FIM-Based Task and Motion Planning for Robotic Bricklaying

Martin Slepicka[1,2], Canberk Yalçıklı[1,2], Mohammed Reza Kolani[1,2], Stavros Nousias[1,2] and André Borrmann[1,2]

[1]Chair of Computing in Civil and Building Engineering, Technical University of Munich, Germany
[2]Georg Nemetschek Institute, Technical University of Munich, Germany

mohammadreza.kolani@tum.de, martin.slepicka@tum.de

Abstract -

Construction robotics has a significant potential to automate a wide range of construction tasks, but it requires a structured pipeline that links high-level design representations to concrete robotic actions. Robotic bricklaying is particularly well-suited to automation because it consists of repetitive, predefined steps. However, it also introduces process-planning challenges that must be resolved at the level of individual components. Fabrication Information Modeling (FIM) has been introduced as an intermediate layer between design and fabrication for construction-scale additive manufacturing. This study extends the Fabrication Information Modeling framework to enable the transformation of high-level architectural design data into detailed, machine-interpretable specifications suitable for robotic bricklaying operations. Specifically, it proposes an integrated FIM-based data and process representation that supports the systematic planning and execution of automated bricklaying tasks. The framework's utility is validated through detailed bricklaying simulations and a successful real-world application within a controlled laboratory environment.

Keywords -

Fabrication Information Modeling; Industry Foundation Classes; Automation in Construction; Robotized Bricklaying

1 Introduction

Robotics in construction is increasingly adopted to automate repetitive, rule-based tasks in domains such as masonry, welding, and structural steelwork. Through their inherent precision and reliability, robotic systems enhance both occupational safety and operational productivity in hazardous, labor-intensive construction environments [1]. Brick wall assembly, characterized by standardized operations, is a prime candidate for robotic automation [2]. However, achieving a fully automated workflow requires addressing the inherent geometric complexities of masonry [3] and establishing a digital system that bridges building design with automated construction processes [4].

Building Information Modeling (BIM) serves as the foundation, providing a semantically rich digital representation of the built asset [5, 6]. Despite the advancements in digital modeling, a significant gap persists between BIM-based design data and robotic task execution. Robots remain unable to autonomously interpret the high-level, human-centric abstractions used to define BIM entities into the precise sequences required for physical construction [7]. While BIM provides a robust geometric and semantic foundation for Industry 4.0 integration, bridging this gap requires a dedicated methodology that extends BIM data into digital manufacturing workflows, transforming static information into actionable robotic instructions [8].

Fabrication Information Modeling (FIM) acts as a BIM-based digital manufacturing framework that transforms high-level design data into detailed, robot-interpretable instructions. By establishing a cyber-physical link between digital models, processes, and the physical asset, FIM enables integration across the design, fabrication, and assembly phases [4].

This study proposes an FIM-based task-planning framework for automated robotic brick wall assembly. The framework enriches the BIM model by incorporating a fabrication information model, which adds semantic fabrication attributes at the brick level and enables robotic execution of the assembly plan. This process translates high-level entities (e.g., wall definitions) into granular data that supports masonry task planning and execution. The proposed approach was integrated into a robotic simulation framework and validated through real-world robotic bricklaying.

The remainder of this paper is organized as follows: Section 2 provides background on robotic bricklaying, fabrication information modeling in various applications, and FIM-based planning and execution, and FIM; Section 3 details the proposed methodology; Section 4 describes the system implementation in simulation and laboratory environments; Section 5 presents the results; and finally, Section 6 concludes the paper.

2 Background

Automation in construction has become a central research focus, as digital technologies and robotic systems are increasingly being investigated for their potential to enhance productivity, safety, and efficiency on construction sites[7]. Within this broader research context, robotic bricklaying remains a promising yet relatively understudied area of construction automation. It provides a valuable basis for investigating task and motion planning, execution strategies, and information requirements in automated construction workflows.

2.1 Robotic bricklaying

Several studies have explored the integration of robotics into bricklaying workflows at different levels of automation. Early methods demonstrated that human–robot collaboration, often facilitated through mobile or augmented systems, can adapt to on-site conditions, thereby enhancing placement accuracy and reducing material waste [9]. Moving beyond collaborative scenarios, the automation of cable-driven robotic systems has been studied using BIM-driven simulation frameworks that optimize robot kinematics, process physics, and site logistics, resulting in collision-free and energy-aware trajectories [10]. Higher levels of automation have been achieved in a robotic masonry context; while Zhou et al. [11] embedded viscoelastic mortar behavior into a force–position feedback controller, enabling physics-informed adjustment of trajectories, velocities, and bond thickness. Complementing these, data-centric pipelines such as the IFC-to-Robot Operating System (ROS) workflow [7] have begun to translate BIM/IFC wall models into brick-level Cartesian pick-and-place tasks and validate them in a simulation environment. Furthermore, [12] presents a multi-robot coordination system for bricklaying that optimizes robot placement, material location, and task decomposition to reduce overall construction time.

