BIM-Integrated Generation of Safety-Aware, Georeferenced Occupancy Grid Maps for Autonomous Masonry Robots

Masoud Shakoorianfard[1], Jan Deubner[1], Christian Richter[1], and Frank Will[1]

[1]Chair of Construction Machinery, TUD Dresden University of Technology, Germany
[masoud.shakoorianfard, jan.deubner, christian.richter1, frank.will]@tu-dresden.de

Abstract –

Bricklaying faces significant labor shortages, for which Mobile Construction Robots (MCRs) offer a promising solution. However, the absence of many building elements and hazards in brickwork environments, such as floor and exterior wall openings, which may be undetectable to robot sensors, poses safety challenges to the navigation system. This paper presents an automated workflow to integrate IFC-based Building Information Modeling (BIM) information into the Robot Operating System 2 (ROS 2) navigation stack (Nav2) by generating compatible 2D Occupancy Grid Maps (OGMs) and safety-zone masks. To do so, navigation-relevant geometric, semantic, and georeferencing information is extracted from the IFC model and stored in dedicated data structures. Then, detectable and undetectable obstacles are classified based on the robot's mechanical structure and sensor configuration, including the traversability threshold and the 2D LiDAR scan plane height. This results in the generation of (1) a static main map containing detectable obstacles and (2) a keep-out mask encoding undetectable hazards, both of which are rasterized automatically and integrated into Nav2, where keep-out regions are enforced as costmap filters for global path planning and local path tracking. Simulation results demonstrate that this approach enhances operational safety in brickwork construction environments by increasing the safety success rate and eliminating intrusions into the safety zones.

Keywords –

Occupancy Grid Map; Safety Zone; Mobile Construction Robot; Navigation; BIM; IFC

1 Introduction

Among various roles in the construction industry, the position of bricklayer is one of the most challenging to fill, mainly due to labor shortages. In Germany, for instance, the number of trainees in bricklaying has fallen by approximately 82% over the last 28 years [1]. The repetitive, physically demanding nature of bricklaying makes bricklayers more susceptible to early retirement due to reduced fitness, resulting in many unfilled jobs.

Although robots have been proven effective in various industries and are increasingly replacing human labor, they are limitedly employed in the construction industry [2]. Among various robotic systems, Mobile Construction Robots (MCRs), consisting of robotic arms and mobile platforms, are a promising option for automating on-site brickwork. In contrast to stationary robotic systems, the mobility capability of MCRs removes limitations on building size. However, the unstructured and dynamic nature of the environment of construction sites requires advanced navigation and control solutions to be integrated into these robotic systems [3]. To address this, MCRs are generally equipped with Simultaneous Localization and Mapping (SLAM) algorithms for navigation, using on-board sensors to build a map and to localize within it. On a brickwork construction site, SLAM-based mapping can be severely degraded by unstructured, irregular layouts, temporary occlusions, and the absence of elements that have yet to be built. Moreover, typical two-dimensional Light Detection and Ranging (2D LiDAR)-based maps do not reveal unsafe "empty" spaces such as open doorways on exterior walls or unrestricted floor openings. Although these areas are clear to the sensors, they pose safety challenges for a heavy MCR. Therefore, addressing such undetectable hazards is essential for safe autonomous navigation.

Building Information Modeling (BIM) provides a rich source of prior knowledge for this purpose. BIM is defined as "the shared digital representation of the physical and functional characteristics of any construction works" [4]. Inherently modeled as an object-oriented system, BIM describes building elements with geometry, semantics, and, in many cases, georeferencing information. Recent research has begun to utilize BIM for robotics purposes, such as high-level task planning [5], path planning [6], or 2D Occupancy Grid Map (OGM) generation [7]. However, the systematic use of BIM to (1) distinguish between obstacles that are detectable by the robot sensors and undetectable hazards, (2) generate dedicated safety zones that can be integrated into the robot navigation system, and (3) leverage the georeferencing information to keep the navigation map aligned with a global site frame, has received limited attention so far.

The present study aims to bridge this gap by integrating information provided by BIM models into the navigation system of a masonry MCR to facilitate safe autonomous navigation during the brickwork phase. The brickwork is planned to be realized using Autoclaved Aerated Concrete (AAC) blocks. The MCR comprises an industrial robotic arm and an omnidirectional mobile platform equipped with 2D LiDAR sensors. The ROS 2 Navigation Stack, Nav2, powers the robot navigation system. The open standard and vendor-neutral Industry Foundation Classes (IFC)-based BIM models are adopted to provide three-dimensional (3D) geometric and semantic information corresponding with various building elements, such as walls, openings, and staircases.

