BIM-based task and motion planning prototype for robotic assembly of COVID-19 hospitalisation units—flatpack house

Yifan Gao[a,c,d], Jiawei Meng[b], Jiangpeng Shu[a,*], Yuanchang Liu[b]

[a] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China

[b] Department of Mechanical Engineering, University College London, London WC1E 6BT, The United Kingdom

[c] Center for Balance Architecture, Zhejiang University, Hangzhou 310058, China

[d] The Architectural Design & Research Institute of Zhejiang University Co. Ltd, Hangzhou 310028, China

* Corresponding author at 866, Yuhangtang Road, Hangzhou, Zhejiang 310058, China

Email address: jpeshu@zju.edu.cn (Jiangpeng Shu, corresponding author)

yfgao91@zju.edu.cn (Yifan Gao)

jiawei.meng@ucl.ac.uk (Jiawei Meng)

yuanchang.liu@ucl.ac.uk (Yuanchang Liu)

BIM-based task and motion planning prototype for robotic assembly of COVID-19 hospitalisation units—flatpack house

Abstract: Fast transmission of COVID-19 led to mass cancelling of events to contain the virus outbreak. Amid lockdown restrictions, a vast number of construction projects came to a halt. Robotic platforms can perform construction projects in an unmanned manner, thus ensuring the essential construction tasks are not suspended during the pandemic. This research developed a BIM-based prototype, including a task planning algorithm and a motion planning algorithm, to assist in the robotic assembly of COVID-19 hospitalisation facilities with prefabricated components. The task planning algorithm can determine the assembly sequence and coordinates for various types of prefabricated components. The motion planning algorithm can generate robots' kinematic parameters for performing the assembly of the prefabricated components. Testing of the prototype finds that it has satisfactory performance in terms of 1) the reasonableness of assembly sequence determined, 2) reachability for the assembly coordinates of prefabricated components, and 3) capability to avoid obstacles.

Keywords: Building information modelling, robotic construction, hospitalisation facility, COVID-19 pandemic, motion planning.

1. Introduction

The pandemic caused by COVID-19 is the most serious global health crisis during the past decades due to the rapid growth in confirmed cases and a massive spike in hospitalisations [1]. To prevent the further spread of the virus, China built two new hospitals (*Leishenshan* and *Huoshenshan*) in Wuhan and also converted existing facilities to 16 module hospitals [2]. These measures effectively reduced the number of death and the spread of the virus [3]. Following these examples, the UK also turned convention centres into seven Nightingale hospitals [4]. Nevertheless, numerous reports from different world regions indicate that many patients have limited access to treatment due to the shortage of hospitalisation facilities [5]. As COVID-19 still looms as a significant threat to human beings, critical infrastructure for pandemic isolation and treatment remains far from adequate globally [3].

The current method for constructing the COVID-19 hospitalisation facilities is by having human workers assemble each prefabricated component (e.g., purlin, pillar, beam) into designated coordinates [2]. However, under the pandemic lockdown situation, it becomes difficult for workers to access the site and conduct construction activities, and a vast number of construction projects have come to a halt [6]. As the Associated General Contractors (AGC) in the US pointed out [7], nearly 90% of the US domestic construction projects were put on hold in 2020, and more than 27% of the construction organisations have either furloughed or laid off employees in 2021. As a result, a certain number of COVID-19 healthcare projects, which should have been constructed to provide the treatment spaces for the infected patients, were not delivered in time [7]. Consequently, technologies that can enable the unmanned assembly of hospitalisation facilities from prefabricated components are in urgent need. This spawns the idea in this research to develop robotic technologies for assembling COVID-19 hospitalisation facilities.

In the literature, empirical evidence has been provided that the use of robotic technologies has the potential to replace human labour and alleviate the impact of lockdown restrictions on the construction progress [8–11]. The evidences are from industry survey [12], systematic review [13–16], mechanical design [17–20], prototype and methodology development [8–11,21–23], algorithmic design [18,19], and commentary [8,24,25]. In these literature, the studies by Terada and Murata [8], Willmann et al. [9], Ding et al. [10], and King et al. [11] are found to have a similar scientific focus to this research — utilising robots to assemble prefabricated building components. Terada and Murata [8] utilised a robotic manipulator to assemble building blocks. Willmann et al. [9] designed a robotic prototype for timber construction, which fosters automation penetration across the digital workflow including timber fabrication, perforation, and connection. Ding et al. [10] and King et al. [11] presented methods that can determine assembly coordinates of building components for robotic construction according to digital blueprints in Rhinoceros.

The COVID-19 hospitalisation facilities are composed of container-type units—flatpack house, where each flatpack house unit is a ward [2]. The current method to assemble the flatpack house units of COVID-19 hospitalisation facilities consists of the following characteristics.

First, there is a fixed sequence for the assembly of the flatpack house unit. The structural frame is assembled first (including the beam, column, and purlin components). Once the frame is in place, the floor and roof panels are assembled next. Then, the wall panels are attached. This is interpreted in greater detail in section 2. Second, the prefabricated components of the flatpack house are made from lightweight steel, which is more suitable for robot assembly compared with components that are made from dense materials such as concrete. However, applying robots to replace human labour in the assembly of flatpack house, there are still difficulties that await to be addressed. The difficulties lie in letting the end-tip of the robot autonomously follow a pre-determined sequence to place the prefabricated components in coordinates where they are needed in space without human intervention. The authors found that the robotic approaches developed in the existing studies [8–11] (as aforementioned) have limited usability for addressing the technical difficulties in assembling flatpack house, given the lack of the following capabilities:

- *Pre-determining the assembly sequence of building components*. In the existing studies [8–11], the assembly coordinates of prefabricated building components were predetermined for the robotic assembly. However, the function of determining the assembly sequence of the prefabricated building components was not considered in their approaches. In such a case, workers are still needed to collaborate with the robotic platforms to manually input the coordinates for the assembly of each building component respectively. Given the pandemic lockdown situation, the robotic assembly of the hospitalisation facilities is expected to be performed automatically without human intervention. The challenge of such a solution lies in the generation of a reasonable assembly sequence of building components, and then the end-tip of the robot can follow the pre-determined sequence to place building components in coordinates where they are needed in space [23]. Therefore, quantitative spatial reasoning of the building components' assembly sequence is a critical issue to consider for the robotic assembly of the hospitalisation facilities.

Given the background, the scientific question of the study is how to determine a mathematical relationship between coordinates of prefabricated components and assembly sequence, with the consideration of their geometry and centroid, for robotic construction.

Therefore, this research aims to overcome the technical challenges and provide a robotic prototype that consists of the following:

1) A task planning algorithm that can determine a mathematical relationship between coordinates of prefabricated components and assembly sequence, with the consideration of geometry and centroid, for robotic construction; and
2) A motion planning algorithm that can analyse the determined assembly sequence and coordinates and generate robots' kinematic parameters for performing the assembly of COVID-19 hospitalisation facilities.

In addition, in the research field of industrial robot, there are a number of recently published studies that have reported the data transmission between product design data and robotic platforms for manufacturing such as Tao et al. [26], Jokić et al. [27], Izagirre et al. [28], Zhang et al. [29], and Li et al. [30]. However, the authors found that there is still relatively little information on the data penetration between the building digital designs and robotic platforms. This research aims to fill in the gap and investigate the data operation method in the proposed robotic prototype for facilitating the information penetration between the building digital designs and robotic platforms.

The rest of the paper is organised as follows: section 2 demonstrates the construction characteristics of COVID-19 hospitalisation facilities, section 3 presents the overall architecture of the proposed prototype, section 4 illustrates the configuration of the robotic platform, section 5 interprets the task planning algorithm, section 6 interprets the motion planning algorithm, section 7 demonstrates the testing results of the developed prototype, section 8 discusses the theoretical and practical implications, while section 9 summarises the findings, notes the limitations, and recommends future research directions.

2. Demonstration of COVID-19 hospitalisation facilities

As documented in Luo et al. [2], the COVID-19 hospitalisation facilities of *Leishenshan* hospital in Wuhan, China are composed of more than 3000 container-type units—flatpack house. Each flatpack house unit is a ward, which has a standard size of 6.0 meters long, 3.0 meters wide, and 2.6 meters tall and consists of prefabricated components including purlins, beams, columns, and panels [2] (see Figure 1).

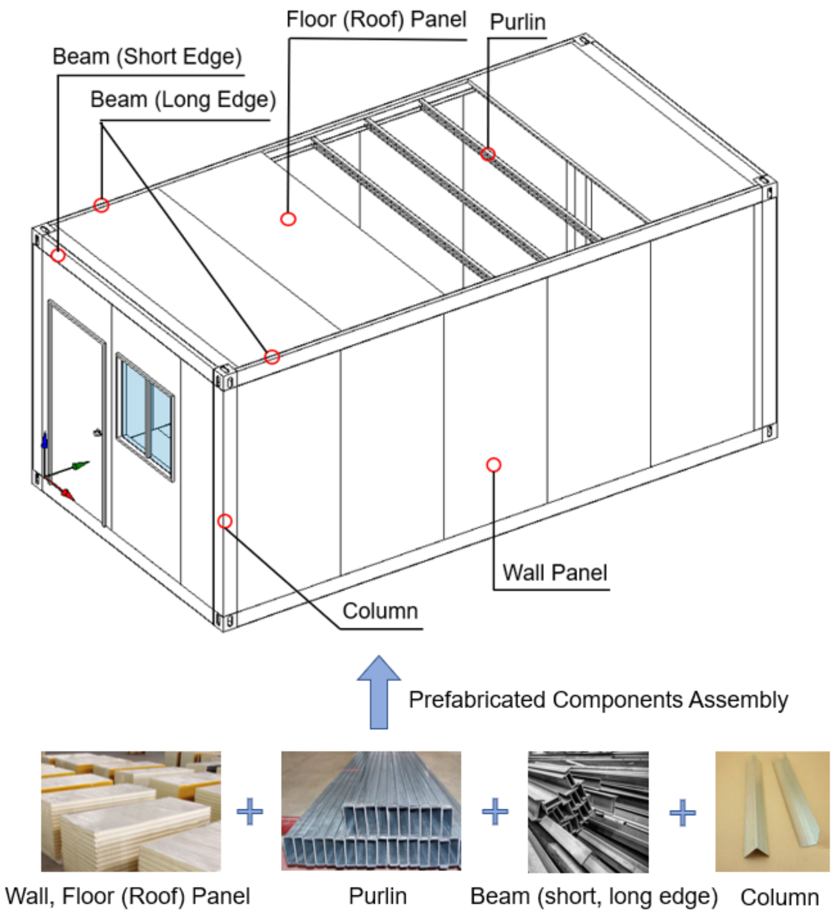


Figure 1. The standard flatpack house unit utilised in *Leishenshan* hospital.

According to Luo et al. [2] and industrial professionals' depiction, the current method to construct the *Leishenshan* hospitalisation facilities consisted of the following characteristics. First, the flatpack house units were assembled on-site and piece by piece (by human workers) from prefabricated components that were made in advance in a factory. Second, the flatpack house units were hoisted into the designated location and lined up side by side (by a mobile crane). In Figure 2, the authors utilise a computer-simulated environment to illustrate the construction procedure. As can be seen, a truck transports the prefabricated components from the factory to the building site, and worker (A) unloads the components from the truck and stores them in the trolley. Then worker (B) assembles the flatpack house using prefabricated components in the trolley. Specifically, the assembly of the flatpack house is threefold. The structural frame is assembled first (including the beam, column, and purlin components). Once the frame is in place, the floor and roof panels are assembled next. Then, the wall panels are attached. When the assembly completes, a mobile crane is used to lift the flatpack house unit from the assembling area and install the unit into the hoisting area.

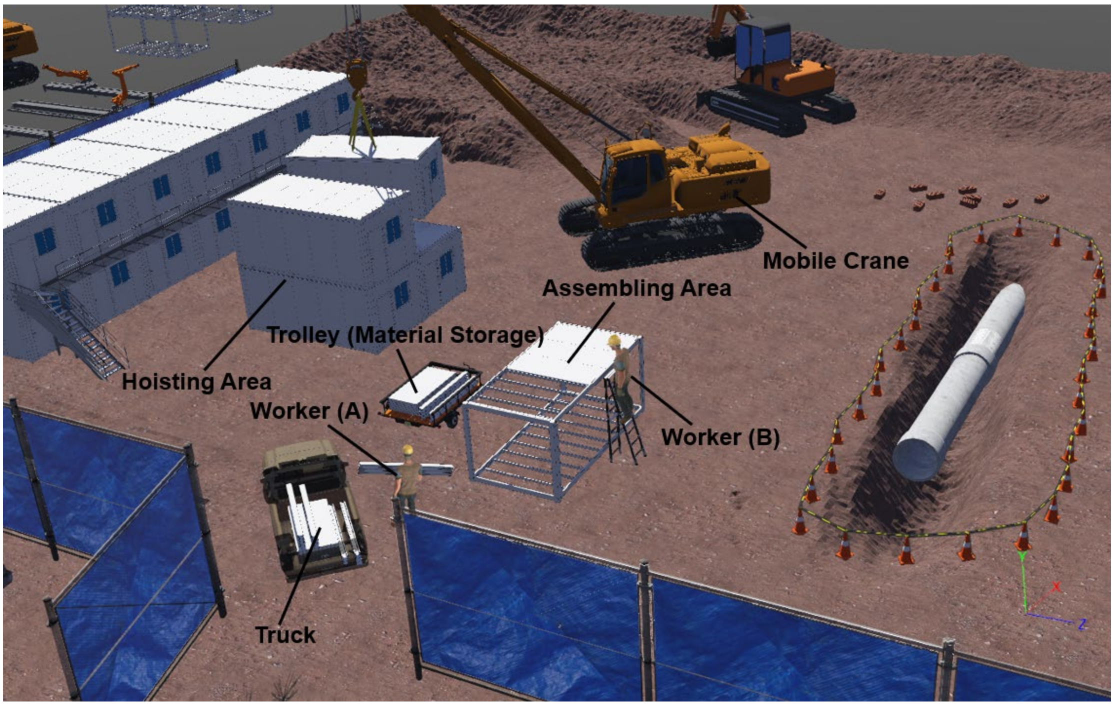


Figure 2. The *Leishenshan* hospital construction procedure (illustrated in a computer-simulated environment).

