

Component-based robot prefabricated construction simulation using IFC-based building information models

Citation for published version (APA):

Zhu, A., Pauwels, P., & de Vries, B. (2023). Component-based robot prefabricated construction simulation using IFC-based building information models. *Automation in Construction*, 152, Article 104899. <https://doi.org/10.1016/j.autcon.2023.104899>

Document license:

CC BY

DOI:

[10.1016/j.autcon.2023.104899](https://doi.org/10.1016/j.autcon.2023.104899)

Document status and date:

Published: 01/08/2023

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

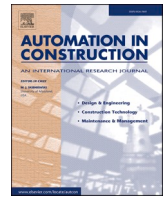
www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Component-based robot prefabricated construction simulation using IFC-based building information models

Aiyu Zhu^{a,*}, Pieter Pauwels^{a,b}, Bauke De Vries^a

^a Department of the Built Environment, Eindhoven University of Technology, the Netherlands

^b Department of Architecture and Urban Planning, Ghent University, Belgium

ARTICLE INFO

Keywords:

Robot construction simulation
BIM
Construction robot
Smart construction

ABSTRACT

Despite the widespread usage of Building Information Modeling (BIM) based simulation in the construction industry, robot construction simulation environments are built on particular modeling tools and can only be simulated with certain robot models. This not only limits construction robot development and also makes it difficult to directly share building information with robots, which impedes the study of robotic construction. To overcome this problem, we propose a new framework for robotic construction simulation which integrates open standard formats and tools to achieve a commonly used robot simulation environment. This provides a new approach to automatically convert IFC to simulation description format (SDF) and generate building components as smart components, which eliminates platform-specific limitations on robotic construction simulation, which further expands the utilization of BIM models.

1. Introduction

Since the 1990s, the productivity of the Architecture, Engineering and Construction (AEC) industry has decreased every year, compared to the growth in the efficiency of the manufacturing industry [1]. A major factor for the growth of efficiency in the manufacturing industry is their development and use of automated robots [2]. Factory-based manufacturing lines allow to upscale fabrication in the manufacturing industry. This finds its way to some parts of the AEC industry, in particular for prefabrication of components in safe and controlled factory environments. However, the majority of the AEC industry has customized production lines that are often labor-intensive, manual, and on-site. Even when components can be prefabricated, a lot of on-site labour and assembly processes need to happen. In other words, the AEC industry is a labor-driven industry with constraints of conventional technology that make it difficult for robotic technologies difficult to intervene, and this is one important factor that stops the AEC industry from increasing its highly needed construction efficiency [3].

Several research and development (R&D) efforts in recent years nevertheless investigate this problem, and look into the application of robotic construction in the AEC industry as one of its most important trends for the future development and industrialization [4]. As Bock and

Linner [5] argued, in the future, throughout the AEC industry, robots will play a prominent role, in prefabricated off-site construction, automated assembly, and specialised on-site operations. In this article and research project, we look particularly into the challenge of *robotic component-oriented, assembly-based on-site construction*. The main hypothesis for this challenge is that components can be prefabricated off-site by robots through a large number of repetitive tasks with high precision [6], and *on-site assembly can then be robotized as well by transferring, positioning, and connecting the required components*. Several attempts have been made to automate robotic construction assembly, including the assembly of wood structures based on robotic arms [7,8]; component transfer using automated tower cranes [9]; automated positioning and installation robots for steel beams [10]; and cable-driven robots for facade installation [11]. These studies demonstrate the functionality of robots to fulfill construction demands in different scenarios.

Current obstacles to robotic construction (assembly) include, but are not limited to: project cost limits, human-machine interaction issues, and compliance with construction regulations [12]. A safe environment in which workers and robots can safely operate together is not easy to guarantee, leading to high costs and challenging regulations. To counter this high risk, it is argued in this work that *a reliable and good construction*

* Corresponding author.

E-mail addresses: a.zhu@tue.nl (A. Zhu), p.pauwels@tue.nl (P. Pauwels).

¹ Faculty of the Built Environment, Eindhoven University of Technology, The Netherlands.

simulation environment is needed, where different scenarios for robot operations in construction projects can be simulated before they are actually put in practice. This lack of a robot construction simulation environment is one of the most significant obstacles to the advancement of robotic construction [13]. The importance of simulation in various industries cannot be overstated, especially when faced with large, complex and potentially costly tasks [14]. For both construction and robots, R&D is expensive, time-consuming, and potentially risky. Realistic simulations can assist us in spotting potential problems before we take actions, thereby enhancing our prediction and planning capabilities and lowering the probability of mission failure by a substantial margin. In the construction field, with the growth of Building Information Modeling (BIM) technology, the construction industry is rapidly digitizing all aspects of design, construction, operation, and maintenance. Thus, there is no lack of simulation platforms for buildings, whether for visualization or process simulation [15]. However, present simulations in the construction industry are not properly connected with robotic simulations, and as stated by Zhang et al. [16], BIM information and robot models need to be better combined with robotic simulations, which is the key target for this research project and article.

Existing tools that incorporate robotic building simulations with BIM have limitations. Those simulation tools are mostly based on developed plug-ins for specific software (such as Rhino, ABB studio) and are used for vendor-specific robots. Because of the diversity of design software, it is difficult to transfer complete BIM information that is not based on an open standard (e.g., IFC), and similarly, it is difficult for specific software for robotics companies to meet the demands for custom robot functionality in R&D for construction. Particularly for present research in robot construction, a common open simulation environment would enable the rapid creation and testing of more construction-oriented robot prototypes, hence achieving the merger of the AEC domain and robotics industries.

The AEC and the robotics industry have their own open standards. In the AEC industry, the Industry Foundation Classes (IFC)² is the open specification for building digital assets advocated by the international community buildingSMART,³ and it has been recognized as the ISO standard for data interaction. IFC is an object-oriented data model, and can be effectively utilized for the description of component-oriented construction processes [17]. As open source robotics software, the robot operation system (ROS)⁴ provides not only a standard robot framework but also a wealth of resources (e.g., robot shape descriptions, common algorithm implementations, physical simulations), an open community, and extensibility for industrial robots (ROS-industrial⁵). With the common resources of the ROS framework, users can personalize their robots with a variety of capabilities. Therefore, a growing number of research and new robotics companies utilize ROS as their R&D platform. Gazebo⁶ is not only compatible with ROS as an open source simulation program for robotics development, but its sophisticated physics engine also provides a platform for realistic simulation. Although both the robotics and AEC industries have well-established open data standards, an open source simulator based on physics engines can extend the efficiency of using BIM models to a more extensive scale (not only for describing geometric information, but also for simulating the physical properties of building components). However, there is no approach that enables the addition of complete IFC-based BIM models to the ROS platform in order to simulate generic robotic construction. It is therefore essential to produce a simulation approach based on open standards and open source software for robot-BIM interaction simulation in order to break the information isolation

between BIM and robot models in simulation.

The main goal of our study is to present a new approach for a universal robotic construction simulation by integrating ROS and BIM in an open simulator environment. The proposed framework can expand the efficiency of using BIM by generating a simulated construction environment with details of building components using IFC-based BIM models. Through a robot construction simulation case study for light steel frame, we present in this paper the entire process from BIM model production to construction task scheduling and construction simulation. Our main contributions are summarized as follows:

- a new approach for component-based robotic construction simulation, which can remedy the limitations of software and robot types in current simulations,
- a pipeline to automatically convert IFC to simulation description format (SDF), which can bridge the gap between BIM models and physics engine-based simulation,
- validation tests for component-based robotic construction with a simulated light steel frame construction scenario.

The article is organized as follows: after this introduction, we provide a literature review of related studies in Section 2. In Section 3, we highlighted the objectives of the study as well as the scope. In Section 4, we present our proposed robotic construction simulation framework. In Section 5, we demonstrate the application of the framework through a case study of the on-site assembly of a light steel structure. In Section 6, we discuss our work and clarify current limitations. In the last Section, we conclude and point out the future work.

2. Literature review

Because this study involves the application of BIM technology, robotics, and robotic simulation in prefabricated construction, the current work will be examined on three aspects in this section: BIM technology integration, robotics in prefabricated construction, and robotic construction simulation.

2.1. BIM technology integration

The Industry Foundation Classes (IFC) is offered by buildingSMART International as an open data model for building information, and so it forms a significant instrument for building information transmission. The IFC standard defines an IFC schema in EXPRESS, XSD, and OWL [18]. It allows to store building information as IFC instance files (SPFF, XML, RDF). The IFC standard is supported by all contemporary BIM tools, yet focusing mostly on the SPFF format. Several other researchers have used the IFC format for exchanging building information [19–21] and significant work has been devoted on linking IFC data with data from other domains [22].