Taken together, existing studies remain project- or platform-specific and only partly address full automation of generating robot-interpretable task data from digital design entities. This leaves robotic brick-wall assembly and construction in need of a framework that can systematically handle robotic process planning down to individual brick components, while resolving discrete process-planning challenges, such as junction assembly scenarios involving T- and L-junctions of connected brick walls within a workspace and an arm robot's reachability.

2.2 Fabrication Information Modeling

Fabrication Information Modelling (FIM) is a BIM-centric digital fabrication methodology introduced to the construction sector by [4, 8]. It is designed to structure

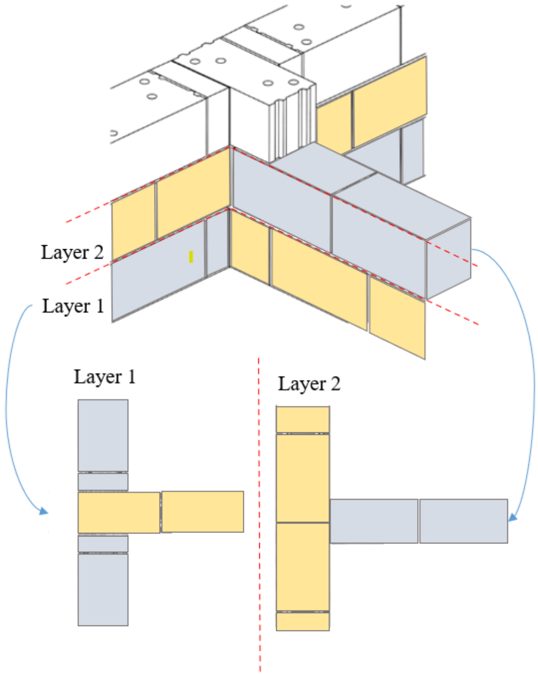

Figure 1. Illustrates the layer-by-layer changing priority wall status between walls and its relevance on the connection point brick placement adaptation [13]

and automate the data flow between high-level BIM geometry and robot-interpretable execution information. Unlike BIM, which primarily describes design intent, constructable element properties, or abstract spatial representations, FIM explicitly models how a structure will be fabricated and processed by deriving fine-grained fabrication data, toolpaths, material-deposition coordinates, sequencing, and machine-control parameters. A key step in FIM's geometry processing is component slicing, where wall or fabrication elements are processed to ordered layers that expose intermediate contours for planning and manipulation [4], enabling FIM to reference fabrication-relevant geometry at multiple levels of granularity. The framework further enables designers to interact with intermediate fabrication representations such as slice contours and path curves, forming a foundational stage for integrating feedback, optimisation, and robotic parameter control functionalities [8]. While representing fabrication and process information via slicing, the Industry Foundation Classes (IFC) schema enables aggregation, allowing multiple geometric representations to co-exist simultaneously [4]. Aggregation capability supports both automation, real-time monitoring and recalibration mechanisms such as RTDE (Real-Time Data Exchange)[14], which integrate material behaviour, nozzle pressure, movement dynamics, and environmental sensor feedback to enable closed-loop process control during extrusion-based Additive Manufacturing [15]. The proposed framework addresses the research gap by approaching the FIM framework from a BIM-to-FIM perspective and extending FIM's established methodology and IFC-based information base for robotic bricklaying [15].

Despite these advances, robotic brick wall assembly still presents multi-scale challenges, particularly because assembly plans often lack the detailed fabrication and execution information required for robotic implementation, especially at geometrically complex conditions such as wall junctions.

