

# AUTONOMOUS MODULAR CONSTRUCTION STRATEGY USING ROBOTIZED CRANE BASED ON DEEP LEARNING AND REINFORCEMENT LEARNING

Yifei XIAO , T. Y. YANG  , Xiao PAN, Fan XIE,  
Amir GHAREMANI BAGHMISHEH


*Department of Civil Engineering, The University of British Columbia, Vancouver, Canada*

## Article History:

- received 3 April 2024
- accepted 2 February 2025

**Abstract.** Modular construction offers significant advantages including faster construction time, higher quality control and less environmental impact. To further enhance its advantages, advanced robotic construction technologies are being developed. This research develops an automated modular construction framework that incorporates the robotic kinematics, deep learning and deep reinforcement learning using a robotized crane. The proposed modular construction strategy utilizes YOLOv5-S for modular container identification and localization. An improved proximal policy optimization (PPO-I) is developed and implemented in this strategy for collision-free three-dimensional (3D) lifting path planning and modular container transportation. States and rewards of the PPO-I and robot kinematics design of a real mobile crane are developed. The feasibility of the proposed modular construction strategy is verified through four case studies in 3D virtual environments. More than 97% success rate is observed meaning that the proposed strategy can be implemented in the robotized crane to localize the modular container and transport it to the target position with collision avoidance. The results indicate the potential of the proposed robotic-assisted modular construction strategy in the field of automated construction.

**Keywords:** modular construction, mobile crane, robotized crane, machine learning, deep learning, deep reinforcement learning, 3D path planning, collision avoidance.

 Corresponding author. E-mail: [yang@civil.ubc.ca](mailto:yang@civil.ubc.ca)

## 1. Introduction

Currently, the construction industry is facing many significant challenges, such as the lack of skilled workers and technology innovation, which lead to low efficiency, and very high labor and equipment costs (Delgado et al., 2019). Modular construction, as an alternative for traditional construction, integrates off-site manufacturing with on-site assembly to reach quality-controlled buildings within a shorter time than traditional construction methods. The slow unproductive site activities are replaced with a quicker and more efficient factory process in modular construction. Panelized and volumetric construction are the two most efficient classes of prefabricated construction because 70% to 95% of the building can be manufactured in the factory before transporting it for on-site assembly (Smith, 2010). These buildings have gained attention in densely populated cities, where construction often has been restricted by shortage of workforce, small working spaces and high requirements on low disturbance during operation (Lawson et al., 2014). The modular con-

struction market is expected to increase its global market value by 49% over the next decade due to increased demand for housing (Research and Markets, 2021). For instance, there is a local plan in Toronto to construct more than 1000 modular buildings before 2030 (City of Toronto, 2019). Modular construction offers significant benefits, including faster on-site construction, less material waste, less site disturbance, and higher utilization of construction material (Chen et al., 2020). The modular buildings are mostly advantageous in hospitals, schools, and apartments applications where repetitive units are preferred (Bi et al., 2020). Despite widespread advantages, most industry players are still reluctant to employ modular construction because of key challenges in the construction and project management which are completely different than those in traditional construction (Chua et al., 2020). The development of high-quality construction depends on innovative design theories, high-performance construction materials, and advanced construction technologies (Qi et al., 2021;

Rothmund et al., 2021; Štefanič & Stankovski, 2018). In order to achieve a fast and high-quality modular construction paradigm, the current construction industry needs to evolve towards digitization and intelligence. This transformation can be realized by implementing off-site robotic-assisted prefabrication and on-site robotic-assisted construction (Asadi et al., 2018; Petersen et al., 2019; Gharbia et al., 2020). Due to the fact that robots can replace various end effectors according to different construction tasks, robots have the abilities to replace human labor in some construction tasks. However, although robots are currently widely used in the manufacturing industry, their application in the construction industry is still in its infancy stage. Therefore, this paper aims to develop a robotic-assisted automated modular construction method for the next-generation construction industry. An automated construction framework that incorporates the robotic kinematics, deep learning and deep reinforcement learning is proposed to achieve an autonomous modular construction using a robotized crane. By implementing the robotic technologies, a faster and cost-effective modular construction process can be achieved. The outcomes of the research will accelerate knowledge exchange on smart infrastructure and cost-effective construction methods in the construction industry.

## 2. Literature review and research overview

### 2.1. Construction robots

Unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) are one of the main categories of robots used in construction. They can be used for various construction tasks, such as exploring high-risk areas, monitoring structural components and transporting light-weight materials. The main application of UAVs in construction is to undertake surveying and monitoring tasks. For example, UAVs have been used to access mud eruption zones (Di Stefano et al., 2018) and inspect building façade (Zhang et al., 2023). However, due to the limited payload of UAVs, they cannot be used for the construction tasks that require lifting heavy construction materials. UGVs have also been developed for exploring and monitoring tasks. For example, a UGV has been implemented to automate inspection process for a bridge (Peel et al., 2018). However, due to the complexity of the construction site environment, their autonomous movement on the construction site remains highly challenging. Robotic arms are another widely investigated robot in the construction field. The robotic arms are currently mainly used for off-site automated prefabrication, such as building component manufacturing. For example, Leong et al. (2023) used a robotic arm to install panels for a small lattice structure. They developed a printing path control algorithm for the robotic arm and used printing head to print the modular lattice structural panels on the structure. They proved that partially replacing labour with robotic arms can accelerated construction and reduced cost. Leng et al. (2023) proposed a human-

robot interaction construction method to accelerate the construction process. They used a six-axis KUKA robotic arm with the help of manual construction to construct a timber structure. Chea et al. (2024) developed a robotic collaborative system for the construction of load-carrying structures. They used two robotic arms to assemble structural components for a small-scale structure. Robotic arms have shown their strong capabilities in building component manufacturing due to their higher degrees of freedom compared to many other robots. However, due to their limited working range and payload capacity, the application of robotic arms in heavy construction tasks that require lifting heavy construction materials remains challenging. Compared to robotic arms, robotic cranes have larger working range and payload capacity, which means that the use of robotic cranes could be an appropriate solution for heavy construction tasks. However, due to the fact that crane robotization is still in the preliminary stage and few robotic cranes are well-developed in the construction industry, a large number of studies on cranes mainly focus on traditional crane lifting path planning. Therefore, the investigation into crane robotization and automated construction using robotized cranes needs to be conducted.

### 2.2. Lifting path planning of cranes

To achieve an automated construction process, lifting path planning is a critical task for robotic cranes. This is because robotic cranes need to plan a feasible lifting path and then transport construction materials to designated position on the construction site without any human controls. The investigations on improving crane operation planning can be divided into three categories, including 1) entity location planning, such as layout optimization (Al-Hussein et al., 2005; Fu et al., 2024); 2) process planning, such as scheduling (Oladugba et al., 2023); 3) physical motion planning, such as lifting path planning (Mousaei et al., 2021; Zhu et al., 2024). To plan the lifting path for the crane, previous studies analyzed dynamic systems and structural diagrams. For example, Ouyang et al. (2020) proposed a composite trajectory planning method for payload swing reduction using Lagrange's kinematics equation. The trajectory planning method introduced by Liu et al. (2019, 2021) utilized dynamic equations of the crane to reduce the unnecessary lifting path. However, these studies only considered the individual configurations of cranes and ignored the existence of obstacles on a construction site. Due to the high congestion of large-scale construction sites, the lifting operation of the crane will directly affect the on-site safety. Therefore, a lifting path that considers the obstacles on a construction site is crucial for crane operations. To achieve a collision-free lifting path planning, one of the most popular methods is the heuristic algorithm. The process of implementing heuristic algorithms is to search iteratively to approximate the exact solution by trial-and-error procedures, which will be effective for complex decision-making problems (Olearczyk et al.,

2014). A number of simulation methodologies and heuristic algorithms have been investigated for the collision-free lifting path planning problems, such as rapid random-exploring tree (RRT) (AlBahnassi & Hammad, 2012; Zhou et al., 2021), rapid random-exploring tree A star (RRT\*) (Mu et al., 2023), genetic algorithm (GA) (Yu & Wang, 2024; Yin et al., 2024), A\* algorithms (Olearczyk et al., 2014), and probabilistic search algorithm (Chang et al., 2012). However, all these path planning algorithms still require crane operators to control the crane to execute the lifting path. In order to achieve an autonomous construction process, a method that allows a robotic crane to plan a feasible lifting path and then execute the path by itself still needs to be developed.

### 2.3. Application of deep learning and reinforcement learning in construction

With the advancement of artificial intelligence (AI) and machine learning (ML), the investigation of deep learning (DL) algorithms in the fields of construction engineering has attracted many researchers' attention. The application of DL in construction can be categorized as four main categories, which are object detection, action recognition, image segmentation, and text mining and processing. Research on object detection mainly includes construction safety, site monitoring, and automatic detection and evaluation. Construction safety with DL algorithms, such as convolutional neural network (CNN), mainly focuses on detection of personal protective equipment (PPE) (Alkaissy et al., 2023; Ludwika & Rifai, 2024). Site monitoring using DL algorithms is mainly about the detection of workers, equipment, and construction materials (Fang et al., 2019; Islam et al., 2024). Automatic detection and evaluation with DL algorithms mainly focus on damage detection on the surface of structural components, such as cracks and special waveform of moisture damage (Kalfarisi et al., 2020; Hu et al., 2024). Research on action recognition involves the detection and prediction of the actions of onsite workers or equipment (Mekruksavanich & Jitpattanakul, 2023; Guo et al., 2023; Heravi et al., 2024; Ekanayake et al., 2024). The key points of workers or equipment will be defined and detected, then followed by the training of DL based action recognition model according to the obtained sequence data of the key points. Research on image segmentation is mainly about segmenting infrastructure surface images for damage detection (Alipour et al., 2019; Sirimewan et al., 2024). Research on text mining and processing focuses on acquisition of valuable information from project documents, which can help project managers to make decisions and identify the causes and effects of accidents (Pan & Zhang, 2020; Shamshiri et al., 2024). Indeed, the application of DL plays a vital role in construction fields, especially for object detection and construction safety. However, the role of DL in the automated construction field is more like the eyes of robots, which is mainly used to recognize objects. Robots cannot plan their motions and autonomously complete construction tasks by only relying on DL. Other

advanced AI algorithms and robotic technologies will be needed for automated construction.