Information on the material and mass of the prefabricated components as well as the number of each component used in a flatpack house is provided in Table 1. According to the information, it is estimated that a flatpack house has an overall weight of 969.2 kg. This exceeds the handling capacity of the top-size industrial robotic manipulator on the market—ABB IRB 8700 (rated payload: 500kg). Mobile cranes might be more suitable than robotic manipulators for the installation of the standard flatpack house unit at the designated location. Therefore, this research focuses on robotising the first stage of hospitalisation facilities construction—assembling the flatpack house from prefabricated components. Note that this research focuses on the robotic assembly of structural components (see Table 1), and does not involve the installation of mechanical, electrical, and plumbing parts. The robot will replace human labour in the assembly process, where the construction difficulties lie in letting the end-tip of the robot autonomously follow a pre-determined sequence to place the prefabricated components in coordinates where they are needed in space without human intervention. To achieve the targeted performance for the robot, a task planning algorithm and a motion planning algorithm will be developed to respectively: 1) determine reasonable assembly coordinates and sequence of prefabricated components; and 2) analyse the determined assembly sequence and generate robots' kinematic parameters for performing the assembly of COVID-19 hospitalisation facilities autonomously without human intervention. Note that the sequence will be determined in accord with the threefold assembly logic as specified (i.e., first the frame, next the floor and roof panels, then the wall panels).

Table 1. Material and mass of prefabricated components, and the number of each component used in a unit.

Component	Material	Mass (kg)	Number
Beam (Long Edge)	Galvanised Steel	55.7	4
Beam (Short Edge)	Galvanised Steel	27.1	4
Column	Galvanised Steel	20.7	4
Purlin	Galvanised Steel	4.8	18
Wall Panel	Metal-skinned Polystyrene	12.9	18
Floor (Roof) Panel	Metal-skinned Polystyrene	16.9	14

3. System overview of the proposed prototype

This research explores how the COVID-19 hospitalisation facilities can be assembled from prefabricated components using robotic manipulators. As discussed, the motion control of a robotic manipulator consists of letting the tip of the end-effector follow a pre-determined "sequence" of "coordinates". Thus, the challenge of such a solution lies in the generation of a reasonable assembly "sequence" of building components, and the precise, automated placement of building components in "coordinates" where it is needed in space. To provide a prototype that can outplay the challenges, this research considers expanding the digital blueprint of the COVID-19 hospitalisation facilities in BIM into robotic control instructions. The process generates a task planning algorithm and a motion planning algorithm. As Ding et al. [10] pointed out, BIM projects contain large quantities of spatial information that can be used to serve build-up activities. For example, BIM projects are composed of loadable families (known as the "library components"), which are the graphical representations of prefabricated building components [19] (Figure 3a). To create a BIM project, the families are spatially integrated (Figure 3b). The process of integrating families into a unifying BIM project creates georeferenced properties that are useful for determining the assembly coordinates and sequence of building components.

(a)

(b)

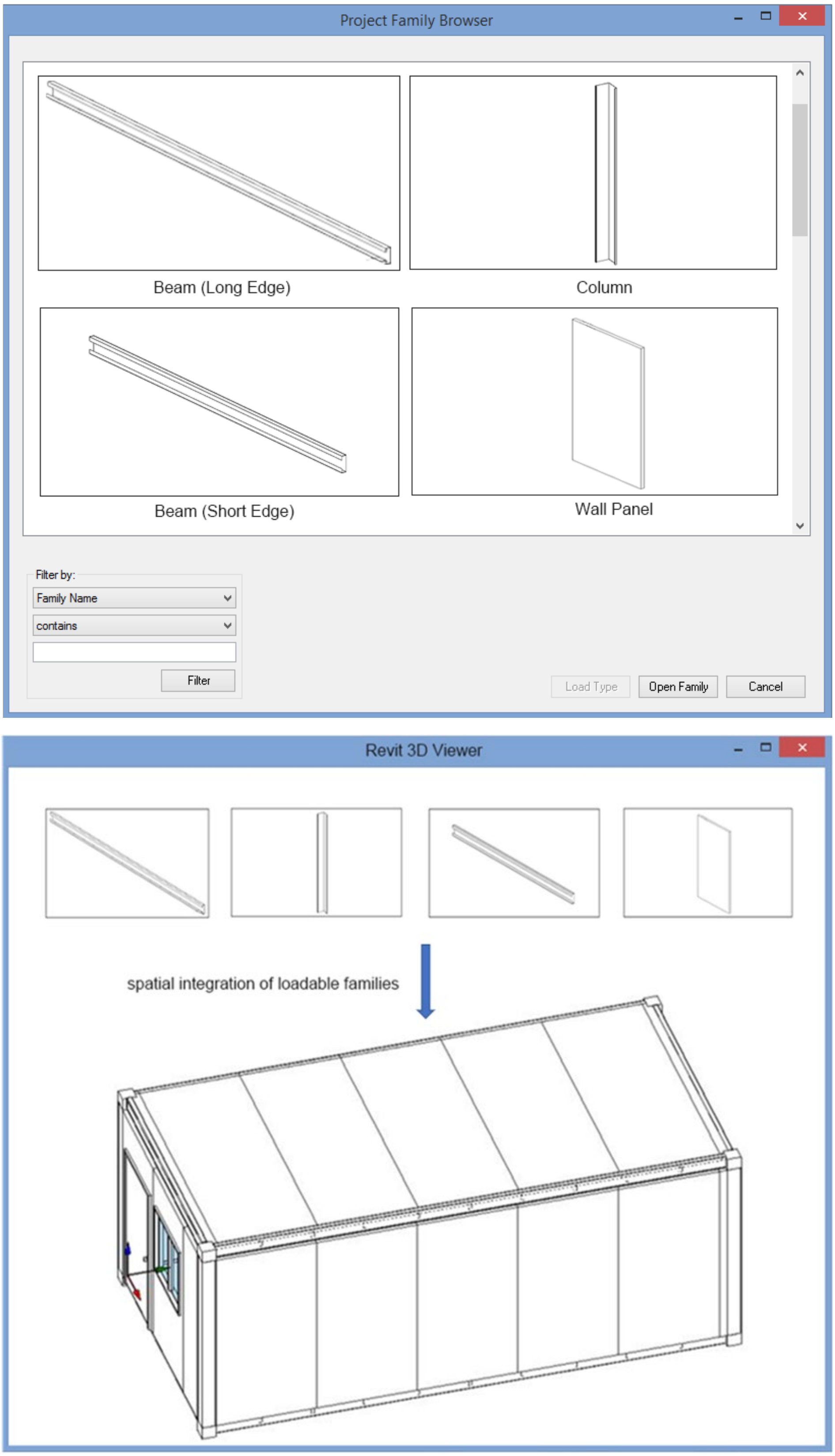


Figure 3. BIM project creation: (a) BIM loadable families; (b) spatial integration of loadable families into BIM.

BIM design tools (e.g., Autodesk Revit) host informative databases for their projects, where Application Programming Interface (API) (e.g., Dynamo) can be utilised to couple with, and democratise, the database for end-users to get access to data and retrieve desired features [31]. In this research, Dynamo and the Robot Operating System (ROS) were utilised to develop the robotic prototype for assembling the prefabricated hospitalisation facilities, where Dynamo's role is to provide the required data input for ROS. A Dynamo-based task planning algorithm was developed to locate the assembly coordinates of the building components in the flatpack house Revit model and then generate a reasonable assembly sequence of the components based on their coordinates. The assembly sequence data constitutes the information required for the robotic motion planning. Then, the Robot Operating System (ROS) executes the motion planning algorithm to analyse the assembly sequence data and generate robots' kinematic parameters for performing the assembly of the flatpack house (including the joint and path parameters). A communication interface was also established to operate the data transmission between the task and motion planning layers. An overview of the architecture of the proposed prototype is provided in Figure 4. The detailed steps are discussed in sections 4-7 below.

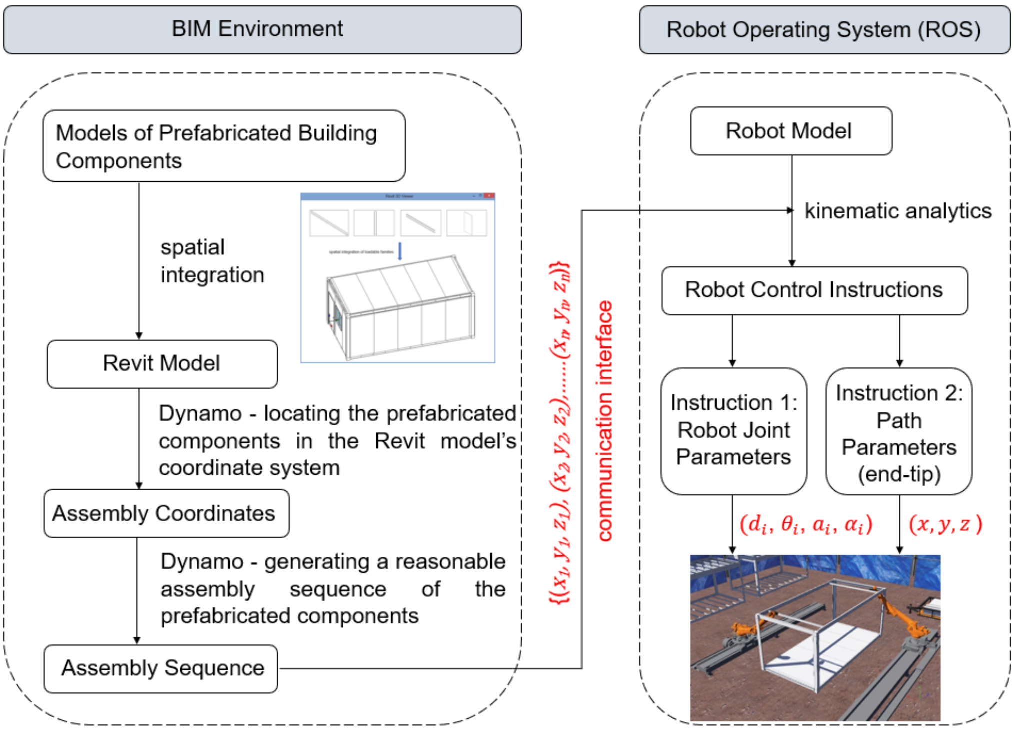
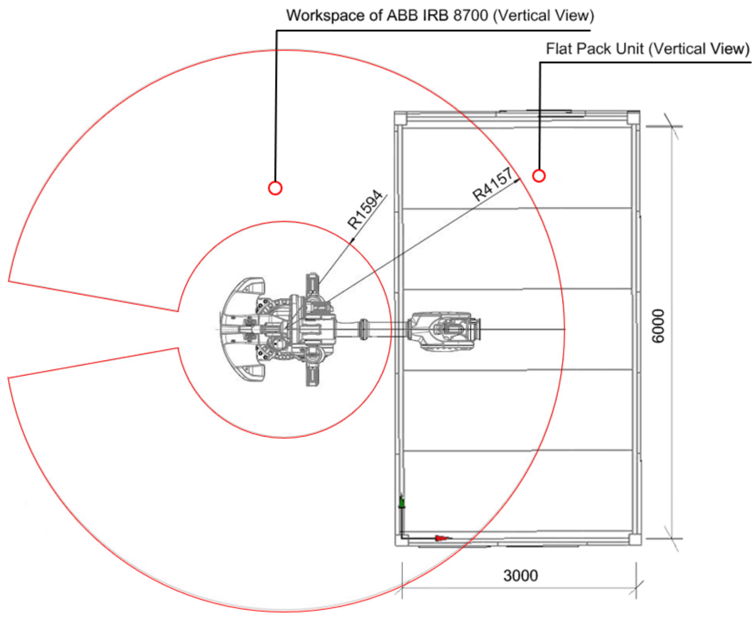


Figure 4. An overview of the architecture of the proposed prototype.

