Enabling Human-Robot Partnerships in Digitally-Driven Construction Work through Integration of Building Information Models, Interactive Virtual Reality, and Process-Level Digital Twins

by

Xi Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2022

Doctoral Committee:

Professor Vineet R. Kamat, Co-Chair
Associate Professor Carol C. Menassa, Co-Chair
Professor SangHyun Lee
Associate Professor Jonathan Wesley McGee

Xi Wang

wangix@umich.edu

ORCID iD: 0000-0002-2583-0356

Dedication

To my mother and father,

who are the best parents in the universe

To my robot Megaboss,

who brought happiness and sorrows to my life

Acknowledgements

I would like to express my sincere gratitude to my academic advisors, Prof. Carol Menassa and Prof. Vineet Kamat, for their guidance and support during my doctoral study. Prof. Menassa not only guided me to obtain a thorough understanding of my research topics but also taught me to present them clearly to the research community. Prof. Kamat taught me to develop penetrating research insights. When I was working on this dissertation, he spent several nights supervising me at the robotic laboratory after-hours so that I can collect more data for this dissertation. I really appreciate that they respected my choice and fully supported me when I decided to switch my research topic to construction robotics, which led to this dissertation. In addition to research, they paid considerable attention to helping me develop soft skills in presentation, proposal writing, interpersonal communication, and leadership. During the five years' time working with them, Prof. Menassa and Prof. Kamat instructed me to conduct my first literature review, to publish my first research paper, and to make my first conference presentation, helping me grow from an undergraduate student into an independent researcher. I am extremely fortunate to have them as my advisors.

I sincerely appreciate the efforts of Prof. Wes McGee in setting up the physical Kuka robot and developing the software and PLC system to enable robot control. He provided valuable guidance for me to understand the hardware system and communication framework and to operate the robot. I also want to thank Prof. SangHyun Lee, who taught my first class at the University of

Michigan. He provided constructive advice and valuable comments to help me develop my research objectives, formulate my research methodology, and improve my dissertation.

I would like to acknowledge the financial support I received from the U.S. National Science Foundation (Awards FW-HTF #2025805 and FW-HTF #2128623), Barbour Scholarship and Rackham Graduate Student Research Grant provided by the Rackham Graduate School, and the John L. Tishman Fellowship from the Department of Civil and Environmental Engineering for me to complete this dissertation.

I am extremely thankful to my colleagues and collaborators Dr. Bharadwaj Mantha, Dr. Kurt Lundeen, Dr. Lichao Xu, Dr. Da Li, Dr. Ci-Jyun Liang, Min Deng, Hongrui Yu, Somin Park, Shuoqi Wang, and many others from the SICIS, LIVE and DPM groups for their critical comments, thorough discussions, and remarkable suggestions on my research. I really enjoyed the time we spent together. Our friendship means a lot to me.

I am deeply appreciative of my friends, Wenxin Hu, Wei Wei, Siqi Zhang, Meng Dong, Xiaoyu Guo, Xiaohang Ji, Huiwen Xu, Hang (Alex) Song, and Bo Fu, who are always available for a talk to share my happiness and sorrows. They brought me food to protect my health while I was working on my research, stayed up all night to debug my code, helped me improve my writing, and provided me with valuable suggestions. Without their support, I could not make it through this challenging time. I hope our friendship can continue to blossom after my graduation.

Last but not least, I would like to take this opportunity to express the deepest gratitude to my beloved parents and family relatives who provided with me a lot of encouragement and spiritual support. The whole family encouraged and supported me to pursue my dream over these years without any complaints. I want to express my sincere gratitude for their understanding and kindness.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

Human cognition plays a critical role in construction work, particularly in the context of high-level task planning and in-field improvisation. On the other hand, robots are adept at performing numerical computation and repetitive physical tasks with precise motion control. The unstructured and complex nature of construction environments and the inability to maintain tight tolerances in assembled workpieces pose several unique challenges to the wide application of robots in construction work. Thus, robotization of field construction processes is best achieved through human-robot partnerships that take advantage of both human and robot intelligence, as well as robots' physical operational capabilities, to overcome uncertainties and successfully perform construction work.

This dissertation explores the pathway of integrating building information models (BIM), interactive virtual reality (VR), and process-level digital twins to enable human-robot partnerships in digitally-driven construction through three levels. At the first level, an interactive and immersive process-level digital twin system in VR that serves as the human-robot collaboration platform is proposed. It integrates visualization and supervision, task planning and execution, and bi-directional communication to enable human workers to remotely collaborate with construction robots in field construction. A human-in-the-loop experiment based on a drywall installation case study was conducted for system verification and to collect user feedback for future improvements. Overall, the system enables human-robot partnerships and reduces the cognitive planning and physical workload of human workers.

At the second level, Building Information Models (BIM) are integrated into the digital twin system to enable closed-loop BIM-driven Human-Robot Collaboration (HRC) in construction. BIM provides digital information to both the robot and its human partners to drive the construction process. In addition, deployment of the system to co-robotically performed construction work is studied. A physical drywall installation case study and three physical experiments (i.e., visual detection and end-effector movement) were conducted to verify the system workflow and to evaluate the system. Building on the previous level, the integration of BIM reduces human co-workers' planning effort and improves construction work accuracy.

Motivated by the programming and human instruction effort required to guide motion sequences in typical robotic work, the third level of this dissertation builds upon the BIM-driven digital twin system and explores how to enable robots to automatically plan their motion sequence. A Scene Distance Matrix (SDM) is proposed to guide robots' sequential decisions in selecting modular construction skill primitives that lead to robot motions. Interactive Learning from Demonstration (LfD) is used to teach robots the mapping from the SDM to the skill primitives. The proposed approach is presented with a case study that contains three scenarios, including exterior wall sheathing, drywall installation, and timber frame construction. A wooden shelf construction task has been used to verify the proposed LfD module and its integration with the BIM-driven digital twin system. It further reduces the planning and programming effort of human workers.

Overall, this research aims to create a scalable pathway to bring human workers in the loop of robotized construction and capitalize on human workers' improvisation ability to handle uncertainties on construction sites. In addition, it explores the integration of BIM and LfD with the interactive digital twin to improve system autonomy in task planning and motion sequencing.

This dissertation establishes the foundation of next-generation construction work by transitioning the role of construction workers from manual task performers to robot supervisors.

Chapter 1 Introduction

As a major economic sector, the construction industry accounts for 13% of GDP globally (Mckinsey Global Institute 2017). However, it is also one of the most labor-intensive and dangerous industries with the highest number of workplace fatalities (CPWR 2018). The introduction of robots has often been proposed to have significant potential to mitigate the issues faced by the construction industry because robots are fast, accurate, and have repetitive operational capabilities and redundant physical power. However, few robots are deployed on construction projects due to technical challenges (e.g., unstructured work environments and loose tolerances) posed by the very nature of the industry (Feng et al. 2015). In the natural course of performing construction work, human workers continuously use their cognitive capabilities and experience to overcome emerging workplace challenges, and thus have a strong potential to support co-robotic construction work. However, a technical workflow to efficiently support such collaboration between human workers and robots for construction work has been missing.

This research aims to create a scalable pathway to bring human workers in the loop and capitalize on human workers' instinctive ability to improvise during the process of handling uncertainties on construction sites. More specifically, this research explores the development of a Digital Twin platform to support human-robot collaboration through the integration of Building Information Modeling (BIM) into the co-robotic construction process. Underlying methods such as robot Learning from Demonstration (LfD) are studied and integrated for task sequencing and

path planning and to improve robot autonomy. An overview of the research is presented in Figure 1.1.

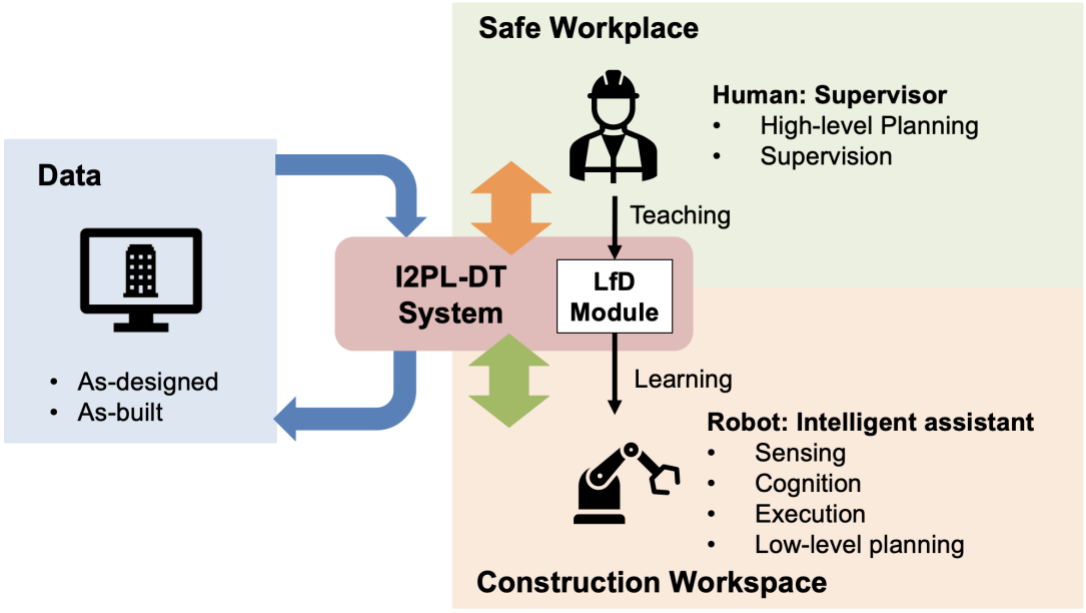


Figure 1.1 Research Overview

1.1 Importance of the Research

The construction industry is ill-famed for its stagnant productivity and lackluster safety record. Over the past twenty years, the annual productivity growth of the construction industry is only 1%, compared to 2.8% of the total economy, causing the productivity of construction to fall way behind other industries over this time span (Mckinsey Global Institute 2017). It is also one of the most labor-intensive and dangerous industries in the U.S., with the highest number of fatalities and the 4th highest nonfatal injury rates among all the major industries (CPWR 2018; Liu et al. 2017). According to the literature, contact with objects and overexertion and body reaction are two main reasons for nonfatal injuries in construction, which account for 33.2% and 27.5% of all reported injuries that lead to days away from work (CPWR 2018).