As the current common format for BIM model description, object-oriented design concepts of IFC can be well coupled with the component-oriented construction of prefabricated construction [17,23]. It has inspired studies on component-oriented prefabricated construction related to IFC-based BIM as we show below. In 2015, Costa and Madrazo [24] proposed connecting component lists with IFC-based BIM models through semantic technologies to enable access to component information during the construction process. Liu et al. [25] demonstrated a BIM-based component-centric construction planning approach to optimize the construction process while working with limited resources. To visualize and simulate the construction process, a method integrating BIM model and construction experience is proposed in [26] to address the assembly sequence planning of components in prefabricated construction. To satisfy the precision and complexity of prefabricated construction and achieve uniformity in the data flow from component design to manufacturing, Liu et al. [27] designed an IFC-based framework to enable the delivery and exchange of information

² <https://www.iso.org/standard/70303.html>.

³ <https://www.buildingsmart.org/>.

⁴ <https://www.ros.org/>.

⁵ <https://rosindustrial.org/>.

⁶ <https://gazebo.org>.

for components. Due to its standard and universal qualities, IFC is increasingly being adopted as a common carrier for building information description and communication. In this study, IFC is utilized as the BIM information format with its adaptability.

Further to the above, BIM is increasingly accompanied with Internet of Things (IoT) research related to prefabricated construction. Such research aims to enable digitization and automation with higher efficiency [28]. Although a large number of BIM and IoT-based studies have emerged in recent years, based on the scope of this study, we mainly review the use of BIM and IoT technologies in the component-based prefabricated construction process. The Internet of Things (IoT) technology is introduced into prefabricated construction, and the potential for building information to be combined with embedded devices to allow construction 4.0 is increasing [29]. Consequently, the concept of smart construction objects (SCOs) is proposed to enable realistic components to carry their own information and interaction by embedding hardware (e.g. QR codes), allowing the information of the components to be interacted with and updated in real time throughout the construction process [30]. Based on the concept of SCOs, [31] further presented several scenarios for their use in smart construction, one of which added component traceability based on changes in component status to fulfill the goal of regulating building processes. Subsequently, SCOs were applied experimentally in some construction scenarios. Li [32] developed a simulation of lean construction using BIM, SCOs, and a platform to evaluate it in a partially prefabricated project [33]. In robotic construction, Zhu et al. [17] established the notion of SCOs as smart components and suggested a mechanism to guide robots to cooperate through component's construction needs. As a consequence, in the robot construction simulation, the smart component-based robotic construction method is a good candidate to be used as the construction logic.

2.2. Robotic construction simulation

Although there exists research and tools on simulation environments in the respective fields of construction (e.g. Synchro 4D, Naviswork [34]) and robotics (e.g. Gazebo [35], V-REP [36]), there is not enough research related to simulation of construction robots, in particular navigation and operation. Current studies on construction simulation [37–40] are mainly based on Rhino software and its Grasshopper programming plug-in, as well as robot plug-ins based on the Rhino platform supported by relevant robotics companies (e.g. Robot component by ABB). The ETH dFab team [41] has created a more comprehensive framework to enable robot planning and simulation of the construction site, which is also based on ROS in Grasshopper and Rhino as simulators. Since this simulation environment focuses primarily on robot task planning, construction planning is absent. Meanwhile, some research has focused on BIM-based robotic construction simulation. For example, Kim et al. [23] has developed a ROS-based robotic construction simulation system that allows robots to follow a construction planning to simulate construction by integrating IFC-based BIM models and construction plans (schedule in XML) into Gazebo. This study introduces a novel concept for BIM-based robot building simulation. Our study elaborates on this work and proposes a workflow that includes a complete and general IFC converter, and creates the construction planning using smart components in a more interactive manner (see original ideas in [17]).

3. Objective and scope

As indicated in the introduction and literature review, the lack of a uniform approach for integrating building information with robot information for simulation prohibits the effectiveness of adopting BIM models and the variety of robot models in existing construction simulations. This can be solved by expressing both the BIM models and the robot models relying on an open standard format and unifying the

formats in order to generate a universal simulation environment. Therefore, this study proposes a generalized component-based robot construction simulation method. A prototype framework is implemented to simulate the robot construction process by splitting the component-oriented BIM files in IFC format and converting them into SDF-based smart components (see Section 4 and Fig. 1).

The purpose of this study is to provide a generic framework for robot construction simulation as a way for expanding the potential for varied robotic applications in construction scenarios via a simulation-based strategy. Furthermore, the study proposes an approach that facilitates the participation of more stakeholders, particularly researchers in the construction and robotics industries, in the development and testing of prototype construction robots through simulation.

The scope of this study is restricted to the simulation of prefabricated construction by robots due to the industrialized and component-oriented features of prefabrication is the main construction behavior is transfer and assembly. Therefore, this study mainly focuses on the implementation of the robot on-site prefabricated construction processes, excluding the actual robot grasping, assembly-related technical details. The simulation of the specific construction behavior of the robot is available in other works [42].

4. Framework of IFC-based robot construction simulation

In this section, we present a systematic description of the IFC-based robot construction simulation. We provide a general overview of the framework (Section 4.1), an introduction to the relevant formats (Section 4.2), and then make an overall elaboration of each part of the framework (Section 4.3, Section 4.4, Section 4.5).

4.1. Framework overview

The two core issues for implementing component-oriented robotic construction simulation are: 1. how to realize the interaction between building information and robots and 2. how to connect to a set of control algorithms that can be used for robots to perform construction. To resolve the above two issues, our framework performs primarily two functions: 1. the implementation of BIM data in the robot simulation environment, and 2. the implementation of a ROS-based component-oriented robot construction method. In this framework, as shown in Fig. 1, IFC is to be adopted as the standard format for BIM model input, and the IFC-based Simulation Definition Format (SDF) for the description of BIM models in a robot simulation platform. The SDF format is crucial for robot simulation, as a result, the BIM information essential for the construction will be parsed and converted into SDF (left in Fig. 1). In addition, the smart component-oriented construction planning system is included in the framework as a controller for robot construction (top-right in Fig. 1). The third and last part of the framework is the specific robot description in the Unified Robot Description Format (URDF) which describes the robot(s) used for construction (bottom-right in Fig. 1). Thus, the framework consists of three main parts: (1) extract, transform and load (ETL) procedure of IFC to SDF, (2) the smart component-oriented construction planning system, and (3) ROS-based simulation.

4.2. Formats

In this part, we will briefly discuss the two kinds of data that are of use to our system.

Industry Foundation Classes (IFC). IFC, as a standard format to describe building information models, describes all the information of a building model in a .ifc file. As a STEP-based format [18,43], the IFC file has 2 sections: HEADER and DATA; the HEADER section contains general information about the file itself, the DATA section contains a list of entities, each with an integer index, type name, and a list of parameters, and the set of available entities. The IFC provides powerful semantics for

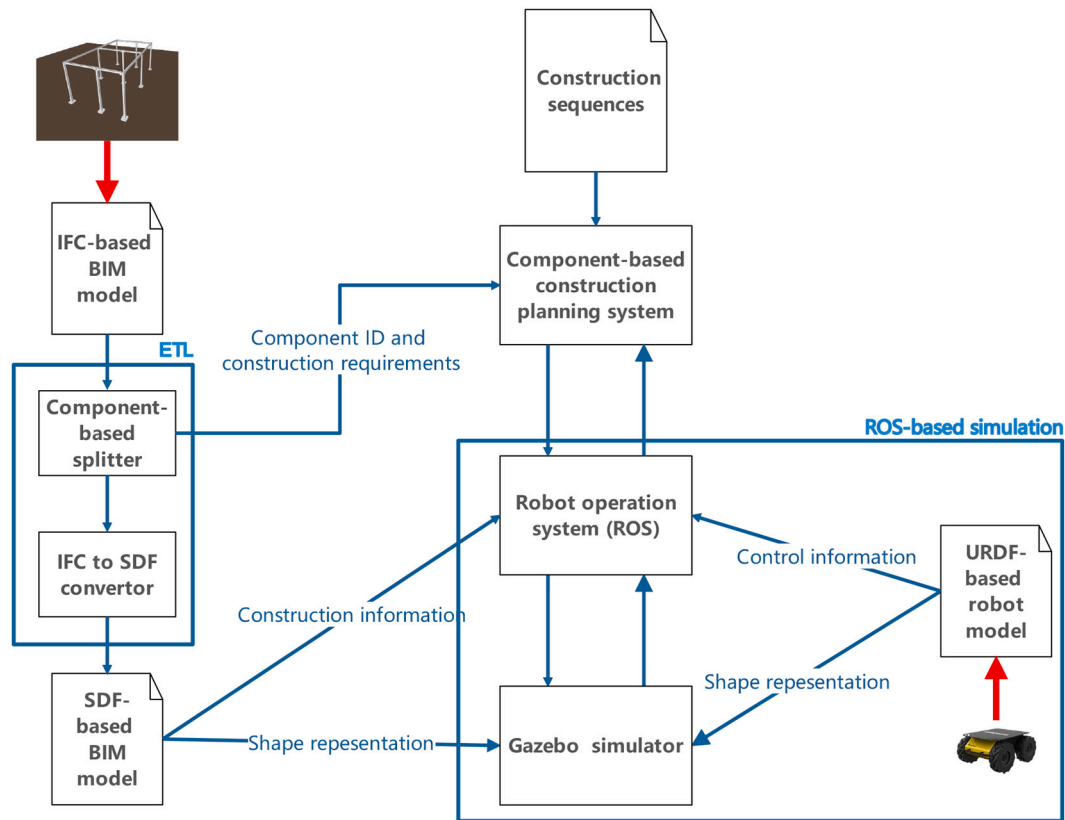


Fig. 1. Framework of robot construction simulation.