2.3 FIM-based robotic execution

Recent work on construction robotics has increasingly explored methods for linking BIM-based representations with robotic task and motion planning. Initial system-oriented studies demonstrated that IFC models can be converted into robot-compatible planning environments, enabling motion planning, localization, and simulation-based task definitions [16]. Parallel graph-based approaches further showed that BIM's semantic and geometric information can be structured in machine-queryable formats to support robotic navigation and information processing [17]. While these efforts contribute important translation mechanisms, they primarily operate at the project or system level, where the focus remains on robot control, simulation environments, or data conversion pipelines, rather than component-scale fabrication. To improve component awareness, smart-component-oriented workflows segment IFC building entities into metadata-rich representations that carry individual states and robot-communication interfaces, enabling automated IFC–SDF conversion and robotic simulation execution [18]. Similarly, RDF-based middle-layer models organize BIM entities into semantic knowledge graphs that support ontology-based querying and planning of robotic tasks [19]. Although these methods improve data standardization and component-level communication, they do not yet provide a fabrication-focused structure that models building elements according to their manufacturing process to be executed in a single manufacturing process.

Overall, existing studies demonstrate multiple methodologies for establishing BIM–ROS bridges across different scales of robotic construction task planning and execution. However, current studies partly address the automation of generating robot-interpretable task data from design entities. This leaves robotic brick-wall assembly in need of a framework that can handle process planning at the individual brick component level, while resolving discrete process-planning challenges of brick walls.

To address these challenges, we developed a FIM framework that is structured as a pipeline that relies on the IFC data structure to manage and utilize fabrication data for robotic brick wall assembly. Within this pipeline, planning utilizes geometric and semantic fabrication information derived from individual wall elements, aggregating them into a structure-level FIM for the composite brick wall. Fabrication information is represented using a Labeled Property Graph (LPG) in the FIM Information Graph, which is suitable for representing IFC data structures while enabling efficient retrieval and updates of task data. The planning loop maps back the fabrication task states and geometric parameters, enabling component-level task validation, adaptation, and feedback exchange with robotic control. The framework is validated in a simulation environment, followed by robotic execution in a real-world experiment, confirming its applicability beyond digital planning.

3 Methodology

The workflow presented in this work builds on the research of M. Slepicka and A. Borrmann [15], which addressed the transformation of high-level design information into detailed, robot-interpretable data for additive manufacturing. In this context, continuous material deposition in additive manufacturing can be conceptualized as an “infinite brick,” whereas the notion of a “finite brick” represents its discretized counterpart, i.e., robotic bricklaying. To support ”finite brick”, the FIM data structure is extended to represent individual bricks and their corresponding tasks for robotized assembly, as illustrated in Fig. 2.

Similar to 3D printing, a bricklaying fabrication model can be defined by discretizing the component into individual layers, where each layer is described by the brick placement surface (slice) – usually a plane – and the path along which the bricks are placed (toolpath). For ”finite brick”, the toolpath (`IfcCurve`) is additionally discretized into a list of individual points (`IfcPointOnCurve`), each representing the center of placement for a brick. For regularly shaped bricks, slices are aligned with the top surface of each brick, toolpaths follow their longitudinal centerline, and the discretized points are positioned at the center of their top surface. Thus, compared with additive manufacturing, an additional discretization step is introduced to provide discrete brick placement positions rather than a continuous toolpath.

For data storage, the IFC schema supports multiple IfcShapeRepresentations definitions per layer (cf. Section 2.2). This approach enables the systematic organization of fabrication information into multiple levels of detail, enhancing the model's adaptability and supporting efficient data retrieval. To fully describe the placement of a cuboid (brick) in three-dimensional space, both position and orientation are required. Rather than defining these

