. In addition, digital twins can also support real-time monitoring and adaptive control to reduce downtime and repeated reprogramming [5]. In practice, however, implementations remain largely decoupled. Robotic simulation platforms typically function as control-centric digital twins, offering motion planning and execution but limited support for geometric authoring or design iteration. Conversely, design-centric modelling tools act as design-oriented digital twins, excelling in geometric authoring and coordination while lacking native capabilities for robot control. This division causes design and robot-control domains to evolve independently, forcing practitioners to rely on disjointed software pipelines and specialist expertise.

To address these limitations, we propose a hybrid digital twin framework that integrates a design-centric layer with a control-centric robotics layer within a unified platform. The design layer supports the import, authoring, and validation of geometric models while embedding fabrication constraints and task sequencing early in the workflow. The control layer provides collision-aware motion planning, kinematic reasoning, real-time sensor feedback, and execution.

2 State of the Art Analysis

2.1 Key Requirement for Hybrid Digital Twin

2.1.1 3D Modelling Capability

In construction projects, building information modelling (BIM) data captures extensive building information, but are typically authored for documentation and coordination rather than direct robotic execution [6]. Consequently, downstream users must reinterpret or refine BIM geometry, adjusting tolerances, segmentations, generate toolpaths to make it fabrication-ready, reflecting a largely top-down workflow where design information flows into production without being inherently production-aware [7]. Conversely, many bespoke or materially driven construction tasks follow a bottom-up approach, where designers work directly with material behavior and fabrication constraints. This allows geometry and manufacturability to co-evolve, enabling unique, project-specific component, most evident in architecture domain [8,9]. A modelling environment to support both top-down BIM inputs and bottom-up fabrication driven design is therefore essential.

2.1.2 Robot Control

At its core, robot control involves computing forward and inverse kinematics, simulating motions, and visualising robot behavior within the virtual environment [10]. Beyond trajectory generation, a good robot control should also extend to perception-driven pipeline, where sensor data are processed to update the robot's understanding of its environment and to adapt motions accordingly [11]. In addition, robot control also governs

the physical hardware. This may occur through offline programming- where joint trajectories are post-processed and compiled into robot-specific code to run in the robot controller, or online programming where libraries translate high-level commands into low-level motor signals for real-time operation. C.J Lian et al. demonstrated this by constructing the communication bridge between a KUKA KR120 simulation and real hardware using PLC & RSI module to read and write data in real time, including sensor data and joint values [12].

2.1.3 Seeing Beyond First Person

In construction, proposed digital twin should also help users expand their spatial awareness for more effective motion planning. A 3D environment is therefore essential, as it provides a unified space for geometric and sensor data which can be updated frequently, allowing users to inspect the scene from any viewpoint, overcoming the single-angle limitations of the physical environment. This benefit is similarly demonstrated by Mori et al. where enhanced spatial perception proved valuable for remote operation in a constrained setting [13]. Having a 3D environment also proved beneficial for training neural network to plan conflict free navigation and manipulation whilst carrying large objects, as evidenced by Cai et al. in the case of transporting long rebar from one position to another [14].

2.2 Digital Twin Platform

Based on the aforementioned requirements, several potential software platforms were identified and compared in Table 1, each presenting distinct advantages and limitations.