describing the AEC domain. Because the object of this study is the independent building components, the IFC entities and their relationships are the focus of our research. The core structure of a standard IFC format file is shown in Fig. 2. A typical IFC file contains: *IfcProject*, *IfcSite*, *IfcBuilding*, and *IfcProduct*. The first three attributes describe the project hierarchy, information about the project, information about the building site, and information about the building on the site, respectively. The *IfcProject* is used to describe the project information, and every IFC file needs to contain the project information. *IfcSite* is used to describe the information of the site where the project is located, including the size and location of the site. *IfcBuilding* is used to describe the overall building information, and the role is to link the complete information of building components to the building. Therefore, an *IfcProduct* has a relative placement within the *IfcBuilding*. The relationship three components are described by an *IfcRelAggregates* entity. After that, all information about the building product is described by *IfcProduct* and linked to its corresponding *IfcBuilding* or *IfcBuildStory* or *IfcSpace* by the property *IfcRelContainedinSpatialStructure*. In *IfcProduct*, the building entities are described by *IfcBuildingElement*, for example: beams (*IfcBeam*), slabs (*IfcSlab*), walls (*IfcWall*), columns (*IfcColumn*) are all subclasses of *IfcBuildingElement*.

Simulation Definition Format (SDF). Simulation Description Format (SDF)⁷ in XML format, as supported by the robot simulator Gazebo,⁸ is commonly used to describe non-robotic objects in a simulation. Thus, SDF can express the objects in a BIM model, but in a robot simulation environment. An SDF-based model simulation environment usually contains two parts: world and model, which are used to realize the simulation for the environment and the objects, and are stored in separate folders on the system. The world is represented by a.world file,

which is mainly used to model physical properties (e.g., light, gravity, wind, etc.) and terrain descriptions throughout the environment. Also all model files need to be declared in the world file and describe their world coordinates. Therefore, the world file can be considered as a description of the site environment (See Fig. 3 (a)). The model is used to simulate the objects, and each object has a single folder that contains three parts: the configuration file, the meshes folder, and the model.sdf file as shown in Fig. 3 (a). The.config file includes the standard information (version, author and model name) as a configuration, the meshes folder stores the 3D geometry and the information of objects is described in the model.sdf file. In the SDF file, as displayed in Fig. 3 (b), the model allows to define the semantics and properties (e.g., name, pose, mass, inertia, etc.), shape representation (DAE or STL supported), and collision parameters. Therefore, SDF can be regarded as an appropriate format for describing construction component information.

4.3. ETL: from IFC to robot world

4.3.1. Handling IFC models

IFC describes the entire building through a unified file. In order to satisfy the SDF format and to realize the component-oriented construction process, the IFC file of a complete building needs to be split into multiple IFCs according to the components, so that each IFC file describes the information of only one component independently. To ensure that the required construction objects are reasonably split up, we follow a 2-step process:

1. Determination of the splitting objects: In the present study, we focus only on the objects that need to be built. In our case, those are the *IfcBuildingElement* objects that need to be split.

⁷ <http://sdformat.org/>.

⁸ <https://gazebo.org/>.

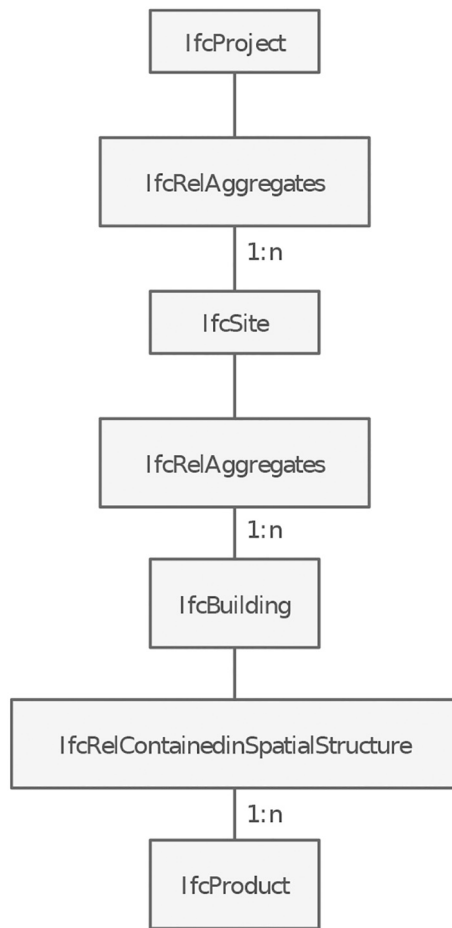


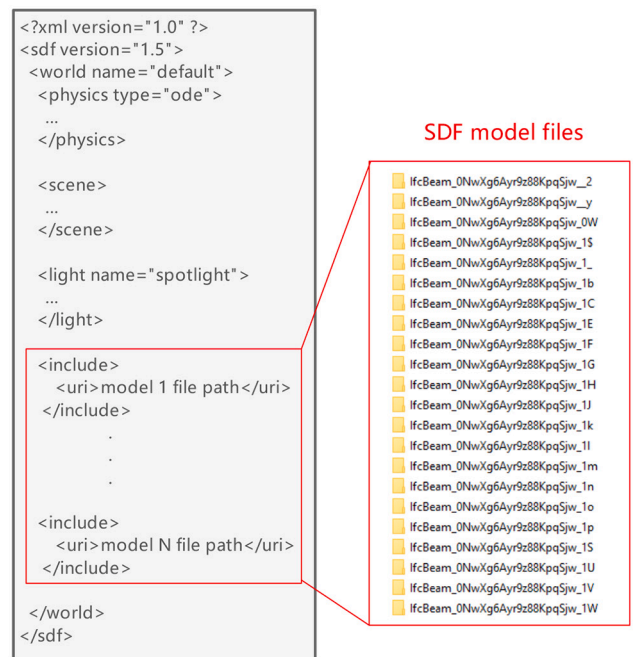
Fig. 2. The structure of IFC.

2. Execution splitting: By declaring the IFC type of an entity, we can get all the information of that entity through *IfcOpenShell*.⁹ For example, if we need to split out all the beams, we can find out all the beams in that model by indexing *IfcBeam* in *IfcOpenShell* and generating a separate IFC file for each *IfcBeam*.

We implemented the above steps into a splitting system for IFC files based on *IfcOpenShell*. This system uses the process as shown in Fig. 4. First, the system will find the required objects (such as a beam, column, etc.) and index the properties corresponding to the objects. After that, it creates a new IFC file for each object, imports the original IFC header file, and the attributes of *IfcProject*, *IfcSite*, and *IfcBuilding* corresponding to the object, and the entity attributes describing the object. After that, it iterates through all the attributes corresponding to the found objects and imports them into the new IFC files corresponding to each object. When all attribute values have been extracted, the splitting process of an object is finished. With this method, we can split the information from any component that we need into a separate IFC file for further usage.

4.3.2. Creating SDF from IFC

For simulated construction, the most important information provided by IFC for one component is: (1) the GUID (used to find the component), (2) the global location in the building (used as the target location for construction), (3) the geometric information (used for collision detection and display), and (4) the properties (used to simulate



(a) World file

(b) Model file

Fig. 3. SDF format.

the real-world physical environment). In order to automatically extract the above information from the IFC files (per object) and generate the SDF files (Fig. 5), we propose a process and developed a Python-based implementation of the converter¹⁰ of IFC to SDF. In Section 4.2, we introduced the basic format of IFC and SDF, so to achieve automated mapping of IFC into SDF, we should first establish the relationship between IFC and SDF data structures. For a BIM model, mainly transforming the shape representation of an object to the SDF format is complex and needs to be described by referencing a graphical format (e. g., DAE or STL). All other information can be described directly in SDF. Therefore, the IFC file to be converted is divided into two parts: semantic information (the GUID, the global location and properties) and geometric information (designed BIM model in DAE or STL format).

As shown in Fig. 3 (b), one SDF model file includes the meta

⁹ <http://ifcopenshell.org>.

¹⁰ https://github.com/ISBE-TUE/ifc_to_sdf.