In recent years, the industry is also facing a severe labor shortage. According to Associated General Contractors of America, 81% of general contractors in the U.S. reported this issue in 2020 (AGC 2020). Due to the fact that the younger generation is reluctant to choose careers in construction because of the dangerous work environments and the perceived high physical demands, the labor shortage is expected to become an increasingly critical issue in the near future (Liang et al. 2019). Despite the labor shortage, people with different physical abilities rarely find opportunities in the field or on job sites because of the physical-demanding nature of typical construction work (Aulin and Jingmond 2011). As a result, construction does not employ a broadly diverse and inclusive workforce. For instance, only 4.5% of construction laborers are female (BLS 2021).

With the rapid development of computational power and artificial intelligence, robotics has demonstrated the potential to mitigate some of the productivity, safety, and labor shortage issues faced by the construction industry and improve the diversity and inclusion of the construction workforce (Davila Delgado et al. 2019; García de Soto et al. 2018). Robots are capable of manipulating heavy objects and performing repetitive tasks without loss of precision or susceptibility to issues such as fatigue experienced by human workers. Robots have already been adopted in several industries to reduce human workers' workload and their exposure to potential hazards, such as manufacturing (Mitsi et al. 2005), nuclear (Qian et al. 2012), healthcare (Barbash and Glied 2010), and rescue (Davids 2002). Studies have found that robotics can also bring economic benefits by increasing quality and productivity while bringing down labor costs (Davila Delgado et al. 2019).

Despite the considerable benefits robotics has brought to other industries, the very nature of the construction industry impedes the application of robotics on construction sites (Lundeen et

al. 2019). To highlight the challenges, a comparison can be considered between the construction and the manufacturing industries, where robots are widely applied to assist with work in the latter.

(1) Robot working environments: Manufacturing is usually conducted in a factory environment that is well-controlled with limited changes or interruptions. However, construction sites are very unstructured with constantly moving equipment and workers, which poses high requirements for robots' perception of ambient environment and adaptability (Lee and Moon 2014);

(2) Products: Manufacturing typically produces a large number of identical products that travel along an assembly line, while the products of construction projects (e.g., buildings) are typically unique and stationary, which requires robots' mobility, accurate localization, and the ability to complete a series of tasks at disparate locations on an expansive job site (Fukuda et al. 1991);

(3) Uncertainties: Manufacturing typically has very tight tolerances, and each workpiece is produced under strict dimensional and quality control and is accurately installed. However, construction workpieces and raw materials are subject to larger deviations, introducing uncertainties to robotic construction processes (Milberg 2006).

Because of the complexity and uncertainties of construction projects, human cognition capabilities and expertise play a crucial role in construction and are difficult for a robot to replicate within a short period of time and for a broad range of conditions. As shown in Figure 1.2, humans can perform creative design and adjust their plans to adapt to changes (Sharif et al. 2016). They are good at perceptual understanding by synthesizing the information around them and learning from their experience (Seong et al. 2019). However, robots have limited creativity, adaptability, and synthesizing ability and they lack commonsense and context related to the real world (Calinon

2009). On the other hand, robots have high physical capability. They are good at minor deviation detection and fast numerical computation. They can perform work with high speed and accuracy. However, humans have comparatively lower physical capability. They cannot detect minor deviations, and their performance is inconsistent and subject to errors (Seong et al. 2019).

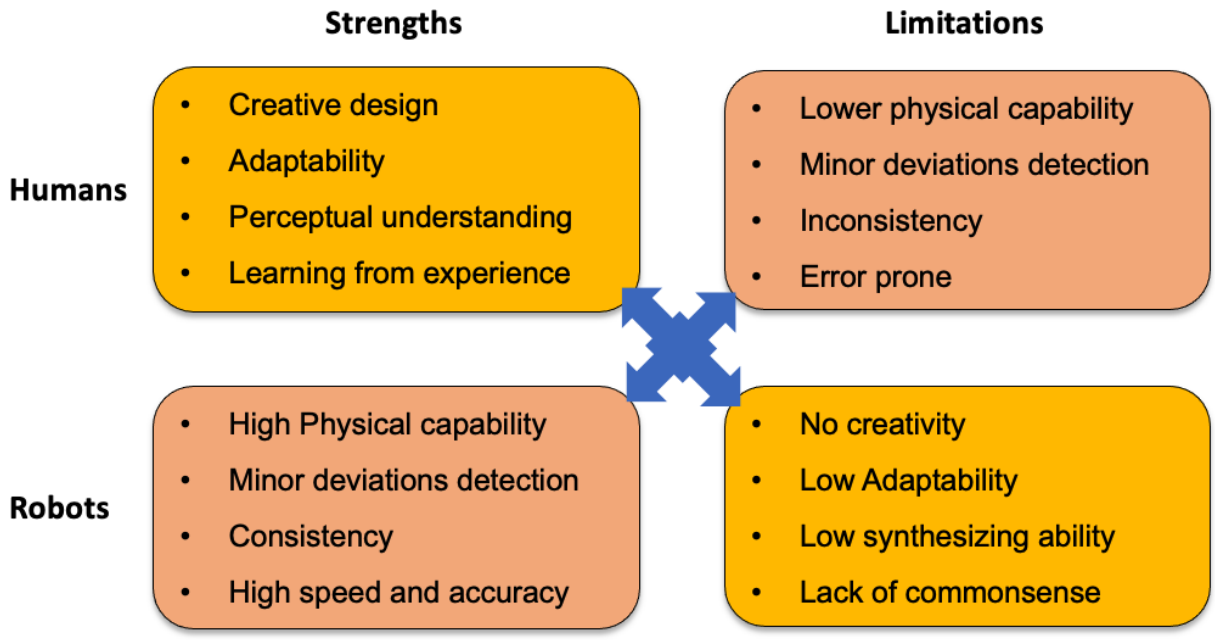


Figure 1.2 Human and Robot Capabilities Comparison

Therefore, human-robot collaboration (HRC) that takes advantage of both humans' and robots' capabilities provides an effective approach for robotized construction. It also offers the potential for construction work to be guided by human workers from remote locations. The need for automated and remote construction work is further stressed with the outbreak of the Covid-19 pandemic, which critically affected construction projects across the world and highlighted the importance of accomplishing work remotely in order to keep the economy going. Unlike most other industries, construction work cannot be done remotely "at-home" and it is hard to keep a safe social distance on-site because certain tasks require the physical coordination of several workers. As a result, disruptions caused by the Covid-19 pandemic resulted in 975,000 job losses in April

2020 and serious economic impacts (ENR 2020). Automated and remote construction thus emerged as a viable option for certain construction work to be performed remotely, and thus reducing the number of workers required on-site to protect their safety.

In summary, there are clear and critical needs to promote remote HRC in construction to mitigate the productivity, safety, inclusion, and labor shortage issues faced by the construction industry. However, a framework that can efficiently support human workers to collaborate with robots from remote locations while ensuring construction work quality and fully exploiting robot capabilities has been missing. This research attempts to remedy this state of affairs and explores three critical aspects: HRC framework for field work, closed-loop BIM integration for digitally-driven field work, and motion sequence Learning from Demonstration (LfD) for unstructured task sequencing in field work.

1.2 Background of the Research

Operating robots has traditionally been prohibitive for many small and medium businesses that typically have no robotics experts on their staff. The development of human-robot interaction techniques together with the emergence of advanced hardware and algorithms are making robot operation easier and thus accessible to people with limited expertise or training in robotics. This section presents an overview of current robot operation techniques and their limitations, with an emphasis on construction robots. It also describes how immersive technologies have promoted the development of effective interaction techniques between humans and robots.

1.2.1 Pre-Programmed Fully Automated Robotic Operation

One of the most traditional ways of operating robots is to program robot functions and sequences of movements for fully automated execution using computer programs or teaching

pendants (OSHA 2021). Pre-programming has been widely used for robotic assembly lines in factories where the robots are fixed in one location and repeat the movements automatically. Such automation significantly increases productivity and reduces labor costs (Wallén 2008). However, pre-programming for fully automated construction robots is much more complex because the physically dispersed and one-of-a-kind nature of the construction product (e.g., buildings) makes the robot trajectories different for each manipulation task (Navon and Retik 1997).