4. Configuration of the robotic platform

The workspace and payload of a robotic manipulator determine if the manipulator is suitable for certain construction tasks [23]. Workspace is the set of all positions that a manipulator can reach, which constitutes a reachable volume between the maximum and minimal working radius of the manipulator [32] (see Figure 5a). However, it was found that when having one side of the flatpack house placed as close as possible to the minimal working radius, the workspace of the top size industrial robotic manipulator on the market—ABB IRB 8700 (work range: 4157mm, rated payload: 500kg)—still cannot fully cover the spatial extent of the flatpack house for assembly, with the far end being out of reach (see Figure 5a). In this case, collaborative construction using dual robotic manipulators—KUKA KR 120 R3100 (work range: 3095mm, rated payload: 120kg)—is considered (see Figure 5b). To provide a reasonable range of workspace for assembling COVID-19 hospitalisation facilities, the manipulators are mounted on KUKA KL 4000 linear unit (maximum translational distance: 8500mm) [33]. Payload is the amount of matter (i.e., mass) that a manipulator can lift [32]. As can be seen in Table 1, the beam (long edge) is the structural component that has the maximal mass (i.e., 55.7kg). Thus, the manipulator to use in this research should meet the criterion of having its payload exceed 55.7kg. The rated payload of KR 120 R3100 is 120kg, which meets the criterion and is robust for the assembly of the flatpack house in this research.

(a)



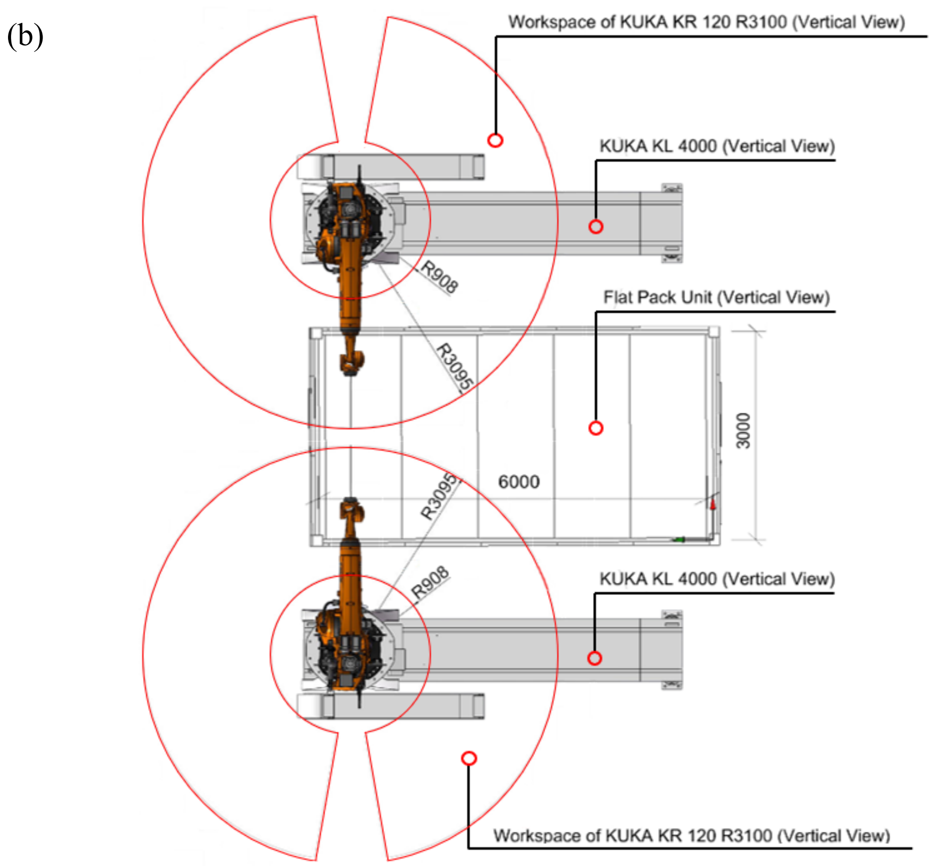


Figure 5. Robotic platforms: (a) ABB IRB 8700; (b) KUKA KR 120 R3100.

The robotic manipulator used in this research is composed of one prismatic joint and six revolute joints (i.e., seven degree-of-freedom (DOF)). A revolute joint enables a relative rotary motion about an axis, and a prismatic joint translates a linear displacement along an axis [34]. Kinematic specification of the joints is presented in Figure 6. Motion range of the joints is provided by KUKA [35] as follows: joint 1 (from 0 to 8.500 m), joint 2 (from -3.227 to 3.227 rad), joint 3 (from -1.483 to 0.872 rad), joint 4 (from -1.361 to 2.093 rad), joint 5 (from -6.106 to 6.106 rad), joint 6 (from -2.181 to 2.181 rad), and joint 7 (from -6.106 to 6.106 rad). The motion range will be utilised as joint constraints for kinematic analytics in this research. The COVAL vacuum gripper is connected to the main body of the manipulator through a flange, which is the end effector and operates utilising vacuum adsorption to hoist building components [36] (Figure 6). The gripper is designed for heavy-duty applications and can withstand a weight of up to 68kg.

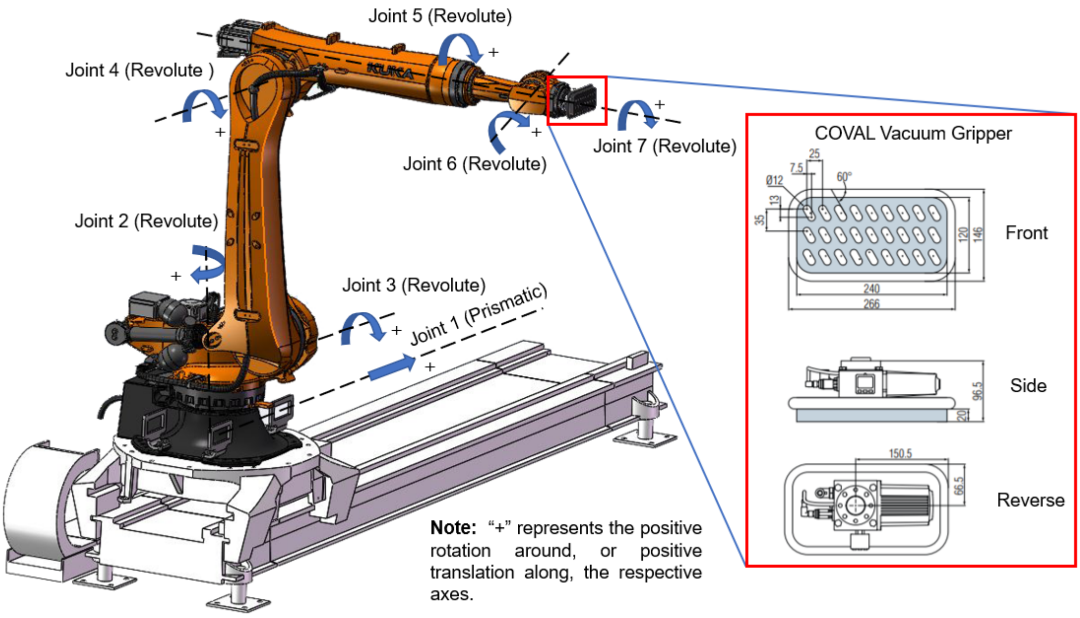


Figure 6. Kinematic specification of the joints.

5. Task planning algorithm: determining assembly coordinates and sequence

In this research, the task planning logic to bring into effect is locating the placement coordinates of the building components for assembly and then generating a reasonable assembly sequence of the components based on their coordinates. The authors utilised the Autodesk Revit API—Dynamo—to develop the task planning algorithm, which is named the Assembly Coordinates and Sequence Determination (ACASD) algorithm. Dynamo is a visual programming environment that extends the parametric analysis capabilities of Revit [37]. The analytic capability of Dynamo is enabled through functional nodes, which are composed of input and output ports and are connected in sequence to form a complete logic [37] (see Figure 7). Users can compile in Python to create nodes for specific functions [31] (e.g., *Assembly Sequence ()* in Figure 7). The compositions of the ACASD algorithm are presented in Figure 7 and Algorithm 1 below. As can be seen, the algorithm consists of three sections, which are discussed in greater detail in the following paragraphs.

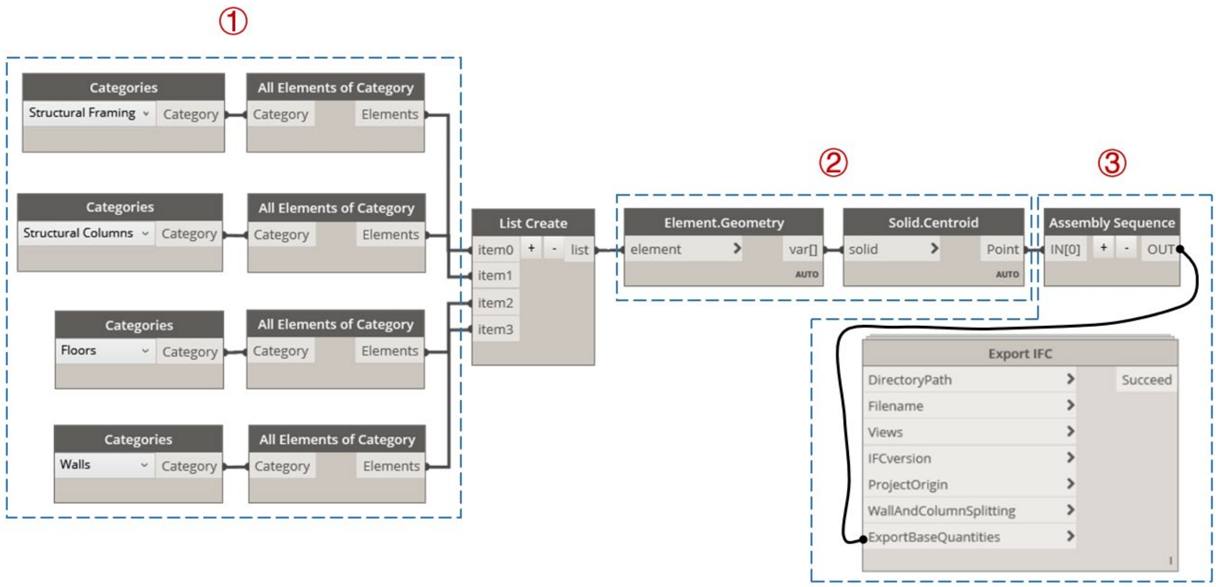


Figure 7. The architecture of the Assembly Coordinates and Sequence Determination (ACASD) algorithm.

Algorithm 1: Assembly Coordinates and Sequence Determination (ACASD) Algorithm

---

orglst ← categories(structural framing);
orglst ← categories(Structural Columns);
orglst ← categories(Floors);
orglst ← categories(Walls);
orglst ← orglst.Geometry();
orglst ← orglst.Centroid();
sortbyZYX(orglst):
  orglst.sort(key = lambda orglst: (orglst.Z, orglst.Y, orglst.X));
  return orglst;
end
input ← IN[0];
for i in input:
  if i.categories() ! = Walls:
    result_frame ← append(sortbyZYX(i));
  else:
    result_wall ← append(sortbyZYX(i));
end
OUT ← {result_frame, result_wall}

Section 1 of the algorithm seeks to let Dynamo identify all building components from the Revit model in Figure 3b. This is enabled by nodes *Categories ()* and *All Elements of Category ()* (Figure 7). *Categories ()* returns the names of structural categories that form the Revit model (e.g., framing, columns, floors). In Revit, the cache keeps track of the building components by attaching a unique identifier ID to each component. *All Elements of Category ()* can read the identifier IDs of all components of the returned categories. The reading result is passed to the *List Create ()* node, which creates a list of the identified components' IDs (Figure 7). Using the nodes *Element.Geometry ()* and *Solid.Centroid ()*, section 2 of the algorithm is designed to locate the identified components in the Revit model's coordinate system (Figure 8). *Element.Geometry ()* takes the list of the identified components' IDs and retrieves the geometry associated with the IDs from Revit. *Solid.Centroid ()* then detects the vertexes of each geometry, computes the centroid for each geometry by averaging the sums of the coordinates of the vertexes, and plots the centroid (represented by black dots in Figure 8). The origin of the reference frame for describing the centroid coordinates was set at the bottom-left corner of the flatpack house (blue arrow—z-axis, red arrow—x-axis, and green arrow—y-axis) (Figure 8).