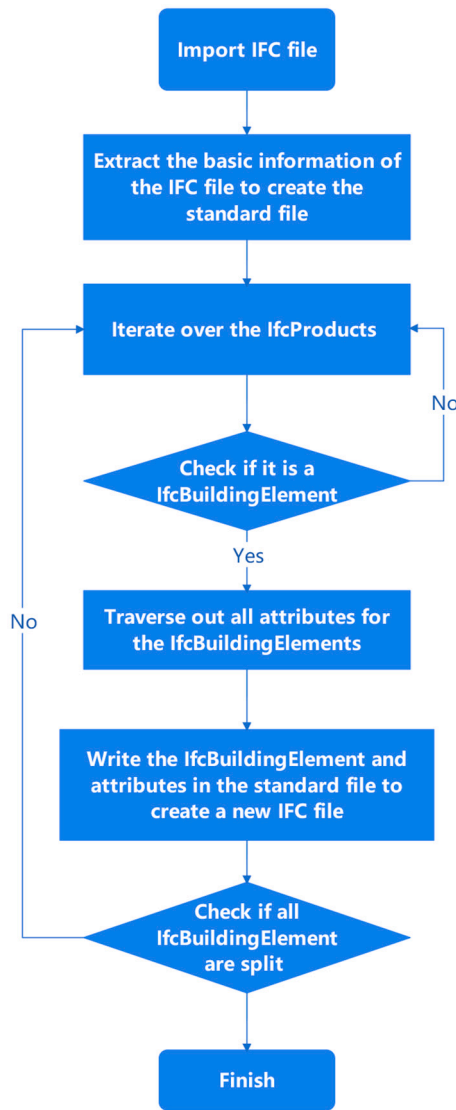


Fig. 4. The process of splitting IFC files.

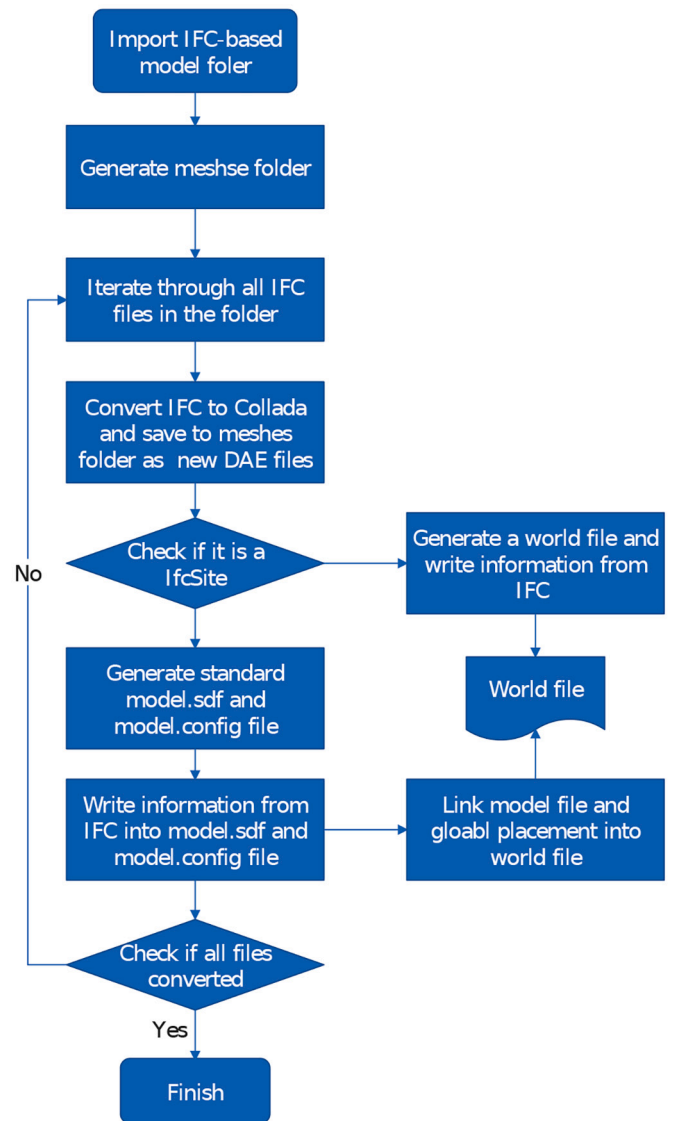


Fig. 5. The process of conversion IFC to SDF.

information (name, placement and static checking) and link information (link's name, properties and geometry). In the SDF model, a model can be made of multiple links (e.g., a robot can be composed of several rigid bodies). Because building components are the basic unit of a building after it has been subdivided, when defining IFC using SDF, each model comprises a single link in our proposed system. Consequently, the GUID information of the entity in IFC can be converted to SDF as both model name and link. This permits the indexing of ROS environment components using GUID. Meanwhile, through `IfcObjectPlacement`, we can get the location information of the component in the building, and therefore map to the pose tag in SDF format. Finally, to make the simulation more realistic to the physical environment, we also need to map the physical property information of the entities in the IFC to the SDF. For example, in Fig. 3 (b), we can extract the weight from `IfcQuantityWeight` and map it to the mass tag in SDF.

To operate in a visualized physical environment, each object needs to define its geometric information for visualization and collision detection (Fig. 3 (b)). In the SDF format, the object can be described with basic geometry (e.g. box, sphere, cylinder), however, SDF format does not support the design of models, so these basic geometries are generally not used to generate complex geometries (e.g., complex surfaces, deformations generated by Boolean operations, combinations of multiple geometries, etc.). In order for the designed 3D model to be used in the

simulation environment, the SDF format supports the import of the designed 3D models through the DAE or STL format with a mesh tag. Therefore, when dealing with the geometric information of IFC, it is necessary to convert the IFC format into a DAE or STL format. With the help of `IfcOpenShell`, we can convert the IFC of each component split into the DAE and store it in the meshes folder only for the corresponding component. Therefore, a DAE model will be generated for each component model. Eventually, the generated DAE model needs to be imported into the SDF model via the mesh tag. With the cases demonstrated in Fig. 6, it shows few examples that the IFC model is converted into SDF files and loaded in Gazebo.

4.4. Smart component-oriented construction planning system

In this section, we describe how to define SDF-based components as smart components and implement the integration of smart component-based construction planning [17] into a ROS-based simulation.

4.4.1. Creating smart components

As a construction-oriented extension of the SCO [30], the smart component gives the component autonomy, awareness, and communication in the construction process, thus realizing the intelligence of the

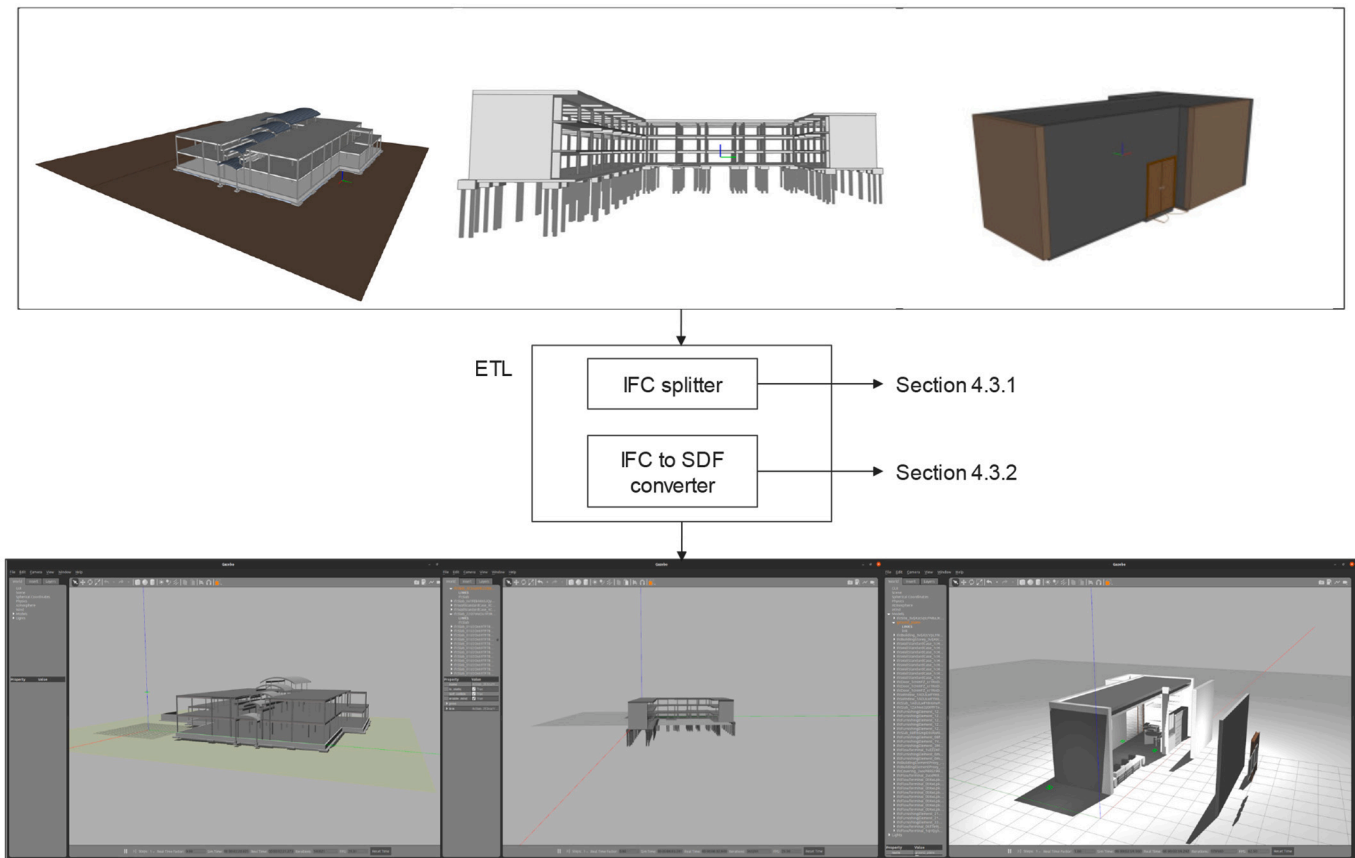


Fig. 6. The example BIM models in Gazebo.

component [17]. The smart component defines the state of the component in different construction processes and the corresponding construction requirements, and gives each component the ability to communicate with the robot so that the component can guide the robot to achieve construction according to its own construction requirements. Consequently, each component has its own state machine to present, record, and update its status. Also, the unique GUID facilitates the interaction between model information and state machine information. For example, when the component is transferred to the target location, the component will return its current coordinates in the simulation environment to determine whether it is consistent with the target location, and if so, the component will return its own GUID to index its corresponding state machine and thus update its state and publish the new state. This allows the component to update its state according to its own spatial changes while sharing it with other objects.