Some construction robots rely on the project digital model. For example, Eversmann et al. (2017) developed a robotic prefabrication system for timber structures. Motion plans of the robots are generated from the digital plan of the structure and the layout of the workspace, whose designs have been optimized for robot prefabrication. Digital plans allow for precise robotized assembly, but such a system requires highly structured workspaces and workpieces, which are unavailable in most construction projects. Therefore, algorithms and techniques have been developed for robots to adapt to the current environmental circumstances and develop motion plans accordingly. Lundeen et al. (2017) developed adaptive manipulation techniques for construction robots by fitting the digital construction models onto as-built geometry. Feng et al. (2015) and Navon (2000) proposed vision-based systems to detect and manipulate arbitrarily placed construction materials on site. However, the adaptability lays on the foundation of significant research and programming work beforehand and is limited to a few specific tasks. As a result, pre-programming construction robots to be fully autonomous is labor-intensive and not cost-efficient (Jen et al. 2008; Navon and Retik 1997).

1.2.2 Lead-Through Robot Operation

Human-robot interaction allows humans to operate the robot for task execution. Human operators lead the robot to adaptively execute tasks by controlling joint angles or end-effector

positions. Given the fact that construction tasks involve less identical repetition, it provides an efficient solution to construction robot operation by notably reducing the programming effort required upfront.

An intuitive method of human-robot interactive operation is to lead the robot through physical contact, which requires human operators to apply physical forces directly to the robot or the object the robot is carrying to guide the robot to corresponding positions. This method has several advantages. First, the robot is carrying the workpiece so human co-workers are relieved from physical stress and can pay more attention to task details, such as the glass curtain wall installation robot (Lee and Moon 2014) and Material Unit Lift Enhancer – MULE135 (Construction Robotics 2022). Second, lead-through retains the agility of human workers and reduces the possible deviation caused by robotic sensors or controller errors. For example, Devadass et al. (2019) developed an adaptive robotics assembly system that allows human co-workers to fine-tune the final workpiece pose to ensure accurate assembly after the robot carries the workpiece to the designed position. Moreover, Yousefizadeh et al. (2019) adapted robot pre-defined trajectories for human intentions and Gil et al. (2013) considered construction workers' habits when designing robots.

These types of robots for physical interaction with humans need to be specifically designed with safeguard functions. However, working alongside construction robots could still be dangerous since robot failures can cause serious accidents or even fatalities when they are carrying large and heavy construction workpieces.

1.2.3 Teleoperation

Teleoperation allows human operators to control robots from remote locations. It can protect human workers from dangerous environments, and thus is popular among construction robots.

Joysticks are powerful for navigating and thus have been widely used for unmanned ground vehicle (UGV) teleoperation (Nicholas Flann et al. 2000). In addition to navigation, they also demonstrate the potential as a device to operate robots with higher degrees of freedom (DOFs) since it has been adopted to operate construction equipment such as cranes and excavators. For example, Jung et al. (2013) developed a joystick teleoperation system for robotic beam assembly such that the operator does not need to board the cabin far above the ground. However, operators have limited perception of the environment while operating remotely, which leads to increased difficulty and reduced accuracy of manipulation.

Haptic and force reflecting devices can reflect contact force to the operator and provide the operator with tactile responses from the environment (Chotiprayanakul et al. 2012; Lee et al. 2007). It allows contact force control and makes teleoperation safer and smoother. Therefore, it has been used in a variety of robotized construction tasks, such as glass window panel fitting (Chung et al. 2010; Lee et al. 2007), steel bridge grit blasting (Chotiprayanakul et al. 2012), and underwater construction (Hirabayashi et al. 2006).

Gesture-based teleoperation controls robots with human gestures. Some systems use vision-based methods to detect human gestures. Yu et al. (2014) proposed a system to guide a mobile construction robot by waving a traffic light baton, which could be detected by a digital camera. Du et al. (2012) controlled the robot end-effector by tracking the hand position with a Kinect camera. Some others use wearable sensors to track human body motions. Kim et al. (2009)

used inclinometers, orientation sensors, and rotary encoders to detect human arm movement to operate an excavator. Seong et al. (2019) tracked dexterous human hand movement with gesture-controlling gloves and replicate the movement on a robotic hand.

Researchers have also developed several types of interfaces for robot teleoperation. For example, the mobile phone has been used to teleoperate robots via text messages (Patra and Ray 2007) or voice commands (Kubik and Sugisaka 2001). Victores et al. (2011) used computer software to control a robot to perform tunnel inspection and maintenance tasks. David et al. (2014) developed a system that allows the user to operate the cutting head of a tunnel boring machine remotely by directly interacting with a primary robot arm. With the popularization of smartphones and tablets, human-robot interaction with multi-touch interfaces has also been presented (Frank et al. 2016; Hashimoto et al. 2011).

With the emergence of low-cost commercial head-mounted devices in recent years, several immersive teleoperation interfaces have also been developed. Whitney et al. (2018) developed the "ROS Reality" package that allows users to teleoperate robots in Virtual Reality (VR). Sukumar et al. (2015) used stereo cameras to create a remote Augmented Reality (AR) environment in VR goggles for immersive teleoperation.

Although teleoperation can reduce programming workload and protect operators from danger, it has several limitations. First, the robot is moving at the same time that the human is operating. While real-time operation has some benefits, it poses additional safety risks since operators have a limited perception of the robot working environment. Second, human operators need to figure out and lead the robot through the full path of manipulation. The effort could be spared in certain cases by making better use of robot intelligence. Lastly, teleoperating robots with multiple DOFs has a steep learning curve. Comprehensive training and expertise are required for

human operators. Furthermore, operation difficulties and safety risks increase in the case of construction robots since they usually manipulate large and heavy workpieces.

1.3 Research Objectives

Lundeen (2019) proposed planning, cognitive, and physical capabilities as three vectors of human and robot capabilities in HRC tasks, which respectively represent high-level operational planning capability, adaptive trajectory and motion planning capability, and physical work capability. In existing construction HRC systems, robots perform physical work and human co-workers perform planning and cognition tasks (Figure 1.3a) (Chung et al. 2010; Liang et al. 2020b; Shum et al. 2013). These systems greatly reduce construction workers' physical stress by relieving them from strenuous physical construction activities. However, the robot's capability is only limited to the physical work that can neither be preprogrammed nor be autonomously performed by the robot because of the complexity and uncertainties of construction tasks (Liang et al. 2021b; Lundeen et al. 2019).

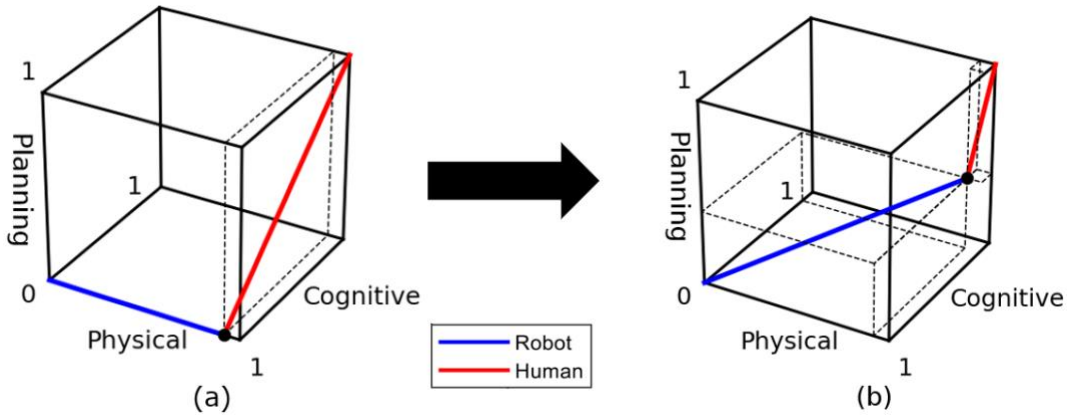


Figure 1.3 Research Impact on Human-Robot Effort Distribution

Therefore, the overall objective of this research is to reduce both the physical and mental effort required of construction workers by enabling human-robot partnerships that leverage robot

planning and cognitive capabilities in addition to their physical prowess in the performance of digitally-driven field construction work (Figure 1.3b).

The specific objectives of this dissertation are as follows:

- Develop and verify a digital twin system to enable human workers to remotely collaborate with construction robots to perform field work.
- Investigate methods to leverage the availability of ubiquitous BIM to improve the autonomy and accuracy of human-robot collaborative construction (HRCC) and record informative construction data to enable future robot learning.
- Automate the digitally-driven construction workflow to interface the digital twin system with various construction tasks.
- Deploy the closed-loop digital twin system to perform physical robotic construction work and evaluate its feasibility and performance in laboratory settings.
- Explore how to teach robots to automatically develop the motion sequences to perform various types of construction assembly tasks.
- Explore a technical and social framework that can serve as a baseline for enabling widespread acceptance and deployment of HRCC in the construction industry.

1.4 Dissertation Outline

This dissertation is a compilation of scientific manuscripts which document this research in creating a digital twin system that connects human workers, robots, BIM, and construction sites for collaborative construction and the subsequent steps to improve system autonomy, accuracy, and intelligence to reduce human effort during the interaction process. The remainder of the dissertation is organized as follows.

Chapter 2 introduces the development of an immersive and interactive process-level digital twin system as a platform for human workers to remotely collaborate with construction robots. A multi-shape drywall installation case study in the Gazebo Simulation environment is presented as a proof-of-concept implementation of the system. In addition, a human-in-the-loop study is conducted for system verification and collecting feedback for future improvements.