Table 1. Software comparison

Software	Robot Control Support	3D Modelling Capability	Physics & Sensor Simulation	Interoperability	BIM Connectivity	Typical Use in Robotics	Acceptable Import File Types
ABB RobotStudio [15]	Native ABB only	Moderate modelling	Basic physics	RAPID, PC SDK, Robot Web Service	Imports CAD; no IFC/RVT; BIM via external pre-processing	ABB programming, cycle optimisation	STEP, IGES, STL, JT, DWG/DXF
Unity [16]	External via ROS2 and C#;	Strong Mesh based. game-engine level world building	Good physics	Python, C#, C++	IFC import via plugins; Revit review via Unity Reflect/Industry;	Digital twins, visualization, VR training, simulation	FBX, OBJ, GLTF/GLB, PNG/JPG, STL, STEP/IGES
Rhino + Grasshopper [17]	External via COMPAS FAB, HAL Robotics, KUKA Prc, Robots, Taco ABB etc.	Excellent CAD/ NURBS/ Mesh; parametric modelling	Limited physics	Python, C#, Grasshopper	IFC import via plugins; Native supports BIM semantics through RhinoInsideRevit	Robotic fabrication, automated design-to-robot workflows	3DM, STEP, IGES, STL, OBJ, DWG/DXF, E57, LAS, CSV
ROS 2 [18]	Full robot planning & control (MoveIt!, ROS control)	limited modelling, depends on simulator	Depends on simulation engine	Python, C++, DDS	BIM→CAD→URDF/SDF conversion; IFC via external converters	Autonomy, planning, SLAM, multi-robot coordination, hardware control	URDF, XACRO, SDF, STL, OBJ, PCD, CSV

Software	Robot Control Support	3D Modelling Capability	Physics & Sensor Simulation	Interoperability	BIM Connectivity	Typical Use in Robotics	Acceptable Import File Types
NVIDIA Isaac Sim [19]	ROS2 native support	High-end 3D via USD ecosystem, omniverse	High-fidelity physics + photoreal sensors	Python, C++, CUDA, USD	BIM via Omniverse Revit/IFC Connectors; maintains metadata; supports IFC → USD workflows	Digital twins, perception, multi-robot simulation	USD/USDC, OBJ, FBX, GLTF, STL, STEP/IGES, URDF, PCD
RoboDK [21]	Native multi-brand controller support;	Limited modelling	Basic physics	Python, C# API, robot specific post-processors	No native BIM; requires external CAD/BIM preprocessing	Offline programming, toolpath generation	STEP, IGES, STL, OBJ, DXF, CSV
MATLAB/Simulink [22]	Robotics System Toolbox, ROS2 support	Depends on external (Unreal Engine)	Good physics	MATLAB, Simulink API, C/ C++	IFC via third-party toolbox; Revit/IFC → MATLAB conversion needed; semantics limited	Controller design, HIL testing, planning	URDF, STL, STEP (converter), PCD, CSV, MAT

Based on the information gathered, Rhinoceros with Grasshopper emerges as a suitable front-end environment for construction robotics due to its ease of use, strong modelling capabilities, broad interoperability across data formats, native BIM support and high compatibility with external software. While it does not natively provide advanced robot control or real-time planning functionalities, its open ecosystem and APIs enable integration with external robotics middleware such as ROS. This allows the design environment to be effectively coupled with a control-centric system, combining intuitive modelling and task authoring with robust robotic planning and execution. This has been achieved by COMPAS FAB, which demonstrated similar capability in the same environment but it requires higher level of software expertise such as python scripting, making it harder to adopt for general user. Overall, a hybrid configuration leverages the strengths of both environments and mitigates their individual limitations, making it well-suited for construction where components

are highly variable and robotic systems are often operated by non-specialist users.

3 Methods

3.1 System Architecture

We develop a hybrid digital-twin framework that links a design-centric layer with a control-centric robotics layer bi-directionally, preserving the strengths of each while connecting them through well-defined data flows as shown in Figure 1. Synchronizing these layers into a hybrid digital twin leverages the strengths of both domains: 1) the intuitive modelling and task authoring capabilities in the design-centric layer, 2) the accessible offline programming in Grasshopper using HAL Robotics, and the robust, intelligent planning + control capabilities of ROS2 on the control-centric layer.