4.4.2. Smart component-based construction planning

Smart component-oriented prefabricated construction on-site is the process of transferring prefabricated components from the yard to the construction location and assembling them, which means that the construction task is completed when all components are built. Thus, the whole construction planning can be treated as two steps: construction sequence and assembly planning. The construction sequence requires the planning of the sequence of all components (work breakdown structure), and the assembly planning involves the planning of the construction tasks of each independent component. According to different construction requirements and construction planning methods (e.g., time priority, cost priority, task priority), different construction procedures can be generated. The creation of the construction sequence is out of scope for this paper, and we only use it as a standard input file (XML). Therefore, only when the current component publishes its status as built, the planning system will release the next component's

construction requirements to the robot and guide it to execute a new round of construction. In assembly planning, based on the concept of smart component [17], the construction task of each component can be mapped to the component state, so that we can use the construction state of the component to guide the robot to execute the corresponding work.

4.5. ROS-based simulation

Using the smart component-based construction planning in Section 4.4.2, here we introduce the method into the ROS-based robot simulation environment¹¹ which contains ROS to provide robot control and Gazebo to provide a physics engine-based visualization simulator. When the SDF-based components are imported into Gazebo, using the interaction mechanism between Gazebo and ROS, we designed a system as demonstrated in Fig. 7 to implement the interaction between the components and the robots as well as the simulation of the construction. When it is the turn of a component to be built, the id and current state of the component will be published to the system, which will assign the required task to the executable robot based on the current state of the component, and then publish the corresponding id of the robot to ROS to control the robot to execute the task. At the same time, ROS will interact with Gazebo in real time to control the robot in the simulated environment. When the robot executes a task in the simulation environment, the system receives feedback and extracts the information of the component in the simulated environment and compares it with the task requirements of the component to check if it is consistent (e.g., when the component is transferred to a target location, the system checks if the position of the component in the simulated environment is consistent with its position in the original BIM model). If it is, the state machine

¹¹ <https://wiki.ros.org/gazebo-ros-pkgs>.

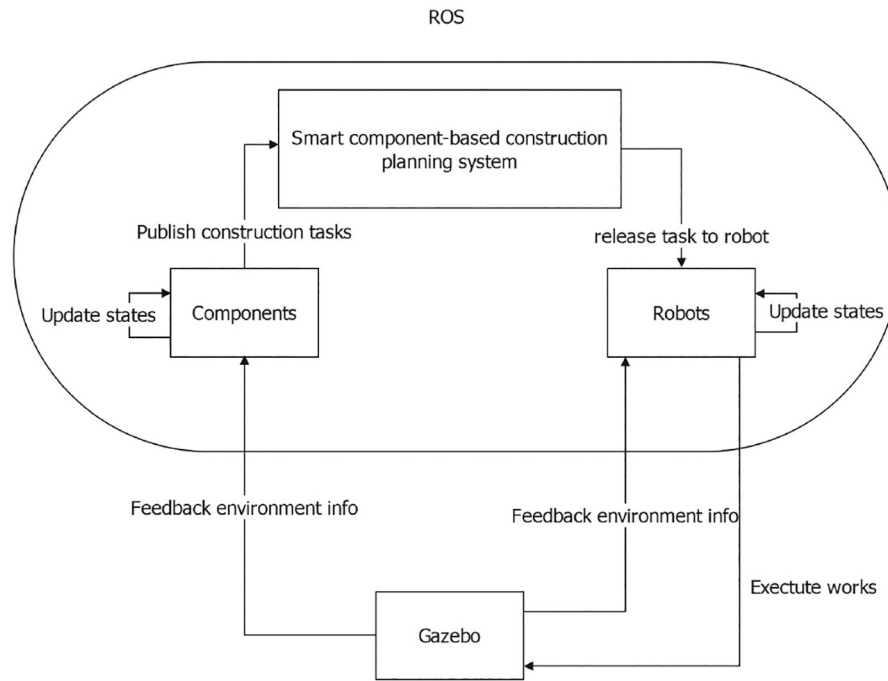


Fig. 7. The system for interaction between components and robots for construction planning system.

corresponding to the component will be updated and the new state will be published to the ROS to guide the robot to execute the new task. If it is not consistent, the robot will repeat the task and return an unsuccessful state for the user to detect the construction. When the robot finishes construction tasks, the component changes status to 'constructed' and the robot status changes to 'idle'. After that, the system will go back to the first step and loop the above procedure until the construction is completed.

5. Case study

In this section, we will demonstrate the specific implementation from model import to construction simulation as described in the framework using a scenario in which a single robot builds a light steel structure. Based on our framework, we need to realize the linkage of the BIM model, construction robot, and construction simulation, thus we will also address these three factors in the case study.

- BIM model creation and processing: we first create a light steel frame in Revit similar to a greenhouse and export it to an IFC4 file. Then, using the approach described in Section 4, we transform the IFC model into an SDF model folder, and ultimately the model can be used in a Gazebo simulation.
- Robot model selection and application: for simulation, we utilize the husky robot,¹² which is recommended by ROS. The husky robot, being a ROS-based robot, has reliable performance in the ROS environment and offers high compatibility and expandability (customizable sensors, robotic arms, and other hardware). In this simulation, the robot has four states: "detect component position", "pick up", "identify component target location" and "assemble". The study of specific methods of execution of the robot's picking up and assembly tasks is not in the scope of this study.
- Smart component-based robot construction simulation: with the communication mechanism of ROS, we realize the interaction

between the components of the light steel frame and the robot, and also design a state machine to realize the construction process.

5.1. BIM model creation and processing

Lightweight steel structures are employed in a variety of small-scale building applications, including sun shelters, bus stops, and functional temporary structures. The individual components are excellent for robotic construction because of their lightweight, small size, and ease of assembly. At the same time, the present general robot model has certain payload and size limitations; thus, we chose a light steel structure model for simulation in this scenario. We certainly aim to extend additional BIM models (see Fig. 6) for simulation with the further growth of construction robot research and development.

In the light steel frame construction (Fig. 8), there are 9 independent foundation components, 9 column components, and 10 beam components. The designer has to export the model to the standard format of

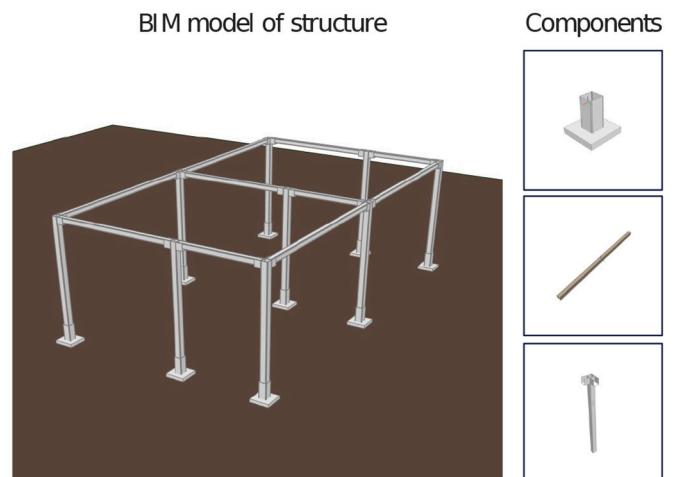


Fig. 8. The BIM model of structure for robotic construction.

¹² <https://clearpathrobotics.com/>.

IFC4 once it has been constructed in the modeling software (here we use Revit), and then the IFC file can be converted to SDF format using the approach described in Section 4.

Initially, the system reads the IFC file's header information, as well as `IfcProject`, `IfcSite`, and `IfcBuilding` information, to create a standard template file for IFC. Then, the information of the components in the model is extracted and inserted into copies of this standard template file to produce IFC files that describe each component separately. Therefore, the system will traverse the model, locate each individual component, extract its information, and write it into the standard template file, finally splitting the original IFC file into 28 independent IFC files that primarily include component information. After that, the 28 component geometries are converted into DAE format for Gazebo display and collision detection. At this stage, all the components will correspond to two files, respectively, the IFC file containing all the information and a DAE file dedicated to describing the shape.