Reinforcement learning (RL) is also a branch of ML algorithms that aims to train an agent to generate proper actions based on a specific environment. The RL algorithm is particularly powerful for autonomously making decisions during complex construction tasks, especially when the prior knowledge of the dynamics model or the environment is unknown. As reported by Asghari et al. (2022), the research of Henze and Schoenmann (2003) was one of the earliest attempts about the applications of RL in construction engineering and management (CEM) projects. From 2005 to 2010, only 14 articles, which implemented RL based algorithms and were related to CEM project, were published. After 2018, however, an obvious growth in the number of RL related publications could be observed (Asghari et al., 2022). This trend indicates that RL has become the most promising approach in the construction field. For example, Lee and Kim (2021) achieved an autonomous construction hoist system by using deep Q network (DQN). The results showed that DQN can optimize the control policy and reduce the waiting time and lifting time during construction. Andersson et al. (2021) implemented proximal policy optimization (PPO) using an energy-optimization approach in the reward function for a forestry crane to undertake single-log grasping tasks. The feasibility of the PPO was verified in a simulated environment and a successful rate of 97% was achieved. Zhu et al. (2023) developed a building information modelling (BIM) based re-configurable simulator for the assembly planning problem. The deep reinforcement learning was implemented and trained through this simulator for robot-based prefabricated construction. Although RL has recently been implemented for autonomous optimization and planning of robots in maintenance and construction (Yao et al., 2020; Chen et al., 2021; Zhu et al., 2023), few studies have focused on the application of RL in motion planning for crane operations, especially for automated modular construction. Therefore, in addition to the lifting path planning, a RL-based crane motion planning approach that can also enable cranes to automatically execute the lifting path needs to be proposed.

### 2.4. Research overview and main contributions

To address the limitations mentioned above, an automated construction framework (presented in Section 3.1) that incorporates the robotic kinematics, deep learning and deep reinforcement learning (DRL) is proposed to achieve an autonomous modular construction using a robotized crane. By implementing the proposed automated construction framework, the robotic crane is able to automatically identify and localize the construction materials, plan a collision-free lifting path, and finally transport the construction materials to the target position on a structure automatically. In the proposed automated construction framework, the YOLOv5-S (Jocher et al., 2021), a DL algorithm, is implemented to identify and localize a modular

container on the construction site. An improved proximal policy optimization (PPO-I), a DRL algorithm, is proposed to conduct 3D lifting path planning with collision avoidance and transport the modular container to the target position. To ensure that the robotized crane can execute the lifting path generated by PPO-I, robot kinematics design is conducted including the determination of Denavit–Hartenberg (DH) parameters, forward kinematics and inverse kinematics. The main contributions of this paper are: 1) A novel automated construction framework is proposed to achieve an autonomous modular construction. This will allow a robotized mobile crane to localize the construction materials and transport them to target positions on structures; 2) A novel improved proximal policy optimization (PPO-I) reinforcement learning algorithm is developed to conduct collision-free lifting path planning. Compared to traditional PPO, PPO-I has a faster convergence speed and a more stable learning process; 3) Robotic kinematics design is conducted to study the movement of multi-degree of freedom kinematic chains of mobile cranes. This will ensure that the lifting path generated by PPO-I can be executed by the robotized crane.

### 3. Methodology

#### 3.1. Automated construction framework

Figure 1 shows the proposed automated construction framework that incorporates the robotic kinematics, DL and DRL algorithms. At the beginning of construction, the DL algorithm is used to identify and localize the construction materials on the construction site. Once the position of the construction material (a 3D coordinate) is determined, this 3D coordinate will be used for robotic kinematics analysis. Through robotic kinematics analysis, the crane motions that can enable the hook of the crane move to this 3D coordinate will be obtained. Then, the crane motions will be sent to the robotized crane to execute, and the hook will move to the position where the construction materials are placed. The crane hooks can be connected to construction materials manually or through robot assisted methods (such as using a robotic arm to connect the hook and construction materials). After the hook is connected to the construction material, the DRL algorithm will start to plan the lifting path for the crane to execute. According to

a specific construction environment, the DRL algorithm will generate a new 3D coordinate that the hook of the crane needs to reach. This newly generated 3D coordinate will be used for robotic kinematics analysis again, and the corresponding crane motions will be obtained and sent to the robotized crane to execute. After the crane executes the crane motions, a new construction environment and states of the crane will be generated. The DRL algorithm will repeatedly execute this crane motion planning process until the construction material is transported to the target position on the structure. In this study, the robotic kinematics design of the crane, the DL algorithm and the DRL algorithm are presented in Section 3.2, Section 3.3 and Section 3.4, respectively.

#### 3.2. Kinematics design of mobile cranes

In the process of robot modelling, a robot is expressed as a series of links and joints. A joint is a mechanism that allows relative motion between various parts of a robot, while a link is a rigid body between two adjacent joints that maintains a fixed relationship between each joint. A traditional mobile crane (Figure 2) normally consists of a crawler or rubber-tired carrier, a turntable (rotary column), an inner boom, an outer boom, several telescopic booms, hoist rope, hook and power systems (such as electric motors, hydraulic pumps or pneumatic system). When designing the kinematics of a robotized crane, the crawler or rubber-tired carrier can be treated as a base link, while the turntable, inner boom, outer boom and the telescopic boom can be treated as a series of links. When numbering the links and joints, the link should be numbered from 0 (base link) to  $n$  (end link) and the joints should also be numbered from 0 to  $n$ . Table 1 summarizes the joint information of the robotized crane. Joint movements are all around the Z axis of the local coordinate system. This section aims to develop the kinematics for the proposed robotized crane. According to the composition of traditional mobile cranes, the proposed robotized crane is designed as Figure 2b. In this study, the crane motions generated by the robotic kinematics represent the joint movements of the robotized crane.

The kinematics design of the proposed robotized crane includes the establishment of local coordinate systems for joints, determination of Denavit–Hartenberg (DH) param-

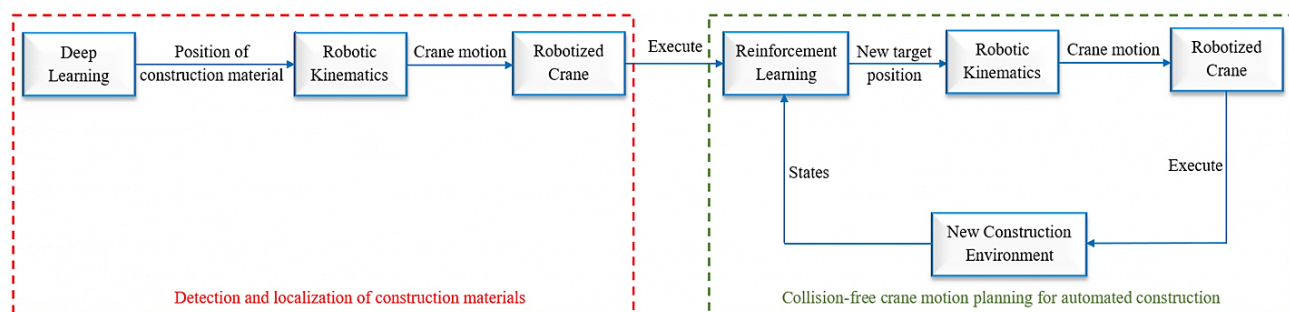
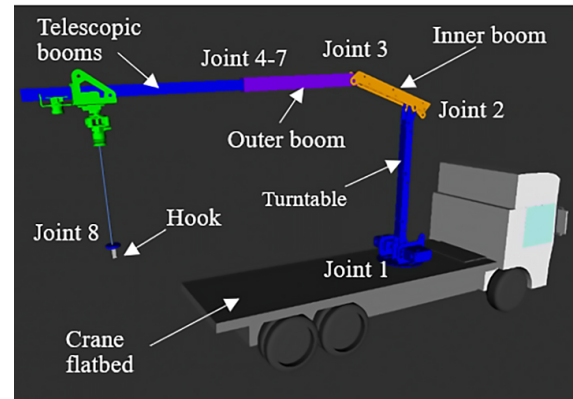


Figure 1. Automated construction framework

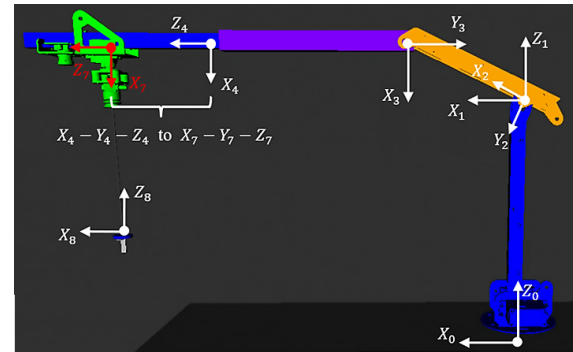


**Table 1.** Joint information of the robotized crane

Joint	Type	Description	Reference local axis (refer to Figure 3)
1	revolute	Connection between truck and turntable, representing the rotation of the turntable	Z axis
2	revolute	Connection between turntable and inner boom, controlling the pitch angle of the inner boom	Z axis
3	revolute	Connection between inner boom and outer boom, controlling the pitch angle of the outer boom	Z axis
4–7	prismatic	Connection for the telescopic boom, representing the extension of telescopic boom	Z axis
8	revolute	Connection used to regulate the orientation of the lifted construction materials	Z axis

**a)** Traditional mobile crane**b)** 3D model of the robotized crane**Figure 2.** Traditional mobile crane and the robotized crane model

eters, calculation of link transformation matrix, forward kinematics (FK) and inverse kinematics (IK). The local coordinate system establishment of the proposed robotized crane follows the procedure presented by Denavit and Hartenberg (Denavit & Hartenberg, 1955). Figure 3 shows the local coordinate systems of the crane components. The solid point in the local coordinate system indicates that the remaining third coordinate axis points out of the plane. The  $X_0$ - $Y_0$ - $Z_0$  coordinate system is set as the world coordinate system in this study. Next step is to define the DH parameters of the proposed robotized crane. The DH parameters describe the geometry of how each rigid link attaches to its child link through a joint. In this paper, the Craig convention (Craig, 2005) is adopted to create the DH table. Four parameters are defined in the DH table including link twist ( $\alpha$ ), link length ( $a$ ), joint angle ( $\theta$ ) and link offset ( $d$ ). Link twist ( $\alpha$ ) is used to describe the angle between two adjacent Z axis, while the joint angle ( $\theta$ ) is adopted to define the angle between two adjacent X axis. Link length ( $a$ ) is used to describe the distance between two adjacent Z axis, while link offset ( $d$ ) is adopted to define the distance between two adjacent X axis. Based on the local coordinate systems shown in Figure 3 and the Craig convention, the DH parameter table for the proposed robotized crane is summarized in Table 2. The end link for FK and IK analysis is defined as the end telescopic boom (local coordinate system  $X_7$ - $Y_7$ - $Z_7$ ).