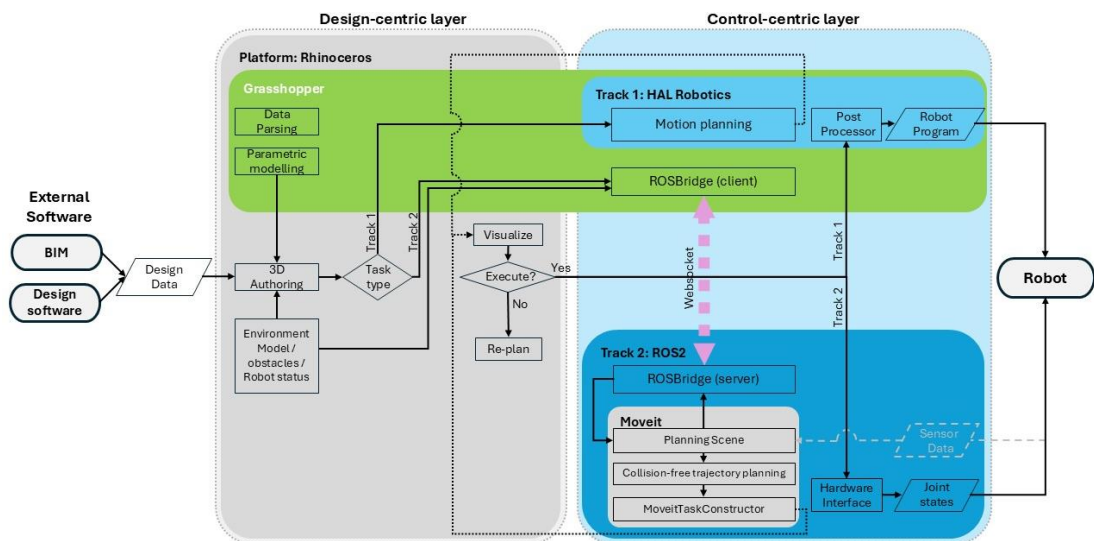


Figure 1. Bi-Directional hybrid digital twin architecture

3.1.1 Design-Centric Layer

The design-centric layer is intended for architects, engineers, and fabricators to import data, manipulate geometry, explore form, and author task logic in a visual, interactive manner. We implemented this layer in Rhinoceros and Grasshopper, a 3D modelling software with a mature computational ecosystem widely adopted across the AEC industry. A major advantage of this environment is its interoperability and extensibility. Rhinoceros natively supports import of a wide variety of 3D formats. Tools such as RhinoInsideRevit further strengthens the pipeline between BIM and 3D modelling [23]. Grasshopper's scripting capabilities and the RhinoCommon API also allow users to build custom components to automate workflows such as G-code generation, working with external libraries and even designing prototype graphical user interface (GUI) with plugins like Human UI [24], thus, facilitating seamless exchange of geometric data between upstream BIM tools and downstream fabrication processes. This openness makes the Rhino a natural front-end for importing upstream design intent and pre-processing geometry for robotic execution.

3.1.2 Control-Centric Layer

The control-centric layer translates design intent into executable robotic actions, performing motion planning, and integrating sensor feedback for adaptive control. In our framework, control operates along two complementary tracks and is used interchangeably depending on the nature of the task.

- Track1: Grasshopper-based control (HAL Robotics):

The first track is HAL Robotics, a plugin that allows users to simulate robot kinematics, generate toolpaths, and visualise robot motion in the design environment [25]. HAL Robotics supports a wide range of robots' brand and is well maintained by the developer. External axis, custom end effector can be easily modelled and integrated into the environment using its extensive commands built inside Grasshopper. Upon successful simulation, it generates a robot program which can be easily uploaded to the controller for execution. While this track provides accessible offline programming and design-intuitive feedback, it does not include intelligent trajectory planning to avoid obstacles or maintain certain constraints, which leads to the following track.

- Track2: ROS 2-based control:

The second track is based on ROS 2, a modular middleware optimized for real-time robotic control. This

includes kinematic and dynamic computation, motion planning with collision avoidance through MoveIt! [26] and perception pipelines that process sensor data to inform closed-loop adjustments. ROS2 is also capable of interfacing directly with hardware, enabling real-time trajectory execution. Furthermore, ROS2 exposes robot states and sensor feedback through standard topics and services, which can be visualised and read and used in Rhino/Grasshopper with a custom plugin subscribing over ROSBridge [27], providing a continuous feedback loop between the design and control layers.