Now, the conversion from IFC to SDF takes place. Similarly, before converting each component, the system extracts the basic project information (guid, name, type) to create a standard SDF folder including the configuration file, SDF file, and mesh folder and then writes the basic project information into the configuration file and SDF file. The system then turns the IFC files of 28 components to SDF folders, as described previously. Furthermore, we need to utilize the information from `IfcSite` to construct the world file, which involves integrating the path of the component file into the world file and declaring the component's world coordinates for the global environment.

This process implements our model from design to simulation in the Gazebo. Using `IfcOpenShell`, we created a python-based system to automate the conversion of IFC to SDF.¹³

5.2. Robot model selection and application

To implement the robot's functions, we used an open-source general-purpose robot in this case, namely the Husky robot. The Husky robot, as shown in Fig. 9, is an open-source mobile robot with a standard mobile base and a range of extra sensors that work with the ROS package to construct maps, maneuver, and avoid obstacles in a simulation environment. Based on the features of prefabricated construction, the robot needs to perform the following tasks in construction: 1. find the component to be built, 2. pick up the component, 3. deliver the component to the target location, 4. assemble the component, and avoid

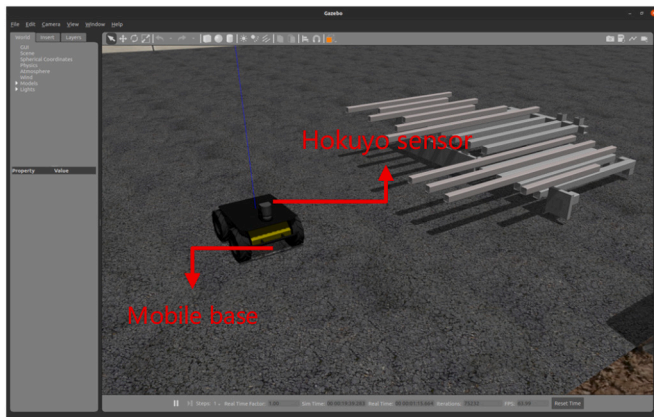


Fig. 9. The simulation environment includes the Husky robot that picks up components based on their state from the yard, and places and assembles them.

obstacles throughout the process to ensure the safety of the construction process. We will use the semantic label (“pick up” and “assemble”) for describing the picking up and assembling activities, as stated previously. As a result, in the simulation environment, robot functions need to integrate navigation and obstacle avoidance.

In general, the navigation stack obtains information from the odometer and sensors on the robot to compute the action data required by the robot in that environment and returns it to the robot for execution. The following packages, which will be briefly explained in the following, are mostly employed in the navigation stack architecture to perform mobile base control, spatial location positioning, and path planning, respectively. To move the robot in the global environment, the move base function package¹⁴ is used to accept the intended movement information and execute it by matching the robot base coordinates with the world coordinates. As a tool to realize robot spatial location, the AMCL package¹⁵ is used on the probabilistic algorithm for robot tracking based on a particle filter to realize robot location on a global map. In this case, global and local planners deployed for both path planning tasks in order to accomplish global path optimization and real-time obstacle avoidance, and function packages with different algorithms can be used in the planner based on the user's needs. Meanwhile, we use the global planner based on the A* algorithm [44] and the local planner based on the dynamic window algorithm [45]. With the above configuration, the Husky robot can achieve autonomous navigation in the Gazebo environment to perform the construction tasks required by the building components.

5.3. Smart component-based robot construction simulation

In this section, we document how to set up a simulation environment and create a component-oriented construction task publishing system using the ROS communication mechanism to create a smart component-based robot construction simulation. The construction scenario is based on a designed environment, as indicated in Fig. 10.

We designate a yard area and arrange the required building components in the center square of the environment. The construction goal for the robot is to assemble components from the yard into the structure in the central square. To begin, we need to initialize the environment, which includes defining the yard location, as well as the appropriate poses of components and the robot's initial position. In practice, the

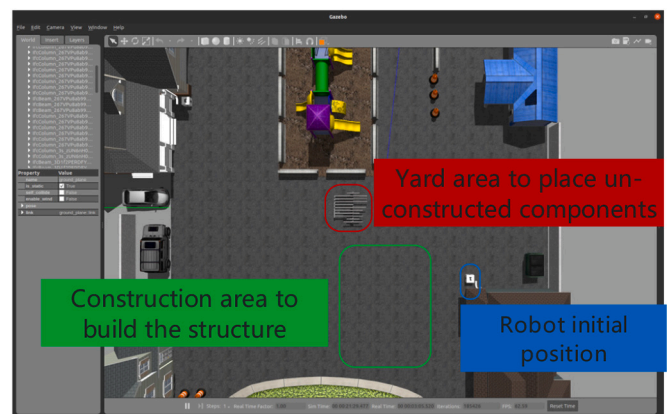


Fig. 10. The simulated construction scenario includes a yard where all to be built components are collected, a robot initial position, and the area where the structure needs to be built.

¹³ https://github.com/ISBE-TUe/ifc_to_sdf.

¹⁴ http://wiki.ros.org/move_base.

¹⁵ <http://wiki.ros.org/amcl>.

components are usually stacked horizontally, so components are designed to be stacked uniformly in the yard area as shown in the figure. The robot's initial position should be similar to a construction site's temporary rest area, or a place where the robot can be repaired and charged. Consequently, we assign that the bottom right (blue area) in the environment depicted in Fig. 10 is the robot repair and charging area as the initial position for the robot.

The suggested simulation environment consists of two main parts, namely a visualization in the rviz application,¹⁶ and a simulation engine in Gazebo. These are standard tools for robot simulation, and the contribution of our work lies mainly in connecting the BIM environment better with these simulation tools. The rviz application is mainly used to visualize the performance of the ROS function packages. The rviz environment (left in Fig. 11) includes the following parts: (1) the known map information of the robot (grey), (2) the location of the objects and their collision volume in the environment scanned by the laser (black lines and the light blue areas), (3) the inflation factor that is used to represent the safety range of collision detection (dark blue), (4) the global paths planned (green lines), and (5) the local paths planned by the laser in real-time according to the current environment (red lines). The right side of Fig. 10 shows the Gazebo simulation environment, which is used to simulate the real-world scenario using ROS.

After the environment has been initialized, the robot construction simulation can be executed. Based on the process of Section 4.4, we constructed the state machine system illustrated in Fig. 12 to accomplish the automated construction based on the components' construction tasks. The process shown in Fig. 12 functions as follows: when the previous component is completed, the system accepts the information and duties of the unbuilt component as the next task to the robot (top-left). The system then sends the task to the robot (top-right), which accepts it and proceeds to the component's position in the yard to pick up the component (bottom-right). When a component is picked up, it enters the state of needing to be transferred. The system accepts the component's target position and sends it to the robot, and the robot begins the transfer task once it reaches the target location (bottom-left). The component's state switches to 'assemble' when the robot reaches the target place, and the robot receives the task from the system and performs the assembly work. When the assembly work is accomplished, the component's construction task is completed (top-left).

With the presented state machine control, a simulation environment is created where the robot can automate the construction of smart component-based construction and realize the construction work of assembling the components into the target structure. As this simulation environment mimics real robot behavior by its embedding of ROS, the

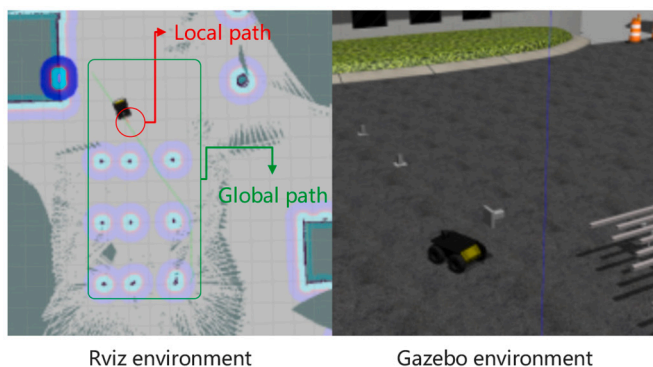


Fig. 11. Rviz and Gazebo are used in the suggested simulation environment.

resulting process can be executed also in a real-world environment, yet this has not been tested.

6. Discussion

As a result of the rapid advancement of robotics, robot-based construction is viewed as one of the potential future solutions for achieving efficient construction. Due to the unique features of the construction industry, such as long construction cycles, large-scale construction, large amounts of investment, and multiple stakeholders, as well as the high costs of development and production in the robotics industry, robots are not widely used in actual construction projects. Despite the fact that the simulation and the actual scenario will diverge, construction simulation can provide novel concepts and platforms for robotic construction research and applications. For stakeholders in the construction industry attempting to apply robots to real-world construction, an open standard simulation can cost-effectively test and optimize the development of robotic construction, while a physics-engine simulation environment can provide a realistic and intuitive representation of robots in construction. It also benefits from the use of open standards (IFC, SDF and etc.) and tools (ifcopenshell, ROS and Gazebo) that can reduce some of the constraints imposed by the modeling software on the user and enable the assessment and adoption of the method by relevant researchers.