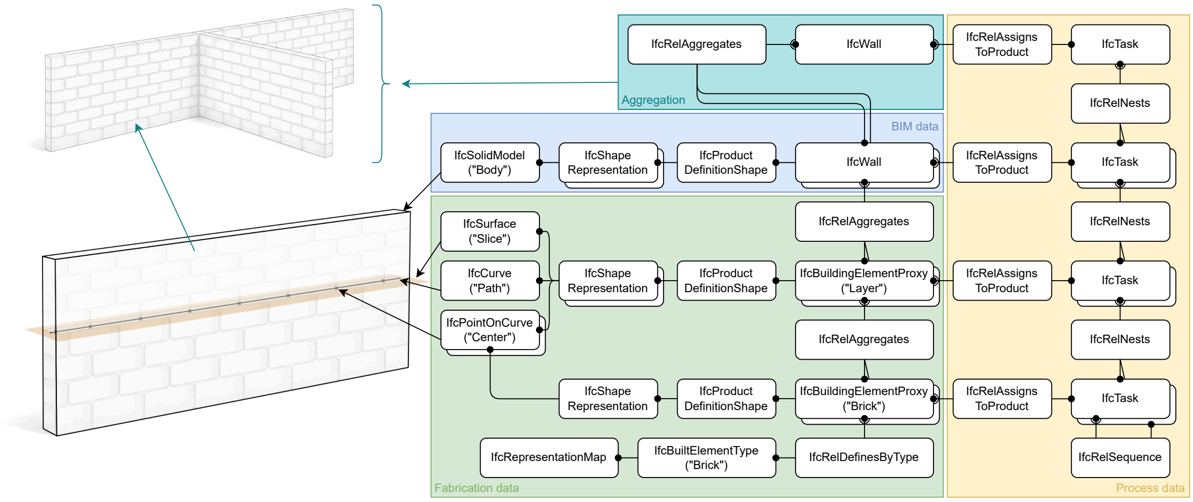

Figure 2. FIM data structure for describing brick walls composed of multiple segments. In the data structure, geometric (left), semantic (middle), and process information (right) are represented.

explicitly, they are captured parametrically through three representations for each layer: the slicing surface (slice) provides the normal orientation, the path provides the lateral orientation, and the point list provides the coordinates. At the same time, this parametric formulation establishes clear rules for discretization from volume to slice, from slice to path, and ultimately to the list of points. For each point, a semantic entity that defines the individual bricks is modeled using `IfcBuildingElementProxy`.

Overall, as defined in [15], a FIM model describes a contiguous component and its corresponding process. For brick walls, we define a contiguous component as the aggregation of wall segments for which all bricks can be placed without violating any placement condition (i.e., each brick must be fully supported). For example, a T-junction of two wall segments built with interlocking bricks at the joint can be regarded as a single contiguous wall. That means, the BIM model is subdivided into multiple FIM models, each representing one contiguous component.

For each FIM model, a new aggregated wall entity is generated that groups all BIM elements belonging to the same fabrication process (see Fig. 2, top). The adaptive planning algorithm identifies how individual wall entities are connected via the relationship `IfcRelConnectsElements` (see Ifc 4.3.2 Documentation [20, Section 5.4.3.48]). With this relationship, it is possible to define the connection type of adjacent wall segments. If the relationship is not present in the model, the adjacent wall segments are regarded as separate components.

In this study, the positional planning algorithm follows an iterative approach. After extracting the relevant BIM data, each wall segment is discretized into individual brick positions, as described in the preceding sections. Brick positions located near junctions are then modified to interlock adjacent segments in an alternating pattern, as illustrated in Fig. 1. Once all brick positions are defined, the bricklaying sequence is determined; here, a straightforward layer-by-layer strategy is adopted. More advanced sequencing algorithms could be applied in future work to optimize construction speed and minimize robot motion.

Aligned with the overall component breakdown structure (wall, wall segment, layer, and individual brick), the process can be modeled in the same hierarchical way. Although the semantic elements can implicitly suggest a task sequence, FIM describes the manufacturing process explicitly using `IfcTask`. As illustrated in Fig. 2 (right), each modeled brick is assigned an individual task representing its placement action. These tasks are organized hierarchically, with brick-level tasks nested within layer tasks, which in turn are nested within component-level tasks and higher-level groupings. In addition, individual tasks are connected through sequencing relationships (`IfcRelSequence`). These relationships can be planned and modeled in a separate planning stage to explicitly define the placement order.

To enable FIM-based robotic bricklaying in ROS, the individual placement tasks must be translated into executable robot instructions. Since each task is linked to the corresponding "Brick" IFC entity, the target pose can be retrieved directly from the FIM model; combined with the known source pose of the next brick, this allows a trajectory to be generated using a standard motion planner. To support visualization and simulation within the ROS environment, SDF files are additionally required to represent the brick entities (cf. Section 2.3). In the FIM approach, however, SDF models are not generated through a one-to-one BIM-to-SDF conversion for every individual brick, but rather once for each brick type occurring in the model. For subsequent use, the corresponding SDF file can also be linked directly within the FIM model by means of an external reference (`IfcExternalReference`).

4 Implementation

This section presents a proof-of-concept implementation. In the implemented pipeline, BIM data is accessed using the open-source library IfcOpenShell, geometric modeling is performed with PythonOCC, and the resulting fabrication information is stored in a Neo4j graph database via Cypher queries. The implementation's validity is subsequently evaluated using a small-scale robotic setup comprising a UR10e robotic arm equipped with a Robotiq 2F-85 two-finger gripper, controlled via ROS2.