3.2 From Design to Production

The hybrid digital twin couples the modelling capabilities of a design environment with the operational intelligence of a control environment in a bi-directional manner. Users can stream design data from BIM or other design software into the design layer to author geometry, simulate fabrication logic, validate kinematic feasibility, and directly deploy robot motions, all within a unified pipeline. Depending on the type of tasks, different tracks will be taken in the control layer. Once the robotic motion is generated, it can be visualized inside the design layer once more before user decides to execute or to make changes to the design to improve validity. Lastly, deploying on the robot will depend on the chosen track in the control layer. HAL robotics path uploads a robot program to the controller while ROS can directly drive the robot via a hardware interface.

3.3 Networking infrastructure

Reliable data exchange is critical given the number of software and hardware components in the proposed framework. To support mobility and uninterrupted robot communication, the system leverages the laboratory's existing 5G network infrastructure, which includes a private 5G base station, a wall-mounted repeater, and multiple distributed routers that provide full wireless coverage. It offers high throughput and low latency required for data-intensive tasks such as streaming sensor information for computation and visualisation within the digital twin environment. Devices connected through this 5G system can also be teleoperated remotely, provided they are linked to a router equipped with a 5G SIM card that has a static IP address, enabling secure, real-time control over long distances [28]. The laboratory's autonomous guided vehicle (AGV) already operates wirelessly on this 5G network. Future work will focus on integrating it into the hybrid digital twin, allowing both stationary and mobile robotic systems to be coordinated within a unified environment

3.4 Incorporating Intelligence to Robot

3.4.1 Collision-Avoidance Feature

Beyond translating geometry into robot instructions, the proposed digital twin also aims to provide capabilities that exceed conventional simulation tools. Traditional robot simulators typically offer waypoint-to-waypoint planning and lacks advanced feature such as collision avoidance or task-specific constraints factors that usually require specialist intervention. Embedding these capabilities directly into the digital twin increases practical value and accessibility for non-robotics users. This is possible through MoveIt!, part of ROS 2 motion-planning framework. MoveIt! provides high-level interfaces for inverse kinematics, constraint-aware planning and trajectory optimisation [26]. In the proposed architecture, a ROS 2 instance serves as the computational backend: the design layer sends target poses, obstacles, and MoveIt! returns feasible, collision-free trajectories.

3.4.2 Automated Workflow

MoveIt! Task Constructor (MTC) can further decomposes complex tasks into modular step known as stages, which can allow AEC practitioners to execute task-level plans without interacting with low-level robotics abstractions [29]. This reduces operational complexity while providing access to advanced planning capabilities traditionally reserved for robotics experts. One example is a pick & place task as shown in Figure 2. Instead of relying on user to teach the entire sequence, an automated reusable template can be setup beforehand on ROS side and called from the hybrid digital twin frontend.

Task Tree	name	time
▼ Motion Planning Tasks		
▼ Pick and Place		
current state	1	0 16.7890
open hand	1	0 0.3624
Go to startingPos	1	0 0.0003
Standby	1	0 0.0012
pick object	1	0 0.0013
move to pre-p...	1	0 0.0216
approach object	1	0 0.0080
allow collision	1	0 0.0022
close hand	1	0 0.0001
attach object	1	0 0.0001
lift object	1	0 0.0018
Move to retreat	1	0 0.0095
computeRetreatIK	1	0 6.7830
GenerateRetreatIK	5	0 0.0086
Move to pre-place	1	0 0.0000
place object	2	0 9.5629
computePlaceIK	17	3 0.0393
GeneratePlaceIK	2	0 0.0197
approach object	2	0 0.0001
open hand	2	0 0.0190
detach object	2	0 0.0003
forbid collision	2	0 0.0002
retract	2	0 0.0002
Go Home	2	0 0.0043