**Figure 3.** Local coordinate systems of the robotized crane**Table 2.** DH parameter table of the robotized crane

$i$	$\alpha_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$
1	$0^\circ$	0	$\theta_1$	$l_1$
2	$-90^\circ$	0	$\theta_2$	0
3	$0^\circ$	$l_3$	$\theta_3$	0
4	$90^\circ$	0	$0^\circ$	$d_4$
5	$0^\circ$	0	$0^\circ$	$d_5$
6	$0^\circ$	0	$0^\circ$	$d_6$
7	$0^\circ$	0	$0^\circ$	$d_7$

Note:  $l_1$  is the length of the turntable.  $l_3$  is the length of the inner boom. The bold symbols represent the degree of freedom of each joint that can be controlled.  $l_2$  is not used in order to match the joint number ( $i$ )

The transformation matrix between a parent joint and a child joint ( $T_{i-1}^i$ ) describes the relative pose (including position and rotation) relation of two coordinate systems, which can be expressed as Eqn (1):

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

In robot kinematics, FK is to calculate the pose of the end joint of a robot given the rotation angle or translation distance of each joint. On the contrary, IK is to calculate the required rotation angle or translation distance of each joint given the pose of the end joint of a robot. The FK of the proposed robotized crane can be solved using Eqn (2). Then, according to the DH table, Eqn (1) and Eqn (2), the results of FK of the proposed robotized crane are shown in Eqn (3). As for IK, the method provided by Klampt (Hauser, 2021) is adopted to calculate IK based on the established local coordinate system and the DH table.

$$T_7^0 = T_1^0 T_2^1 T_3^2 \dots T_6^5 T_7^6; \quad (2)$$

$$T_7^0 = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2+\theta_3) & -\sin(\theta_1) & \cos(\theta_1)\sin(\theta_2+\theta_3) & p_x \\ \sin(\theta_1)\cos(\theta_2+\theta_3) & \cos(\theta_1) & \sin(\theta_1)\sin(\theta_2+\theta_3) & p_y \\ -\sin(\theta_2+\theta_3) & 0 & \cos(\theta_2+\theta_3) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where

$$p_x = \cos(\theta_1) * [(d_4 + d_5 + d_6 + d_7)\sin(\theta_2 + \theta_3) + l_3\cos(\theta_2)]; \quad (4)$$

$$p_y = \sin(\theta_1) * [(d_4 + d_5 + d_6 + d_7)\sin(\theta_2 + \theta_3) + l_3\cos(\theta_2)]; \quad (5)$$

$$p_z = (d_4 + d_5 + d_6 + d_7)\cos(\theta_2 + \theta_3) + l_1 - l_3 * \sin(\theta_2). \quad (6)$$

### 3.3. Modular container identification and localization using deep learning algorithm

In this study, to achieve automatic control of the mobile crane, a computer vision algorithm needs to be embedded into the control framework to automatically identify modular containers in a complex construction environment. Once the modular containers are detected successfully, the image coordinate of the modular containers can then be transferred to a world coordinate by using the camera intrinsic matrix and extrinsic matrix. As this research aims to achieve real-time control performance, the selected CNN-based vision algorithm should achieve real-time performance together with high accuracy and robustness. In this regard, a novel real-time light-weight object detector is established based on the architecture of YOLOv5-S. Overall, the structure of YOLOv5 consists of three components: 1) backbone network, 2) model neck, and 3) model head. The backbone network is used to generate rich feature maps of input images. During this process, the resolution of the original images reduces, while their feature depth (channel) increases. The neck component is responsible to pool and concatenate information from feature map pyramids at different resolutions. This helps the algorithm better generalize the detection of objects at various

scales. The head component is applied in the final stage, in order to regress the output as bounding box locations, class labels, and objectness scores. More detailed illustrations of the YOLOv5-S architecture can be referred to the Jocher et al. (2021) and MMYOLO Contributors (2022). After the modular container is identified, an image-based 2D coordinate ( $u, v$ ) will be obtained. By using Eqn (7), this image-based 2D coordinate ( $u, v$ ) can be converted to a 3D coordinate under the world coordinate system. Then, this 3D coordinate will be used for robotic kinematics analysis (IK analysis). It should be noted that coordinate ( $u, v$ ) is the center of the bounding box. Therefore, the calculated 3D coordinate in the world coordinate system is also the coordinates of this center.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P_{in} P_{ex} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (7)$$

where  $[u, v]$  is the image-based 2D coordinate of the modular container.  $[X, Y, Z]$  is the 3D coordinate of the modular container under the world coordinate system.  $P_{in}$  is the camera intrinsic matrix and  $P_{ex}$  is the camera extrinsic matrix.  $s$  is a scale factor.

### 3.4. Motion planning using deep reinforcement learning algorithm

In this section, an improved PPO algorithm (PPO-I) is proposed to control the robotized crane to transport the modular container to the desired location with path planning. The PPO-I framework, the approaches that are adopted to improve the performance of traditional PPO (Schulman et al., 2017), the state design and the reward function design are introduced.

Figure 4 shows the framework of PPO-I implemented in this study. Two neural networks (actor network and critic network) are employed in PPO-I. The actor network is made up of an input layer (14 neurons), two hidden layers (64 neurons each) and an output layer (3 neurons). The critic network is made up of an input layer (14 neurons), two hidden layers (64 neurons each) and an output layer (1 neuron). The inputs of these two neural networks are the states (e.g., position and direction of movement) of the agent (e.g., the robotized crane). The actor network will generate actions (e.g., incremental movements) to be taken by the agent, while the critic network will calculate a value ( $V_\phi$ ) to evaluate the states. After the agent executes the actions, a reward and a new state will be obtained. The reward and  $V_\phi$  will be used to update the parameters of the actor network. At the same time, the parameters of the critic network will be updated based on the previous state and the new state. Gradient ascent and gradient descent are used to update parameters of the actor network and critic network, respectively. This will ensure that the actions generated by the action network will maximize the total reward and the critic network will estimate the state value more accurately. The parameters for



- Learning rate decay: At the beginning of training, the agent learns a policy very slowly if the learning rate is set too small, which will increase the time consumption to have a convergence of PPO algorithm. At the end of training, a too large learning rate will cause the PPO algorithm to be unable to converge. Learning rate decay can enhance the stability at the end of training and improve the efficiency of algorithm convergence.
- In traditional PPO algorithm, the epsilon parameter of Adam Optimizer is set to  $1 \times 10^{-8}$  and the activation function is "Relu". Compared to  $1 \times 10^{-8}$  and "Relu", through experiment with these parameters, it is found that the use of epsilon parameter  $1 \times 10^{-5}$  and the activation function "Tanh" can make the reward reach its maximum level faster and increase the magnitude of maximum rewards.

An incorrect state design will cause the actor network to be unable to generate proper actions for the agent to execute, and also cause the critic network to be unable to correctly predict the value of the input states. This will finally affect the convergence of PPO-I training. In this study, the states of the robotized crane are designed based on the real observations required by the crane operator when operating the crane, which are summarized below. In PPO-I, the output of the actor network is the incremental movements of the end of the telescopic boom, while the output of critic network is a scalar used to evaluate the goodness of the states. States are the input of both actor network and critic network. The 14 neurons in the input layer of the actor network and critic network represent four vector states (states 1–4, each containing three elements) and two scalar states (states 5–6).

- 1) Current position of the hook ( $XYZ_c$ ): a real-time 3D coordinate of the hook at each time step. The position is updated after the robotized crane executes a new action output by actor network.
- 2) Target position of the hook ( $XYZ_t$ ): a 3D coordinate of the destination where the hook needs to reach when transporting construction materials. The tar-

get position is only updated when the environment is reset at the beginning of each training episode.

- 3) Current direction of hook movement ( $\mathbf{v}_{cdm}$ ): a 3D direction unit vector from the real-time hook position to target position. The current direction unit vector is updated after the robotized crane executes a new action output by actor network.
- 4) Target direction of hook movement ( $\mathbf{v}_{tdm}$ ): a 3D direction unit vector from the initial hook position to target position. The target direction unit vector is only updated when the environment is reset at the beginning of each training episode.
- 5) Relative distance ratio: a ratio of distance defining how far the hook is from the target position, which can be calculated using  $\frac{d_{cht}}{d_{iht}}$ , where  $d_{cht}$  is the distance from current hook position to the target position.  $d_{iht}$  is the distance from initial hook position to the target position.
- 6) Normalized collision distance: a virtual lidar sensor with 6 m working range is installed on the hook for collision detection. This state will provide a distance ratio between the hook and detected nearest object to 6 m. This state is 1 when no obstacles are detected, and it is equal to 0 when collision occurs. When this state value continues to decrease, it indicates that the current object lifting path cannot effectively avoid obstacles and the crane needs to find a new transportation direction.

When implementing the DRL algorithm, the learning objective is to train a DRL model to maximize the total reward. In some easy tasks, a simple discrete reward (DR) can be used to effectively train the DRL model. In complex tasks, both continuous reward (CR) and DR will be needed. Table 3 shows the DRs and CRs used in this study. CRs consider the proximity to target position and the direction when transporting a modular container. DRs consider the influence of reaching and staying at target position, collision, working space and simulation steps.

**Table 3.** Summary of reward function

Reward No.	Reward description	Reward value
CR 1	Positive reward for proximity to target position	$p_1 \times e^{-\left \frac{L_{pt}}{10}\right }, (p_1 > 0)$
CR2	Positive reward for matching direction	$p_2 \times \mathbf{v}_{cdm} \cdot \mathbf{v}_{tdm}, (p_2 > 0)$
CR 3	Negative reward for unnecessary time consumption	$p_3 \ (p_3 < 0)$
DR 4	Positive reward for reaching and staying at target position	$p_4 \ (p_4 > 0)$
DR 5	Negative reward for collision	$p_5 \ (p_5 < 0)$
DR 6	Negative reward when the hook exceeds the allowable working space	$p_6 \ (p_6 < 0)$
DR 7	Negative reward when simulation steps exceeds the maximum allowable steps	$p_7 \ (p_7 < 0)$

Note:  $p_1$  to  $p_7$  are the adjustable parameters that define the magnitude of each reward function.  $L_{pt}$  is the distance between the material lifted and the target position.  $\mathbf{v}_{cdm}$  is the current direction of movement at each time step.  $\mathbf{v}_{tdm}$  is the target direction of movement.