Several limitations remain in this study to achieve a more comprehensive simulation of robot construction. First, this study abstracted the construction actions ("pick up" and "assembly") in case study. As a construction planning, the abstraction of specific construction tasks is reasonable, but in real-world scenarios, the robot requires motion planning for grasping and assembly actions in order to achieve more accurate simulation. Second, due to the diversity of building types and the complexity of the corresponding construction processes (especially cast-in-place), the focus in this study is only on prefabricated construction-based robotic construction simulation as a prototype for starting, but further expansion to more building types is needed to satisfy a wider range of construction simulation demands.

7. Conclusion and future works

This study presents a method for using IFC-based BIM models in a robotic simulation environment and executing automated construction simulations based on the notion of component-oriented construction. The proposed framework integrates an IFC-based ETL processing, a construction controller based on smart components and a ROS-based simulation platform. It leverages BIM models in a robotic aspect to achieve the IFC-based BIM models in robotic prefabricated construction simulation. Different from the previous construction simulations, this concept relieves the dependence on specific software. Based on the open standard, expands robot types and facilitates customized construction robots for robotic construction research.

Further effort should explore the unique actions of robots to execute construction based on the existing simulation environment, and first realize the specific construction actions of robots in the simulation environment, and then extend them to actual robots. Meanwhile, we will expand the number and types of robots in the framework to allow for additional testing and studying of multi-robot building collaboration. Finally, future work can aim to integrate new building types in the simulation environment by expanding the present framework.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Aiyu Zhu reports financial support was provided by China Scholarship Council (No.202007720036).

¹⁶ <http://wiki.ros.org/rviz>.

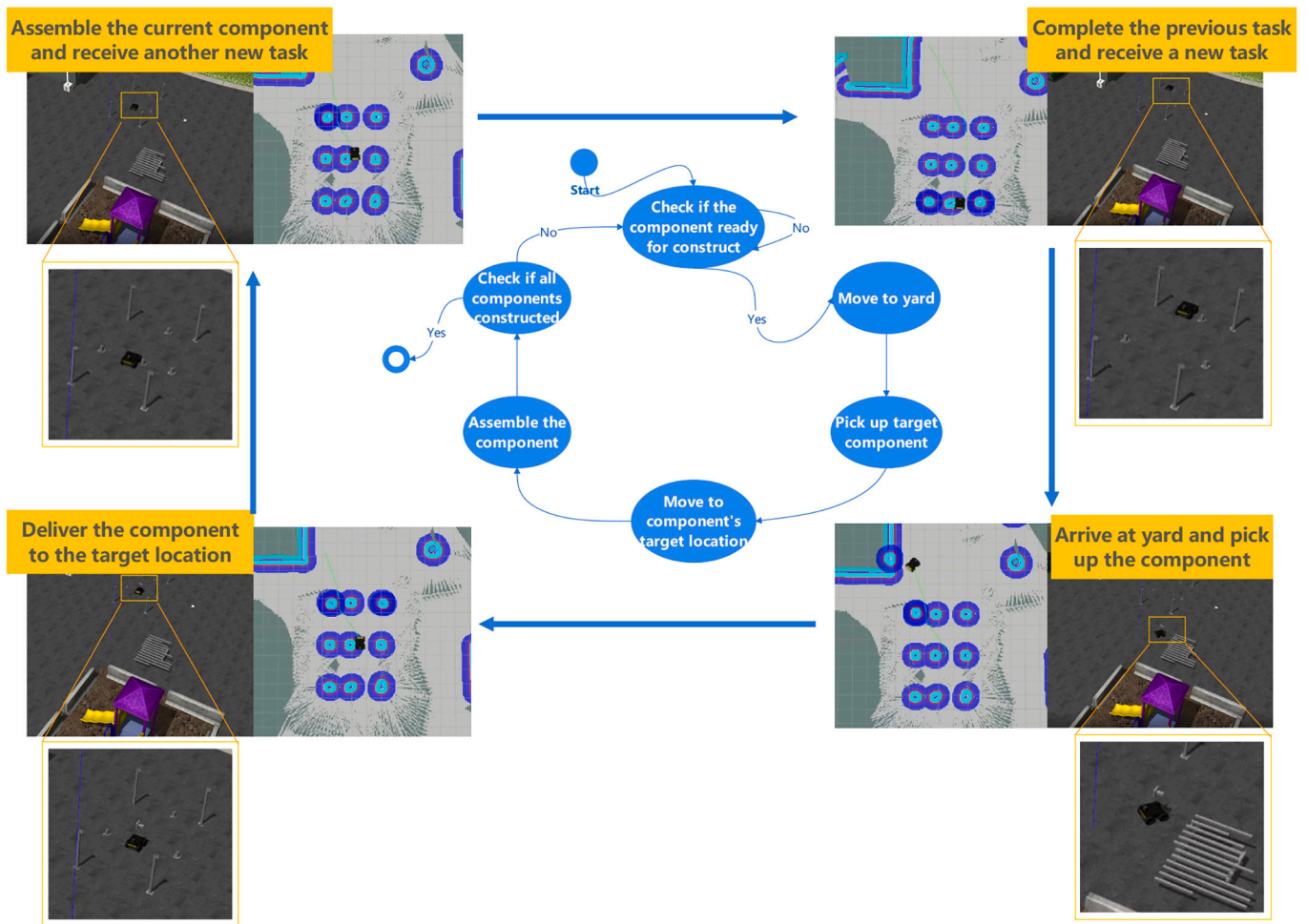


Fig. 12. The construction process goes in four steps: (1) receiving a task; (2) giving task to robot; (3) picking up component from yard; and (4) delivering and placing component in target location.

Data availability

No data was used for the research described in the article.

Acknowledgments

The support provided by the China Scholarship Council (No.202007720036) during a study of Aiyu Zhu to the Eindhoven University of Technology is acknowledged.

References

- [1] T. Bock, The future of construction automation: technological disruption and the upcoming ubiquity of robotics, *Autom. Constr.* 59 (2015) 113–121, <https://doi.org/10.1016/j.autcon.2015.07.022>.
- [2] Wallén, J., The History of the Industrial Robot Technical Report 2008 URL <http://iu.diva-portal.org/smash/get/diva2:316930/FULLTEXT01.pdf>.
- [3] N. Melenbrink, J. Werfel, A. Menges, On-site autonomous construction robots: towards unsupervised building, *Autom. Constr.* 119 (2020), 103312, <https://doi.org/10.1016/j.autcon.2020.103312>.
- [4] E. Gambao, C. Balaguer, A. Barrientos, R. Saltaren, E. Puente, Robot assembly system for the construction process automation, in: *Proceedings of International Conference on Robotics and Automation* vol. 1, 1997, pp. 46–51, <https://doi.org/10.1109/ROBOT.1997.620014>.
- [5] T. Bock, T. Linner, *Robot-Oriented Design: Design and Management Tools for the Deployment of Automation and Robotics in Construction*, Cambridge University Press, 2015, <https://doi.org/10.1017/CBO9781139924146>.
- [6] C. Kasperzyk, M.-K. Kim, I. Brilakis, Automated re-prefabrication system for buildings using robotics, *Autom. Constr.* 83 (2017) 184–195, <https://doi.org/10.1016/j.autcon.2017.08.002>.
- [7] R. Naboni, A. Kunic, A. Kramberger, C. Schlette, Design, simulation and robotic assembly of reversible timber structures, *construction, Robotics* 5 (2021) 13–22, <https://doi.org/10.1007/s41693-020-00052-7>.
- [8] J. Koerner-Al-Rawi, K.E. Park, T.K. Phillips, M. Pickoff, N. Tortorici, Robotic timber assembly, *construction, Robotics* 4 (2020) 175–185, <https://doi.org/10.1007/s41693-020-00045-6>.
- [9] G. Lee, H.-H. Kim, C.-J. Lee, S.-I. Ham, S.-H. Yun, H. Cho, B.K. Kim, G.T. Kim, K. Kim, A laser-technology-based lifting-path tracking system for a robotic tower crane, *Autom. Constr.* 18 (2009) 865–874, <https://doi.org/10.1016/j.autcon.2009.03.011>.
- [10] H. Nam, W. Choi, D. Ryu, Y. Lee, S. Han Lee, B. Ryu, Design of a bolting robot for constructing steel structure, in: *2007 International Conference on Control, Automation and Systems*, 2007, pp. 1946–1949, <https://doi.org/10.1109/ICCAS.2007.4406667>.
- [11] R. Hu, K. Iturralde, T. Linner, C. Zhao, W. Pan, A. Pracucci, T. Bock, A simple framework for the cost–benefit analysis of single-task construction robots based on a case study of a cable-driven facade installation robot, *Buildings* 11 (2021), <https://doi.org/10.3390/buildings11010008>.
- [12] M. Gharbia, A. Chang-Richards, Y. Lu, R.Y. Zhong, H. Li, Robotic technologies for on-site building construction: a systematic review, *J. Build. Eng.* 32 (2020), 101584, <https://doi.org/10.1016/j.jobbe.2020.101584>.
- [13] J. Buchli, M. Gifftaler, N. Kumar, M. Lussi, T. Sandy, K. Dörfler, N. Hack, Digital in situ fabrication - challenges and opportunities for robotic in situ fabrication in architecture, construction, and beyond, *Cem. Concr. Res.* 112 (2018) 66–75, <https://doi.org/10.1016/j.cemconres.2018.05.013>.
- [14] M.I. Kellner, R.J. Madachy, D.M. Raffo, Software process simulation modeling: why? What? How? *J. Syst. Softw.* 46 (1999) 91–105, [https://doi.org/10.1016/S0164-1212\(99\)00003-5](https://doi.org/10.1016/S0164-1212(99)00003-5).
- [15] S. Azhar, A. Nadeem, J.Y. Mok, B.H. Leung, Building information modeling (BIM): a new paradigm for visual interactive modeling and simulation for construction projects, in: *Proc., First International Conference on Construction in Developing Countries* vol. 1, 2008, pp. 435–446. URL: <https://www.researchgate.net/publication/283118367>. URL:.