BIM-to-FIM pipeline initialization. As a starting point for the iterative pipeline, a BIM model must be provided. Using IfcOpenShell, the model is first filtered for brick walls. All brick walls that are connected via an `IfcRelConnects` relation are then grouped and aggregated. Then, for each aggregated wall, a separate FIM model is created by setting up a Neo4j database and populating it with all relevant IFC data for that wall (cf. Fig. 2, "Aggregation" and "BIM data"). As a subsequent step, the geometric information of all wall segments is derived to enable slicing operations analogous to the 3D printing methodology described in Section 2.

Segment-wise wall discretization. Using PythonOCC, the volumetric model – typically represented as a BRep or an extruded surface – is subdivided into horizontal slices, referred to as "fabrication layers", with a spacing defined by the brick height. For each layer, a centerline is computed as an alignment curve along which brick positions can be planned. These alignment curves are then discretized into a set of points representing the positions of individual bricks. The discretization is performed according to the selected brick-laying pattern, for example, the pattern shown in Fig. 1, in which every second layer is shifted by half a brick length along the centerline to create an interlocking arrangement. The spacing between adjacent brick positions is determined by the brick length plus a 1 cm clearance space. In this study, the brick dimensions were set to 12 × 4.8 × 3.6 cm. In addition, bricklaying was simplified to dry stacking to focus on the pick-and-place workflow, so no additional allowance for mortar application was required. In this planning stage, the discretization is performed naively, placing only full bricks even when they exceed segment boundaries, providing a straightforward discretization of all individual wall segments.

Brick placement. In accordance with the discretization, semantic entities are instantiated for each layer and for each discretized point using the `IfcBuildingElementProxy` class. The layer entities are designated as `IfcFabricationLayer`, while the brick entities are labeled `IfcBrick`. For all instances, properties can be defined to capture the most relevant characteristics, such as identifiers or static parameters. The geometric representations created in the previous steps are attached to the semantic entities as illustrated in Fig. 2. And, to describe the individual bricks efficiently, type elements (`IfcBuiltElementType`) are instantiated for each type of brick including a representation map that defines the brick geometry.

Rule-based adaptations of brick positions and shapes. In a second iteration, all brick positions located near a segment boundary are reexamined. Two cases are distinguished: either the brick is located near a junction, indicated by an `IfcRelConnects` relationship, or it lies close to the boundary (end) of the wall segment. If the position is near the wall end, it is evaluated whether the corresponding brick extends beyond the wall's boundary representation. When this occurs and an adjustment of the brick length is required, the brick entity is reassigned to a different type entity with suitable dimensions, creating a new type entity if necessary. If the resulting brick length falls below a predefined threshold, a minimum length is enforced and the adjacent brick is shortened accordingly. This adaptation is constrained by the requirement that the modified arrangement must still satisfy the interlocking brick pattern, thereby ensuring proper interlocking across layers. For each updated brick, the position of the corresponding center point is then recomputed parametrically and updated in the model.

In the alternative case, where a brick location is close to a junction, the brick is repositioned so as to interlock with the adjacent wall segment (see Fig. 1, bottom), while the length of the neighboring brick within the same wall segment is adjusted accordingly. Simultaneously, the affected bricks in the connected wall segment are modified to accommodate the interlocking brick. As in the previous case, shape-related changes are handled through the associated type entities, whereas positional changes are reflected in the point list. Across layers, this interlocking is alternated between the involved wall segments. Within the connected wall segment, bricks that overlap with the interlocking brick are identified and trimmed as required. If a brick