## 4. Model training and results

### 4.1. Modelling strategies

In this section, the details of modelling strategies, the DL model training and the DRL model training are introduced. A novel light-weight modular timber structure proposed by Yang et al. (2022) is modelled and used in three case studies as the prototype building for modular construction task. The assembly procedure of the modular tall timber building is shown in Figure 5. In this study, the proposed automated construction framework aims to conduct the modular construction task at stage 2. In addition, another larger building is also simulated to further verify the feasibility of the DL and DRL algorithms. In each case study, the well-trained DL and DRL models are implemented for the modular container localization and transportation respectively.

All models in the dynamic simulation environment are created by writing unified robotics description format (URDF) and performing a simulation in PyBullet (Coumans et al., 2016). To improve the effectiveness of the DL and DRL models for practical applications, the following modelling strategies are adopted in this study: (1) the dynamic simulation environment is constructed based on 3D point cloud data of a real construction site; (2) all objects (including structural model and the robotized crane, etc.) in the simulation environment are full scale (1:1 scale); (3) the robotized crane is modelled based on the exact specifications of the COPMA 510\_ENG crane (COPMA Articulated Cranes, 2023). There are two action strategies for the modular construction task. Action Strategy 1 is to use the DL algorithm to identify the 3D coordinates of the modular container, and then control the robotized crane to move the hook to the identified modular container location using IK. Action Strategy 2 is to apply the DRL algorithm to transport the modular container to the target position on the structure. To improve robustness of Action Strategy 2, the location of the modular container is placed randomly within the pick-up zone and the target position is randomly assigned within the 3D box space (as shown in Figure 6a). According to the building layout, 8 crane setup positions serving within 8 zones are defined as shown in Figure 6b. Next to each setup position, there is a container pick-up zone where the modular containers are placed pri-

or to assembly. The robotized crane will not move once it is set up in the setup zone and start to transport the modular containers within the corresponding pick-up zone. In each of the container pick-up zones, there is a virtual visual sensor system composed of a camera and a lidar sensor. The camera is used to identify the coordinates of the modular container in X and Y axes, while the lidar sensor is used to identify the coordinates of the modular container in Z axis. The camera and lidar sensor are fixed positioned at known positions within each pick-up zone. Based on the location information of cameras and lidar sensors, and the detection results of YOLOv5 (using cameras) and coordinate in Z axis (using lidar), the 3D coordinate of the modular container can be obtained. In addition, another lidar sensor is installed on the hook of the robotized crane to detect obstacles within a 6 m radius.

During the modelling and training, several assumptions are made, including:

- Action outputs by the actor network are the incremental movements of the end of the telescopic boom.
- The rotation of the joint 2 or the joint 3 in a real traditional crane is driven by an actuator that provides one-dimensional linear motion. However, the linear motion of an actuator can be converted to a rotation angle of the joint 2 or the joint 3. Hence, a revolute type of joint is applied for both joint 2 and joint 3 when writing the URDF file.
- The movements of all the joints of the robotized crane are continuous and simultaneous.
- In Action Strategy 1, after the hook reaches the location of the container, it is assumed that on-site human laborers will assist in connecting the hook to the container. 5) Joint 9 is added into the robotized crane model for orientation control of lifted object.

### 4.2. Training results

In order to deploy the YOLOv5-S algorithm as an effective vision sensor in the proposed DRL framework, there are several steps to train the algorithm successfully including: 1) training data generation and preparation, 2) anchor boxes determination, 3) training and validation. Each step is explained as follows.

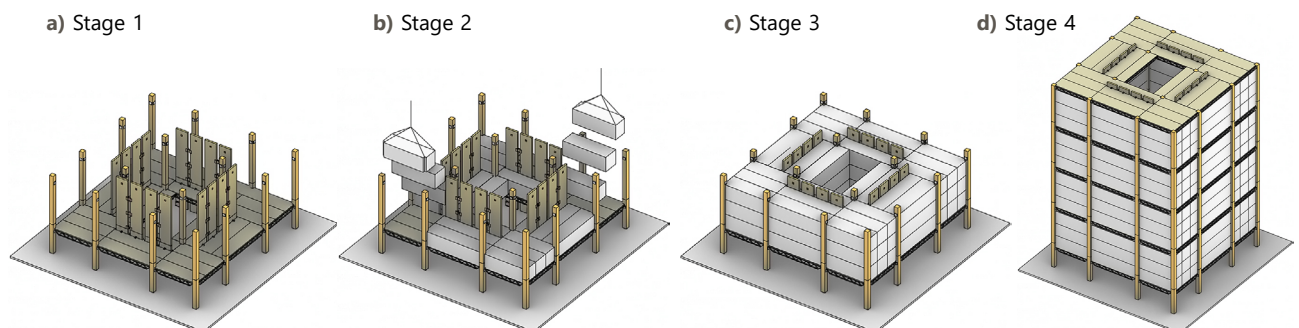
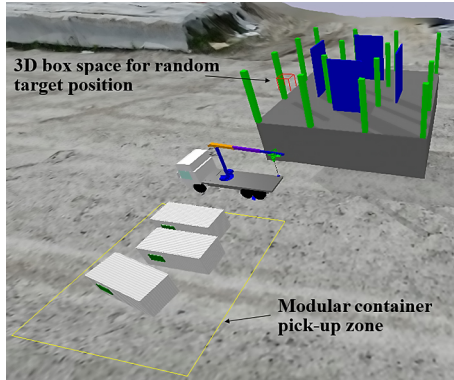


Figure 5. Assembly procedure of the modular tall timber building (Yang et al., 2022)



a) 3D training environment



b) Construction scheme

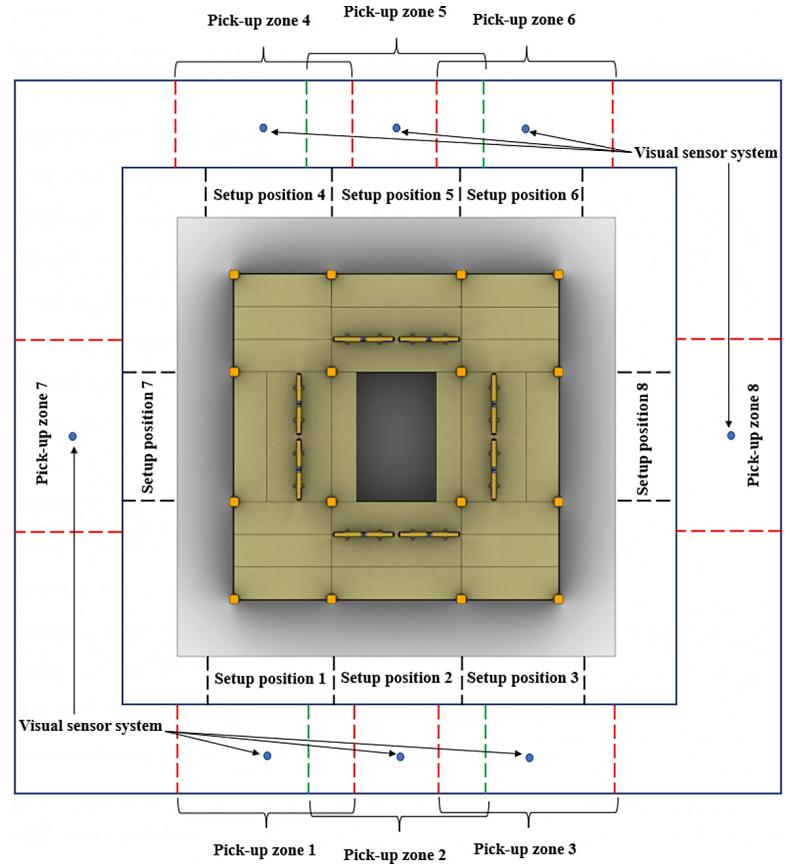


Figure 6. Construction scheme and 3D training environment

In Action Strategy 1, to create a database for training, images were first captured using virtual cameras in PyBullet. Then, standard data augmentation techniques such as cropping, horizontal flipping, small scaling, etc., are applied to the collected images to enlarge the training database. As a result, about 1806 images were generated, which were divided into 80% for training, and 20% for testing of the YOLOv5-S algorithm. In Action Strategy 2, after the training data are prepared, it is required to determine appropriate anchor box sizes to train the YOLO algorithms successfully and efficiently. For this purpose, k-means clustering algorithm is applied to the training dataset to determine the suitable anchor box dimensions. Anchor boxes are pre-defined bounding boxes of different shapes and sizes that are used as references for predicting the bounding boxes of objects in the image. Objects in images come in various sizes and aspect ratios. By using multiple anchor boxes with different scales and aspect ratios, YOLO can more effectively detect objects of different shapes and sizes. The estimated anchor box di-

mensions are summarized in Table 4. These anchor boxes will be used by the network as the basis to predict the bounding boxes.

The training is conducted using the stochastic gradient descent with momentum (SGDM) algorithm. The initial learning rate is set to 0.00125. The momentum factor is 0.937. The weight decay is 0.0005. The batch size is set to 16, based on the hardware configuration of an RTX 2070 GPU (with 8GB VRAM). The number of training epoch is set to 300. After training, the performance of the CNN object detector is typically evaluated as the mean average precision (mAP), which is the mean of the average precision across all the classes. To evaluate the average precision for each class, it is first required to calculate the Precision-Recall curve, where the Recall = True Positives / (True Positives + False Negatives), and Precision = True Positives / (True Positives + False Positives). To evaluate the precision and recall, an Intersection-over-Union (IoU) threshold must be assigned. If the IoU value between a predicted bounding box and its ground truth box is higher than

Table 4. Anchor box dimensions

Index	1	2	3	4	5	6	7	8	9
Width [pixels]	115	164	233	107	177	210	148	234	203
Height [pixels]	158	113	99	216	161	141	205	182	225

the IoU threshold, this predicted box will be considered a “True Positive”. Once the precisions and recalls are evaluated, the average precision for each class is evaluated as the area enclosed by the precision-recall curve. An average precision close to 1 indicates that the trained CNN algorithm is capable of retrieving the object of interest from most images at high precision. Table 5 presents a summary of three commonly used mAP metrics. The metric, mAP(IoU = 0.5) and mAP(IoU = 0.75), means that the mAP is determined at an IoU threshold of 0.5 and 0.75, respectively. The metric, mAP(IoU = [0.5:0.05:0.95]), is determined as the mean of the mAP taken at an IoU threshold starting from 0.5 to 0.95 with an increment of 0.05. It is noted that the mAP values are over 90% in all the mAP metrics presented, indicating that the YOLOv5-S algorithm can detect the container module precisely. In addition, inference on testing images is also presented in Figure 7. The modular containers in all the images are successfully localized with a precise boundary box.