Chapter 3 describes the integration of BIM into the digital twin collaboration platform to improve the accuracy and autonomy of the human-robot collaboration process and to save as-built construction data. The deployment of the system onto the physical industrial robotic arm for in-laboratory construction tasks is investigated and used to verify the system. Robot movement and visual detection accuracy are used to evaluate the system performance.

Chapter 4 presents a robotic high-level motion sequence LfD framework. By learning from human demonstrations in the interactive digital twin system, robots automatically make sequential decisions in selecting construction skill primitives to perform construction work. Three different types of simulated construction tasks and their corresponding number of human demonstrations required are used to present the LfD process. A wooden shelf construction task in Gazebo simulation is conducted to verify the LfD-integrated digital twin system.

Lastly, Chapter 5 provides a summarizing conclusion of the dissertation as a whole. Specifically, it discusses the significance and contributions of the research. Finally, future research directions are articulated.

Chapter 2 Interactive and Immersive Digital Twin for Human-Robot Collaborative Construction

2.1 Introduction

Robots are adept at manipulating objects with high precision and perform tasks repetitively, and thus offer a promising alternative to relieve construction workers from physically demanding and repetitive tasks (Liang et al. 2019; Xu and Garcia de Soto 2020). Robots also allow some construction work to be conducted remotely and facilitate social distancing on-site so that construction projects are not significantly interrupted by unexpected circumstances such as the Covid-19 pandemic. Robots' reasoning intelligence in scene understanding, motion planning, and adaptability experienced rapid growth in recent years because of the progress in artificial intelligence and computational power (Brosque et al. 2020). However, construction robots face several challenges due to the unstructured and complex nature of construction environments and relatively loose tolerances of construction projects, which may lead to frequent robot failures while performing tasks on-site (Lundeen et al. 2019; Milberg 2006).

While robots are competent in accurate and repetitive manipulation of heavy workpieces, detection of minor deviations, and numerical computation, human beings are better at creative planning and sequencing based on domain knowledge, experience, and perceptual understanding (Seong et al. 2019; Sharif et al. 2016). Considering drywall installation as an example, when the wall frame deviates from the design, the human carpenter will tune the drywall panel or adjust nailing angles to ensure that the panel is firmly connected to the wall frame, which is acceptable

to a certain extent (NAHB 2011). Although human workers can quickly improvise a new plan in such situations, it is difficult for a robot to make decisions adaptively when it comes across unknown situations. Therefore, human expertise and abilities to improvise (i.e., adjust the plan according to the current circumstance that differs from the design) play a crucial role in overcoming these uncertainties and are indispensable during the construction process, making it unrealistic to completely replace construction workers with robots (Kyjanek et al. 2019).

Despite the significant promise of HRC, current techniques for humans to interact with large, industrial, construction robots are still inefficient (Kyjanek et al. 2019). One of the most common HRC methods for construction robots is teleoperation. However, it suffers from limited perception and accuracy reduction (Roldán et al. 2019). When it comes to mobile robotic arms with multiple degrees of freedom (DOFs) carrying large and heavy workpieces, a significant amount of training is required for operators since any errors or lapses can cause collisions and other safety issues (Hashimoto et al. 2011). Another typical HRC method in construction is to lead the robot by putting forward physical forces on the robot itself or the workpiece carried by the robot, which is also fraught with safety issues since the worker needs to intimately share the workspace with the robot (Chung et al. 2010). Moreover, both of these techniques do not take advantage of robot intelligence in reasoning and still require human workers to continuously perform manual tasks during the whole work process.

In order to overcome the limitations of existing HRC techniques and allow construction workers without robot programming expertise to seamlessly communicate with and intuitively operate robots for on-site construction work, an interactive and immersive process-level digital twin (I2PL-DT) system has been proposed to enable human-robot collaborative construction (HRCC). The human co-worker is responsible for high-level task planning and supervision and

the robot undertakes detailed workspace sensing and monitoring, cognition (i.e., motion planning), and physical execution of the work (Figure 2.1).

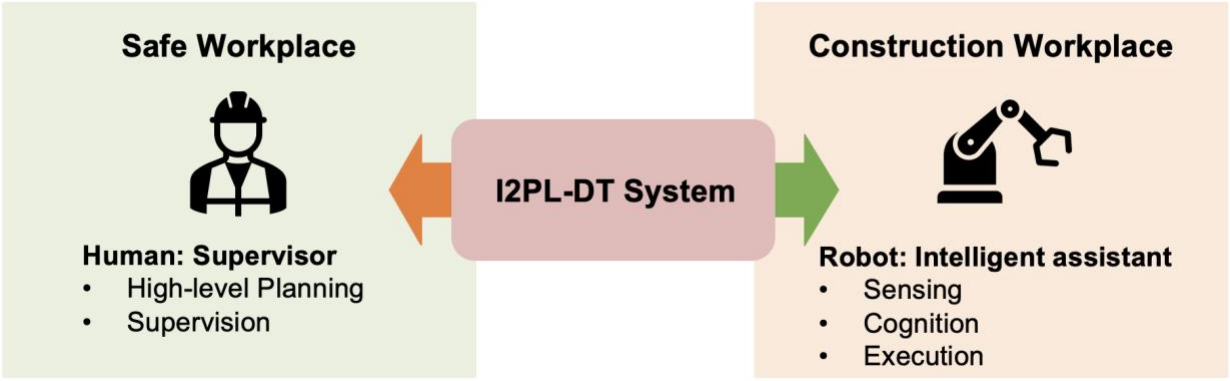


Figure 2.1 Chapter 2 Research Objective

The characteristics of the enabled collaborative workflow are as follows: (1) the workspace is continuously sensed and monitored by the robot and the information can be visualized by human workers through the VR digital twin; (2) human co-workers can perform high-level task planning and send task objectives and commands to the robot intuitively with the VR interface; (3) the robot can automatically develop collision-free motion plans and demonstrate the plans to human upon receiving requests from human; (4) human co-workers can preview the motion plans and approve a desirable plan for execution; (5) the robot can physically execute the approved plan to perform the task while the human worker supervising the execution process in VR. As shown in Figure 2.2, the overall goal of this chapter is to improve robot cognitive capability during the HRCC process to reduce cognitive effort.

Allowing workers to interact with on-site robots from remote locations has the potential to reduce the number of on-site workers or facilitate their physical separation. In addition, with the help of immersive VR, women and people with disabilities (e.g., wheelchair users) can also perform construction work in collaboration with construction robots, offering potentially game-

changing benefits towards making the construction workforce more inclusive. In order to evaluate the system and obtain feedback for future improvements, a drywall installation case study involving imperfect rough carpentry (wall framing) together with a human-in-the-loop experiment is conducted.

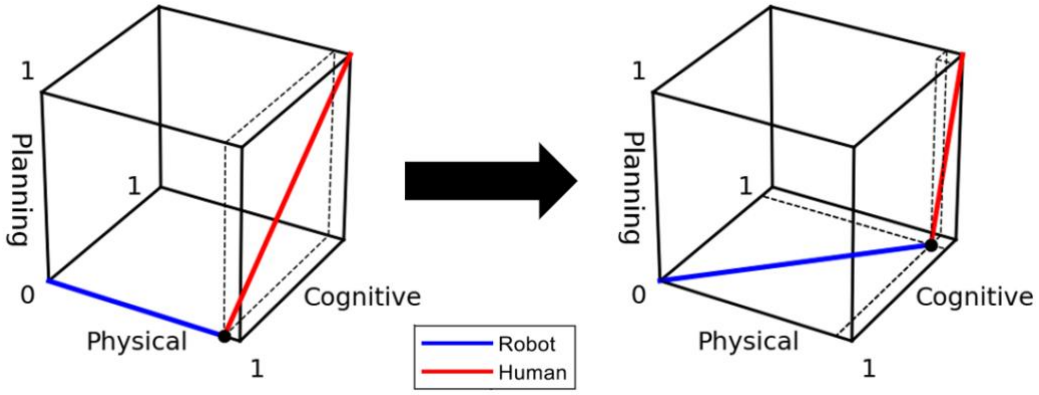


Figure 2.2 System Impact on Human-Robot Effort Distribution (Chapter 2)

2.2 Related Work

2.2.1 Digital Twins in Robotics Applications

The concept of digital twins has become increasingly popular in recent years with the growth of sensing and computing capabilities and visualization technologies (Bilberg and Malik 2019). Digital twins include a virtual representation of their original entity (i.e., the physical world) that contains necessary and pertinent information from it. Most importantly, digital twins also include data communication capabilities that connect and synchronize the digital world with the original entity and exchange information between them (Deng et al. 2021a). Such communication capabilities differentiate digital twins from 3D simulations and are inevitable elements of digital twins (Grieves 2014; Wang et al. 2017).

Digital twins have been used for several robotic applications in the manufacturing industry. For example, Kuts et al. (2019) proposed an Industrial Digital Twin to program motions for an industrial robot arm to repeat in manufacturing process. Bilberg and Malik (2019) used a digital twin-based simulation for dynamic task sequence arrangement and allocation between a human and a robot in an assembly cell. Digital twins have also been used for HRC assembly system validation (Malik and Brem 2021) and safety protection while humans shared the workspace with robots (Maragkos et al. 2019). Liang et al. (2022) also developed a synchronization system to connect construction robots and digital twin simulations. However, the application of digital twins to construction robots is still limited.