- [16] J. Zhang, H. Luo, J. Xu, Towards fully BIM-enabled building automation and robotics: a perspective of lifecycle information flow, *Comput. Ind.* 135 (2022), 103570, <https://doi.org/10.1016/j.compind.2021.103570>.
- [17] A. Zhu, P. Pauwels, B. De Vries, Smart component-oriented method of construction robot coordination for prefabricated housing, *Autom. Constr.* 129 (2021), 103778, <https://doi.org/10.1016/j.autcon.2021.103778>.
- [18] International Organization for Standardization, 16739-1, Industry foundation classes (IFC) for data sharing in the construction and facility management industries—part 1: data schema, 2018. URL: <https://www.iso.org/standard/70303.html>.
- [19] M. Golparvar-Fard, F. Peña-Mora, S. Savarese, Automated progress monitoring using unordered daily construction photographs and IFC-based building information models, *J. Comput. Civ. Eng.* 29 (2015) 04014025, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000205](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000205).
- [20] H. Hamledari, E. Rezaezadeh Azar, B. McCabe, IFC-based development of as-built and as-is BIMs using construction and facility inspection data: site-to-BIM data transfer automation, *J. Comput. Civ. Eng.* 32 (2018) 04017075, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000727](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000727).
- [21] A.M. Tanyer, G. Aouad, Moving beyond the fourth dimension with an IFC-based single project database, *Autom. Constr.* 14 (2005) 15–32, <https://doi.org/10.1016/j.autcon.2004.06.002>.
- [22] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, R. Klein, The IFC to linked building data converter-current status, in: *Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018)* vol. 2159, 2018, pp. 34–43. CEUR Workshop Proceedings. URL: <http://ceur-ws.org/Vol-2159/04paper.pdf>.
- [23] S. Kim, M. Peavy, P.-C. Huang, K. Kim, Development of BIM-integrated construction robot task planning and simulation system, *Autom. Constr.* 127 (2021), 103720, <https://doi.org/10.1016/j.autcon.2021.103720>.
- [24] G. Costa, L. Madrazo, Connecting building component catalogues with BIM models using semantic technologies: an application for precast concrete components, *Autom. Constr.* 57 (2015) 239–248, <https://doi.org/10.1016/j.autcon.2015.05.007>.
- [25] H. Liu, M. Al-Hussein, M. Lu, BIM-based integrated approach for detailed construction scheduling under resource constraints, *Autom. Constr.* 53 (2015) 29–43, <https://doi.org/10.1016/j.autcon.2015.03.008>.
- [26] Y. Wang, Z. Yuan, Research on BIM-based assembly sequence planning of prefabricated buildings, in: *ICCREM 2017*, 2017, pp. 10–17, <https://doi.org/10.1061/9780784481059.002>.
- [27] Y. Liu, M. Li, B.C. Wong, C.M. Chan, J.C. Cheng, V.J. Gan, BIM-BVBS integration with openBIM standards for automatic prefabrication of steel reinforcement, *Autom. Constr.* 125 (2021) 103654, <https://doi.org/10.1016/j.autcon.2021.103654>.
- [28] Z. Li, G.Q. Shen, X. Xue, Critical review of the research on the management of prefabricated construction, *Habitat Int.* 43 (2014) 240–249, <https://doi.org/10.1016/j.habitatint.2014.04.001>.
- [29] M.A. Hossain, A. Nadeem, Towards digitizing the construction industry: State of the art of construction 4.0, in: *Proceedings of the ISEC* vol. 10, 2019, <https://doi.org/10.14455/ISEC.res.2019.184>.
- [30] Y. Niu, W. Lu, K. Chen, G.G. Huang, C. Anumba, Smart construction objects, *J. Comput. Civ. Eng.* 30 (2016) 04015070, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000550](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000550).
- [31] Y. Niu, W. Lu, D. Liu, The application scenarios of smart construction objects (SCOs) in construction, in: *Proceedings of the 20th International Symposium on Advancement of Construction Management and Real Estate*, Springer, 2017, pp. 969–980, https://doi.org/10.1007/978-981-10-0855-9_86.
- [32] X. Li, G.Q. Shen, P. Wu, H. Fan, H. Wu, Y. Teng, RBL-PHP: simulation of lean construction and information technologies for prefabrication housing production, *J. Manag. Eng.* 34 (2018) 04017053, [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000577](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000577).
- [33] X. Li, G.Q. Shen, P. Wu, T. Yue, Integrating building information modeling and prefabrication housing production, *Autom. Constr.* 100 (2019) 46–60, <https://doi.org/10.1016/j.autcon.2018.12.024>.
- [34] J.I. Avendaño, S. Zlatanova, A. Domingo, P. Pérez, C. Correa, Utilization of BIM in steel building projects: a systematic literature review, *Buildings* 12 (2022) 713, <https://doi.org/10.3390/buildings12060713>.
- [35] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(IEEE Cat. No. 04CH37566) vol. 3, IEEE, 2004, pp. 2149–2154, <https://doi.org/10.1109/IROS.2004.1389727>.
- [36] E. Rohmer, S.P. Singh, M. Freese, V-rep: A versatile and scalable robot simulation framework, in: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IIEEE*, 2013, pp. 1321–1326, <https://doi.org/10.1109/IROS.2013.6696520>.
- [37] R. Weissenböck, Robotic design-fabrication - exploring robotic fabrication as a dynamic design process, in: *Real Time - Proceedings of the 33rd eCAADe Conference - Volume 2, CUMINCAD*, 2015, pp. 309–318, <https://doi.org/10.52842/conf.ecaade.2015.2.309>.
- [38] S. Sharif, V. Agrawal, L. Sweet, Adaptive industrial robot control for designers, in: *Proceedings of the 35th eCAADe Conference, CUMINCAD*, 2017, pp. 151–158, <https://doi.org/10.52842/conf.ecaade.2017.2.151>.
- [39] A. Das, I. Worre Foged, M.B. Jensen, Designing with a robot-interactive methods for brick wall design using computer vision, in: *Proceedings of the 38th eCAADe Conference, CUMINCAD*, 2020, pp. 605–612, <https://doi.org/10.52842/conf.ecaade.2020.2.605>.
- [40] O.W. Chong, J. Zhang, R.M. Voyles, B.-C. Min, BIM-based simulation of construction robotics in the assembly process of wood frames, *Autom. Constr.* 137 (2022), 104194, <https://doi.org/10.1016/j.autcon.2022.104194>.
- [41] R. Rust, G. Casas, S. Parascho, D. Jenny, K. Dörfler, M. Helmreich, A. Gandia, Z. Ma, I. Ariza, M. Pacher, B. Lytle, Y. Huang, COMPAS FAB: robotic fabrication package for the compas framework, gramazio Kohler research, ETH Zürich, https://github.com/compas-dev/compas_fab/.
- [42] B. Belousov, B. Wibranek, J. Schneider, T. Schneider, G. Chalvatzaki, J. Peters, O. Tessmann, Robotic architectural assembly with tactile skills: simulation and optimization, *Autom. Constr.* 133 (2022), 104006, <https://doi.org/10.1016/j.autcon.2021.104006>.
- [43] buildingSMART, IFC specifications database, 2022. URL: <http://technical.buildingsmart.org/standards/ifc/>. last accessed date: 2023/04/17.
- [44] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Syst. Sci. Cybernet.* 4 (1968) 100–107, <https://doi.org/10.1109/TSSC.1968.300136>.
- [45] M. Seder, I. Petrovic, Dynamic window based approach to mobile robot motion control in the presence of moving obstacles, in: *Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE*, 2007, pp. 1986–1991, <https://doi.org/10.1109/ROBOT.2007.363613>.