This work introduces the following key contributions:

- **BIM-integrated map generation:** An automatic pipeline that converts IFC-based BIM models of brickwork construction sites into Nav2-compatible 2D OGMs, using direct extraction of geometric and semantic information from the IFC-based BIM model instead of site mapping or image-based map generation.
- **Safety-aware modelling of undetectable obstacles:** A classification of building elements into free spaces, detectable obstacles, and undetectable obstacles for a masonry MCR equipped with 2D LiDAR sensors, and generation of corresponding safety zones that are integrated as keep-out regions into Nav2 costmaps.
- **Georeferenced navigation maps:** The extraction of high-quality georeferencing information from the BIM model and anchoring the generated OGMs in a global frame accordingly, thereby facilitating the future Global Navigation Satellite System (GNSS)-based localization of the MCR.
- **Building-scale evaluation:** The demonstration of the approach in a building-scale BIM-driven simulation and the analysis of navigation behavior in unsafe scenarios, comparing maps with and without BIM-derived safety zones.

The organization of this paper is as follows: Section 2 provides an overview of related work regarding BIM-integrated navigation systems in construction robotics. Section 3 elaborates upon the presented workflow and its implementation. Section 4 gives results and evaluations, discusses findings, and outlines limitations. Finally, the work is concluded in Section 5 with an outlook.

2 Related Work

The core idea of deriving occupancy representations for robotic navigation directly from BIM models has been investigated in several studies, an overview of which is presented in this section.

A representative graph-based method is presented in [6], where semantic, geometric, and topological information contained in the BIM model is utilized to generate obstacle-aware navigation graphs for 2D or 3D navigation, depending on the agent profile. However, the resulting graphs are not directly compatible with the occupancy grid and costmap representations in Nav2.

A grid-based method is proposed in [8] and [9], where indoor environments are discretized into planar grids for path planning. The approach is further extended to multi-storey buildings by modeling vertical transition elements, such as staircases and elevators. In [8], various hazard distributions in the vicinity of obstacles are applied on top of the generated grids. Although both studies efficiently use BIM semantics, the agent profile, including sensor configuration, is not considered in the map generation algorithm. Consequently, the resulting maps are more applicable to human agents than to heavy construction robots.

Another widely used solution renders selected BIM elements into a color-coded raster image of the environment [10], [11]. In this approach, irrelevant BIM model entities are first filtered out, and a geometry library is adopted to visualize the remaining elements as a 3D color-coded model. A horizontal section plane, whose height is determined by the storey height and robot sensor configuration, is applied to this model, and a top-down raster image is generated by adjusting the virtual camera. The resulting image is then converted to the Nav2-supported map format using image processing techniques. Although intuitive, this method treats all visually empty regions as free space, and it neither considers obstacles undetectable by robot sensors (including floor openings) nor utilizes BIM semantic information to distinguish between detectable and undetectable hazards, making it unsuitable for safety-critical brickwork applications.

Beyond these BIM-based navigation and map-generation approaches, several frameworks integrate BIM into mobile robots and 3D mapping. The BIM-SLAM framework [12] combines BIM models with multi-session 3D LiDAR SLAM to support long-term indoor mapping and to detect and reconstruct elements not present in the original BIM model. In [13], a BIM-enabled digital twin framework is proposed that combines BIM, autonomous robots, 3D mobile mapping, IoT sensing, and indoor positioning for real-time indoor environment monitoring. Both mentioned studies demonstrate the potential of integrating BIM into robotic mapping for lifecycle applications such as long-term localization and monitoring. However, they neither target the construction phase of masonry projects nor generate safety-aware 2D OGMs for Nav2 with explicit modeling of undetectable hazards, such as floor openings and open exterior doorways.

Most of the applications and case studies discussed in the above research correspond to the operational phase of construction projects. In contrast, this research targets the construction phase, specifically, robotic masonry. In this context, missing elements such as doors, windows, and staircases, combined with the high accuracy requirement in bricklaying, demand precise, safety-aware navigation maps that explicitly model undetectable hazards.

3 Development of the BIM-Based Occupancy Grid Map Generator

This section presents the automated BIM-integrated OGM generation workflow that links an IFC-based BIM model to the Nav2-powered navigation system of the masonry MCR. To realize the workflow, navigation-relevant geometric, semantic, and, when available, georeferencing information is extracted from the BIM model and stored in dedicated JavaScript Object Notation (JSON) data structures. The JSON file is the input to the OGM creator software tool, which generates a static OGM and the corresponding safety-zone mask. Finally, these maps are loaded into Nav2 and utilized for localization, path planning, and enforcing safety constraints in the vicinity of undetectable hazards.