Figure 2. MTC stages

3.5 Digital Twin Setup & Data Exchange

The early prototype system comprises two computers:

one running Rhinoceros for the design layer and the other running ROS 2 for the control layer as shown in Figure 3. Communication between the two is achieved via ROSbridge, using a custom Grasshopper-based client that connects to the ROS 2 server over a WebSocket (TCP) interface and exchanges messages encoded in the ROSbridge JSON protocol. Through this interface, the design layer transmits target goals and environmental data to iteratively update the MoveIt! planning scene in ROS 2. Collision-aware trajectories generated by MoveIt! are then streamed back to Rhinoceros for real-time visualisation and validation. A custom GUI built in the design layer further abstracts this process, enabling users to interact with the hybrid digital twin without direct exposure to ROS-specific configurations. This bidirectional data exchange allows live robot states and sensor feedback to be visualised in the design environment, while updated geometry and scene information can be rapidly propagated to the control layer for replanning. As such, real-time data exchange constitutes a core capability of the proposed hybrid digital twin, supporting adaptive behaviour and integrated design-to-execution workflows.

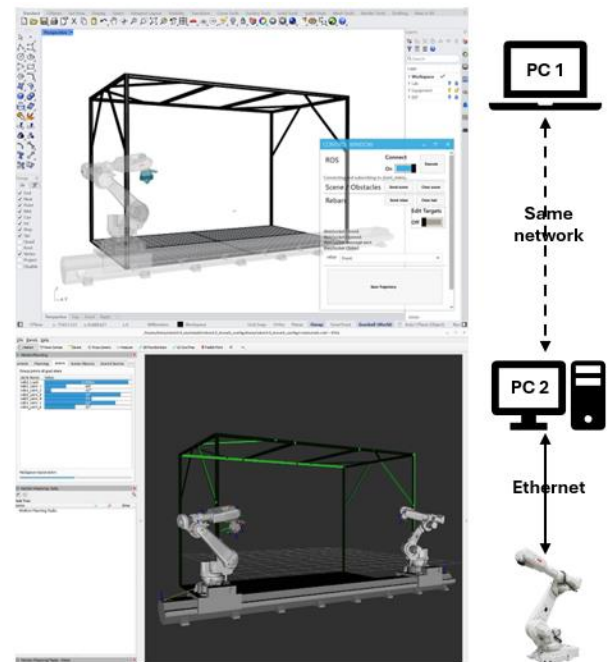


Figure 3. Bi-Directional hybrid digital twin setup

3.6 Verification Experiment

To assess the practical value of the proposed hybrid digital twin framework, we designed an experiment comprising several robotic processes, each representing a different construction modality. This multi-process demonstration shows that the framework can seamlessly

coordinate diverse tasks—additive and subtractive manufacturing as well as assembly within a single pipeline. The goal is to illustrate that heterogeneous robotic operations can be executed across different machines without changing software environments or learning separate programming interfaces. Using a concrete panel as an example, the workflow consisted of three stages: (1) milling architectural surface details using a subtractive robot, (2) 3D printing an unconventional formwork, and (3) employing an assembly robot to place reinforcement bars intelligently into the formwork with obstacle avoidance.

4 Results

4.1 Fabrication Processes

4.1.1 Subtractive Manufacturing: CNC Milling

Using the proposed digital twin framework, a CNC milling operation was first executed. The toolpaths were authored in the design layer and transferred to the control layer via Track 1, implemented through HAL Robotics, for kinematic simulation and offline motion planning. After successful validation, a robot-specific RAPID program was generated using HAL's post-processor and uploaded to the robot controller via the Robot Web Services (RWS) interface for execution. During operation, a laser tracker-based calibration procedure was performed, and live joint states were streamed back into Rhinoceros at 20–50 Hz through a custom C# script in Grasshopper. This enabled real-time visualization of the robot's motion within the design environment, allowing discrepancies to be identified and adjustments to be made, when necessary, as shown in Figure 4. Upon completion of the milling process, the panel was transferred to the subsequent fabrication stage.