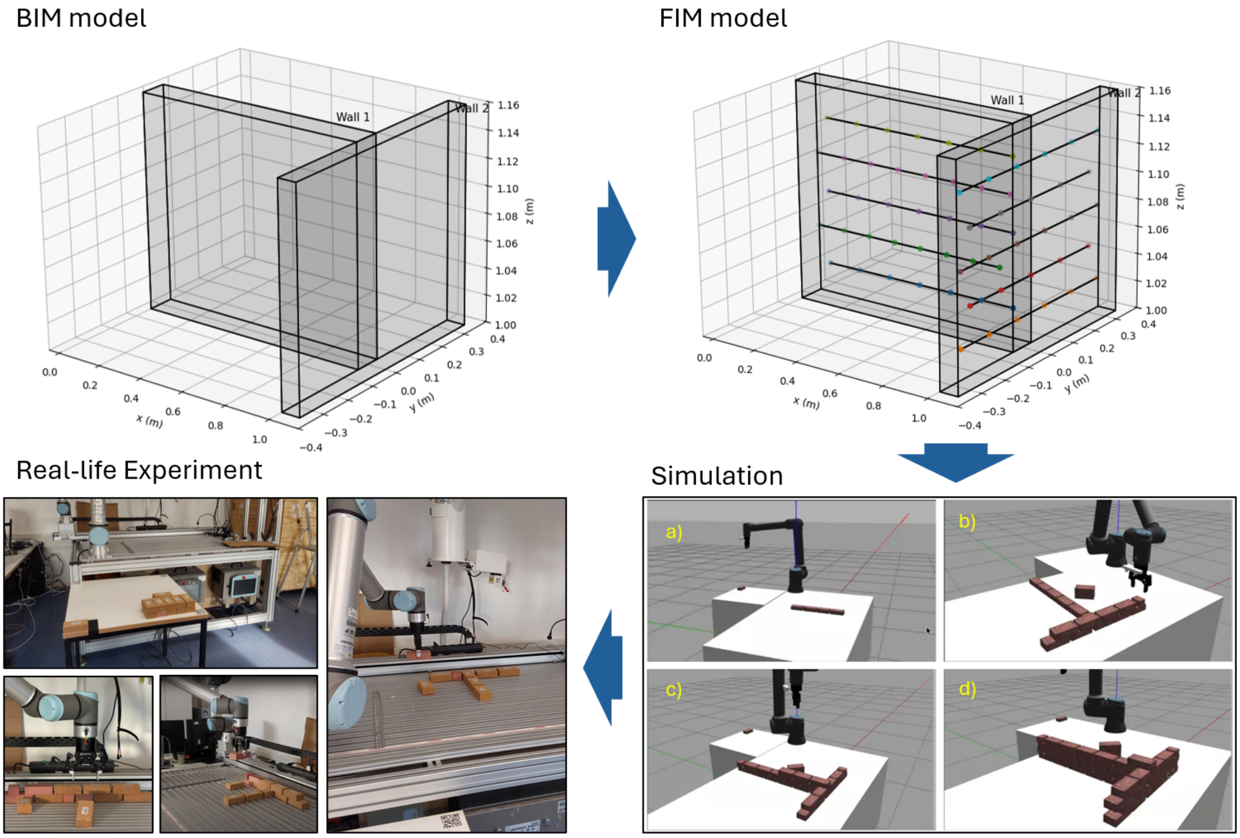

Figure 3. BIM model (top left); FIM model (top right); (a) Sequencing starts with priority wall, (b-c) Priority wall changes with layer change, (d) Successful execution in simulation (bottom right); and real-world execution using the UR10e robot (bottom left).

is split during this process, an additional discretization point and a corresponding `IfcBuildingElementProxy` are created. All resulting modifications are propagated back to the model by updating center point positions and brick dimensions accordingly.

Task planning. As an additional information layer built on top of the brick positions, the bricklaying sequence can be represented within FIM. To achieve this, each ”IfcBrick” instance is associated with an `IfcTask` instance that abstractly describes the corresponding brick placement action (see Fig. 2). Consecutive placement actions are then linked using `IfcRelSequence`. In this study, sequencing is performed in a straightforward, layer-by-layer manner.

Each `IfcTask` instance represents the action of placing a single brick, includes attributes describing the task's execution state, and is linked to an `IfcTaskTime` instance that stores scheduling information. The task state can take the values *NOTSTARTED*, *STARTED*, or *COMPLETED*. The associated task time specifies the planned execution time and enables tracking of the actual execution duration.

FIM-based Robot Execution. Once the BIM-to-FIM pipeline is complete, the FIM model is ready for execution. Robot-interpretable commands are then generated sequentially for each modeled task. Each task is linked to an ”IfcBrick” instance, from which the brick's position and orientation are retrieved via a simple Cypher query and transformed into a robot target pose. By additionally specifying a source position and orientation for the brick, motion paths can be generated that guide the robot to the pickup location (source pose) and from there to the drop location (target pose). Finally, gripper actions are defined to fully describe the pick-and-place action.

The streamed robotic commands consist of Tool Center Point (TCP) orientations, Cartesian movements, and gripper actions. The generated task sequences are recorded in JSON or YAML and transmitted via TCP/IP to ROS2 as granular brick action commands for execution. By streaming task data per brick instead of exporting a single static command file, the robot controller remains continuously connected and synchronized with the component-level database during execution.