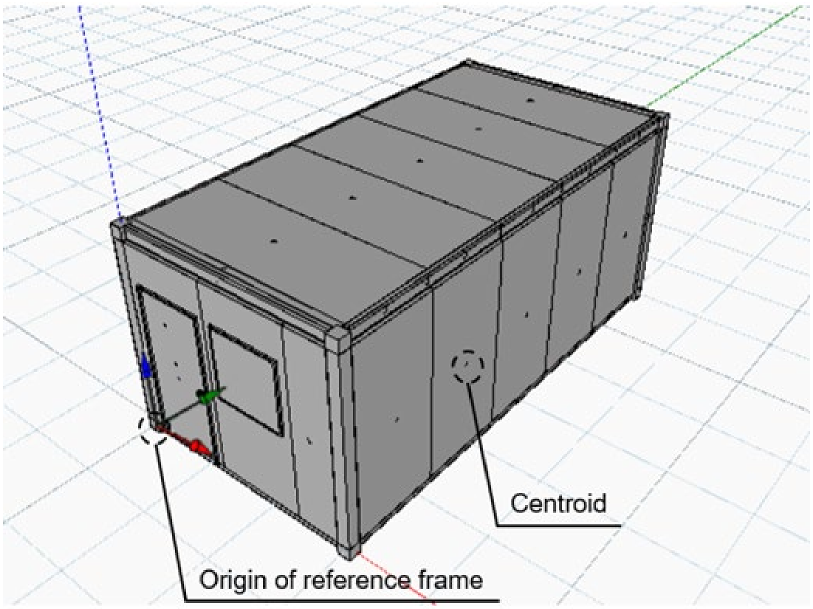
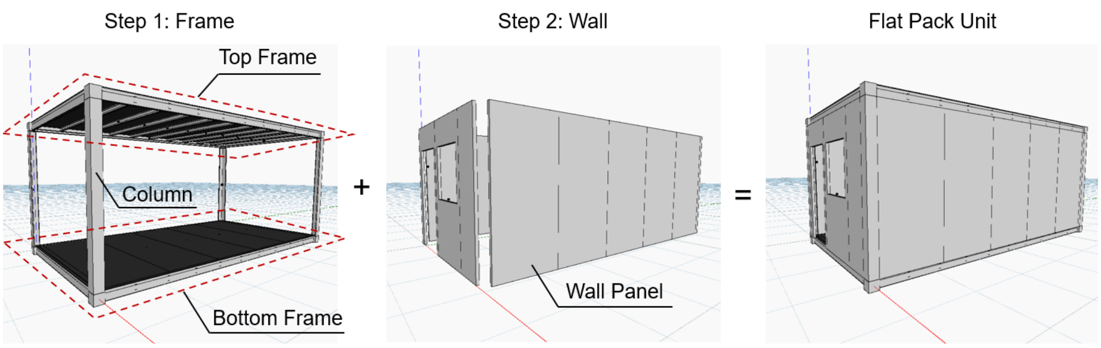


Figure 8. Centroids of the building components are plotted in Dynamo.

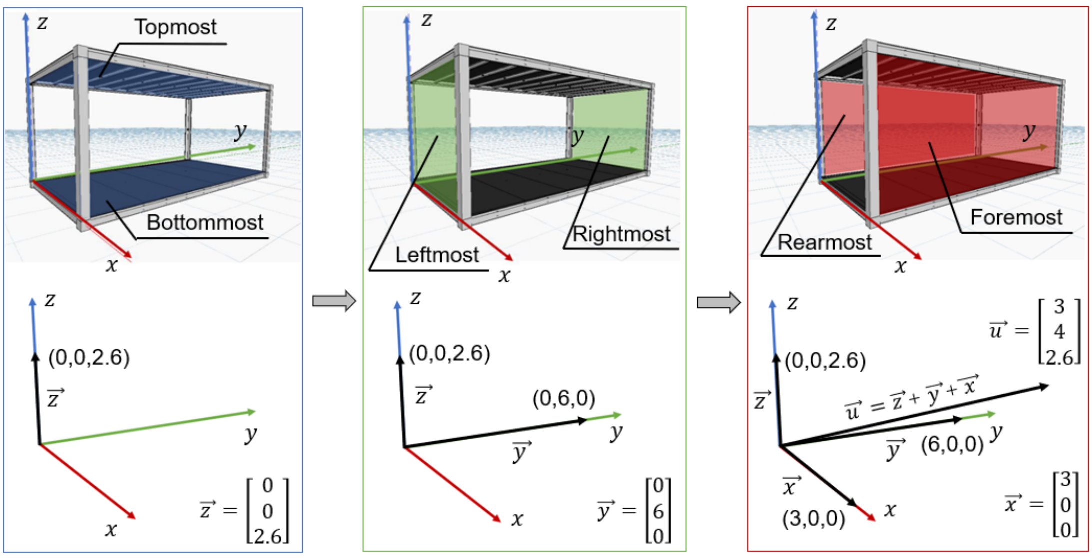
Section 3 of the algorithm concerns the creation of a reasonable assembly sequence for the building components. The procedure for constructing a flatpack house can be presented as a sequence of subtasks [2]: the frame is first assembled and then the wall panels are enclosed

(Figure 9a). The frame consists of bottom and top frames as transverse bearing constitution and columns as vertical supporting (Figure 9a), where the assembly sequence incorporates the bottom-to-top, left-to-right, and back-to-front processing (Figure 9b). The processing derives a vector that points along in sequence the z-axis, the y-axis, and the x-axis. Given that the size of the flatpack house is 6.0 meters long, 3.0 meters wide, and 2.6 meters tall [2], the vector $\vec{u} = \vec{z} + \vec{y} + \vec{x} = (0, 0, 2.6) + (0, 6, 0) + (3, 0, 0) = (3, 6, 2.6)$ (Figure 9b). The node *Assembly Sequence ()* in section 3 was defined to derive the vector $\vec{u}$ as introduced. When running the *Assembly Sequence ()* node, the vector $\vec{u}$ is applied to detecting in sequence the z-, y-, then x-coordinate values of the building components' centroids. This process first generates a list of building components arranged in ascending order along the z-axis (from the bottommost to the topmost). Then, for components that indicate the same size of z-coordinate values, the left-to-right, back-to-front procedure makes inferences to sort the components along the y-axis (from the leftmost to the rightmost), then the x-axis (from the rearmost to the foremost). Following the process, the assembly sequence for the frame is determined. For wall panels, the assembly sequence is determined in the same manner.

(a)



(b)



Bottom-to-top (along z-axis) Left-to-right (along y-axis) Back-to-front (along x-axis)

Figure 9. Assembly process of a flatpack house: (a) the frame and wall sequence; (b) the bottom-to-top, back-to-front, left-to-right processing.

The output of the *Assembly Sequence ()* node is: “result_frame = []; result_wall = []; OUT = result_frame + result_wall”. The “result_frame” paradigm contains the centroid coordinates of the frame components sorted in the assembly sequence order. The “result_wall” paradigm contains the centroid coordinates of the wall panels sorted in the assembly sequence order. The “OUT = result_frame + result_wall” paradigm indicates that the sortation of frame components comes before the listing of wall panels’ centroid coordinates. This is consistent with the aimed assembly sequence that the frame is first assembled and then the wall panels are enclosed. Subsequently, the outputs are supplemented to form an analysable data file for the ROS executions, which constitutes the information required for the robotic motion planning. This research used Industry Foundation Classes (IFC) as the interoperable data format between the task planning layer (in BIM) and the motion planning layer (in ROS).

The communication interface in Figure 10 was designed for data transmission between the task and motion planning layers, which is enabled by functional nodes at Dynamo, IFC and ROS terminals. First, the node *Export_IFC ()* at the Dynamo terminal takes the sorted assembly sequence list and exports it to an IFC file. Then, the node *IfcAxis2Placement3D ()* at IFC terminal organises data entries in a string form that can be parsed by the ROS system as follows: “#IFC Identifier = ifcPropertyStringValue(Parameter Label).placement(Parameter Content)” (see Figure 10). The node *Subscriber ()* at the ROS terminal is responsible for parsing the assembly coordinates and sequence data from the IFC tags. The robots' joint and trajectory parameters for performing the assembly of the prefabricated components will be generated based on the IFC data provided (see "Joint Interface" and "Trajectory Interface" in Figure 10). Finally, the node *Publisher ()* at the ROS terminal publishes the generated joint and trajectory parameters for controlling the robotic manipulator.

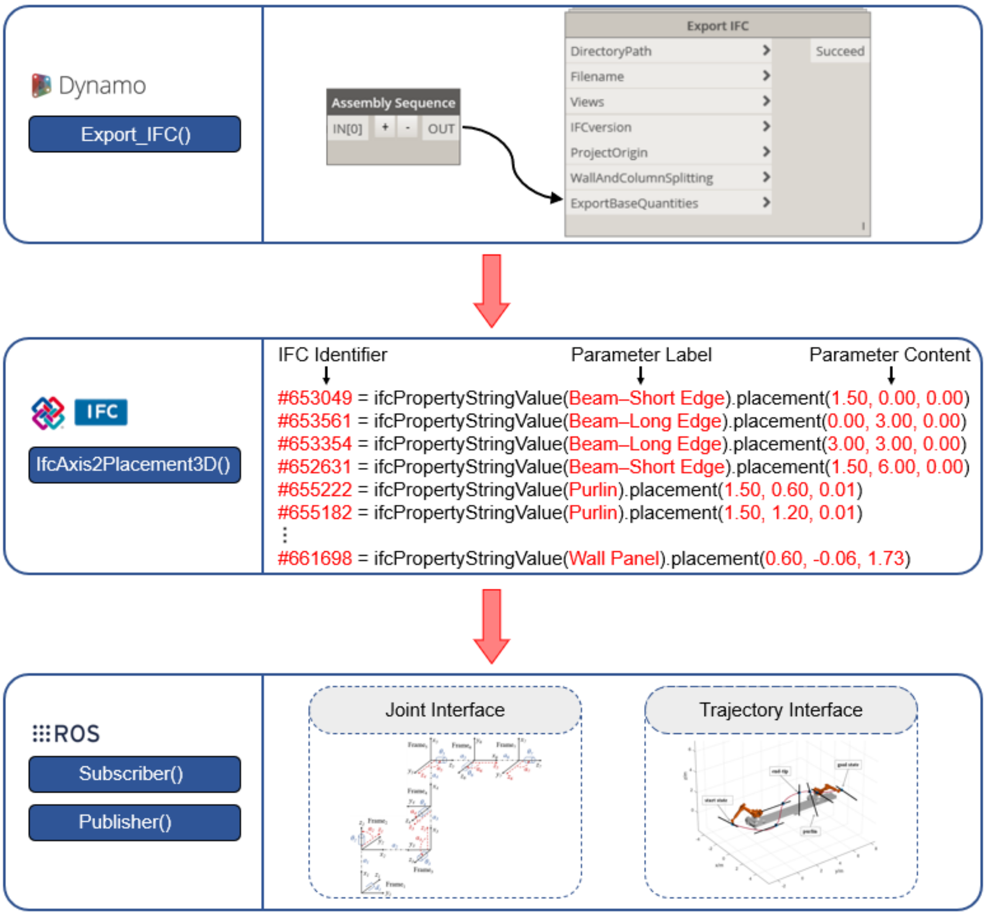


Figure 10. Communication interface for data transmission between the task and motion planning layers.

6. Motion planning algorithm: generating kinematic parameters while avoiding obstacles

The kinematic equation is fundamental to robotic motion planning, which can be used to compute values for the joints that achieve a desired position for the end-tip of a manipulator [34]. To derive the kinematic equation of the manipulator used in this research, reference frames, which are used to specify movements of each joint, were attached to the joints (as specified in

Figure 6) following the right-hand convention rules [38,39] (Figure 11).

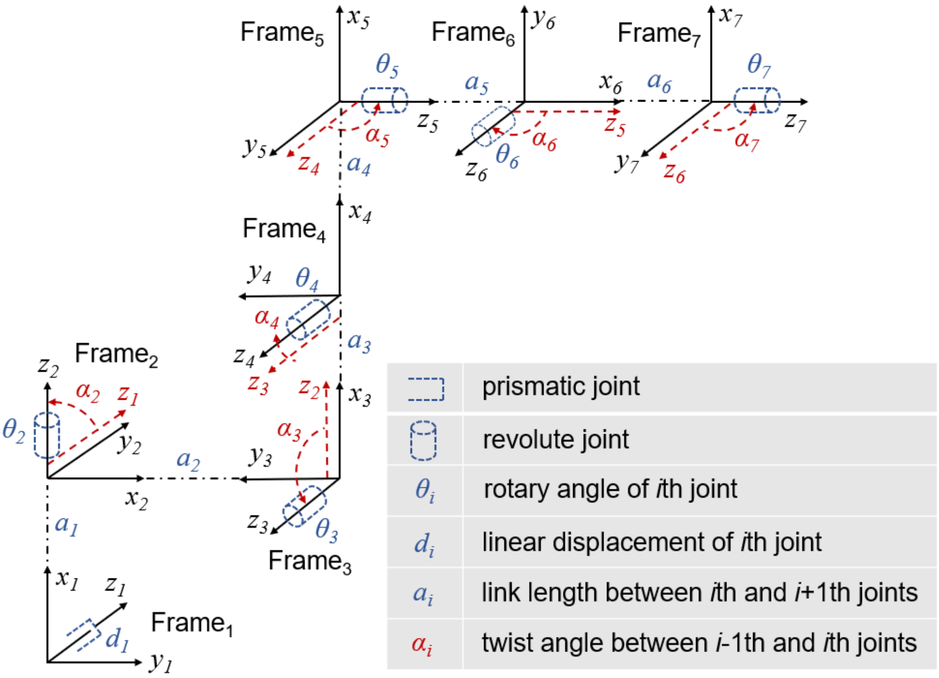


Figure 11. Reference frames are attached to the robotic manipulator.