4.1.2 Additive Manufacturing: 3D Printing

In this stage, the formwork of the concrete panel was first sliced to produce printing toolpath using predefined extrusion parameters (bead width, layer height) in the design layer as pre-processing step. Design data can also be imported if preferred. Using a predefined workObject reference frame, the toolpath generated was re-aligned and sent to HAL Robotics for motion planning followed by uploading to the robot for execution. In the physical world, after the CNC-milled panel was transported and positioned, the printing began operation. In theory, the robot can be jogged with the digital twin to three points to define a workObject (local reference frame) and compensate for transport-induced misalignment. Future vision-based detection can also be employed to automate this calibration and further reduce human intervention. However, due to hardware limitation at the point of

research, this feature was omitted. During printing, robot states were monitored real-time, ensuring the robot does not enter undesirable configurations seen in Figure 5.

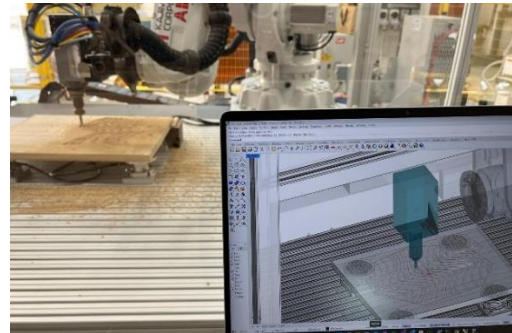


Figure 4. CNC milling process

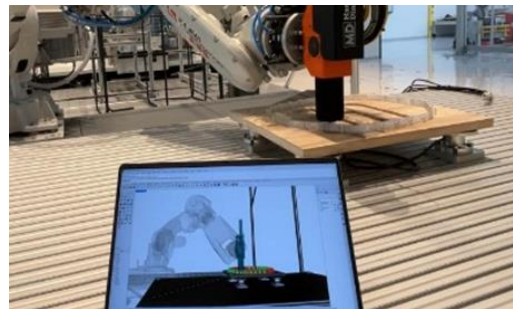


Figure 5. 3D printing with live monitoring

4.2 Material Handling Process

4.2.1 Pick & Place of Rebar

The final part of the demonstration highlights the intelligent path-planning capabilities enabled by ROS 2 as control layer. After the 3D-printed component was placed within reach of the third robot arm equipped with a gripper, the surrounding environment, particularly the laboratory's cage was streamed into the control layer via track 2 (ROS 2) for obstacle-aware planning shown in Figure 6. Using this environmental data, ROS computed a collision-free trajectory that allowed the robot to retrieve a rebar from within the cage and accurately place it onto the 3D-printed formwork. To facilitate easy control of robot, a customized GUI is designed to prompt user to select key poses such as the pick location, place location, and an intermediate waypoint. The selected data is then sent to the control layer via ROSBridge to be used for planning. At the control layer, MoveIt! automatically generates the optimal trajectory using the data received, enabling high-quality motion without demanding deep robotics expertise. The trajectory is then visualized inside the 3D environment in Rhinoceros. Once validated, the robot motion can be executed directly, which in our case,

via the External Guided Motion (EGM) on the ABB robot.