3.1 Information Exporting from the IFC-based BIM model

An IFC-based BIM model typically contains a wide variety of entities that digitally represent various aspects of a construction project. Depending on the application, however, only a subset of these entities is relevant. This requires navigating through the model and filtering out the irrelevant instances. The IfcOpenShell library is used to access the model and extract relevant attributes and property sets. For the brickwork application, spatial and building elements relevant to navigation and safety, such as walls, openings, and transport elements, are investigated. Assuming consistency, the extracted information is stored in dedicated data structures, used for OGM and safety-zone mask generation (Figure 1).

The developed information exporter tool traverses the IfcBuildingStorey structure and processes contained IfcProduct elements relevant to navigation. In IFC models, object poses are typically defined by nested IfcLocalPlacement entities. For each placement, the translation P is read from Location, and the orientation R is reconstructed from Axis (local Z) and RefDirection (X') attributes. An orthonormal basis is obtained via cross products ($Y = Z \times X'$, $X = Y \times Z$) and stored as a homogeneous transform $T = [R\ P;\ 0\ 1]$. By recursively chaining placements along the containment hierarchy, all relevant element poses are expressed in the selected storey frame and stored in the dedicated data structures.

In addition to local placements, the georeferencing information contained in the IFC model is investigated. According to the Level of Georeferencing (LoGeoRef) concept [14], multiple georeferencing sources and levels may be present in a single model, each with varying quality. In this work, only high-quality georeferencing at LoGeoRef40 and LoGeoRef50 is considered. At LoGeoRef40, the project origin and orientation are obtained from the IfcGeometricRepresentationContext entity via the WorldCoordinateSystem and the TrueNorth attributes, respectively. At LoGeoRef50, the transformation parameters and the target Coordinate Reference System (CRS) are derived from the IfcMapConversion entity attributes, namely Eastings, Northings, OrthogonalHeight, XAxisAbscissa,

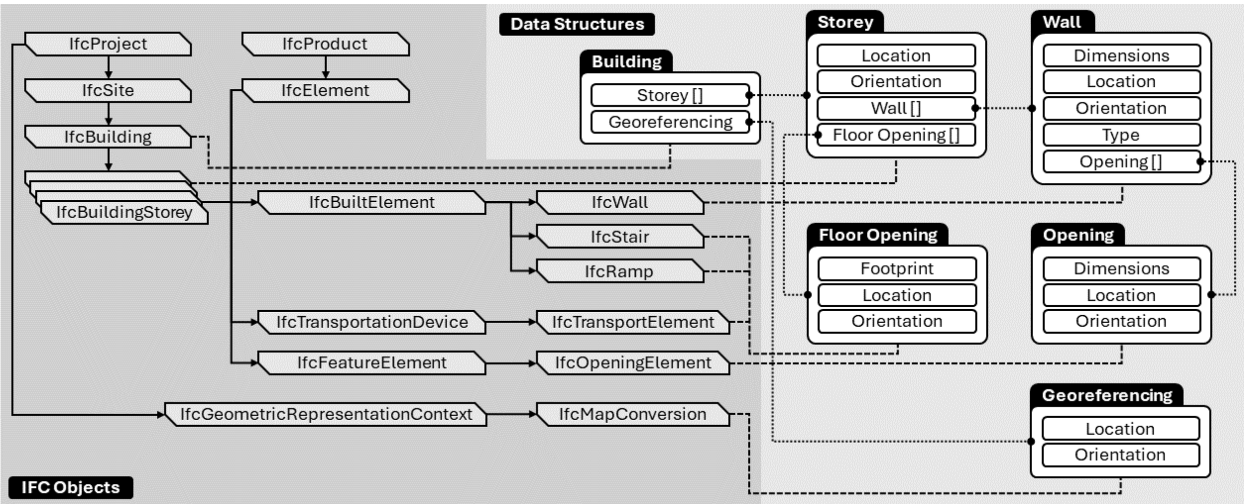

Figure 1. Correspondences between the defined data structures and IFC model entities according to IFC 4.3.

XAxisOrdinate, and TargetCRS. If multiple levels are available, priority is given to the highest LoGeoRef. When such information is available, the origin and yaw are stored in a dedicated data structure that can be used to anchor the OGMs to a georeferenced project frame.

The current implementation assumes straight walls. The dimensions of IfcWall elements representing straight walls can be calculated based on the IfcExtrudedAreaSolid entity, accessed through the Representation attribute of the IfcWall entity. The IfcExtrudedAreaSolid provides three main attributes; the ExtrudedDirection attribute defines the direction of extrusion in the wall local coordinate system. The Depth attribute gives the extrusion distance, and the SweptArea attribute specifies a 2D rectangular extrusion profile. Hence, the Dimensions data field in the data structures can be populated based on the extrusion distance and profile dimensions, with the extrusion direction defining the correspondences between the local axes and the wall length, thickness, and height. The same procedure is also applied to the IfcOpeningElement. The Pset_WallCommon property set is used to access the IsExternal property, which indicates whether a wall is exterior and allows distinguishing hazards near exterior openings from those inside the building.