2.2.2 Immersive Augmented Reality (AR), VR, and Mixed Reality (MR) Technologies in HRC

With the emergence of low-cost commercial immersive devices, immersive technologies, including AR, VR, and MR have been introduced to facilitate HRC from different aspects, including robot teleoperation (Sukumar et al. 2015; Whitney et al. 2018), joint angle control (Kuts et al. 2019), task objectives specification (Roldán et al. 2019; Wang et al. 2020), trajectories planning (Kyjanek et al. 2019), and robots' intention indication (El Hafi et al. 2020; Walker et al. 2018).

Immersive technologies have also been utilized to study HRC in the construction industry. In a beam welding task, AR was used to show target welding positions so that the human operator can adjust the beam position for robotic welding (Tavares et al. 2019). Several studies have been conducted to study construction workers' reactions when they share the workspace with robots in order to develop safe HRC mechanisms (Kim et al. 2015; You et al. 2018). There are extensive studies in construction using immersive technologies for visualization, design, safety, and training purposes (Li et al. 2018; Liu et al. 2020). Zhou et al. (2020) used VR-based robot teleoperation for

civil engineering applications. However, the application of immersive technologies for construction robot operation is limited in practice.

2.2.3 Comparison of I2PL-DT and Existing Studies

An efficient HRC system for construction must possess the following properties: first, human co-workers should be able to assist on-site construction robots to overcome loose tolerances and design deviations through effective guidance, instructions, and communication mechanisms; second, the system should relieve human physical and mental effort by transitioning the role of human workers from physical task performers to robot supervisors; last but not least, the system should ensure the safety of both human workers and construction site property with safeguards and collision avoidance mechanisms.

Based on the nature of the construction industry, we propose seven characteristics useful for HRC systems in construction, as shown in Table 2.1. The proposed characteristics include necessary information and functions that support human workers' remote interaction with on-site construction robots. Several closely related prior studies from a variety of applications are selected and the proposed characteristics they included are summarized in Table 2.2. The scale of the presented case study and the scale of objects manipulated in each study are summarized in the last column.

Table 2.1 Highlighted Characteristics for HRC Systems in Construction

Number	Characteristics
1	Human interaction from remote locations
2	Real-time visualization of the physical environment (if remote)
3	Augmented information useful for supervision purposes
4	Hierarchical task planning (high-level human task planning and improvisation for uncertainties and low-level robot automation for cognitive planning)
5	Robot motion plan preview and evaluation
6	Real-time robot execution process and status supervision
7	Bi-directional communication between the human and the robot

Table 2.2 Prior Studies Characteristics Summary

Study	Type	Characteristics							Case (Object) Scale
		1	2	3	4	5	6	7	
Kyjanek et al. (2019)	AR		–	work progress, detailed robot status		trajectory line	directly	√	Prefabrication (Small)
Roldán et al. (2019)	VR	√			drag end effector		directly	√	Tabletop (No)
Kuts et al. (2019)	VR	√		detailed robot status	specify end effector	robot movement	robot synchronization	√	Tabletop (No)
Ong et al. (2020)	AR		–	plan assistance	select workpiece feature		directly	√	Tabletop (Small)
Frank et al. (2016)	MR		–	estimated object pose	drag block on tablet		directly		Tabletop (Small)
Sukumar et al. (2015)	AR	√	stereo vision	actual and target position			directly through camera	√	Tabletop (Small)
Zhou et al. (2020)	VR	√	virtual screens				with teleoperation	√	Tabletop (Small
Cichon and Roßmann (2018)	VR	√	point clouds + models	detailed robot status				√	Tabletop (Small
This study	**VR**	√	**point clouds + meshes + models**	**workspace BIM, high-level robot status**	**object target pose**	**robot + object movement**	**robot + object synchronization**	√	**Drywall Installation (Large)**

Although there are several existing studies utilizing immersive technologies or digital twins for robot operation, they cannot be directly applied to construction projects. On one hand, most of these systems are at a tabletop scale with a fixed robotic arm manipulating small objects. The same HRC approaches cannot be simply scaled up to construction tasks where both the robot workspace and the workpieces are much larger than typical human workers. For instance, specifying the end-effector position as task goal (Characteristic 4) or previewing the trajectory line for plan evaluation (Characteristic 5) is sufficient for manipulating small objects but is not adequate when workpieces are large (e.g., drywall panels). When the robot is manipulating a large object, it is critical to show how the object will move along with the robot to human co-workers during both the plan preview and execution supervision processes (Characteristics 5 and 6) to evaluate whether there are safety concerns. On the other hand, the execution process of construction robots involves significant uncertainties and less repeatability. As a result, HRC for construction robots needs a more intuitive approach that allows frequent human intervention rather than setting up a series of movements for the robot to repeat over an extended period of time, which should be considered for system design.

It should be noted that these HRC systems are highly configurable and customizable. Characteristics are implemented differently in each study, depending on their scale, applications, and focus. Take Characteristic 3 as an example, previous studies mainly show augmented information such as detailed robot status (e.g., joint angles), work progress (e.g., progress percentage), or workpiece-related information. This study focuses on providing more intuitive high-level robot status (e.g., finding motion plans) and environment-related information that facilitates human inspection and supervision. For Characteristic 2, this study combines 3D BIM, reconstructed 3D meshes, and point clouds to enable real-time visualization while reducing computational resources. We used markers for Characteristic 1 to show whether the system allows

remote operation and Characteristics 7 since the content of bi-directional communication varies depending on the system needs.

In an effort to remedy the identified gaps in knowledge and current capabilities, the objective of the presented research is to develop an HRCC system that is capable of conducting construction tasks with large involved workpieces and can offer interactive communication abilities to construction workers without robot programming expertise. Towards this end, an I2PL-DT HRCC system that covers all seven characteristics is proposed.

2.3 Technical Approach

2.3.1 System Overview

The proposed I2PL-DT system integrates an immersive VR interface for human interaction, middleware for computation and communication, and a robot operational environment (ROE) for sensory data collection and construction task execution. The system framework is presented in Figure 2.3. The immersive VR interface, developed on the Unity platform, allows human workers to interact with robots remotely with an augmented telepresence experience. The ROE is the construction environment in which the robot performs tasks. It consists of the robot, the construction workspace, and sensors in the environment. The immersive VR interface is connected to the ROE via the Robot Operating System (ROS) as the middleware (Quigley et al. 2009). The middleware acts as the bridge between the human and the robot in ROE. It receives and processes data from both the immersive VR interface and the ROE, performs computation based on the information presented, and publishes processed data to corresponding clients.

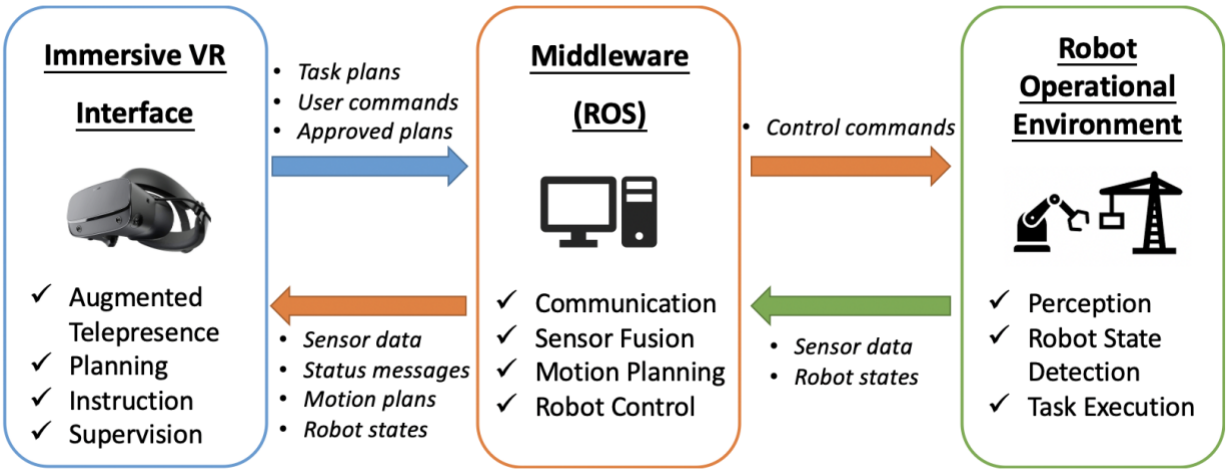


Figure 2.3 HRC System Framework

The general system workflow is shown in Figure 2.4, in which the role of the human and the robot (i.e., middleware, sensors, and the actual robot in ROE) are illustrated. *Workspace sensing and monitoring* are conducted as the system is initiated. The as-built workspace environment and robot states captured by the sensors in the ROE are processed by the middleware and sent to the VR interface continuously to be relayed in the human view. The human co-worker can perform *site inspection* by comparing this as-built workspace geometry with the as-designed BIM in VR and based on this inspection, perform *high-level task planning* to make decisions on work sequence, the workpiece to manipulate, installation position, etc. High-level planning is achieved by interacting with objects and the information dashboard (billboard) in VR. At this stage, the human can test and compare different options without physical stress or risks from repetitively manipulating heavy construction materials. The human co-worker can confirm the task plan and send it to the middleware if satisfied.