A total number of  $3 \times 10^6$  steps are set for DRL training. The results show that a successful rate of 98.8% was achieved. To demonstrate the performance of the proposed PPO-I algorithms, the training results of PPO-I are

compared with other three reinforcement learning algorithms that also utilize actor-critic methods, which are Actor-Critic (AC), Advantage Actor-Critic (A2C) and traditional PPO. According to the training results, only PPO and PPO-I can successfully train the robotized crane to transport the modular container to the target position on the structure. Figure 8 presents the training results including normalized cumulative reward curve, loss of actor network and loss of critic network. According to the normalized cumulative reward curve, it shows that both PPO and PPO-I can eventually converge to the maximum reward level as the total training steps increase. However, it is obvious that PPO-I (around  $1 \times 10^6$  steps) reaches the maximum reward level faster than PPO (around  $2 \times 10^6$  steps), which indicates that PPO-I has better training performance than traditional PPO. Throughout the training process, the cumulative rewards of AC and A2C never reach the maximum reward level. This proves that AC and A2C fail to train the robotized crane to transport the modular container to the target position on the structure. The loss function of a neural network can reflect the accuracy of predicted value obtained from this neural network. The smaller the loss, the smaller the difference between the predicted output generated by neural network and the expected output. According to Figure 8b and Figure 8c, the losses of two neural networks in PPO-I have better convergence than traditional PPO, as a more stable and smaller loss is obtained in PPO-I in both actor network and critic network.

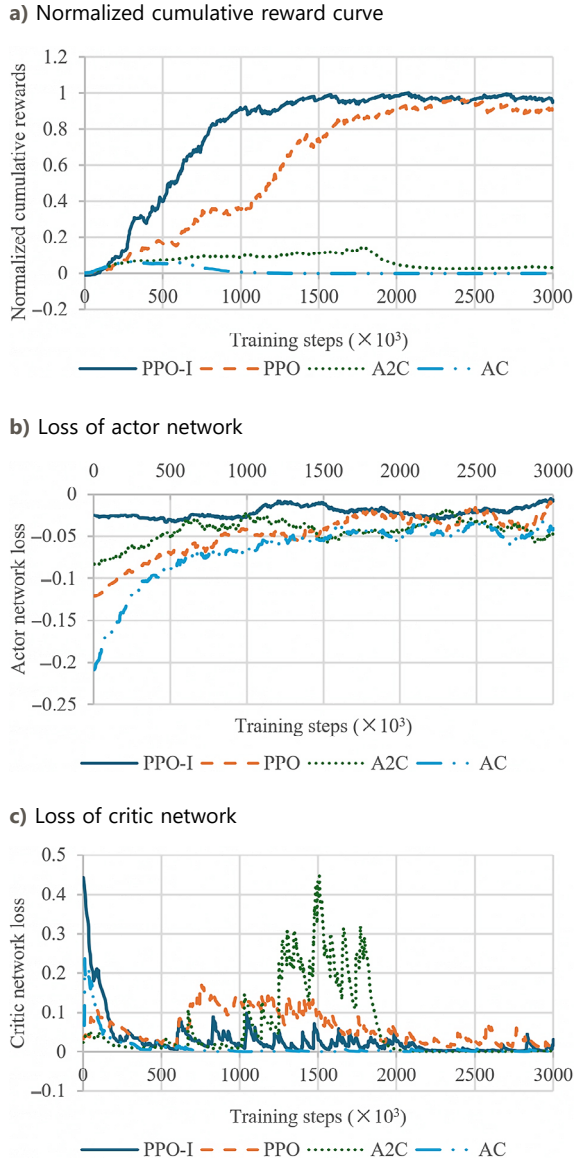
**Table 5.** Summary of the mAP metrics

mAP(IoU = [0.5:0.05:0.95]) (Primary challenge metric)	0.96
mAP(IoU = 0.5) (PASCAL VOC metric)	1.0
mAP(IoU = 0.75) (Strict metric)	1.0



**Figure 7.** Sample detection results performed by YOLOv5-S





## 5. Model evaluation and future study

### 5.1. Case studies

In this section, four case studies were conducted to verify the feasibility and performance of the DL and DRL models trained previously. The complexity of construction environment gradually increases from case 1 to case 4. In these case studies, the distance between the obstacle and the lifted modular container is randomly selected between 1.2 m and 27 m to represent typical construction environment.

The details of each case study are introduced as follow:

- Case 1: the robotized crane works at setup position 1, 3, 4 and 6. The objective is to lift and transport the modular container to the target position (shown in Figure 10).
- Case 2: the robotized crane works at setup position 2, 5, 7 and 8. The objective is to lift and transport the modular container to the target position (shown in

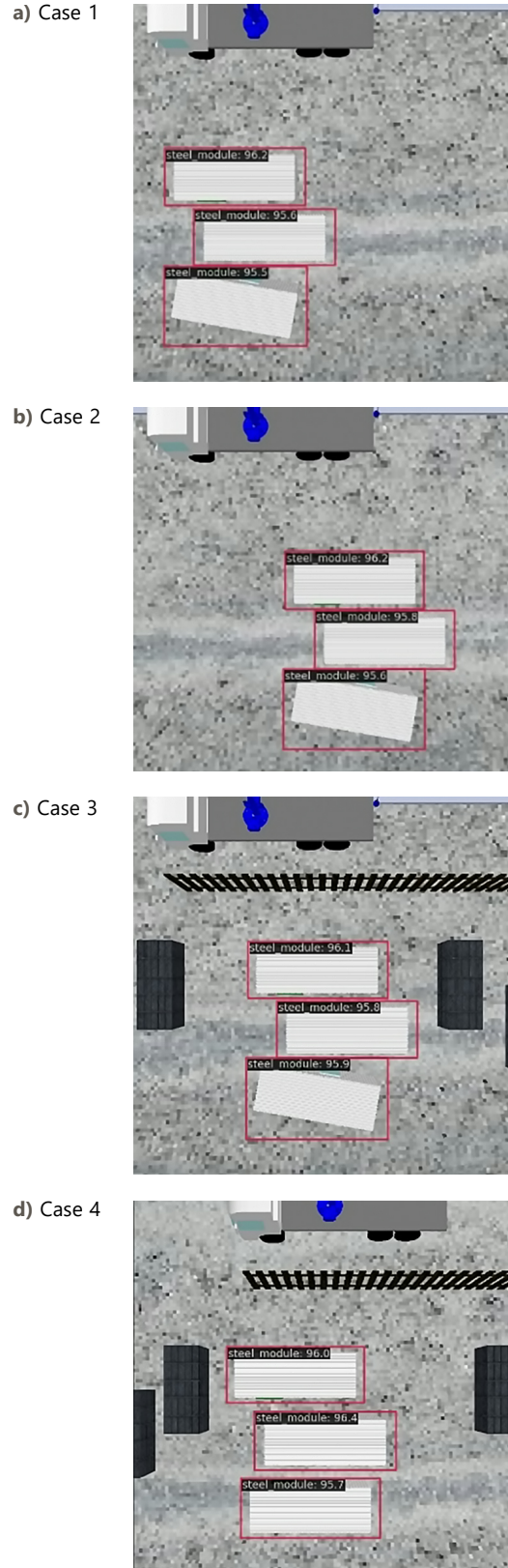
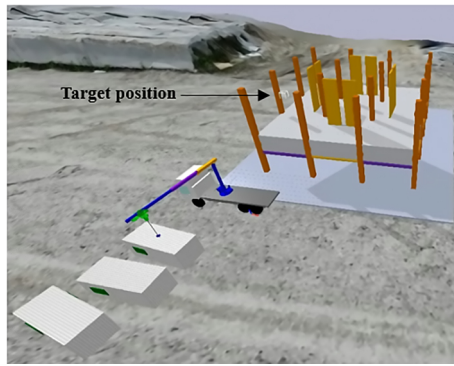


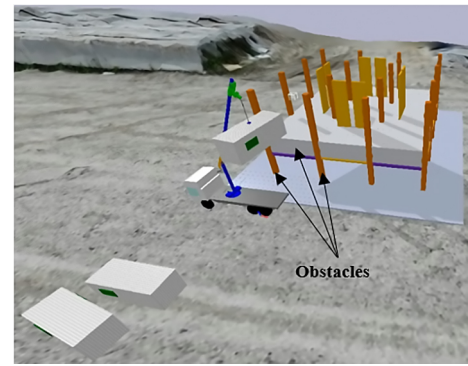
Figure 9. Identification results of modular container for the four cases

Figure 11). Compared to the case 1, Case 2 is more complex because of the existence of a central wall (obstacles) which limits the space for modular container movement.