As to the floor openings, IfcStair, IfcRamp, and IfcTransportElement objects are investigated. A single IfcStair object represents a stair assembly aggregating stair flights and landings, represented by IfcStairFlight and IfcSlab elements, respectively. The footprint of each stair flight is exported from the Representation attribute as a closed polyline on the ground plane, typically rectangular in shape. The footprint is interpreted as a floor opening, provided that the building storey and stair flight element share a common ground plane. The same procedure is also applied to the IfcRamp and IfcRampFlight entities. Regarding the IfcTransportElement object, the footprint is derived from the IfcBoundingBox geometry extracted from the Representation attribute. This is later used to identify vertical shafts as undetectable hazards for generating safety-zone masks.

3.2 Occupancy Grid Map Generation

Nav2 supports static 2D OGMs as grayscale raster images, typically in Portable Gray Map (PGM) format, paired with a YAML file that provides map metadata, including resolution and origin. The raster image encodes a top-down representation of the environment, in which free space is represented by white pixels, occupied space by black pixels, and intermediate gray values indicate unknown regions.

The JSON file containing the exported IFC model information is used as the input to the OGM generator. For a selected building storey, all contained elements are iterated and assigned one of the three following labels: “Free” for navigable space, “DO” for detectable obstacle, or “UDO” for undetectable obstacle. Detectability is primarily determined based on the 2D LiDAR scan plane at height h_s; elements intersecting the scan plane are labeled as “DO,” while hazards not observable at h_s (e.g., floor openings and exterior wall openings) can be treated as “UDO.” The geometric and semantic decision criteria utilized for label assignment are summarized in Table 1.

In Table 1, h_{trav} is the maximum obstacle height that can be traversed by the robot. h_s denotes the height of the 2D LiDAR scan plane on the mobile platform and is assumed to satisfy $h_s > h_{trav}$. h_{cls} is the minimum vertical clearance required by the MCR, including the mobile platform and the robotic arm, to pass through an opening. h_w is the wall height, and z_o, h_o are the vertical position and height of a wall opening, respectively. For wall openings, the semantic criterion IsExternal is taken from the host wall Pset_WallCommon property set. The Floor Opening object type includes footprints of stairs, ramps, and transport elements that are reduced to footprint polygons. The proposed labeling logic is therefore robot-profile-dependent by design.

In the second step, a grayscale image is generated using OpenCV and initialized as free space. The image size and the map origin are determined from the 2D extents of the selected storey geometry and the chosen map resolution. All element footprints are projected onto the storey XY-plane and converted to pixel coordinates. Wall objects with the label “DO” are rasterized as black-

Table 1. Geometric and semantic criteria in the decision-making step for obstacle label assignment.

Object Type	Geometric Criteria	Semantic Criteria	Label
Wall	$h_w \leq h_{trav}$	$IsExternal == FALSE$	Free
	$h_w \leq h_{trav}$	$IsExternal == TRUE$	UDO
	$h_w \geq h_s$	$-$	DO
	$h_{trav} < h_w < h_s$	$-$	UDO
Wall Opening	$z_o \leq h_{trav} \wedge (z_o + h_o) > h_{cls}$	$IsExternal == FALSE$	Free
	$z_o \leq h_{trav} \wedge (z_o + h_o) > h_{cls}$	$IsExternal == TRUE$	UDO
	$z_o \leq h_{trav} \wedge (z_o + h_o) \leq h_{cls}$	$-$	UDO
	$z_o \geq h_s$	$-$	DO
	$h_{trav} < z_o < h_s$	$-$	UDO
Floor Opening	$-$	$-$	UDO

filled polygons. Subsequently, wall opening elements labeled “Free” and “UDO” are rasterized as white-filled polygons to reflect the free space observable at the 2D LiDAR scan height h_s. The resulting pixel matrix is then exported as a PGM image paired with a YAML file containing the map metadata. The generated OGM, referred to as the main map, represents the navigable space and static obstacles detectable by the robot sensors.

In the last step, a safety-zone mask is generated using the same image extents, origin, and resolution as the main map. A grayscale image is initialized as free space, and all objects labeled “UDO” are rasterized as black-filled polygons, including external low walls, non-traversable walls below the 2D LiDAR scan plane, undetectable wall openings, and floor opening footprints. This results in a pixel matrix which can be exported as a PGM image and a YAML file. The generated mask complements the main map by enforcing safety constraints through marking undetectable hazards as keep-out regions.