The pose relationship (i.e., position and orientation) between two successive joints can be derived from attached frames using Denavit-Hartenberg (D-H) notation [40]. DH notation consists of four transformation parameters d_i, θ_i, a_i, and α_i (see Figure 11), which gives a standard methodology to write the kinematic equations of a robotic manipulator [40]. The parameter d_i notates the linear displacement along z-axis in the ith frame. The parameter θ_i notates the rotary angle around the z-axis in the ith frame. The parameter a_i notates the link length between ith and $i+1$th frames along the x-axis in the ith frame. The parameter α_i notates the twist angle between the z-axes in the $i-1$th and ith frames. For the 7 DOF manipulator, d_i and θ_i are variables altering as the joints operate, and a_i and α_i are constants reflecting the mechanical structure of the manipulator (e.g., link length) (see Table 2).

Table 2. D-H parameters of 7 DOF robotic manipulator.

ith frame (joint)	d_i (m)	θ_i (rad)	a_i (m)	α_i (rad)
1 (prismatic)	d_1	0	$a_1 = 1.334$	0
2 (revolute)	0	θ_2	$a_2 = 0.330$	$\alpha_2 = 1.570$
3 (revolute)	0	θ_3	$a_3 = 1.350$	$\alpha_3 = 1.570$
4 (revolute)	0	θ_4	$a_4 = 0.115$	$\alpha_4 = 1.570$

5 (revolute)	0	θ_5	$a_5 = 1.420$	$\alpha_5 = 1.570$
6 (revolute)	0	θ_6	$a_6 = 0.308$	$\alpha_6 = 1.570$
7 (revolute)	0	θ_7	0	$\alpha_7 = 1.570$

The transformations (i.e., d_i, θ_i, a_i, and α_i) along the serial frames form the kinematic equation (T), which were derived by multiplying the homogeneous transformation matrices of d_i, θ_i, a_i, and α_i [40]:

$$T = \prod_{i=1}^{7} \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where T is the manipulator end-effector's Cartesian coordinate; d_i, θ_i, a_i, and α_i are associated with each joint's reference frame system.

In construction sites, motion planning methods can help the robotic manipulator autonomously and safely manoeuvre around assembled parts. Based on the kinematic equation (T) derived, a motion planning algorithm was applied. Considering the trade-off between computational time and path quality, the authors used a redefined sampling-based motion planning algorithm—Rapidly Exploring Random Tree Star (RRT*) [41].

Given χ as the 3-dimensional configuration space for our motion planning problem and χ_{obs} as the known obstacle space, the collision-free space can be calculated by $\chi_{free} = \chi \backslash \chi_{obs}$. The start state x_{start} (i.e., the pick-up coordinate), the goal state x_{goal} (i.e., the assembly coordinate), the 3-dimensional configuration space χ and the known obstacle space χ_{obs} are required as inputs in this algorithm.

The detailed process of the RRT* algorithm is presented in Algorithm 2 and explained below:

Algorithm 2: Rapidly Exploring Random Tree Star (RRT*) (χ, χ_{obs}, x_{start}, x_{goal}, N)

$V \leftarrow x_{start}, x_{goal}$;

$E \leftarrow \emptyset$;

for $i = 1, 2, \ldots, N$ do

  $x_{rand} \leftarrow$ Sample(χ_{free}, N);

  $x_{nearest} \leftarrow$ Nearest(V, x_{rand});

$x_{new} \leftarrow$ Steer($x_{nearest}$, x_{rand}, d_{step});

if CollisionFree($x_{nearest}$, x_{new}, χ_{obs}) then

$X_{near} \leftarrow$ Near(V, x_{new}, r);

$\eta \leftarrow$ Line($x_{nearest}$, x_{new}, s);

(x_{min}, x_{new}) $\leftarrow$ Parent(x_{new}, X_{near}, η);

$V \leftarrow V \cup \{x_{new}\}$;

$E \leftarrow E \cup \{(x_{min}, x_{new})\}$;

$G \leftarrow$ Rewire (G, X_{near}, x_{new});

end

end

return $G(V, E)$;

- First, the function Sample(χ_{free}) generates a random state x_{rand} inside the collision-free space χ_{free} based on a uniform distribution. Specifically, a uniform distribution means that the probability of the potential position of x_{rand} at any point within χ_{free} is equal;
- Second, a comparison between the randomly sampled state x_{rand} and the rest states in the set of nodes V is performed to find the nearest state $x_{nearest}$ to x_{rand};
- Third, the function Steer($x_{nearest}$, x_{rand}, d_{step}) generates a new state x_{new} that is closer to $x_{nearest}$ by connecting x_{rand} and $x_{nearest}$ with a steering function;
- Fourth, the function CollisionFree($x_{nearest}$, x_{new}, χ_{obs}) checks if there is any collision between the straight path from x_{new} to $x_{nearest}$ and the known obstacle space χ_{obs};
- Fifth, if there is no collision found, the function Near(V, x_{new}, r) collects a set of states, which locates within a spherical space that uses x_{new} as the centre and a predefined parameter r as the radius;
- Sixth, the function Line($x_{nearest}$, x_{new}, s) connects x_{new} and $x_{nearest}$ with a straight line. The length of the straight line is equal to another predefined parameter step size s;
- Seventh, the function Parent(x_{new}, X_{near}, η) selects the state with the minimum cost-to-go c_{min} from the set X_{near} as the parent state x_{min};

- Eighth, the new state x_{new} is added to the set of nodes V, and the new edges that connect x_{min} and x_{new} is added to the set of edges E;
- Ninth, the function Rewire(G, X_{near}, x_{new}) keeps adding or removing some edges between x_{new} and the states in X_{near} to ensure the path is optimised and has a minimum cost;
- Last, this algorithm computes the global graph $G(V, E)$, where the optimised global path $\theta = [x_1, \cdots x_N]$ is embedded within this global graph after repeating N times.

Overall, RRT* keeps sampling random nodes x_{rand} within the collision-free space χ_{free}, and then reviews the global graph $G(V, E)$ through measuring the potential cost-to-go to every node $x \in V$, which locates within the spherical space near the newly sampling node x_{new}. An example of applying RRT* to perform motion planning for a robotic manipulator to assemble the main beam (short edge) is illustrated in Figure 12. Figures 10a and 10b show a simulated construction environment, where the two robotic manipulators are simultaneously assembling the main beams (short edge). In Figure 12a, the end-tip of the manipulators are at their start points to pick up the prefabricated beams from the trolleys. In Figure 12b, the end-tip of the manipulators are at their goal points to place the prefabricated beams at the designated location. Figures 10c and 10d show the corresponding motion planning problem solved by RRT* in ROS Rviz. As can be seen, the start point is represented in green, the goal point is represented in red, the obstacle space is represented in grey, tree branches of the RRT* algorithm are represented in orange, and the resulting optimised assembly path is represented in blue. Note that each manipulator's base coordinate system is marked as a world coordinate system, and the prefabricated components' assembly coordinates as determined via the task planning algorithm are unified into the manipulator's base coordinate system for motion planning.

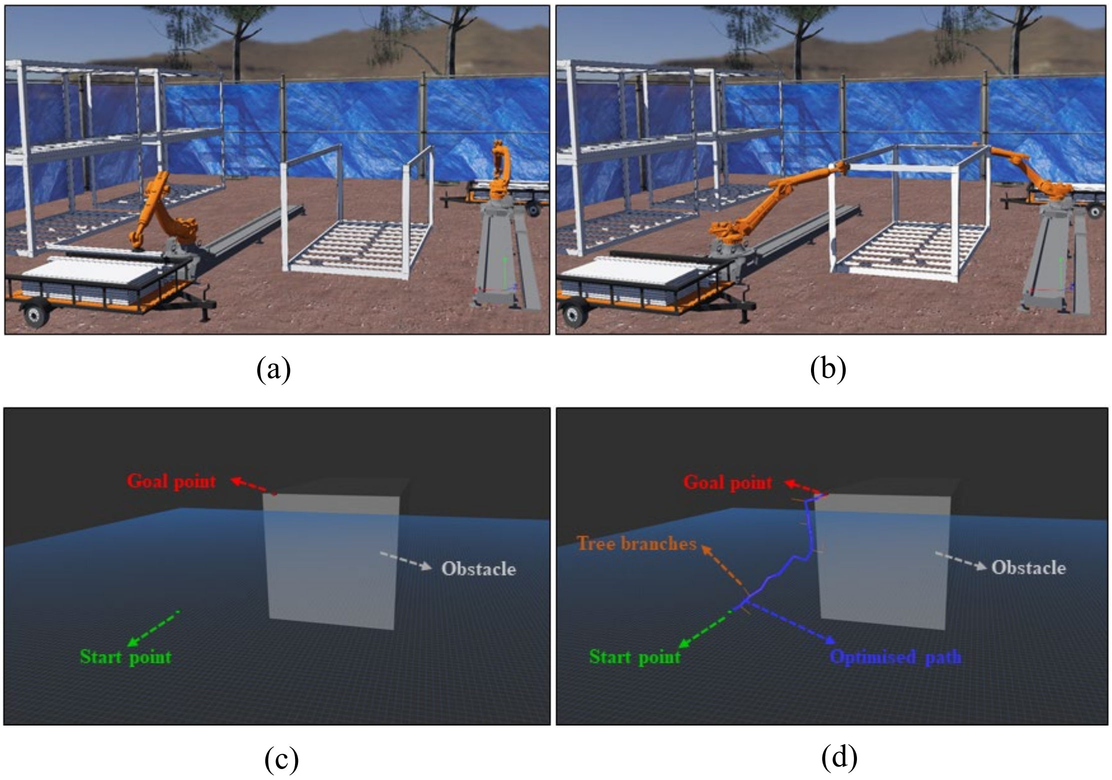


Figure 12. Using robotic manipulators to perform an assembly task in a simulated construction site: (a) and (b) show a simulated construction environment, where the two robotic manipulators are simultaneously assembling the main beams (short edge); (c) and (d) show the corresponding motion planning problem solved by RRT* in ROS Rviz.

7. Testing of the prototype

Tests were performed to evaluate the developed prototype in terms of 1) reasonableness of assembly sequence determined for a given flatpack house BIM model, 2) reachability for the assembly coordinates of prefabricated components, and 3) capability to avoid obstacles.

7.1 Reasonableness of assembly sequence

The authors recreated the standard flatpack house unit in Autodesk Revit, and applied the ACASD algorithm of the prototype to locate prefabricated components in the Revit model's reference frame and determine the assembly coordinates and sequence for each component. The result is provided in Table 3 below. As can be seen, the ACASD algorithm can create a reasonable assembly sequence for the flatpack house. First, the frame is assembled (components 1-44, Table 3), and then the wall panels are enclosed (components 45-62, Table 3). The frame consists of bottom and top frames as the transverse bearing constitution and columns as vertical supporting, where the sequence implies a bottom-to-top, left-to-right, and back-to-front assembly logic. This is reflected in Table 3: components 1-20 for the bottom frame, components 21-24 for the column, and components 25-44 for the top frame.

Table 3. Assembly sequence and coordinates of the prefabricated building components.

Sequence	Components	Coordinates	Sequence	Components	Coordinates
1	Beam (Short Edge)	(1.50, 0.00, 0.00)	32	Purlin	(1.50, 2.40, 2.61)
2	Beam (Long Edge)	(0.00, 3.00, 0.00)	33	Purlin	(1.50, 3.00, 2.61)
3	Beam (Long Edge)	(3.00, 3.00, 0.00)	34	Purlin	(1.50, 3.60, 2.61)
4	Beam (Short Edge)	(1.50, 6.00, 0.00)	35	Purlin	(1.50, 4.20, 2.61)
5	Purlin	(1.50, 0.60, 0.01)	36	Purlin	(1.50, 4.80, 2.61)
6	Purlin	(1.50, 1.20, 0.01)	37	Purlin	(1.50, 5.40, 2.61)
7	Purlin	(1.50, 1.80, 0.01)	38	Roof Panel	(1.50, 0.05, 2.66)
8	Purlin	(1.50, 2.40, 0.01)	39	Roof Panel	(1.50, 0.70, 2.66)
9	Purlin	(1.50, 3.00, 0.01)	40	Roof Panel	(1.50, 1.85, 2.66)
10	Purlin	(1.50, 3.60, 0.01)	41	Roof Panel	(1.50, 3.00, 2.66)
11	Purlin	(1.50, 4.20, 0.01)	42	Roof Panel	(1.50, 4.15, 2.66)
12	Purlin	(1.50, 4.80, 0.01)	43	Roof Panel	(1.50, 5.30, 2.66)
13	Purlin	(1.50, 5.40, 0.01)	44	Roof Panel	(1.50, 5.95, 2.66)
14	Floor Panel	(1.50, 0.05, 0.06)	45	Wall Panel	(1.80, -0.06, 1.17)
15	Floor Panel	(1.50, 0.70, 0.06)	46	Wall Panel	(1.83, 6.05, 1.17)
16	Floor Panel	(1.50, 1.85, 0.06)	47	Wall Panel	(2.65, -0.06, 1.30)
17	Floor Panel	(1.50, 3.00, 0.06)	48	Wall Panel	(3.05, 0.65, 1.30)
18	Floor Panel	(1.50, 4.15, 0.06)	49	Wall Panel	(-0.05, 0.66, 1.30)
19	Floor Panel	(1.50, 5.30, 0.06)	50	Wall Panel	(3.05, 1.80, 1.30)
20	Floor Panel	(1.50, 5.95, 0.06)	51	Wall Panel	(-0.05, 1.81, 1.30)
21	Column	(-0.03, -0.03, 1.30)	52	Wall Panel	(3.05, 2.95, 1.30)
22	Column	(3.03, -0.03, 1.30)	53	Wall Panel	(-0.05, 2.96, 1.30)
23	Column	(-0.03, 6.03, 1.30)	54	Wall Panel	(3.05, 4.10, 1.30)
24	Column	(3.03, 6.03, 1.30)	55	Wall Panel	(-0.05, 4.11, 1.30)
25	Beam (Short Edge)	(1.50, 0.00, 2.60)	56	Wall Panel	(3.05, 5.25, 1.30)
26	Beam (Long Edge)	(0.00, 3.00, 2.60)	57	Wall Panel	(-0.05, 5.26, 1.30)
27	Beam (Long Edge)	(3.00, 3.00, 2.60)	58	Wall Panel	(3.05, 5.87, 1.30)
28	Beam (Short Edge)	(1.50, 6.00, 2.60)	59	Wall Panel	(-0.05, 5.88, 1.30)
29	Purlin	(1.50, 0.60, 2.61)	60	Wall Panel	(0.66, 6.05, 1.30)
30	Purlin	(1.50, 1.20, 2.61)	61	Wall Panel	(2.65, 6.05, 1.30)
31	Purlin	(1.50, 1.80, 2.61)	62	Wall Panel	(0.60, -0.06, 1.73)