As an extra layer of safety, the whole process can be simulated first in a physics-based simulator, e.g., Gazebo, to detect potential issues, such as collisions or reachability problems. The simulation replicates the real robotic laboratory setup. After the simulation, the developed FIM-based fabrication can then be executed in real life using the robotic arm system.

In this study, the experiment was constrained to the reachable workspace of a single UR10e robot, with each wall discretized into a 6×6 brick grid. A T-junction case consisting of two connected walls was evaluated over 6 layers. T-junction handling resulted in 14 of 72 bricks being replanned as shown in Figure 3. After the simulation was executed without collisions, the process was successfully executed in real life using a UR10e robot arm equipped with a Robotiq 2-finger gripper 2F-85. The results of both simulation and real execution are illustrated in Fig. 3.

5 Results

The developed pipeline is responsible for the complete workflow, from geometric and semantic processing of BIM wall entities to producing the composite structure's FIM for robotic execution. Therefore, in this section, the end-to-end applicability of the developed FIM pipeline is demonstrated in the following steps: *i) Fabrication-relevant geometric and semantic information processing and mapping. ii) Deploying the adaptive planning loop. iii) Simulation execution. iv) Real-life robotic execution.*

6 Conclusion and Future Work

In this paper, a methodology for automating the robotic brick wall assembly task is presented as a BIM-to-FIM task and motion planning pipeline. The framework extends fabrication information modeling by enriching IFC-based wall geometry and semantic relationships to generate robot-interpretable process data, while leveraging this semantic information to enable rule-based adaptations to process-planning complexities, such as wall junctions. The previously introduced slicing approach is further extended to discrete Cartesian brick-placement tasks, and an adaptive planning method is proposed to maintain consistency. A streaming interface synchronizes the robot controller with FIM, ensuring that execution reflects the current fabrication state while preserving prior adaptations. The proposed approach was validated in the same simulation and real-life experiment with the UR10e setup, demonstrating that the pipeline bridges the gap between design and robotic execution in an integrated, automated workflow. However, the current evaluation should be interpreted as a proof of concept, as it was conducted in a constrained setup involving a single robot workspace, a 6*6 brick grid, and a six-layer T-junction configuration. Accordingly, the present results primarily demonstrate functional feasibility and workflow integration rather than quantitative performance in terms of execution time, planning efficiency, or geometric placement deviation.

The proposed approach demonstrates that brick-level process adaptation and junction handling can be automated directly by leveraging BIM semantics with an FIM-based planning process, while presenting the framework as a fabrication-centered, component-aware planning pipeline.

Overall, this work demonstrates that Fabrication Information Modeling can serve as a synchronized intermediate layer for robotic brick wall assembly, supporting automated sequencing and geometry-based task adaptation within a unified model. While highlighting FIM's task-agnostic and adaptable, the developed methodology opens the door to more comprehensive scenarios, such as multiple junctions and geometric complexities.

Future work will therefore extend the framework to larger and more diverse wall assemblies, including openings, corners, and multiple junction configurations, in order to assess its scalability under more realistic construction conditions. Furthermore, it will be extended by integrating process feedback mechanisms, such as sensor output and camera data, to enable continuous, process-aware, adaptive planning. In addition, future studies will incorporate quantitative evaluation metrics, including planning time, execution time, task success rate, and deviation between planned and executed brick placement. The current layer-by-layer sequencing strategy will also be extended toward optimization-based sequencing methods to improve process efficiency and reduce unnecessary robot motion.

Contribution statement

Martin Slepicka: Conceptualization, Writing, Reviewing, Supervision. **Canberk Yalçıklı**: Implementation, Writing. **Mohammad Reza Kolani**: Conceptualization, Writing, Reviewing, Supervision. **Stavros Nousias**: Reviewing, Supervision. **André Borrmann**: Writing, Reviewing.

Acknowledgements

The presented research was funded by the TUM Innovation Network "CoConstruct" and the German Research Foundation (DFG) in the frame of the Research Unit FOR 5672, grant number 517965147.

References

[1] Juan Manuel Davila Delgado, Lukumon Oyedele, Anuoluwapo Ajayi, Lukman Akanbi, Olugbenga

Akinade, Muhammad Bilal, and Hakeem Owolabi. Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. *Journal of building engineering*, 26:100868, 2019.