The middleware processes the high-level task plan into specific goals for robot *motion planning*. The motion planner of the robot then generates several collision-free motion plans to achieve the goals. In the meantime, the human co-worker can visualize the planning status

messages of the robot from the billboard in VR. The motion plans are converted into robot states and published to VR for *demonstration* while the human co-worker *previews and evaluates* the plans on the virtual robot model. The motion plans are shown as full-scale realistic animations showing the robot and workpiece movements that will happen at the execution stage. After viewing the motion plan, the human co-worker can approve the plan if satisfied or request a new plan demonstration.

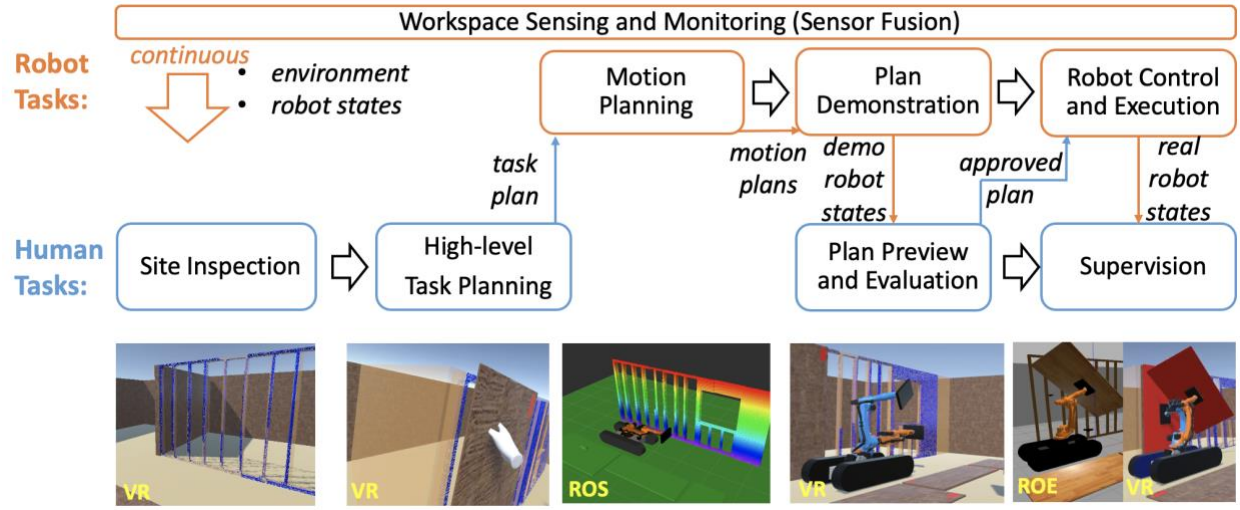


Figure 2.4 System Workflow and Human-Robot Roles Distribution

At the same time, the middleware is notified of which motion plan has been approved by the human co-worker. It starts to *control* the actual robot in the ROE to *execute* the approved motion plan. Joint states of the actual robot are continuously captured by the encoders on the robot actuators and sent to the middleware, which are then relayed to the VR interface. Another virtual robot in VR synchronizes its joint states with the actual robot in the ROE in real-time based on the state messages received. Therefore, the human co-worker can *supervise* the actual robot states as it executes the task. In addition, the human co-worker can also obtain robot execution status messages from the billboard in VR.

In the remainder of this section, the technical approaches for developing the immersive VR interface and middleware are discussed in detail. The establishment of the ROE varies case-by-case and thus is discussed later in the case study.

2.3.2 Immersive VR Interface

This study uses an immersive VR interface for several reasons. First, compared to AR and MR, immersive VR allows human co-workers to be present at remote locations away from the construction site, something that is particularly helpful to reduce construction site congestion and improve safety. For example, it can facilitate social-distancing requirements during periods such as the Covid-19 pandemic without compromising the progress of the work. Second, immersive VR provides realistic experiences to users. Human co-workers can navigate in the immersive VR environment and can observe objects from different perspectives just as they would do in the real world. This overcomes the limited field of view and depth perception of traditional teleoperation approaches and provides freedom for human operators to easily switch observation perspectives (Chen et al. 2007; Roldán et al. 2019). Furthermore, users can overcome some constraints of the real world within immersive VR. For example, human co-workers can defy gravity to "fly" near the roof or move construction materials around without being encumbered by their physical weight. They can also receive augmented information that cannot be directly obtained from the real world. Studies also show that the robot operator's situational awareness is improved while working in VR (Roldán et al. 2017; Ruiz et al. 2015).

2.3.2.1 Immersive Virtual Environment (IVE) Construction

The IVE is the digital twin of the ROE, where the human co-workers can perceive real-time construction workspace conditions, robot states, and augmented information such as the as-

designed building geometry from remote locations. It consists of a virtual construction environment and two full-scale virtual robots (Figure 2.5). One robot demonstrates the motion plan to the human co-worker, referenced as the "planning" robot in the rest of the paper (Figure 2.5a). The other robot, referred to as the "supervising" robot (Figure 2.5b), is synchronized with the actual robot (Figure 2.5c) so that the human co-worker can supervise the actual robot's execution process. The robot digital model, represented in URDF format, has the same size and configuration as the actual robot. It is first loaded in ROS and then transferred to VR via the ROS# library to be loaded as a game object. The VR robot models preserve the kinematic properties of the actual robot and can be controlled by subscribing messages from the middleware.

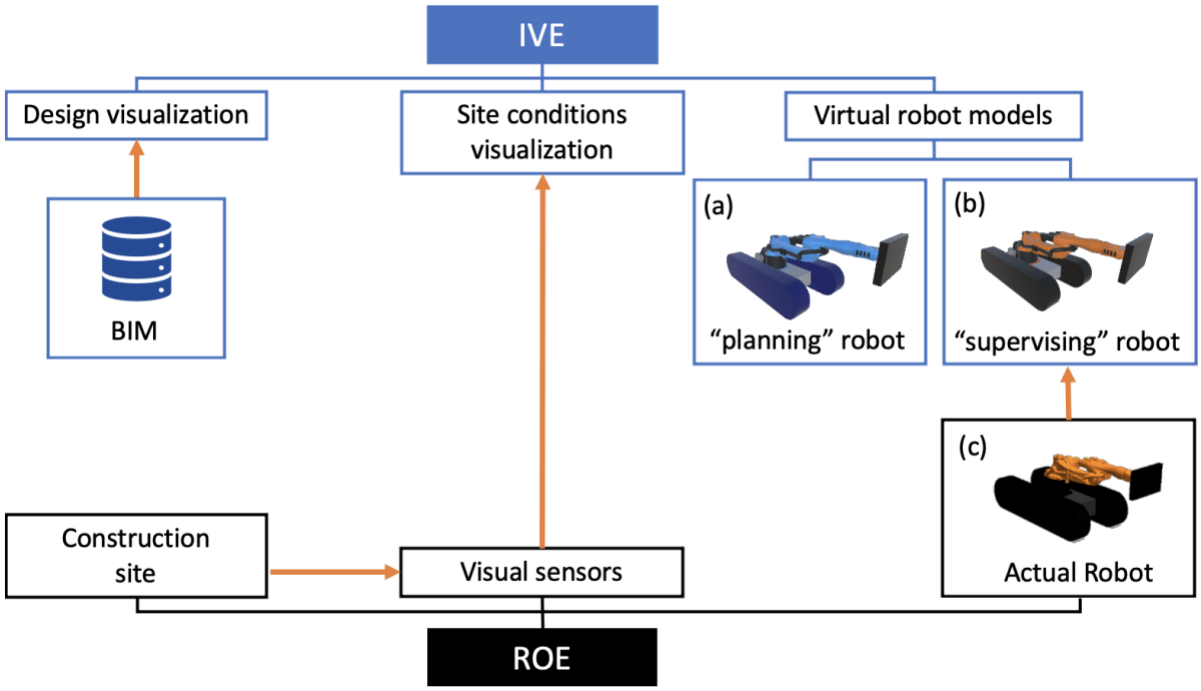


Figure 2.5 IVE Components Scene Graph

Some studies used 3D Computer-Aided Design (CAD) building models, such as BIM as VR construction environments (Du et al. 2018). It can be loaded into IVE conveniently. However, this approach does not reflect the latest construction site conditions because the as-built structure

may deviate from the design. It also cannot capture the moving workers and temporary equipment and structures on-site during the construction process, which should be considered for task planning. Point clouds of real-time construction site conditions can be captured using laser scanners or depth cameras but it is expensive to render large point cloud data in VR because of its high refresh rate (Fang et al. 2016). Wang et al. (2019) generated BIM from point clouds, which can be imported into the IVE. However, the dataset labeling and training processes consume significant labor and computation resources.

In order to visualize actual construction site geometry in near real-time (i.e., with minimal delays caused by automatic data processing and electronic transmission (US DOD 2005)) while reducing the computational load and time delay, this study proposes a hybrid approach to create the IVE of the construction site. Components in the construction environment are first grouped into three categories, non-critical components, already-built structures, and dynamic objects, as shown in Figure 2.6. Non-critical components indicate objects outside the robot workspace (e.g. walls outside the workspace) or components inside the robot workspace but with limited deviations from design that do not affect the human co-worker's decision-making or robot's operation processes (e.g., ground floor). For non-critical components, their as-designed BIM is directly used in the VR scene as a realistic working environment for the human worker.