3.3 Integration into ROS 2 Navigation Stack

As the last step, the generated OGMs are integrated into the Nav2. To do so, the map_server ROS 2 node is used to load the main map and to provide it to the Nav2 via topic and service interfaces. The global and local costmap servers subscribe to this static map and to robot sensor data, including 2D LiDAR sensors, to generate costmaps representing obstacle and hazard distributions. The local costmap is typically configured as a rolling window around the robot and is updated at a higher rate, primarily from real-time sensor readings, while the global costmap mainly provides global context. The global planner utilizes the global costmap to compute a collision-free path from the current pose to a goal pose. At the same time, the controller uses the local costmap to generate motion commands for path tracking. Robot pose estimation in the static map is performed using Adaptive Monte Carlo Localization (AMCL) by matching LiDAR observations to features in the static main map.

Nav2 offers costmap filter plugins to augment costmaps with additional, map-annotated constraints. Regarding the integration of safety-zone masks, the Keepout Filter is enabled in both the global and local costmaps. By applying the masks to the local costmap, it is ensured that keep-out constraints are enforced for hazards that are undetectable by the 2D LiDAR sensors. The filter consumes (1) a CostmapFilterInfo message provided on the costmap_filter_info topic specifying the filter type and the mask topic name, and (2) a filter mask provided as an OccupancyGrid message on the configured mask topic. Based on this information, safety zones are marked in the costmaps without modifying the static main map, paving the way for the global planner and the controller to utilize the filtered costmaps to avoid keep-out regions during path planning and path tracking.

4 Results and Discussion

4.1 Simulation Setup and Map Generation

The proposed workflow is evaluated through experiments conducted in the Gazebo simulation environment. The simulation environment consists of the masonry MCR, simulated 2D LiDAR sensors (20 m range), and a building model. The masonry MCR comprises an omnidirectional mobile base, storage space, and an industrial robotic arm (Figure 2). The mobile platform allows traversal over small obstacles up to a maximum height of h_{trav} =10 mm. For environment sensing, two 270° 2D LiDAR sensors are mounted diagonally on the robot body at a height of h_s=300 mm, providing full surrounding coverage. The Gazebo LiDAR sensor plugin is utilized to generate simulated scan data, and Nav2 powers the robot navigation system.

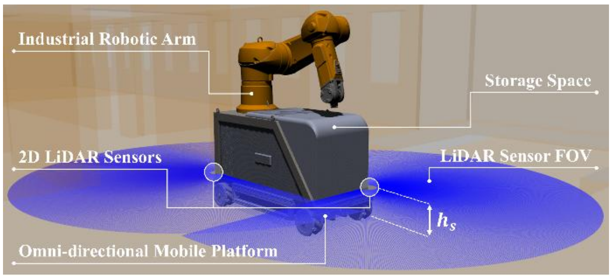

Figure 2. The 3D model of the masonry MCR.

As building input, the “Office Building” IFC-based BIM model [15] made by the Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT), is adopted. The model contains five storeys, and the ground floor is selected for the simulation. The IfcConvert tool (IfcOpenShell) is utilized to filter irrelevant elements and convert the remaining geometry into a 3D model that can be imported into Gazebo. In parallel, geometric and semantic information is exported from the corresponding IfcBuildingStorey and used as input to the OGM generation pipeline. Based on the exported data and the labeling rules, a static main OGM and a safety-zone mask are generated automatically (Figure 3), each paired with YAML metadata. The generated main map and mask are exported at a resolution of 0.016 m/pixel with origin (x, y, yaw) = (-0.5 m, -0.5 m, 0.0 rad) in the map frame. The Nav2 default parameter values are used for the remaining map parameters.

To reflect a sensor-driven local navigation setting, the local costmap is configured as a rolling window updated from LiDAR observations. Safety-zone constraints are enforced through the generated mask, which is applied to both global and local costmaps in order to ensure that hazards undetectable at the LiDAR scan height are treated as keep-out regions during planning and tracking.