7.2 Reachability for the assembly coordinates

The robotic manipulator used in this research is composed of one prismatic joint and six revolute joints (Figure 11). The relationship between the manipulator joint coordinates and the end-effector's Cartesian coordinate is given by the kinematic equation (T) (1) as derived above. Thus, the problem of whether a given assembly coordinate is kinematically reachable for the robotic manipulator can be solved by formulating the following equation:

$$\begin{bmatrix} d_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{bmatrix} = T^{-1}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \qquad (2)$$

where (d_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7) are the manipulator joint variables, which are subject to the following motion constraints as mentioned in section 4: $0 \leq d_1 \leq 8.500\ m$; $-3.227 \leq \theta_2 \leq 3.227$ rad; $-1.483 \leq \theta_3 \leq 0.872$ rad; $-1.361 \leq \theta_4 \leq 2.093$ rad; $-6.106 \leq \theta_5 \leq 6.106$ rad; $-2.181 \leq \theta_6 \leq 2.181$ rad; $-6.106 \leq \theta_7 \leq 6.106$ rad ; T^{-1} is the inverse operation of the kinematic equation (T) (1); (x, y, z) is a given assembly coordinate. Note that the assembly coordinates as determined via the task planning algorithm are the geometric centroids of the prefabricated components. The Cartesian coordinate of the manipulator end-effector represents the position where a component is adsorbed by the vacuum gripper. To provide the appropriate (x, y, z) input for equation (2), the assembly coordinates are converted to the Cartesian coordinate of the manipulator end-effector based on the geometric features of the prefabricated components.

Equation (2) is to find a set of (d_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7) which satisfies a given assembly coordinate (x, y, z). If a solution can be found, the coordinate (x, y, z) is reachable for the manipulator. The 62 assembly coordinates in Table 3 were tested given equation (2) and the motion range of each joint. The results indicated that using dual robotic manipulators KUKA KR 120 R3100 (see Figure 5b) enlarges the workspace and can fully cover the spatial extent of the flatpack house for assembly, where solutions exist for the 62 assembly coordinates. Figure 13 showcases eight examples of assembly coordinates reached by robotic manipulators in a simulated construction environment. The corresponding solutions for Figure 13 examples are provided in Table 4.

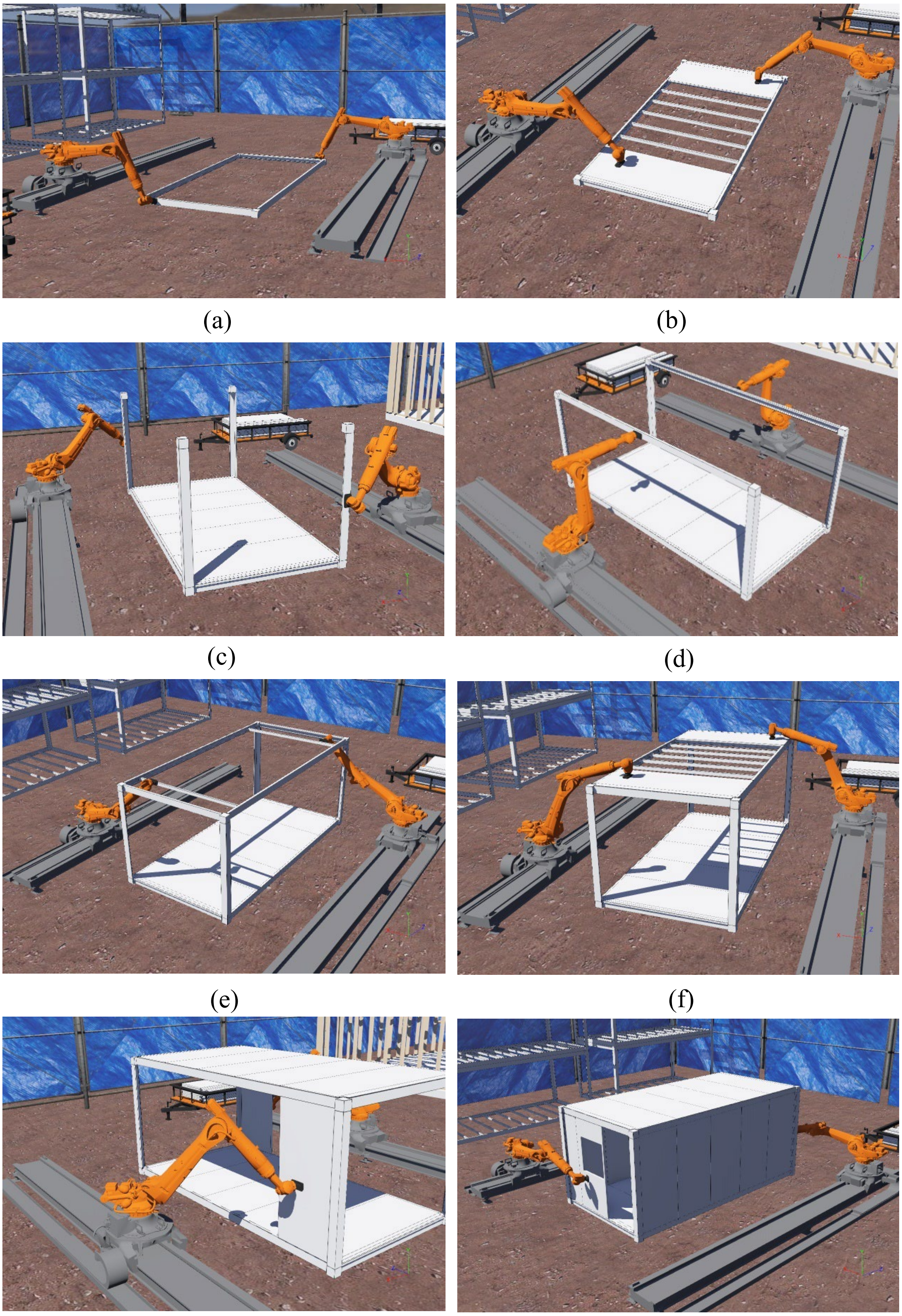


Figure 13. Examples of assembly coordinates reached by robotic manipulators in a simulated

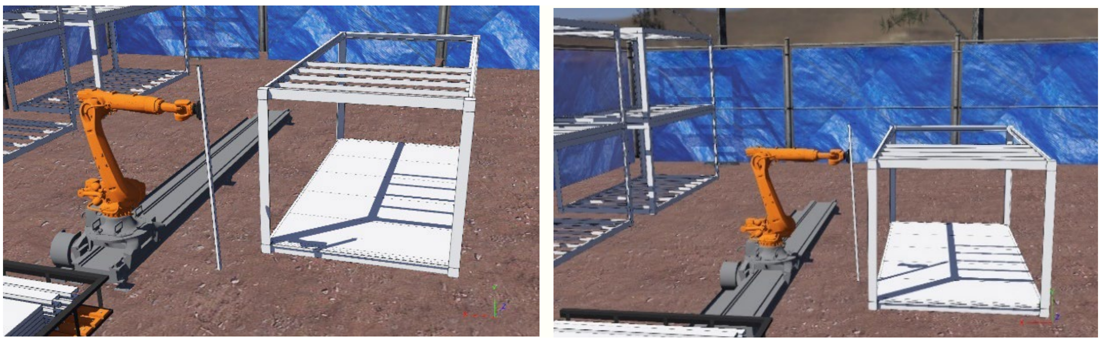
construction environment: (a) bottom frame main beam (short edge); (b) bottom frame floor panel; (c) column; (d) top frame main beam (long edge); (e) top frame purlin; (f) top frame roof panel; (g) wall panel (long edge); (h) wall panel (short edge).

Table 4. Corresponding solutions for Figure 13 examples.

Solution	d_1 (m)	θ_2 (rad)	θ_3 (rad)	θ_4 (rad)	θ_5 (rad)	θ_6 (rad)	θ_7 (rad)
Figure 13a	0.909	0.000	-1.483	0.371	-1.600	1.234	-2.700
Figure 13b	1.950	0.000	-1.483	0.705	0.000	0.763	0.000
Figure 13c	6.239	0.572	-0.862	0.183	-1.954	-2.006	0.765
Figure 13d	4.318	0.010	-0.028	-0.013	0.232	-0.043	0.232
Figure 13e	2.000	-0.174	-1.232	1.907	-1.681	1.706	-0.889
Figure 13f	1.999	0.000	-0.499	0.785	0.000	1.857	0.000
Figure 13g	3.455	-0.671	-0.754	-0.113	0.805	-1.040	0.485
Figure 13h	0.922	0.000	-1.483	1.174	4.674	1.128	4.394

7.3 Capability to avoid obstacles

Testing the developed prototype in the simulated construction environment also finds that it has satisfactory obstacle avoidance performance. The motion planning RRT* algorithm successfully recognised the robotic manipulator and already-in-place prefabricated components as obstacles and generated a series of optimised robotic motions to avoid the obstacles. Figure 14 presents an example of robotic motion optimisation for top frame purlin assembly. The optimisation routine as illustrated sequentially in the sub-figures is interpreted in the figure caption.



(a) (b)

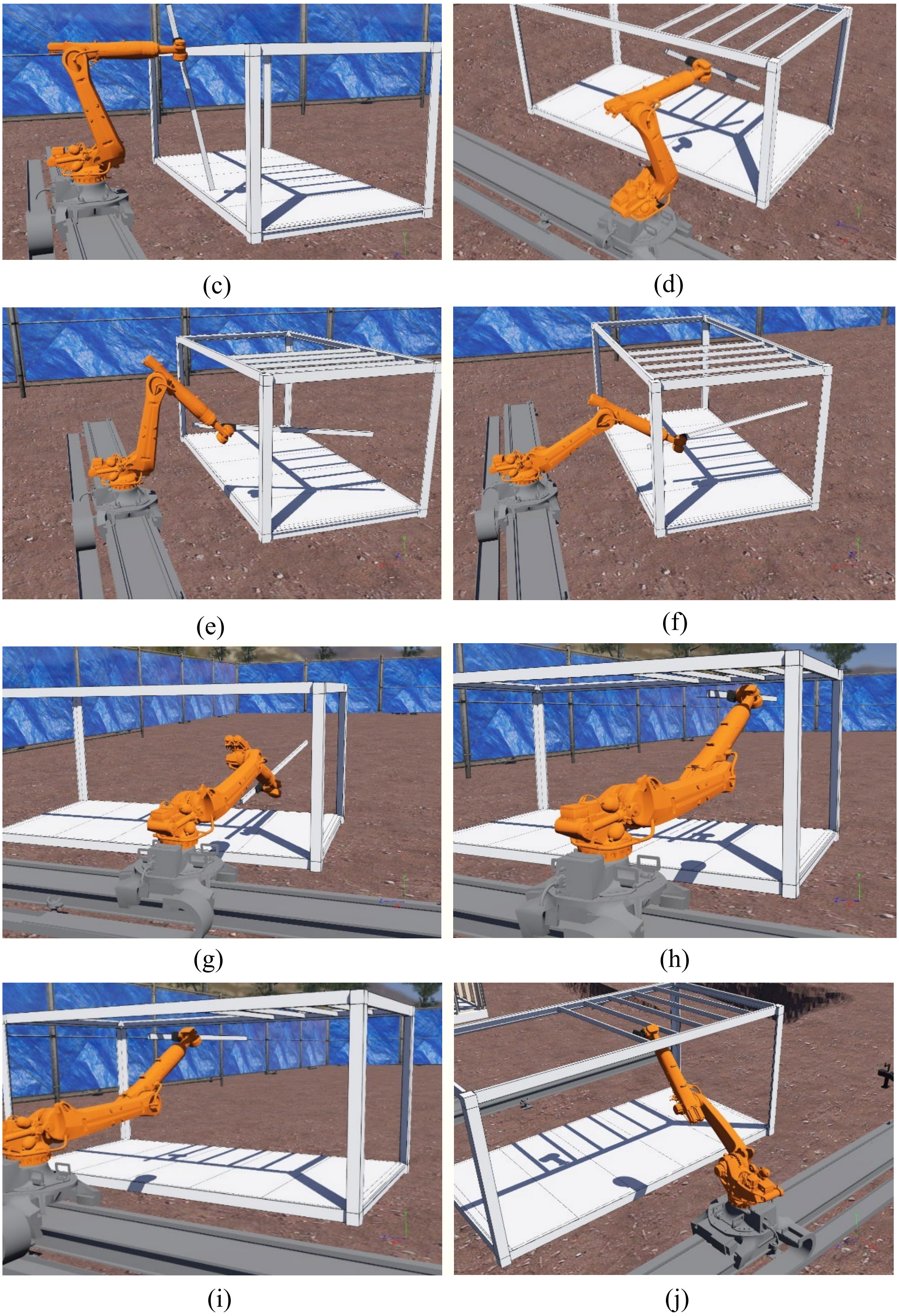


Figure 14. An example of robotic motion optimisation for top frame purlin assembly: (a) and (b) show that the robotic manipulator hoists vertically the purlin and moves forward along joint 1 axis while having its joint 3 rotate backward and joint 4 rotate forward to keep the end-tip (as