Already-built structures are static building components or temporary structures inside the workspace (e.g., columns and formwork) and are closely related to the human co-worker's decision-making or robot operation process. The as-built geometry of these structures is captured by depth cameras or laser scanners on-site as point clouds. The point clouds are reconstructed into 3D meshes in the middleware and sent to the immersive VR interface via ROS# to be loaded as scene objects in IVE (Bischoff 2020). The reasons for converting point clouds of already-built

structures into 3D meshes are two-fold. First, it could significantly reduce the system computational load of refreshing large-size real-time point cloud data at every frame. Second, it allows colliders to be added to the already-built structures for collision avoidance during the high-level task planning process. Their BIM is also loaded into the IVE. However, these models are set as semi-transparent and are only used for visualization purposes to show human co-workers any discrepancies between the as-designed and the as-built structure.

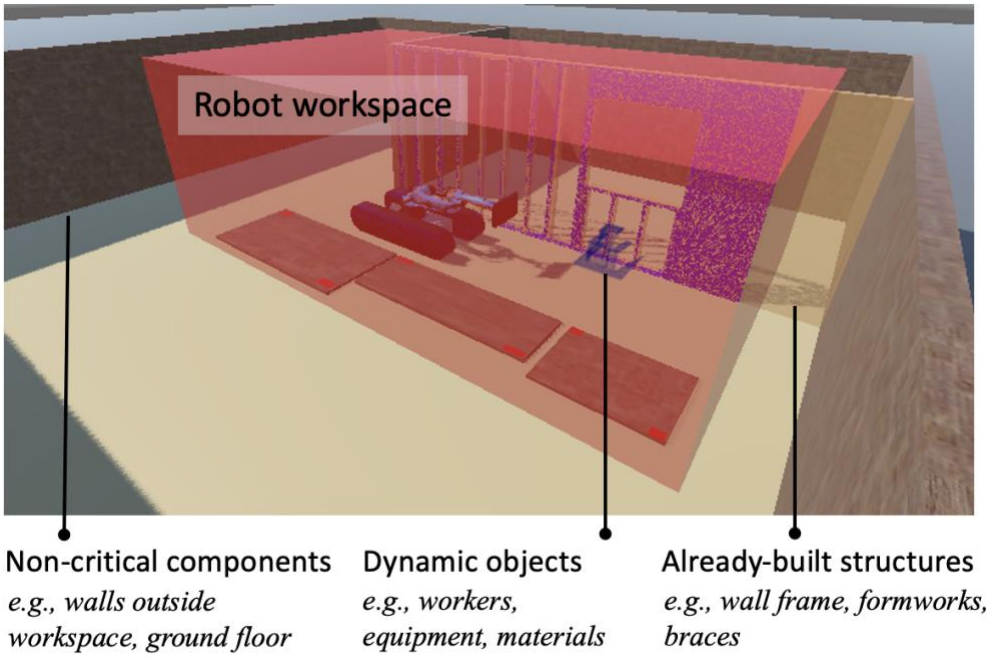


Figure 2.6 Hybrid IVE Construction

The dynamic objects include human workers and movable equipment that might intrude into the robot workspace and obstacles that temporarily stay in the robot workspace, which will affect human co-workers' decision-making and robot operation. It is critical to track these objects in near real-time because they may be present and move in the robot workspace at any time. Once dynamic objects appear in the robot workspace, their point clouds captured by depth cameras or laser scanners on-site are rendered in the VR scene so that the human co-worker can view construction environment conditions in real-time.

In the proposed approach, the categories of the components are decided by manually defining regions or selecting components. However, the proposed framework can be integrated with building components detection and recognition algorithms (Bassier et al. 2019; Sharif et al. 2017; Wang et al. 2020) to automatically detect and classify components from point clouds into proposed categories. In addition, the Greedy Projection Triangulation algorithm has been used for point cloud 3D reconstruction (Marton et al. 2009). Nevertheless, other point cloud reconstruction approaches could be used based on the needs of different cases.

2.3.2.2 VR Interface Development

The immersive VR interface acts as a visualization tool for augmented telepresence experience, a planning tool for human co-workers to perform high-level task planning, and a supervision tool for robot motion plan evaluation and real-time status supervision. It contains several interactive elements for the human worker to perform task planning, guide the robot, and receive information. One of them is the interactive billboard. It shows instructions to human co-workers and system messages. It may contain an internal user interface (e.g., buttons and sliders) inside the VR scene as a supplement for handheld controllers to provide the human worker with additional functions sending commands and interacting with the system.

The interface also includes some task-specific interactive elements as part of the VR scene. For example, for the construction assembly activities, the construction workpieces to be installed (e.g., bricks, panels) are included as interactive game objects in the VR interface. These interactive materials are of the same size and position as the actual construction materials on-site and can be grabbed, moved, and suspended in the air, based on task needs. Human co-workers can use these elements to perform high-level task planning, indicate task goals, and guide the robot.

Several features have been developed in the interface to facilitate decision-making and interaction processes (Figure 2.7). The first feature is scene scale and viewpoint elevation adjustment. The human co-workers are given the ability to adjust the scene scale to be larger or smaller than the real world with handheld controllers during the interaction process (Figure 2.7a). The contracted scene can be used for general planning and supervision, while the enlarged scene can be used for detailed inspection and material pose fine-tuning. Furthermore, the huma co-worker can adjust the elevation of their viewpoint to move around at any desired elevation to obtain an overview of the construction environment and inspect the geometry from the roof level (Figure 2.7b).

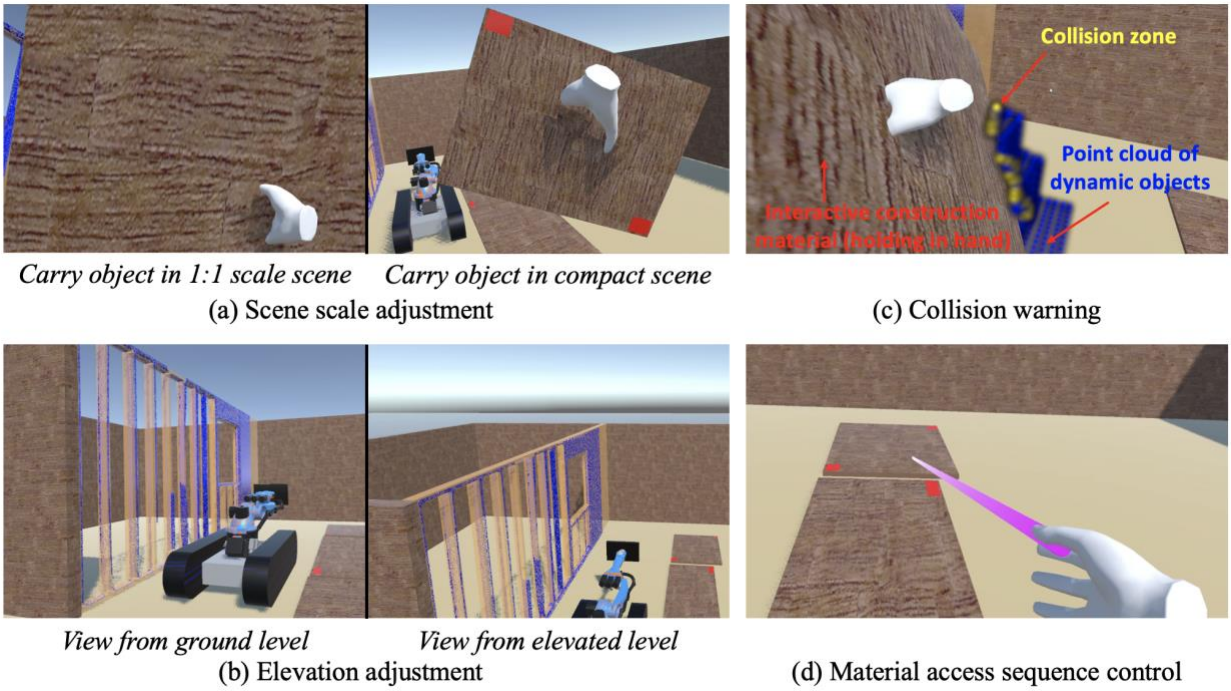


Figure 2.7 Immersive VR Interface Interaction Features

The second feature is collision avoidance and checking. Colliders are added for the interactive construction materials, BIM of non-critical components, and the 3D meshes of already-built structures. It provides collision protection at human co-workers' high-level planning stage

because the worker cannot place the construction materials in collision with the built structures. In addition, when workers place the construction materials in collision with the dynamic objects, the part of the point cloud with collision will change its color as a warning. As shown in Figure 2.7c, the interactive construction material (wood panel) that the user is holding in hand collides with the point cloud of stacked boxes. As a result, the part of the point cloud that collides with the panel changes its color from blue to yellow.

The third included feature is material access sequence control. Following the practical convention, the system only allows the user to interact with materials stacked on the surface (Figure 2.7d). Once the material on the surface is removed, the material lying underneath is then set to be interactable. It should be noted that although this paper mainly discusses pick-and-place related cases, the system can be generalized to many different types of construction tasks (e.g., nailing, joint filling) after configuration.

2.3.3 Middleware

ROS is used as the middleware for the proposed system, which is an integrative open-source robotic software framework (Quigley et al. 2009). It supports and can communicate with a variety of sensors, hardware, and robots. However, it is impractical for construction workers without robot programming expertise to operate robots directly through ROS. One of the reasons is that ROS is developed as a tool to facilitate robot programming. Although some software libraries in ROS provide operator interfaces, their availability and functionality are limited (Roldán et al. 2019). It is insufficient and is not intuitive to use when it comes to complex construction tasks that typically involve several procedures and objects. Therefore, in our framework, ROS is utilized as the middleware for communication between the human and the robot, robot motion planning, sensor fusion, and robot control. In this section, the techniques to establish the

communication framework and to conduct robot motion planning and robot control are introduced. Sensor fusion varies case-by-case and thus will be discussed later in the case study.