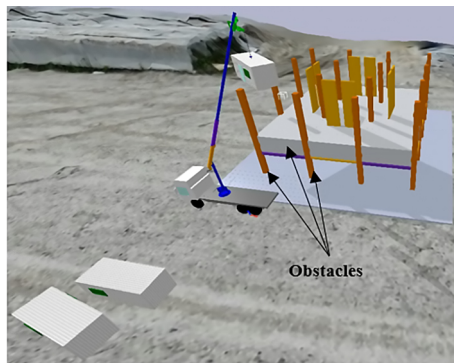
a) Lifting



b) Transporting



c) Avoiding obstacles



d) Reach the target position

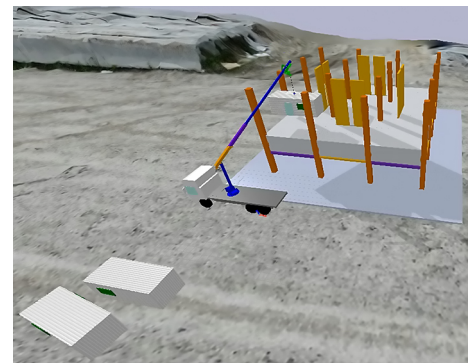
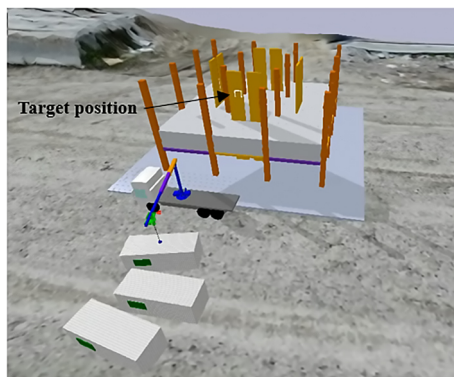
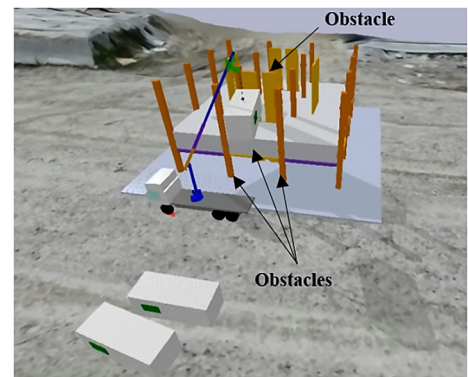


Figure 10. Construction procedure of case 1

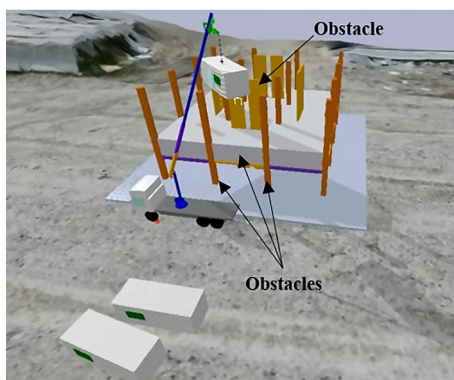
a) Lifting



b) Transporting



c) Avoiding obstacles



d) Reach the target position

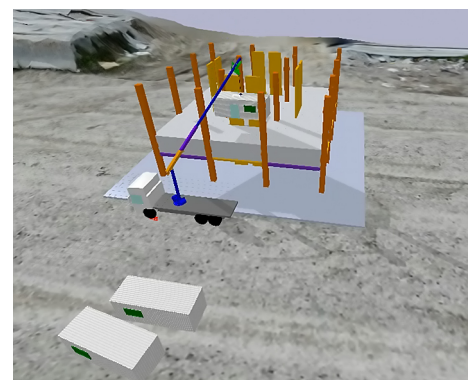


Figure 11. Construction procedure of case 2



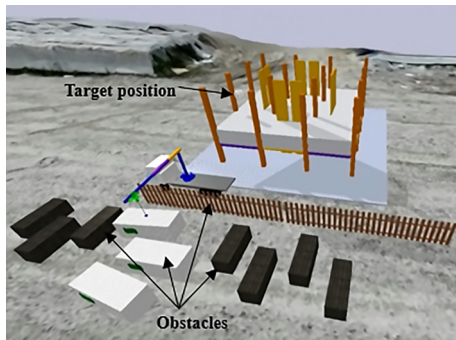
- Case 3: the robotized crane works at setup position 1, 3, 4 and 6. The objective is to lift and transport the modular container to the target position (shown in Figure 12). In addition, it is common to see obstacles such as temporary construction guardrails and construction materials in a real construction site. Therefore, a long wooden guardrail and several wooden construction materials are added as obstacles in case 3.
- Case 4: a new prototype modular timber building is simulated in PyBullet to further verify the feasibility and performance of the DL and DRL models. The objective is to lift and transport the modular container to the target position (shown in Figure 13).

For each case, 100 construction scenarios with various modular container pick up positions and target positions are randomly set for DL and DRL model evaluation. In each scenario, there are three modular containers to be lifted by the robotized crane. The crane will start lifting the modular container that is closest to the crane. During the lifting process, all environmental objects (including the unmoved modular containers, the robotized crane, and all structural components) will be considered as obstacles. To get start, YOLOv5-S is utilized to identify the modular containers, as shown in Figure 9. As shown in Figure 9, all three modular containers have been successfully detected in the four cases. After successful detection of the modular containers, the 3D coordinate of the modular container can be obtained using the procedure introduced in Section 3.3. With this 3D coordinate, the robotized crane can start to transport the modular container to the target position on

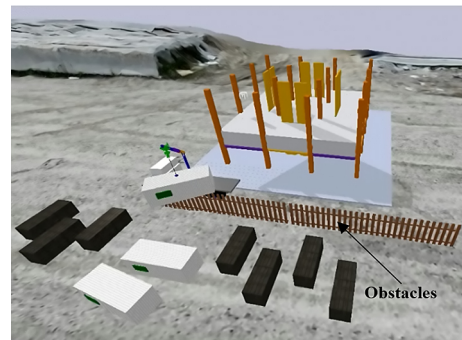
the structure. Figures 10–13 represent the lifting motions in cases 1–4, respectively. In each case, the operation includes lifting, transporting, avoiding obstacles, and placing the modular container to the target position. During construction, the states (including the current position and the target position of the hook, the current and the target direction of hook movement, the relative distance ratio, and the normalized collision distance) are obtained from the virtual environment and used as the inputs of actor network in PPO-I. Based on the continuous update of the states, PPO-I will continuously determine the next actions that the actor network should output for crane to execute. Based on the simulation results shown in Figures 10–13, the proposed framework can be used to autonomously lift the modular container from the initial position to the target position without collision.

To quantify the performances of the DRL model in the four cases, the success ratio and action coordination ratio ( $\xi_{ac}$ ) of each case are summarized in Table 6. The action coordination ratio, which can be calculated using Eqn (11), indicates how large the average action (average coordinates change in X, Y and Z axis) that the actor network outputs at each step. Normally, the larger the action coordination ratio is, the shorter the time it takes for the robotized crane to complete the task. However, an excessive action coordination ratio may cause the robotized crane to collide with obstacles. In this study, the action coordination ratios for the four cases are approximately 0.75 indicating that the hook movement in 3D coordinate system is about 0.75 m per step. Since the distance

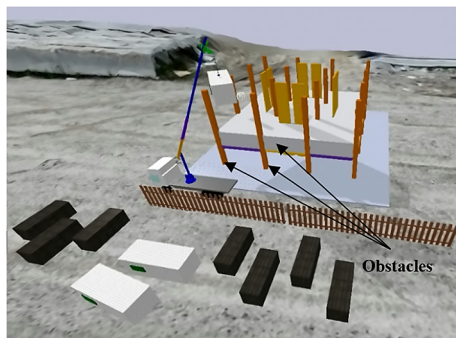
a) Lifting



b) Transporting and avoiding obstacles



c) Transporting and avoiding obstacles



d) Reach the target position

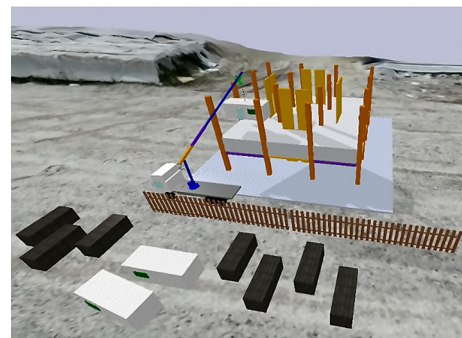
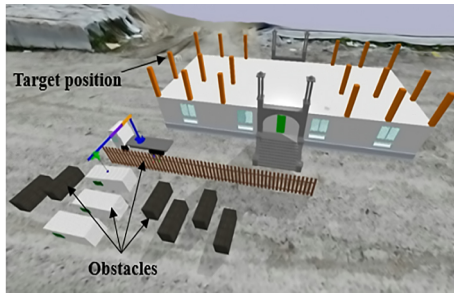


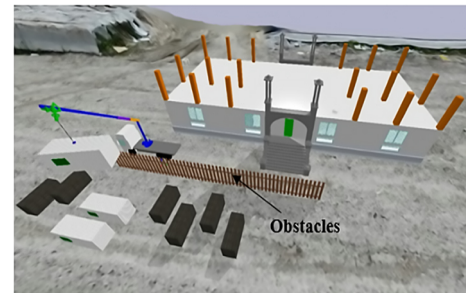
Figure 12. Construction procedure of case 3



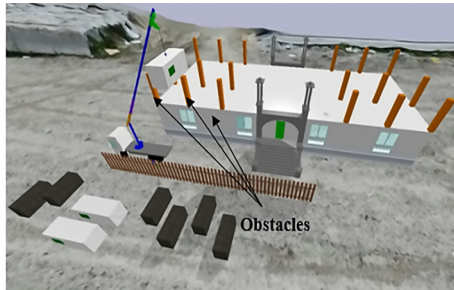
a) Lifting



b) Transporting and avoiding obstacles



c) Transporting and avoiding obstacles



d) Reach the target position

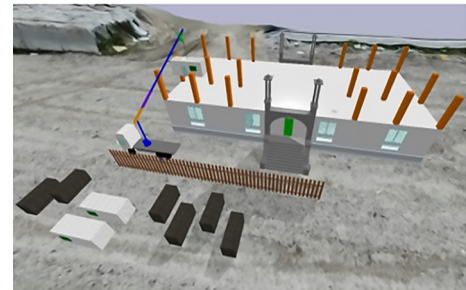


Figure 13. Construction procedure of case 4

between the obstacles and the lifted modular container in these case studies ranges from 1.2 m to 27 m, the average hook movement of 0.75 m per step is reasonable. This is because it will require too many action steps to be executed by the robotic crane if the coordination ratio is too small (e.g. 0.1 m). On the other hand, if the coordination ratio is too large (e.g. 2 m), the action generated by the actor network may cause collision with other objects in the construction site.

$$\xi_{ac} = \frac{1}{n \cdot N} \frac{\sum_{j=1}^{N=100} \sum_{i=1}^n (|a_x| + |a_y| + |a_z|)}{3}, \quad (11)$$

where  $n$  is the total steps in one construction scenario.  $N$  is the total number of construction scenarios.