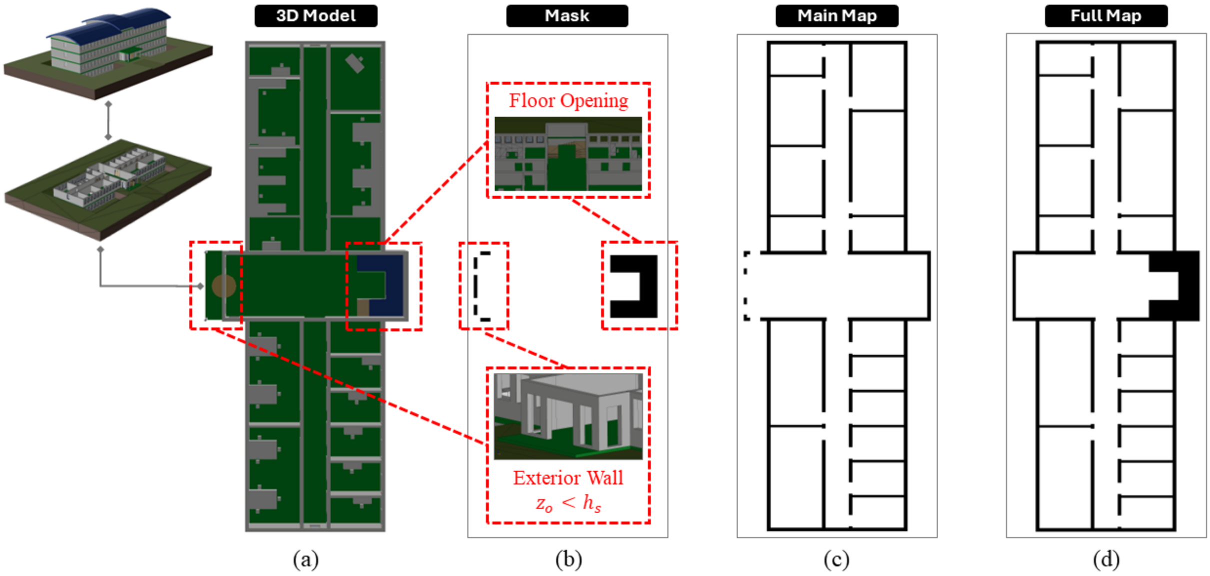

Figure 3. (a) The 3D geometric representation of the ground floor of the "Office Building" IFC-based BIM model, (b) The automatically generated safety-zone mask OGM containing keep-out regions, (c) The automatically generated static main OGM, (d) The combination of main and mask maps as a unified full OGM.

4.2 Experimental Protocol and Results

As to the evaluation of the effectiveness of the safety-zone mask, a randomized set of 10 goal poses is automatically generated at the safety-zone boundaries, each of which navigates the robot toward either a floor opening or an exterior wall opening to intentionally stress the navigation system near undetectable hazards. Each experiment is executed from an identical initial pose, and the same set of goal poses is used for both following test conditions:

- **Condition 1 (Full map):** a single static OGM is used, in which all hazards are encoded as occupied regions, and no mask is applied (Figure 3-d).
- **Condition 2 (Main map + mask):** The static main OGM and the safety-zone mask are generated based on the obstacle labels. The mask is applied as a keep-out filter to both global and local costmaps.

For each experiment and condition, three outcome measures are recorded:

1. Goal reached: Determined based on the Nav2 goal status and the final pose tolerance.
2. Safety success: Defined as no overlap between the robot footprint and any safety-zone polygon at the final pose.
3. Overlap: Defined as the intersection area between the robot footprint and the safety-zone polygon at the final pose, divided by the footprint area.

Table 2 summarizes the outcomes across the simulation experiments.

Table 2. Navigation and safety outcomes across 10 simulation experiments.

Condition	Goal reached (%)	Safety success (%)	Mean overlap (%)	Std. overlap (%)
1	50	0	15.17	7.24
2	0	100	0	0

The results show that the robot reached 50% of the goals in Condition 1; however, none of the experiments were safety-successful, and the robot footprint overlapped the safety-zone polygon with a mean value of 15.17% (std. 7.24%). In Condition 2, applying the BIM-derived keep-out mask eliminated safety-zone intrusions, as evidenced by a 100% safety success and 0% overlap (std. 0%). None of the goal poses are reached in Condition 2, indicating that the poses sampled at the keep-out boundary became infeasible, as expected, under the enforced safety constraints. Figure 4 shows two representative experiments from the randomized goal pose set, illustrating a safety-zone intrusion in Condition 1 and successful keep-out enforcement in Condition 2.