well as the purlin) in a distance from the obstacle (i.e., already-in-place flatpack house unit); (c), (d), (e), (f), and (g) show that joints 3, 4, 5, 6, and 7 of the manipulator adjust their respective angles cooperatively to achieve the pose in (h) and in the meantime avoid the purlin-obstacle collision; (h) and (i) show that the robotic manipulator hoists horizontally the purlin and moves forward along joint 1 axis to approach beneath the assembly coordinate of the purlin; (j) shows that joints 4, 5, 6, and 7 of the manipulator adjust their respective angles cooperatively to place the purlin at the designated assembly coordinate.

8. Discussion

In scrutinising the scientific question as proposed, the findings suggest that the question has been answered in this research. The robotic prototype was developed to reflect the construction characteristics and difficulties of COVID-19 hospitalisation facilities, which consists of a task planning algorithm and a motion planning algorithm that can respectively: 1) derive a vector that can determine the mathematical relationship between coordinates of prefabricated components and assembly sequence, with the consideration of geometry and centroid, for robotic construction; and 2) analyse the determined assembly sequence and generate robots' kinematic parameters for performing the assembly of COVID-19 hospitalisation facilities autonomously without human intervention.

As presented in section 7, the developed prototype was tested in three aspects: 1) determination of the assembly sequence, 2) reachability for the assembly coordinates of prefabricated components, and 3) capability to avoid obstacles. To quantitatively evaluate the assembly sequence determination performance, the coordinates of each building component in the sorted sequence list (Table 3) were scrutinised (section 7.1). As can be seen, the coordinates were arranged in ascending order along the z-axis (from the bottommost to the topmost), the y-axis (from the leftmost to the rightmost), then the x-axis (from the rearmost to the foremost). This implies a reasonable bottom-to-top, left-to-right, and back-to-front assembly logic. To quantitatively evaluate whether the assembly coordinates of each component are reachable, the kinematic analysis on the robotic platform was performed (section 7.2). The results showed that the robot joint solution (d_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7) existed for all the assembly coordinates of the 62 prefabricated components. The authors further ran simulations to test whether the joint solutions produce weird robot poses. The results indicated that the robot configurations for all the assembly coordinates were reasonable and no weird poses were observed. Eight showcase examples are provided in Figure 13. In addition, the collision avoidance capability of the developed prototype was tested (section 7.3). The motion planning RRT* algorithm successfully recognised the robotic manipulator and already-in-place prefabricated components as obstacles and generated a series of optimised robotic motions to avoid the obstacles (see an example presented in Figure 14).

The original innovation of this research is to provide the following three outcomes for the research community:

1) The Assembly Coordinates and Sequence Determination (ACASD) algorithm (open source: https://github.com/yifanrepo/ACASD). In the existing studies [8–11], the assembly coordinates of prefabricated building components were predetermined for the robotic assembly. However, the function of determining the assembly sequence of the prefabricated building components was not considered in their approaches. The ACASD algorithm can be used to determine both the assembly coordinates and sequence for a given flatpack house BIM model.
2) The seven degree-of-freedom (DOF) robotic manipulator kinematic equation. The robotic manipulators used in the previous studies [8–11] in this field only have six revolute joints (i.e., six DOF) and the bases of their manipulators are fixed. In this research, the manipulator has an additional prismatic joint based on the six revolute joints (i.e., seven DOF), which enables the base of the manipulator to move along a linear track to attain a higher range of workspace. The seven DOF equation developed in this research incorporates the manipulator base's moving capability in the kinematic design, which is more suitable for construction-related scenarios.
3) The virtual environment for simulating the assembly of flatpack house using robotic manipulators (open source: https://github.com/yifanrepo/virtual-construction). The authors rigorously modelled the environment to reflect physical effects (e.g., gravity) and workplace resources (e.g., machinery, prefabricated building components, and workers).

In a real project, the robotic prototype developed in this research can be set up as the digital representation of the project and uses its self-contained task and motion planning algorithms to generate useful data input for instructing the robotic assembly. Once the robotic manipulators are set up in the real world, the ROS environment of the prototype can be used to recreate the project's real-world situations (Figure 15). This is enabled by using 3D modelling to create digital companions for the on-site physical objects, which provides a way to project the workplace settings into the prototype's digital world. Therefore, the digital environment forms a one-to-one correspondence mapping of the physical objects' shape, texture, location, and motion (see Figure 15). This ensures that the properties of the physical objects can be well transferred to their digital counterparts and the virtual representation of the workplace is efficient for spatial reasoning and motion planning. Specifically, the workflow of our prototype is as follows. First, the ROS terminal receives the assembly sequence and coordinates data from BIM, and marks the pick-up and assembly locations of the prefabricated building components in its spatial reference system (see Figure 15). The pick-up and assembly locations represent the start and goal states for motion planning respectively. Then, the RRT* algorithm generates the manipulators' joint parameters for performing the assembly of the flatpack house based on the marked start and goal states in the ROS environment, and publishes joint control signals to the digital processors of the robotic manipulators in the real world. In this situation, the end-tips of the manipulators will be driven by the signals to pick up the prefabricated components from the trolleys and follow a pre-determined sequence to transfer each component to the assembly coordinates (see Figure 15). Note that as electrical outlets are not always available in the exact location where the manipulators are positioned on-site, extension cords can be used to reach the manipulators' locations and supply the necessary power to get the assembly job done. When the assembly of a flatpack house unit completes, a mobile crane is used to lift the unit from the assembling area and install the unit into the hoisting area (see Figure 15).

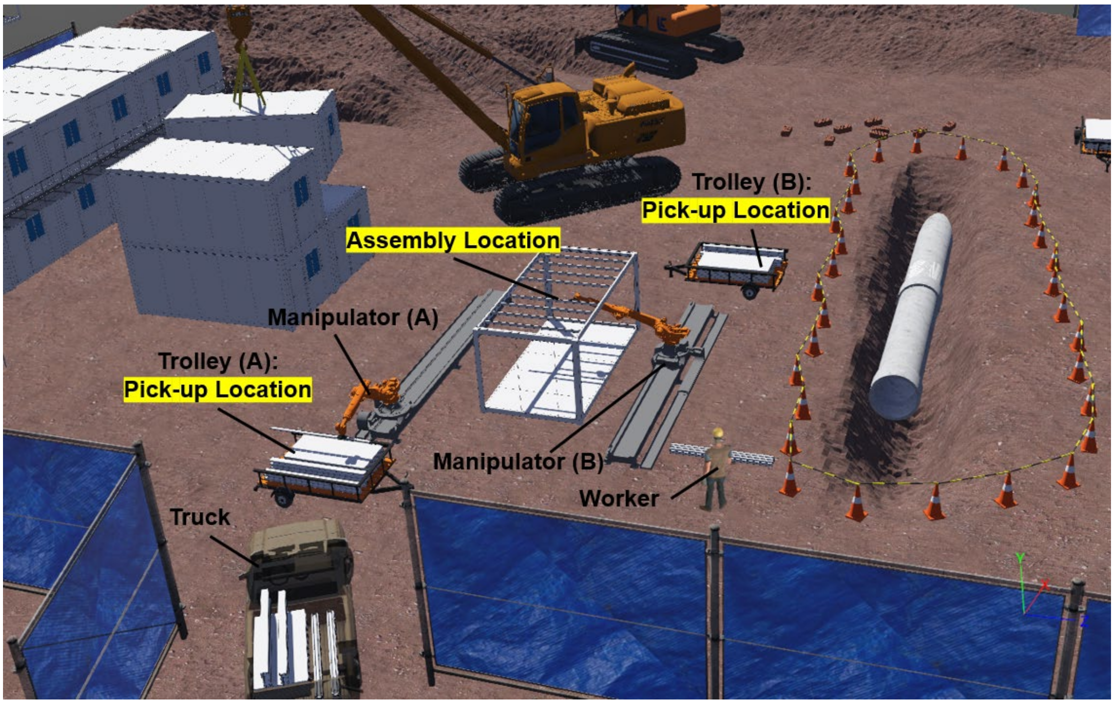


Figure 15. The robotic platform for constructing prefabricated hospitalisation facilities.

The practical value of our robotic prototype is twofold:

1) Once the robotic manipulators are set up in a real project, the prototype can publish useful joint control signals to the digital processors of the manipulators for the assembly of COVID-19 hospitalisation facilities.
2) In a real project, the robotic manipulators can replace human labour in the assembly of flatpack house, where one worker is needed to operate the data transmission inside the robotic prototype. In this case, the number of workers required in the construction procedure can be significantly reduced compared with the traditional method, which contributes to the mitigation of COVID-19 spread on construction sites.

The authors acknowledge that although our prototype was tested in a ROS environment rather than the actual implementation, the environment was rigorously modelled to provide a convincing experimental condition. First, great care was taken to make the physical properties of the environment as close to that of the real world, which consisted of density, gravity, damping, dimension, and material colour and texture. Second, in a real project, the on-site construction resources can be set up in one-to-one correspondence with the settings in the ROS environment (see Figure 15) (e.g., the number and pick-up locations of the prefabricated building components). This ensures that the properties of the digital objects in ROS can be well transferred to their physical counterparts in the real world and the virtual representation of the workplace is efficient for spatial reasoning and motion planning. Third, the environment included an accurate kinematic representation of the robot so that the planned motions as verified in the environment can be well transferred to the reality for assembling prefabricated hospitalisation facilities. The authors carried out a pilot trial in the lab to investigate whether the verified robot motions are achievable in the real world. The results showed that the joint motions and end-tip outreaches as planned and verified for the robot in ROS could be achieved in the reality. Figure 16 shows an example of reaching a predetermined end-tip coordinate for assembling the purlin component in the lab setting. As can be seen, the end-tip coordinate is (1.400, 0.300, 1.100) in the manipulator’s base coordinate system, which could be achieved with joint parameters (1.400 m, -0.671 rad, -0.824 rad, 1.013 rad, 0.625 rad, -1.040 rad, 0.485 rad).

end-tip coordinate
(1.400, 0.300, 1.100)

joint	parameter
joint 1	1.400 m
joint 2	-0.671 rad
joint 3	-0.824 rad
joint 4	1.013 rad
joint 5	0.625 rad
joint 6	-1.040 rad
joint 7	0.485 rad

Figure 16. Pilot trial in the lab.

9. Conclusion and future work

This research presents a BIM-based prototype for robotic assembly of the standard unit of COVID-19 hospitalisation facilities—flatpack house—with prefabricated components. The development of the prototype consisted of a task planning algorithm and a motion planning algorithm. The task planning algorithm—Assembly Coordinates and Sequence Determination (ACASD)—is designed to utilise the spatial information contained in a BIM model to locate the assembly coordinates for the prefabricated components. Then the ACASD algorithm determines the assembly sequence by following a bottom-to-top, left-to-right, and back-to-front logic according to the relative positions of the prefabricated components in the BIM model. The motion planning algorithm—Rapidly Exploring Random Tree Star (RRT*)—incorporates the manipulator base's moving capability in the kinematic analysis, and regards the manipulator and already-in-place prefabricated components as obstacles to generate a series of optimised robotic motions.

Different types of tests were performed to assess the developed prototype and the corresponding results demonstrated that the prototype has satisfactory performance in all the tests. First, the prototype can create a reasonable assembly sequence for the flatpack house. Second, using dual robotic manipulators KUKA KR 120 R3100 enlarges the workspace and can fully cover the spatial extent of the flatpack house unit for assembly, where the assembly coordinates of the 62 prefabricated components are kinematically reachable. Third, the motion planning algorithm successfully recognised the robotic manipulator and already-in-place prefabricated components as obstacles and generated a series of optimised robotic motions to avoid the obstacles.