Table 6. Quantitative analysis results

Case #	Success ratio of each case	$\xi_{ac}$
1	1	0.76
2	0.97	0.74
3	0.99	0.75
4	0.99	0.77

## 5.2. Limitations and future study

This research proposes an automated construction framework which includes the robotic kinematics, DL and DRL algorithms for automatic modular construction. The YOLOv5-S is used to detect and localize the modular containers on the construction site. A novel PPO-I reinforcement learning algorithm is proposed to conduct 3D collision-free lifting path planning for the robotized crane to trans-

port the construction materials. The robotic kinematics is designed to ensure that the crane hook can move to the positions generated by DL and DRL. The results of four case studies demonstrate that the proposed automated construction framework can guide the robotized crane to automatically perform modular construction, which shows the potential of this research in actual construction practice.

Despite the advantages mentioned above, some limitations of the current research are summarized: 1) The proposed automated construction framework can be applied to other types of cranes (such as tower cranes). However, the robotic kinematics designed in this study is based on the specifications of mobile cranes. The robotic kinematics need to be redesigned when other types of cranes are used. 2) The current research is conducted in a 3D virtual environment in PyBullet. The real-world application of the proposed robotics kinematics, DL and DRL algorithms can be investigated. 3) In the proposed automated construction framework, the crane motions (joint movements) generated by the robotic kinematics are achieved through the control algorithms (for example, proportional – integral – derivative (PID) controller) in the simulation environment. In real-world implementations, nonlinear effects of a mobile crane cannot be ignored. Hence, advanced nonlinear control algorithms can be developed to realize high-precision control.

## 6. Conclusions

In this paper, an automated construction framework that incorporates the robotic kinematics, DL and DRL algorithms is proposed to achieve an autonomous modular construction using a robotized crane. In order to robot-

ize a traditional mobile crane, the robot kinematics design of a real mobile crane is conducted including the establishment of local coordinate system, determination of DH parameters, FK and IK analysis. The YOLOv5-S is implemented as a DL algorithm to identify and localize modular containers on the construction site, while an improved PPO (PPO-I) is used as a DRL algorithm to transport the modular container and conduct 3D lifting path planning with collision avoidance. In order to obtain a practical DL model, the dynamic simulation environment is constructed based on 3D point cloud data of a real construction site. On the other hand, to obtain a practical DRL model, all objects in the simulation environment are 1:1 scale and the robotized crane is modelled based on the exact specifications of the COPMA 510\_ENG crane. Both discrete rewards and continuous rewards are defined, and the states of the robotized crane are designed based on the real observations required by the crane operator when controlling a mobile crane. The training results show that the YOLOv5-S algorithm can detect the container module precisely with mAP more than 0.96 and PPO-I algorithm can correctly plan 3D lifting paths and transport modular containers to the target position with a success rate of 98.8%. Finally, four case studies were conducted in 3D virtual environments to verify the feasibility and performance of the DL and the DRL models. The results proved that the proposed construction strategies can accurately identify the modular containers and then transport them to the target position on the structure with a success rate of more than 97%.

## Author contributions

Yifei Xiao: writing – original draft; writing – editing; conceptualization; data curation; formal analysis; investigation; methodology; project administration; validation. T. Y. Yang: funding acquisition; resources; supervision; project administration; paper review. Xiao Pan: writing – original draft. Fan Xie: model preparation. Amir Ghahremani baghmisheh: writing – original draft.

## Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- AlBahnassi, H., & Hammad, A. (2012). Near real-time motion planning and simulation of cranes in construction: Framework and system architecture. *Journal of Computing in Civil Engineering*, 26(1), 54–63. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000123](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000123)
- Al-Hussein, M., Alkass, S., & Moselhi, O. (2005). Optimization algorithm for selection and on site location of mobile cranes. *Journal of Construction Engineering and Management*, 131(5), 579–590. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2005\)131:5\(579\)](https://doi.org/10.1061/(ASCE)0733-9364(2005)131:5(579))
- Alipour, M., Harris, D. K., & Miller, G. R. (2019). Robust pixel-level crack detection using deep fully convolutional neural networks. *Journal of Computing in Civil Engineering*, 33(6), Article 04019040. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000854](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000854)
- Alkaiassy, M., Arashpour, M., Golafshani, E. M., Hosseini, M. R., Khanmohammadi, S., Bai, Y., & Feng, H. (2023). Enhancing construction safety: Machine learning-based classification of injury types. *Safety Science*, 162, Article 106102. <https://doi.org/10.1016/j.ssci.2023.106102>
- Andersson, J., Bodin, K., Lindmark, D., Servin, M., & Wallin, E. (2021, September). Reinforcement learning control of a forestry crane manipulator. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2121–2126). IEEE. <https://doi.org/10.1109/IROS51168.2021.9636219>
- Asadi, E., Li, B., & Chen, I. M. (2018). Pictobot: A cooperative painting robot for interior finishing of industrial developments. *IEEE Robotics & Automation Magazine*, 25(2), 82–94. <https://doi.org/10.1109/MRA.2018.2816972>
- Asghari, V., Wang, Y., Biglari, A. J., Hsu, S. C., & Tang, P. (2022). Reinforcement learning in construction engineering and management: A review. *Journal of Construction Engineering and Management*, 148(11), Article 03122009. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0002386](https://doi.org/10.1061/(ASCE)CO.1943-7862.0002386)
- Bi, J., Zong, L., Si, Q., Ding, Y., Lou, N., & Huang, Y. (2020). Field measurement and numerical analysis on wind-induced performance of modular structure with concrete cores. *Engineering Structures*, 220, Article 110969. <https://doi.org/10.1016/j.engstruct.2020.110969>
- Chang, Y. C., Hung, W. H., & Kang, S. C. (2012). A fast path planning method for single and dual crane erections. *Automation in Construction*, 22, 468–480. <https://doi.org/10.1016/j.autcon.2011.11.006>
- Chea, C. P., Bai, Y., & Zhou, Z. (2024). Design and development of robotic collaborative system for automated construction of reciprocal frame structures. *Computer-Aided Civil and Infrastructure Engineering*, 39(10), 1550–1569. <https://doi.org/10.1111/mice.13145>
- Chen, Z., Popovski, M., & Ni, C. (2020). A novel floor-isolated re-centering system for prefabricated modular mass timber construction – Concept development and preliminary evaluation. *Engineering Structures*, 222, Article 111168. <https://doi.org/10.1016/j.engstruct.2020.111168>
- Chen, S., Dong, J., Ha, P., Li, Y., & Labi, S. (2021). Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, 36(7), 838–857. <https://doi.org/10.1111/mice.12702>
- Chua, Y. S., Liew, J. R., & Pang, S. D. (2020). Modelling of connections and lateral behavior of high-rise modular steel buildings. *Journal of Constructional Steel Research*, 166, Article 105901. <https://doi.org/10.1016/j.jcsr.2019.105901>
- City of Toronto. (2019). *HousingTO 2020–2030 action plan*. <https://www.toronto.ca/community-people/community-partners/affordable-housing-partners/housingto-2020-2030-action-plan/>
- COPMA Articulated Cranes. (2023). *COPMA document*. <https://www.cps-group.com/downloads/copma/>
- Coumans, E., Bai, Y. P., & PyBullet, A. (2016). *A Python module for physics simulation for games, robotics and machine learning*.
- Craig, J. J. (2005). *Introduction to robotics: mechanics and control*. Pearson Education.
- Delgado, J. M. D., Oyedele, L., Ajayi, A., Akanbi, L., Akinade, O., Bilal, M., & Owolabi, H. (2019). Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. *Journal of Building Engineering*, 26, Article 100868. <https://doi.org/10.1016/j.jobe.2019.100868>