4.3 Discussion

The results in Table 2 indicate that enforcing safety zones through the keep-out mask improves safety-aware navigation in brickwork environments. In Condition 1, hazards are encoded only in the static map. However, the local costmap is configured as a rolling, sensor-driven representation, primarily updated from 2D LiDAR observations. Consequently, hazards not observable at

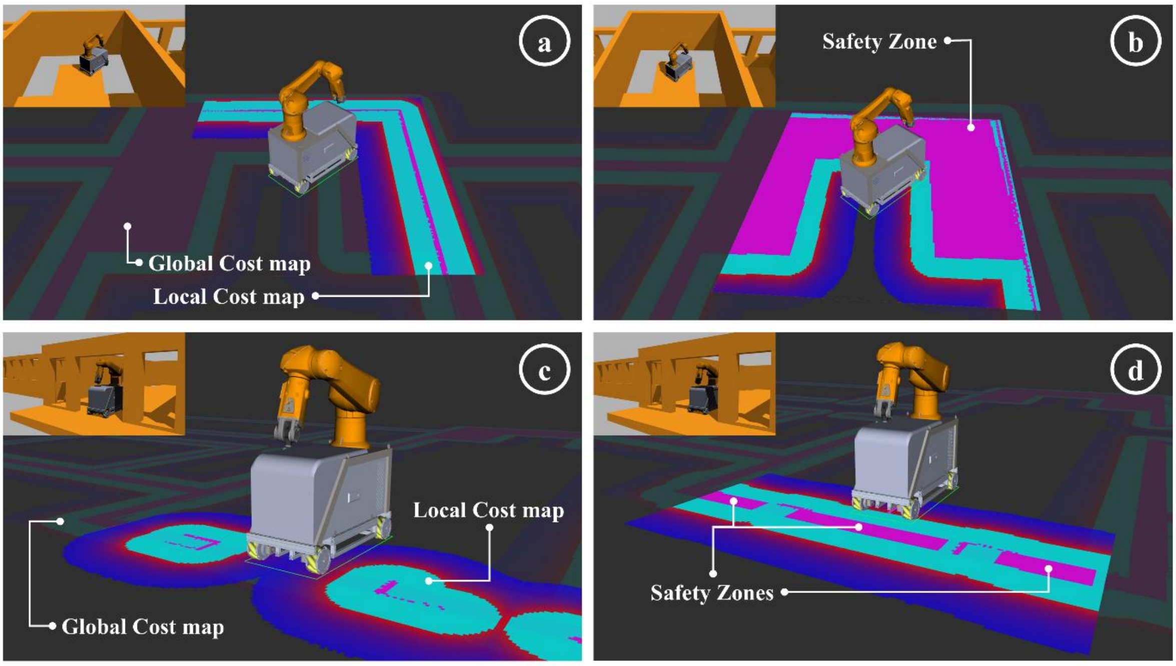

Figure 4. The final pose of the robot in two representative experiments in RViz2 and Gazebo, (a) Experiment I with Condition 1, (b) Experiment I with Condition 2, (c) Experiment II with Condition 1, (d) Experiment II with Condition 2.

the 2D LiDAR scan height, including floor openings and exterior wall openings, may not be reflected in the local costmap, and, therefore, may not be avoided reliably during local path tracking by the controller. In Condition 2, the safety-zone mask is applied as a keep-out constraint to both global and local costmaps. This ensures that undetectable hazards are consistently represented as keep-out regions during both global path planning and local path tracking, resulting in improved safety success and reduced overlap with safety zones. As the goal poses were intentionally sampled at safety-zone boundaries, enforcing keep-out constraints resulted in the infeasibility of these goals. In such cases, the controller stops outside the safety zone without satisfying the goal tolerance, emphasizing the effectiveness of the applied safety-zone mask. In real masonry deployment, navigation goals are provided by a task-level planner from the planned block placement sequence and are filtered by the MCR reachability and keep-out constraints. Hence, the goal poses are selected in the free space outside safety zones and are expected to remain feasible.

Furthermore, separating the main map, which is consistent with 2D LiDAR scans, from the safety-zone mask enables modular integration. The main map remains suitable for scan-to-map localization and standard obstacle avoidance, while safety constraints can be updated by regenerating or modifying the mask without changing the main map. This is particularly applicable to construction-phase environments, where missing elements and temporary conditions can create hazards undetectable by robot sensors.

This evaluation is conducted in simulation and, consequently, does not capture all sources of sensing noise, actuation uncertainty, and site dynamics present in real deployments. Moreover, the outcomes depend on the sensor configuration and selected threshold parameters (e.g., 2D LiDAR scan height and traversability), which must be adapted according to the MCR hardware.

5 Conclusions and Future Work

This paper presented an automated workflow to integrate IFC-based BIM information into the navigation system of a masonry MCR powered by the ROS 2 navigation stack. Navigation-relevant geometric, semantic, and georeferencing information is exported from the BIM model and utilized to generate safety-aware georeferenced Nav2-compatible 2D OGMs. The presented method distinguishes between the obstacles detectable by robot sensors and undetectable hazards, such as floor and exterior wall openings, based on the 2D LiDAR scan plane height and robot-specific threshold parameters. As a result, two OGMs are automatically generated: (1) a static main map that remains consistent with 2D LiDAR scans and supports localization and standard obstacle avoidance, and (2) a safety-zone mask that encodes undetectable hazards as keep-out regions and is applied to Nav2 costmaps through filtering.