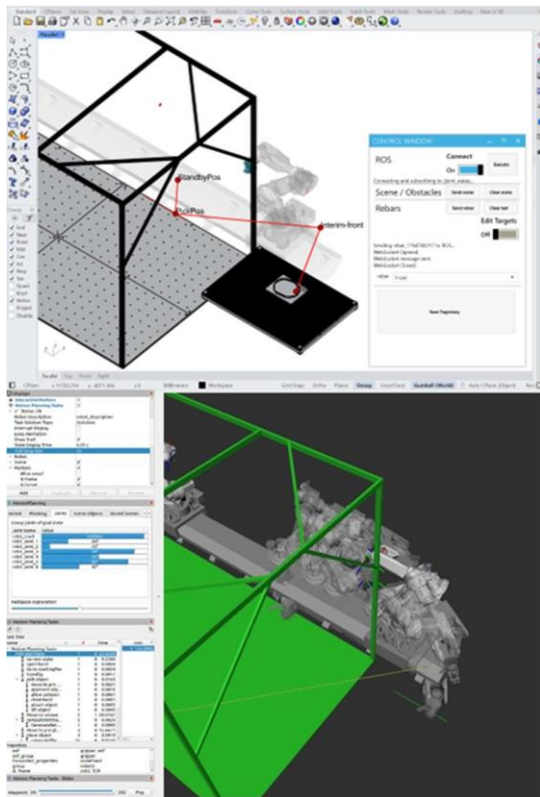


Figure 6. Pick & place application

5 Discussion

5.1 Implications to the Construction Industry

The proposed hybrid digital twin framework offers the AEC industry a practical entry point for adopting robotics, while also providing a foundation for future development. While robotics has demonstrated significant potential to improve productivity across off-site fabrication and on-site construction tasks, its adoption remains limited due to the high programming barrier and reliance on specialised robotics engineers. By enabling construction practitioners to interface directly with design data, simulation, and robot control through familiar tools, the framework lowers this barrier and broadens access to robotic operation and development.

5.2 Limitations & Future work

Although the framework successfully connected different layers into a unified environment, much of the tools are still manually created for a specific operation, which constrains scalability and reusability. Also, at

present, only a subset of robotic control parameters, particularly those exposed through ROS are accessible from the design layer, limiting fine-grained tuning by non-expert users.

Another limitation of the current study is the absence of a systematic quantitative evaluation of the proposed workflow. The case studies presented in this paper primarily serve as proof-of-concept demonstrations to validate the interoperability between the design environment (Rhinoceros/Grasshopper) and the control layer (ROS2 and MoveIt!) across multiple fabrication processes. Future work will focus on conducting application-specific benchmarking, including measurements of setup time, task execution time, and fabrication accuracy, as well as comparisons with conventional robotic programming workflows. This will also include the study of automated workObject calibration to further enhance the workflow. A key next step is to modularise the framework so that users can create abstraction packages that plug seamlessly into either Grasshopper or ROS, enabling broader reuse and customisation. Expanding the framework to integrate mobile robots and dynamically updating environments will further increase relevance to real world conditions. Lastly, applying the framework to a wider range of construction tasks will also help validate its generality and scalability.

6 Conclusion

The experiments demonstrate that the proposed hybrid digital twin framework enables multiple robotic operations within a single platform, offering higher-level abstraction and simplified control of diverse hardware. Once configured, the framework supports the fabrication of varied design instances without additional effort, making it well suited to high-mix, low-volume production. Integration with ROS further extends system capabilities, validating the effectiveness of the framework for robotic fabrication and its potential to improve productivity in construction.

References

- [1] Davila Delgado, J. M., Oyedele, L., Ajayi, A., Akanbi, L., Akinade, O., Bilal, M., Owolabi, H., Robotics and automated systems in construction: Understanding industry-specific challenges for adoption., *Journal of Building Engineering*, 26, 100868, 2019
- [2] Bilancia, P., Schmidt, J., Raffaeli, R., Peruzzini, M., Pellicciari, M., An Overview of Industrial Robots Control and Programming Approaches., *Applied Sciences*, 13(4), 2582, 2023