- Denavit, J., & Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22(2), 215–221. <https://doi.org/10.1115/1.4011045>
- Di Stefano, G., Romeo, G., Mazzini, A., Iarocci, A., Hadi, S., & Pelphrey, S. (2018). The Lusi drone: A multidisciplinary tool to access extreme environments. *Marine and Petroleum Geology*, 90, 26–37. <https://doi.org/10.1016/j.marpetgeo.2017.07.006>
- Ekanayake, B., Ahmadian Fard Fini, A., Wong, J. K. W., & Smith, P. (2024). A deep learning-based approach to facilitate the as-built state recognition of indoor construction works. *Construction Innovation*, 24(4), 933–949. <https://doi.org/10.1108/CI-05-2022-0121>
- Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. (2020). Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *2020 International Conference on Learning Representations*.
- Fang, W., Zhong, B., Zhao, N., Love, P. E., Luo, H., Xue, J., & Xu, S. (2019). A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network. *Advanced Engineering Informatics*, 39, 170–177. <https://doi.org/10.1016/j.aei.2018.12.005>
- Fu, Y., Bu, J., Lin, J., Liu, J., & Zhang, C. (2024). Selection and layout optimization of double tower cranes. *Buildings*, 14(6), Article 1705. <https://doi.org/10.3390/buildings14061705>
- Gharbia, M., Chang-Richards, A., Lu, Y., Zhong, R. Y., & Li, H. (2020). Robotic technologies for on-site building construction: A systematic review. *Journal of Building Engineering*, 32, Article 101584. <https://doi.org/10.1016/j.jobe.2020.101584>
- Jocher, G., Stoken, A., Borovec, J., NanoCode012, Chaurasia, A., TaoXie, Changyu, L., Abhiram V, Laughing, tkianai, yxNONG, Hogan, A., lorenzomamma, AlexWang1900, Hajek, J., Diaconu, L., Marc, Kwon, Y., oleg, wanghaoyang0106, Defretin, Y., Lohia, A., ml5ah, Milanko, B., Fineran, B., Khromov, D., Yiwei, D., Doug, Durgesh, & Ingham, F. (2021). *ultralytics/yolov5: v5.0-YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations*. <https://zenodo.org/records/4679653>
- Guo, H., Zhang, Z., Yu, R., Sun, Y., & Li, H. (2023). Action recognition based on 3D skeleton and LSTM for the monitoring of construction workers' safety harness usage. *Journal of Construction Engineering and Management*, 149(4), Article 04023015. [https://doi.org/10.1061/\(JCEMD4.COENG-12542](https://doi.org/10.1061/(JCEMD4.COENG-12542)
- Hauser, K. (2021). *Kris' locomotion and manipulation planning toolbox - Klamp't*. <https://github.com/krishauser/Klampa>
- Henze, G. P., & Schoenmann, J. (2003). Evaluation of reinforcement learning control for thermal energy storage systems. *HVAC&R Research*, 9(3), 259–275. <https://doi.org/10.1080/10789669.2003.10391069>
- Heravi, M. Y., Jang, Y., Jeong, I., & Sarkar, S. (2024). Deep learning-based activity-aware 3D human motion trajectory prediction in construction. *Expert Systems with Applications*, 239, Article 122423. <https://doi.org/10.1016/j.eswa.2023.122423>
- Hu, K., Chen, Z., Kang, H., & Tang, Y. (2024). 3D vision technologies for a self-developed structural external crack damage recognition robot. *Automation in Construction*, 159, Article 105262. <https://doi.org/10.1016/j.autcon.2023.105262>
- Islam, M. S., Shaqib, S. M., Ramit, S. S., Khushbu, S. A., Sattar, M. A., & Noor, D. S. R. H. (2024). *A deep learning approach to detect complete safety equipment for construction workers based on YOLOv7*. arXiv preprint arXiv:2406.07707. <https://doi.org/10.48550/arXiv.2406.07707>
- Kalfarisi, R., Wu, Z. Y., & Soh, K. (2020). Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization. *Journal of Computing in Civil Engineering*, 34(3), Article 04020010. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000890](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000890)
- Lapan, M. (2018). *Deep reinforcement learning hands-on: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd.
- Lawson, M., Ogden, R., & Goodier, C. I. (2014). *Design in modular construction* (Vol. 476). CRC Press. <https://doi.org/10.1201/b16607>
- Lee, D., & Kim, M. (2021). Autonomous construction hoist system based on deep reinforcement learning in high-rise building construction. *Automation in Construction*, 128, Article 103737. <https://doi.org/10.1016/j.autcon.2021.103737>
- Leng, Y., Shi, X., Hiroatsu, F., Kalachev, A., & Wan, D. (2023). Automated construction for human-robot interaction in wooden buildings: Integrated robotic construction and digital design of iSMART wooden arches. *Journal of Field Robotics*, 40(4), 810–827. <https://doi.org/10.1002/rob.22154>
- Leong, Z., Chen, R., Xu, Z., Lin, Y., & Hu, N. (2023). Robotic arm three-dimensional printing and modular construction of a meter-scale lattice façade structure. *Engineering Structures*, 290, Article 116368. <https://doi.org/10.1016/j.engstruct.2023.116368>
- Liu, Z., Yang, T., Sun, N., & Fang, Y. (2019). An antiswing trajectory planning method with state constraints for 4-DOF tower cranes: design and experiments. *IEEE Access*, 7, 62142–62151. <https://doi.org/10.1109/ACCESS.2019.2915999>
- Liu, Z., Sun, N., Wu, Y., Xin, X., & Fang, Y. (2021). Nonlinear sliding mode tracking control of underactuated tower cranes. *International Journal of Control, Automation and Systems*, 19, 1065–1077. <https://doi.org/10.1007/s12555-020-0033-5>
- Ludwika, A. S., & Rifai, A. P. (2024). Deep learning for detection of proper utilization and adequacy of personal protective equipment in manufacturing teaching laboratories. *Safety*, 10(1), Article 26. <https://doi.org/10.3390/safety10010026>
- Mekruksavanich, S., & Jitpattanakul, A. (2023). Automatic recognition of construction worker activities using deep learning approaches and wearable inertial sensors. *Intelligent Automation & Soft Computing*, 36(2), 2111–2128. <https://doi.org/10.32604/iasc.2023.033542>
- MMYOLO Contributors. (2022). *MMYOLO: OpenMMLab YOLO series toolbox and benchmark*. <https://github.com/open-mmlab/mmyolo>
- Mousaei, A., Taghaddos, H., Nekouvaght Tak, A., Behzadipour, S., & Hermann, U. (2021). Optimized mobile crane path planning in discretized polar space. *Journal of Construction Engineering and Management*, 147(5), Article 04021036. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0002033](https://doi.org/10.1061/(ASCE)CO.1943-7862.0002033)
- Mu, G., Zhang, M., Wu, G., Li, Z., & Pan, L. (2023). Optimisation of overhead crane path based on RRT-A\* fusion improvement algorithm. *International Journal of Simulation and Process Modelling*, 21(1), 63–74. <https://doi.org/10.1504/IJSPM.2023.139808>
- Oladugba, A. O., Gheith, M., & Eltawil, A. (2023). A new solution approach for the twin yard crane scheduling problem in automated container terminals. *Advanced Engineering Informatics*, 57, Article 102015. <https://doi.org/10.1016/j.aei.2023.102015>
- Olearczyk, J., Bouferguène, A., Al-Hussein, M., & Hermann, U. R. (2014). Automating motion trajectory of crane-lifted loads. *Automation in Construction*, 45, 178–186. <https://doi.org/10.1016/j.autcon.2014.06.001>
- Ouyang, H., Tian, Z., Yu, L., & Zhang, G. (2020). Motion planning approach for payload swing reduction in tower cranes with double-pendulum effect. *Journal of the Franklin Institute*, 357(13), 8299–8320. <https://doi.org/10.1016/j.jfranklin.2020.02.001>
- Pan, Y., & Zhang, L. (2020). BIM log mining: Learning and predicting design commands. *Automation in Construction*, 112, Article 103107. <https://doi.org/10.1016/j.autcon.2020.103107>

- Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning* (pp. 1310–1318). PMLR.
- Peel, H., Luo, S., Cohn, A. G., & Fuentes, R. (2018). Localisation of a mobile robot for bridge bearing inspection. *Automation in Construction*, 94, 244–256. <https://doi.org/10.1016/j.autcon.2018.07.003>
- Petersen, K. H., Napp, N., Stuart-Smith, R., Rus, D., & Kovac, M. (2019). A review of collective robotic construction. *Science Robotics*, 4(28), Article eaau8479. <https://doi.org/10.1126/scirobotics.aau8479>
- Qi, B., Razkenari, M., Costin, A., Kibert, C., & Fu, M. (2021). A systematic review of emerging technologies in industrialized construction. *Journal of Building Engineering*, 39, Article 102265. <https://doi.org/10.1016/j.jobbe.2021.102265>
- Research and Markets. (2021). *Modular construction – global market trajectory & analytics*. <https://www.researchandmarkets.com/reports/4805338/modular-constructionglobal-market-trajectory>
- Rothmund, P., Kim, Y., Heisser, R. H., Zhao, X., Shepherd, R. F., & Keplinger, C. (2021). Shaping the future of robotics through materials innovation. *Nature Materials*, 20(12), 1582–1587. <https://doi.org/10.1038/s41563-021-01158-1>
- Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). *High-dimensional continuous control using generalized advantage estimation*. arXiv preprint arXiv:1506.02438. <https://doi.org/10.48550/arXiv.1506.02438>
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347. <https://doi.org/10.48550/arXiv.1707.06347>
- Shamshiri, A., Ryu, K. R., & Park, J. Y. (2024). Text mining and natural language processing in construction. *Automation in Construction*, 158, Article 105200. <https://doi.org/10.1016/j.autcon.2023.105200>
- Sirimewan, D., Bazli, M., Raman, S., Mohandes, S. R., Kineber, A. F., & Arashpour, M. (2024). Deep learning-based models for environmental management: Recognizing construction, renovation, and demolition waste in-the-wild. *Journal of Environmental Management*, 351, Article 119908. <https://doi.org/10.1016/j.jenvman.2023.119908>
- Smith, R. E. (2010). *Prefab architecture: A guide to modular design and construction*. John Wiley & Sons.
- Štefanič, M., & Stankovski, V. (2018, December). A review of technologies and applications for smart construction. *Proceedings of the Institution of Civil Engineers-Civil Engineering*, 172(2), 83–87. <https://doi.org/10.1680/jcien.17.00050>
- Tucker, G., Bhupatiraju, S., Gu, S., Turner, R., Ghahramani, Z., & Levine, S. (2018, July). The mirage of action-dependent baselines in reinforcement learning. In *International Conference on Machine Learning* (pp. 5015–5024). PMLR.
- Yang, T. Y., Lepine-Lacroix, S., & Ghahremani Baghmisheh, A. (2022). Novel high-performance tall modular mass timber buildings. In *5th International Conference on Earthquake Engineering and Disaster Mitigation (5th ICEEDM)*, Yogyakarta, Indonesia.
- Yao, L., Dong, Q., Jiang, J., & Ni, F. (2020). Deep reinforcement learning for long-term pavement maintenance planning. *Computer-Aided Civil and Infrastructure Engineering*, 35(11), 1230–1245. <https://doi.org/10.1111/mice.12558>
- Yin, J., Li, J., Yang, A., & Cai, S. (2024). Optimization of service scheduling problem for overlapping tower cranes with cooperative coevolutionary genetic algorithm. *Engineering, Construction and Architectural Management*, 31(3), 1348–1369. <https://doi.org/10.1108/ECAM-08-2022-0767>
- Yu, G., & Wang, S. (2024). Research on lifting path planning algorithms for intelligent cranes. *Journal of Engineering Research and Reports*, 26(7), 57–63. <https://doi.org/10.9734/jerr/2024/v26i71193>
- Zhang, C., Wang, F., Zou, Y., Dimyadi, J., Guo, B. H., & Hou, L. (2023). Automated UAV image-to-BIM registration for building façade inspection using improved generalised Hough transform. *Automation in Construction*, 153, Article 104957. <https://doi.org/10.1016/j.autcon.2023.104957>
- Zhou, Y., Zhang, E., Guo, H., Fang, Y., & Li, H. (2021). Lifting path planning of mobile cranes based on an improved RRT algorithm. *Advanced Engineering Informatics*, 50, Article 101376. <https://doi.org/10.1016/j.aei.2021.101376>
- Zhu, A., Dai, T., Xu, G., Pauwels, P., De Vries, B., & Fang, M. (2023). Deep reinforcement learning for real-time assembly planning in robot-based prefabricated construction. *IEEE Transactions on Automation Science and Engineering*, 20(3), 1515–1526. <https://doi.org/10.1109/TASE.2023.3236805>
- Zhu, A., Zhang, Z., & Pan, W. (2024). Developing a fast and accurate collision detection strategy for crane-lift path planning in high-rise modular integrated construction. *Advanced Engineering Informatics*, 61, Article 102509. <https://doi.org/10.1016/j.aei.2024.102509>