Simulation results demonstrated, under the tested edge-case conditions, that applying the BIM-derived safety-zone mask in both global and local costmaps improves safety-aware navigation near undetectable hazards compared to a single, full map baseline by eliminating safety-zone intrusions.

Although developed for robotic masonry, the workflow can be applied to other construction-phase MCR applications that require safe navigation in partially completed buildings, such as robotic spraying or layout marking for electrical installations. However, further development is necessary to validate the workflow on a real MCR and to assess its robustness under more realistic site conditions beyond the current simulation-based proof-of-concept. In addition, the dependency on sensor configuration and threshold parameters will be investigated across various robot setups. From the modeling perspective, the current implementation assumes straight wall and stair geometries, which is planned to be extended to support more complex geometries. Finally, improving the pipeline robustness to incomplete or inconsistent IFC data through geometry processing fallbacks and utilizing extracted georeferencing information for GNSS-assisted localization remain promising directions.

Acknowledgements

Co-funded by the European Union

Freistaat SACHSEN

This project is co-financed from tax revenues on the basis of the budget adopted by the Saxon State Parliament.

References

[1] Anzahl der Auszubildenden im Maurerhandwerk bis 2024. Statista. On-line: https://de.statista.com/statistik/daten/studie/206344/umfrage/anzahl-der-auszubildenden-im-maurerhandwerk/, Accessed: 12/03/2025

[2] C. Richter, J. Kortmann, J. Mischke, F. Will, and J. Otto. Status Quo of Construction Robotics: Potentials, Applications and Challenges. In *Construction Logistics, Equipment, and Robotics*. pages 149–156, Raitenhaslach, Germany, 2024.

[3] L. Zeng, S. Guo, J. Wu, and B. Markert. Autonomous mobile construction robots in built environment: A comprehensive review. *Developments in the Built Environment*, 19:100484, 2024.

[4] ISO 29481-1:2016(en). Building information models — Information delivery manual — Part 1: Methodology and format. On-line: https://www.iso.org/obp/ui/en/#iso:std:iso:29481:-1:ed-2:v1:en, Accessed: 12/18/2025.

[5] K. Kim and M. Peavy. BIM-based semantic building world modeling for robot task planning and execution in built environments. *Automation in Construction*, 138:104247, 2022.

[6] A. Hamieh, A. Ben Makhlouf, B. Louhichi, and D. Deneux. A BIM-based method to plan indoor paths. *Automation in Construction*, 113:103120, 2020.

[7] B. Huang *et al.*. Development of BIM Semantic Robot Autonomous Inspection and Simulation System. In *9th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pages 35–40, Shenzhen, China, 2023.

[8] Y.-H. Lin, Y.-S. Liu, G. Gao, X.-G. Han, C.-Y. Lai, and M. Gu. The IFC-based path planning for 3D indoor spaces. *Advanced Engineering Informatics*, 27(2):189–205, 2013.

[9] M. Xu, S. Wei, S. Zlatanova, and R. Zhang. BIM-based indoor path planning considering obstacles. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2-W4:417–423, 2017.

[10] J. Husemann, M. Kunz, M. Suiker, and K. Berns. From BIM to Autonomous Navigation: Using BIM Models to Enable Autonomous Navigation and Localization in Construction Environments. *ISARC Proceedings*, 2025:143–150, 2025.

[11] J. Wang, Q. Lu, H. Xu, and H. Zhu. BIM-to-Robot Mapping: Constructing IFC-Referenced Occupancy Grids and Semantic Metadata for Rule-Based Navigation. In *8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, 780–783, 2025.

[12] M. A. Vega Torres, A. Braun, and A. Borrmann. BIM-SLAM: Integrating BIM Models in Multi-session SLAM for Lifelong Mapping using 3D LiDAR. In *40th International Symposium on Automation and Robotics in Construction*, Chennai, India, 2023.

[13] X. Hu and R. H. Assaad. A BIM-enabled digital twin framework for real-time indoor environment monitoring and visualization by integrating autonomous robotics, LiDAR-based 3D mobile mapping, IoT sensing, and indoor positioning technologies. *Journal of Building Engineering*, 86:108901, 2024.

[14] C. Christian and G. Hendrik. Level of Georeferencing (LoGeoRef) using IFC for BIM. *Journal of Geodesy, Cartography and Cadastre*, 2019.

[15] KIT IFC Examples. IFC Wiki. On-line: https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples, Accessed: 12/18/2025.