Overall, this research highlights the significance of using robotic technologies to deliver construction projects under pandemic circumstances, and provides a prototype that can be used to generate reasonable task and motion planning for robotic assembly of COVID-19 hospitalisation units—flatpack house. Meanwhile, it is potentially useful for other emergency cases that utilise the flatpack house as the standard unit (e.g., earthquake, deluge), which further extends the generalisability of the research outcome. On the other hand, the present research contributed to the existing literature in addition to the mentioned practical implications. More specifically, the developed prototype fills in the following knowledge gaps: 1) determining the assembly sequence of building components, and 2) generating robots' kinematic parameters for performing the assembly of COVID-19 hospitalisation facilities, while incorporating the robot base's moving capability in the kinematic design to attain a reasonable range of workspace.

However, this research has the following limitation and subsequent research needs to be conducted shortly. This research aimed at the assembly steps of determining the coordinates and sequence for the prefabricated components, and using robotic manipulators to place the components at the designated location in the designated order. There is a further auxiliary procedure involved, which is to bolt the prefabricated components. This paper is a part of an ongoing research project. In the subsequent research, the authors will investigate further the use of a collaborative robot (e.g., aerial operation robot) to assist in screwing bolt connections after the components are placed at the designated locations.

Acknowledgement

This work was partially supported by China's National Natural Science Foundation under Grant No. (52108179).

References

[1] A.M. Anter, D. Oliva, A. Thakare, Z. Zhang, AFCM-LSMA: New intelligent model based on Lévy slime mould algorithm and adaptive fuzzy C-means for identification of COVID-19 infection from chest X-ray images, Advanced Engineering Informatics. 49 (2021) 101317. https://doi.org/10.1016/j.aei.2021.101317.

[2] H. Luo, J. Liu, C. Li, K. Chen, M. Zhang, Ultra-rapid delivery of specialty field hospitals to combat COVID-19: Lessons learned from the Leishenshan Hospital project in Wuhan, Automation in Construction. 119 (2020) 103345. https://doi.org/10.1016/j.autcon.2020.103345.

[3] L.-K. Chen, R.-P. Yuan, X.-J. Ji, X.-Y. Lu, J. Xiao, J.-B. Tao, X. Kang, X. Li, Z.-H. He, S. Quan, L.-Z. Jiang, Modular composite building in urgent emergency engineering projects: A case study of accelerated design and construction of Wuhan Thunder God Mountain/Leishenshan hospital to COVID-19 pandemic, Automation in Construction. 124 (2021) 103555. https://doi.org/10.1016/j.autcon.2021.103555.

[4] V. Bushell, L. Thomas, J. Combes, Inside The O2: the NHS Nightingale Hospital London education center, Journal of Interprofessional Care. 34 (2020) pp.698-701. https://doi.org/10.1080/13561820.2020.1823949.

[5] R. Alizadehsani, Z. Alizadeh Sani, M. Behjati, Z. Roshanzamir, S. Hussain, N. Abedini, F. Hasanzadeh, A. Khosravi, A. Shoeibi, M. Roshanzamir, P. Moradnejad, S. Nahavandi,

F. Khozeimeh, A. Zare, M. Panahiazar, U.R. Acharya, S.M.S. Islam, Risk factors prediction, clinical outcomes, and mortality in COVID-19 patients, Journal of Medical Virology. 93 (2021) pp.2307-2320. https://doi.org/10.1002/jmv.26699.

[6] A. Alsharef, S. Banerjee, S.M. Uddin, A. Albert, E. Jaselskis, Early Impacts of the COVID-19 Pandemic on the United States Construction Industry, International Journal of Environmental Research and Public Health. 18 (2021) 1559. https://doi.org/10.3390/ijerph18041559.

[7] K. Simonson, Construction Data, The Associated General Contractors of America. (2021). https://www.agc.org/learn/construction-data (accessed August 21, 2021).

[8] Y. Terada, S. Murata, Automatic Modular Assembly System and its Distributed Control, The International Journal of Robotics Research. 27 (2008) pp.445-462. https://doi.org/10.1177/0278364907085562.

[9] J. Willmann, M. Knauss, T. Bonwetsch, A.A. Apolinarska, F. Gramazio, M. Kohler, Robotic timber construction — Expanding additive fabrication to new dimensions, Automation in Construction. 61 (2016) pp.16-23. https://doi.org/10.1016/j.autcon.2015.09.011.

[10] L. Ding, W. Jiang, Y. Zhou, C. Zhou, S. Liu, BIM-based task-level planning for robotic brick assembly through image-based 3D modeling, Advanced Engineering Informatics. 43 (2020) 100993. https://doi.org/10.1016/j.aei.2019.100993.

[11] N. King, M. Bechthold, A. Kane, P. Michalatos, Robotic tile placement: Tools, techniques and feasibility, Automation in Construction. 39 (2014) pp.161-166. https://doi.org/10.1016/j.autcon.2013.08.014.

[12] J.M. Davila Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, H. Owolabi, Robotics and automated systems in construction: Understanding industry-specific challenges for adoption, Journal of Building Engineering. 26 (2019) 100868. https://doi.org/10.1016/j.jobe.2019.100868.

[13] J. Czarnowski, A. Dąbrowski, M. Maciaś, J. Główka, J. Wrona, Technology gaps in Human-Machine Interfaces for autonomous construction robots, Automation in Construction. 94 (2018) pp.179-190. https://doi.org/10.1016/j.autcon.2018.06.014.

[14] S. Cai, Z. Ma, M.J. Skibniewski, S. Bao, Construction automation and robotics for high-rise buildings over the past decades: A comprehensive review, Advanced Engineering Informatics. 42 (2019) 100989. https://doi.org/10.1016/j.aei.2019.100989.

[15] H. Ardiny, S. Witwicki, F. Mondada, Construction automation with autonomous mobile robots: A review, in: 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), 2015: pp. 418–424. https://doi.org/10.1109/ICRoM.2015.7367821.

[16] Q. Chen, B. García de Soto, B.T. Adey, Construction automation: Research areas, industry concerns and suggestions for advancement, Automation in Construction. 94 (2018) pp.22-38. https://doi.org/10.1016/j.autcon.2018.05.028.

[17] K. Jung, B. Chu, D. Hong, Robot-based construction automation: An application to steel beam assembly (Part II), Automation in Construction. 32 (2013) pp.62-79. https://doi.org/10.1016/j.autcon.2012.12.011.

[18] O. Davtalab, A. Kazemian, B. Khoshnevis, Perspectives on a BIM-integrated software platform for robotic construction through Contour Crafting, Automation in Construction. 89 (2018) pp.13-23. https://doi.org/10.1016/j.autcon.2018.01.006.

[19] O. Kontovourkis, G. Tryfonos, C. Georgiou, Robotic additive manufacturing (RAM) with clay using topology optimization principles for toolpath planning: the example of a building element, Architectural Science Review. 63 (2020) pp.105-118. https://doi.org/10.1080/00038628.2019.1620170.

[20] M. Taghavi, H. Heredia, K. Iturralde, H. Halvorsen, T. Bock, Development of a Modular End Effector for the installation of Curtain Walls with cable-robots, Journal of Facade Design and Engineering. 6 (2018) pp.1-8. https://doi.org/10.7480/jfde.2018.2.2067.

[21] P. Martinez, R. Ahmad, M. Al-Hussein, A vision-based system for pre-inspection of steel frame manufacturing, Automation in Construction. 97 (2019) pp.151-163. https://doi.org/10.1016/j.autcon.2018.10.021.

[22] N. Malik, R. Ahmad, M. Al-Hussein, Generation of safe tool-paths for automatic manufacturing of light gauge steel panels in residential construction, Automation in Construction. 98 (2019) pp.46-60. https://doi.org/10.1016/j.autcon.2018.11.023.

[23] C. Zhou, B. Tang, L. Ding, P. Sekula, Y. Zhou, Z. Zhang, Design and automated assembly of Planetary LEGO Brick for lunar in-situ construction, Automation in Construction. 118 (2020) 103282. https://doi.org/10.1016/j.autcon.2020.103282.

[24] Y. Yu, H. Guo, Q. Ding, H. Li, M. Skitmore, An experimental study of real-time identification of construction workers' unsafe behaviors, Automation in Construction. 82 (2017) pp.193-206. https://doi.org/10.1016/j.autcon.2017.05.002.

[25] R.J. Scherer, P. Katranuschkov, BIMification: How to create and use BIM for retrofitting, Advanced Engineering Informatics. 38 (2018) pp.54-66. https://doi.org/10.1016/j.aei.2018.05.007.

[26] B. Tao, X. Zhao, S. Yan, H. Ding, Kinematic modeling and control of mobile robot for large-scale workpiece machining, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture. 236 (2020) pp.29-38. https://doi.org/10.1177/0954405420933708.

[27] A. Jokić, M. Petrović, Z. Miljković, Semantic segmentation based stereo visual servoing of nonholonomic mobile robot in intelligent manufacturing environment, Expert Systems with Applications. 190 (2022) 116203. https://doi.org/https://doi.org/10.1016/j.eswa.2021.116203.

[28] U. Izagirre, I. Andonegui, I. Landa-Torres, U. Zurutuza, A practical and synchronized data acquisition network architecture for industrial robot predictive maintenance in manufacturing assembly lines, Robotics and Computer-Integrated Manufacturing. 74 (2022) 102287. https://doi.org/https://doi.org/10.1016/j.rcim.2021.102287.

[29] Y. Zhang, J. Xiao, Z. Zhang, H. Dong, Intelligent Design of Robotic Welding Process Parameters Using Learning-Based Methods, IEEE Access. 10 (2022) pp.13442-13450. https://doi.org/10.1109/ACCESS.2022.3146404.

[30] C. Li, P. Zheng, S. Li, Y. Pang, C.K.M. Lee, AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop, Robotics and Computer-Integrated Manufacturing. 76 (2022) 102321. https://doi.org/https://doi.org/10.1016/j.rcim.2022.102321.

[31] F. Tang, T. Ma, J. Zhang, Y. Guan, L. Chen, Integrating three-dimensional road design

and pavement structure analysis based on BIM, Automation in Construction. 113 (2020) 103152. https://doi.org/10.1016/j.autcon.2020.103152.

[32] N. Melenbrink, J. Werfel, A. Menges, On-site autonomous construction robots: Towards unsupervised building, Automation in Construction. 119 (2020) 103312. https://doi.org/10.1016/j.autcon.2020.103312.

[33] E. Kiraci, A. Palit, M. Donnelly, A. Attridge, M.A. Williams, Comparison of in-line and off-line measurement systems using a calibrated industry representative artefact for automotive dimensional inspection, Measurement. 163 (2020) 108027. https://doi.org/10.1016/j.measurement.2020.108027.

[34] M. Nahangi, J. Yeung, C.T. Haas, S. Walbridge, J. West, Automated assembly discrepancy feedback using 3D imaging and forward kinematics, Automation in Construction. 56 (2015) pp.36-46. https://doi.org/10.1016/j.autcon.2015.04.005.

[35] KUKA, KUKA KR 120 R3100, (2021). https://www.kuka.com/en-cn/products/robotics-systems/industrial-robots/kr-quantec (accessed August 21, 2021).

[36] M. Huang, L. He, D. Choi, J. Pecchia, Y. Li, Picking dynamic analysis for robotic harvesting of Agaricus bisporus mushrooms, Computers and Electronics in Agriculture. 185 (2021) 106145. https://doi.org/10.1016/j.compag.2021.106145.

[37] S. Fernández-Rodríguez, J.P. Cortés-Pérez, P.P. Muriel, R. Tormo-Molina, J.M. Maya-Manzano, Environmental impact assessment of Pinaceae airborne pollen and green infrastructure using BIM, Automation in Construction. 96 (2018) pp.494-507. https://doi.org/10.1016/j.autcon.2018.10.011.

[38] K. Zhang, Y. Zhu, C. Lou, P. Zheng, M. Kovač, A Design and Fabrication Approach for Pneumatic Soft Robotic Arms Using 3D Printed Origami Skeletons, in: 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft), 2019: pp. 821–827. https://doi.org/10.1109/ROBOSOFT.2019.8722719.

[39] K. Renuka, N. Bhuvanesh, J. Reena Catherine, Kinematic and Dynamic Modelling and PID Control of Three Degree-of-Freedom Robotic Arm BT, in: G. Kumaresan, N.S. Shanmugam, V. Dhinakaran (Eds.), Advances in Materials Research, Springer Singapore, Singapore, 2021: pp. 867–882. https://doi.org/10.1007/978-981-15-8319-

3_87.

[40] M.M. Alam, S. Ibaraki, K. Fukuda, S. Morita, H. Usuki, Identification of a Kinematic Model of a 6DOF Industrial Manipulator With Angular Positioning Deviation “Error Map” of Rotary Axes, in: 2020 International Symposium on Flexible Automation, Online, 2020: pp. 45–49. https://doi.org/10.1115/ISFA2020-9655.

[41] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, The International Journal of Robotics Research. 30 (2011) pp.846-894. https://doi.org/10.1177/0278364911406761.