- [3] Zhao, C., Chen, J.-Y., Sun, T., Fan, W., Sun, X.-Y., Shao, Y., Guo, G.-Q., Wang, H.-L., Robotic motion planning for autonomous in-situ construction of building structures., *Automation in Construction*, 171, 105993, 2025
- [4] Dolinšek, B., Duhovnik, J., Robotic assembly of rebar cages for beams and columns., *Automation in Construction*, 8(2), 195–207, 1998
- [5] AlBalkhy, W., Karmaoui, D., Ducoulombier, L., Lafhaj, Z., Linner, T., Digital twins in the built environment: Definition, applications, and challenges., *Automation in Construction*, 162, 105368, 2024
- [6] Anane, W., Iordanova, I., Ouellet-Plamondon, C., BIM-driven computational design for robotic manufacturing in off-site construction: an integrated Design-to-Manufacturing (DtM) approach., *Automation in Construction*, 150, 104782, 2023
- [7] Momeni, M., Relefors, J., Khatry, A., Pettersson, L., Papadopoulos, A. V., Nolte, T., Automated fabrication of reinforcement cages using a robotized production cell., *Automation in Construction*, 133, 103990, 2022
- [8] Carpo, M., The Second Digital Turn: Design Beyond Intelligence., 2017
- [9] Bruun, E. P. G., Ting, I., Adriaenssens, S., Parascho, S., Human–robot collaboration: a fabrication framework for the sequential design and construction of unplanned spatial structures., *Digital Creativity*, 31(4), 320–336, 2020
- [10] Siciliano, B., Robotics: Modelling, Planning and Control., 2010
- [11] Luo, J., Zhou, X., Zeng, C., Jiang, Y., Qi, W., Xiang, K., Pang, M., Tang, B., Robotics Perception and Control: Key Technologies and Applications., *Micromachines*, 15(4), 531, 2024
- [12] Liang, C.-J., McGee, W., Menassa, C., Kamat, V., Bi-Directional Communication Bridge for State Synchronization between Digital Twin Simulations and Physical Construction Robots., *37th International Symposium on Automation and Robotics in Construction*, 2020
- [13] Mori, Y., Wada, M., Development of 3D Digital-Twin System for Remote-Control Operation in Construction., *41st International Symposium on Automation and Robotics in Construction (ISARC 2024)*, 2024
- [14] Cai, H., Hu, R., Quek, S. T., Chae, S., Yeoh, J. K. W., Neural Network-Based Estimation of Rebar Deflection for Robotic Rebar Installation of Vertical Structures., *42th International Symposium on Automation and Robotics in Construction*, 2025
- [15] ABB. RobotStudio Suite. On-line: <https://www.abb.com/global/en/areas/robotics/products/software/robotstudio-suite>, Accessed: 20/10/2025.
- [16] Unity Technologies. Unity Reflect. On-line: <https://learn.unity.com/project/introduction-to-unity-reflect>, Accessed: 22/10/2025.
- [17] Robert McNeel & Associates. Rhino and Grasshopper Developer Documentation. On-line: <https://developer.rhino3d.com/en/>, Accessed: 22/10/2025.
- [18] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W., Robot Operating System 2: Design, architecture, and uses in the wild., *Science Robotics*, 7(66), eabm6074, 2022
- [19] NVIDIA Developer. Isaac Sim. On-line: <https://developer.nvidia.com/isaac/sim>, Accessed: 14/10/2025.
- [20] Optimizing BIM Workflows Using USD at Every Design Phase., *NVIDIA Technical Blog*, 2023
- [21] RoboDK. RoboDK API. On-line: <https://robodk.com/doc/en/RoboDK-API.html>, Accessed: 12/10/2025.
- [22] MATLAB and Simulink for Robot Manipulators. MathWorks. On-line: <https://www.mathworks.com/solutions/robotics/robot-manipulators.html>, Accessed: 03/10/2025.
- [23] Robert McNeel & Associates. Rhino.Inside®.Revit. On-line: <https://www.rhino3d.com/inside/revit/1.0/>, Accessed: 22/10/2025.
- [24] Robert McNeel & Associates. HumanUI. On-line: <https://www.food4rhino.com/en/app/human-ui>, Accessed: 23/11/2025.
- [25] Gobin, T., Andraos, S., Schwartz, T., Vriet, R., A framework for modelling, simulating and parametrically programming heterogeneous industrial machines., *Construction Robotics*, 6(2), 101–108, 2022
- [26] Coleman, D., Sucan, I., Chitta, S., Correll, N., Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study., 2014
- [27] Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O. C., Rosbridge: ROS for Non-ROS Users., *Robotics Research*, 100, 493–504, 2017
- [28] Lee, H. J., Krishnan, A., Brell-Cokcan, S., 5G in construction: from deployment to evaluation for robotic applications., *Construction Robotics*, 8(1), 2024
- [29] Gornier, M., Haschke, R., Ritter, H., Zhang, J., MoveIt! Task Constructor for Task-Level Motion Planning., *2019 International Conference on Robotics and Automation (ICRA)*, 190–196, 2019