[2] Z. Dakhli and Z. Lafhaj. Robotic mechanical design for brick-laying automation. *Cogent Engineering*, 4(1):1361600, 2017. doi:10.1080/23311916.2017.1361600.

[3] A. Fingrut, K. Crolla, and D. Lau. Automation complexity - brick by brick. *Proceedings of the International Conference on Computer-Aided Architectural Design Research in Asia*, January 2019. doi:10.52842/conf.caadria.2019.1.093.

[4] M. Slepicka, S. Vilgertshofer, and A. Borrmann. Fabrication information modeling: Closing the gap between building information modeling and digital fabrication. In *Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC)*, 2021. doi:10.22260/isarc2021/0004.

[5] M. Poljansek. Building information modelling (BIM) standardization (EUR 28977 EN). Technical report, Publications Office of the European Union, 2017.

[6] F. Craveiro, J. P. Duarte, H. Bartolo, and P. J. Bartolo. Additive manufacturing as an enabling technology for digital construction: A perspective on construction 4.0. *Automation in Construction*, 103:251–267, 2019. doi:10.1016/j.autcon.2019.03.011.

[7] M. Brinkhoff and M. R. Kolani. Deriving fabrication information from BIM for automated robotic task planning. In *Proc. of Forum Bauinformatik 2024*, Sep 2024. doi:https://doi.org/10.15480/882.13486.

[8] M. Slepicka, S. Vilgertshofer, and A. Borrmann. Fabrication information modeling: Interfacing building information modeling with digital fabrication. *Construction Robotics*, 6(2):87–99, 2022. doi:10.1007/s41693-022-00075-2.

[9] K. Dörfler, T. Sandy, M. Giftthaler, F. Gramazio, M. Kohler, and J. Buchli. *Mobile Robotic Brickwork*, pages 204–217. Springer International Publishing, Cham, 2016. ISBN 978-3-319-26378-6. doi:10.1007/978-3-319-26378-6_15.

[10] T. Bruckmann and R. Boumann. Simulation and optimization of automated masonry construction using cable robots. *Advanced Engineering Informatics*, 50:101388, August 2021. doi:10.1016/j.aei.2021.101388.

[11] Y. Zhou, B. Huang, B. Dong, Y. Wen, and M. Duan. Dynamic robotic bricklaying force-position control considering mortar dynamics for enhanced consistency. *Automation in Construction*, 174:106090, 2025. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2025.106090.

[12] Mohammad Reza Kolani, Stavros Nousias, and André Borrmann. Coordinating robotized construction using advanced robotic simulation: The case of collaborative brick wall assembly. *arXiv preprint arXiv:2405.17171*, 2024. doi:10.35869/proceedings_egice2024.

[13] KLB Klimaleichtblock GmbH. Walls and wall connections. `https://en.klb-klimaleichtblock.de/walls-and-wall-connections.html`, 2026.

[14] M. Slepicka, J. Helou, and A. Borrmann. Real-time data exchange (rtde) robot control integration for fabrication information modeling. In *Proceedings of the 40th ISARC*, Jul 2023. doi:https://doi.org/10.22260/ISARC2023/0016.

[15] M. Slepicka and A. Borrmann. Fabrication information modeling for closed-loop design and quality improvement in additive manufacturing for construction. *Automation in Construction*, 168:105792, 2024. doi:10.1016/j.autcon.2024.105792.

[16] S. Kim, M. Peavy, P.-C. Huang, and K. Kim. Development of bim-integrated construction robot task planning and simulation system. *Automation in Construction*, 127:103720, April 2021. doi:10.1016/j.autcon.2021.103720.

[17] P. Pauwels, R. De Koning, B. Hendrikx, and E. Torta. Live semantic data from building digital twins for robot navigation: Overview of data-transfer methods. *Advanced Engineering Informatics*, 56:101959, 2023. doi:10.1016/j.aei.2023.101959.

[18] A. Zhu, P. Pauwels, and B. De Vries. Component-based robot-prefabricated construction simulation using ifc-based building information models. *Automation in Construction*, 152:104899, 2023. doi:10.1016/j.autcon.2023.104899.

[19] X. Wang, S. Wang, C. C. Menassa, V. R. Kamat, and W. McGee. Automatic high-level motion sequencing methods for enabling multi-tasking construction robots. *Automation in Construction*, 155:105071, 2023. doi:10.1016/j.autcon.2023.105071.

[20] buildingSMART International. Ifc 4.3.2 documentation, 2024. URL `https://ifc43-docs.standards.buildingsmart.org`.