2.3.3.1 Communication

The immersive VR interface and middleware communicate by exchanging ROS messages, which include a variety of formats based on message types. The communication is established using ROS#, which is an open-source library developed for connecting ROS and Unity (Siemens 2021). ROS can exchange messages with robots and their embedded sensors and environmental sensors with the MQTT communication protocol (Liang et al. 2020a). ROS can also communicate with robotics simulation software if ROE is in simulation. For example, an open-source meta-package, gazebo_ros_pkgs, can be used to exchange messages between ROS and the robot and sensor emulators in the Gazebo simulator (Open Robotics 2021).

2.3.3.2 Robot Motion Planning

The motion planning method discussed in this study is based on the mobile industrial arm manipulator, which is a general case for construction robotics. Industrial robotic arms offer high DOFs and payload and have high flexibility to be configured for a variety of complex construction tasks (Bock 2007; Liang et al. 2020b).

The motion plan is considered separately for the robot mobile base and the robotic arm. The robotic arm movement is given a higher priority than the mobile base movement. In other words, the robot will only move its base if its arm cannot find a motion plan directly to reach the target position from the target base location. This setting aims at reducing the localization error caused by frequent robot base movement. The robotic arm motion plan is generated by MoveIt

(Chitta et al. 2012). The point cloud sensing data of the environment is processed into a 3D occupancy grid map with Octomap for collision avoidance (Hornung et al. 2013).

The task plan received from VR is first converted into the corresponding robot end-effector pose in ROS. Then, the Open Motion Planning Library (OMPL) (Sucan et al. 2012), which integrates several cutting-edge sampling-based motion planning algorithms, is used together with the Flexible Collision Library (Pan et al. 2012) to generate kinematics (i.e., position, velocity, and acceleration) of each joint to reach the goal without collision. If the robot is carrying a workpiece, the workpiece is considered as part of the robot during the motion planning and collision checking process. As a result, both the robot and the workpiece carried by the robot will not collide with the environment or with each other. The motion plan is only considered to be successful if it is collision-free.

The algorithm for motion planning is shown in Figure 2.8. After receiving the target $\boldsymbol{T}$, the robot first checks whether there are any dynamic objects in its workspace by checking if there are point clouds other than the ones that represented the already-built building structure. If any point cloud of dynamic objects $\boldsymbol{PCL_D}$ is detected, the system will save it for future comparison at the execution stage. Then, the robotic arm attempts to find a motion plan $\boldsymbol{MP}$ to reach $\boldsymbol{T}$ from its original base location $\boldsymbol{B_0}$ (i.e., location its base stays at without moving). If the robotic arm cannot find a plan after several attempts, it will try to move its base to a target base location $\boldsymbol{B_T}$ while holding the arm at an intermediate object carrying pose $\boldsymbol{P_c}$ (Figure 2.9). The user can define the criteria to determine a set of $\boldsymbol{B_T}$ options $\boldsymbol{S_{BT}}$, which may contain one or multiple $\boldsymbol{B_T}$ near the target. For example, in our system, it is the nearest available location to the target on a specific path.

Algorithm 1: Pseudo Code for Motion Planning

Input: target T,
point cloud of dynamic objects PCL_D,
robot current pose P,
intermediate carrying pose (relative to robot base) P_c
Output: motion plan MP, success indicator $planSuccess$

if $\exists$ PCL_D **then**
  Save PCL_D
end
if $\exists$ *Plan(P to T)* **then**
  $MP \leftarrow$ Plan(P to T)
  $planSuccess \leftarrow$ True //success (without moving base)
  return
end
$B_0 \leftarrow$ getBaseLocation(P)
$S_{BT} \leftarrow$ getBaseLocations(T)
find B_T in S_{BT}, $\exists$ $MP_m \leftarrow$ Plan(B_0 to B_T)
if no B_T found **then**
  $planSuccess \leftarrow$ False // fail (no valid base movement path)
  return
end
$P_0 \leftarrow$ getPose(P_C, B_0)
$P_T \leftarrow$ getPose(P_C, B_T)
$MP_A \leftarrow$ Plan(P to P_0)
$MP_B \leftarrow$ Plan(P_T to T)
$MP \leftarrow$ concatenatePlan(MP_A, MP_m, MP_B)
$planSuccess \leftarrow$ True //success (after moving base)

Figure 2.8 Pseudo Code for Motion Planning

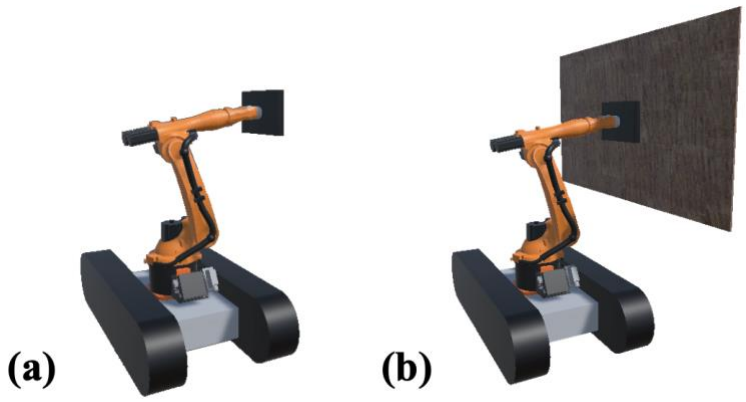


Figure 2.9 Example of Intermediate Carrying Pose (a) Without Object (b) With Object

First, the robot checks whether there is a valid pathway to move its base from $\boldsymbol{B_0}$ to $\boldsymbol{B_T}$ For robots that move forward and backward along a straight line, this process can be achieved by

setting a bounding box along the robot pathway and checking the occupancy within that bounding box using point clouds captured by on-site depth cameras. For robots that can move around the workspace freely, more advanced path planning algorithms like Dijkstra's Algorithm are needed to find a valid path (Dijkstra 1959). If multiple $\boldsymbol{B_T}$ options are determined, the robot will attempt all the potential options in $\boldsymbol{S_{BT}}$ until a valid pathway is found.

Once a valid pathway from $\boldsymbol{B_0}$ to $\boldsymbol{B_T}$ is found, the robotic arm will generate its motion plan $\boldsymbol{MP_A}$ to move to the intermediate carrying pose $\boldsymbol{P_0}$ (i.e., $\boldsymbol{P_c}$ with the base at $\boldsymbol{B_0}$) in preparation for the base movement. It will then generate another motion plan $\boldsymbol{MP_B}$ from the intermediate carrying pose $\boldsymbol{P_T}$ (i.e., $\boldsymbol{P_c}$ with the base at $\boldsymbol{B_T}$) to $\boldsymbol{T}$. These two robotic arm motion plans, together with the robot base movement plan $\boldsymbol{MP_m}$, are combined into the robot final motion plan $\boldsymbol{MP}$.

The system can generate and save several motion plans. The user can specify the cost functions (e.g., time duration) to sort motion plans so that the plans can be demonstrated within a certain order (e.g., time duration from short to long) in the VR interface based on users' preferences. To view the arm plan in VR, we extract discrete joint states from the generated motion plan and publish it to VR at its timepoint specified in the motion plan to move the "planning" robot in VR. For the mobile base movement plan visualization, we simulate the base movement plan in the middleware by selecting discrete location points along the base movement path, publish it to VR at a given frequency, and have the "planning" robot move to certain points while maintaining its arm pose as the intermediate carrying pose.

2.3.3.3 Robot Control and Execution

The approved trajectory plan is converted into robot control commands with the *ros_control* package, which generates output to actual robot actuators with PID controllers according to the motion plan (Chitta et al. 2017). When the robot is executing the task, joint states

and location data from the encoders of the robot actuators are obtained and sent to VR. As the "supervising" robot in VR receives the data, it adjusts its joint states and base location to synchronize itself with the actual robot.

The construction environment is relatively open and dynamic. Even though the robot is not designed to share the workspace with workers and other moveable equipment, they may still accidentally intrude into the robot workspace. Therefore, instead of blindly following the trajectory approved by the human co-worker in VR, safeguard functions are needed during the execution process to prevent accidents.

As mentioned above, at the start of the planning stage, the point cloud of the dynamic objects is saved for future use. These objects are considered for collision avoidance during motion planning. Therefore, it is acceptable if these objects stay at the same place during the execution process. However, if point clouds other than the previously detected ones are detected in the workspace, it means that either the workspace is intruded or the previously detected objects move after motion planning. As soon as an intrusion is detected, the robot will stop emergently. The human co-worker can inspect the site condition and request the robot to replan its motion based on the latest environment. If the robot workspace is very large, it can be separated into different areas. The robot will only stop if a certain area is intruded upon. In addition to the system safeguard functions, the human co-worker has the privilege to stop the robot with the handheld controller at any time.

2.4 Case Study and Experiments

A drywall installation case study involving imperfect rough carpentry (wall framing) is used to demonstrate the proposed I2PL-DT HRC system. For some complex systems, there are several distinct and interdependent technologies and subsystems that need to come together before

the system can be computationally analyzed or applied in real-world settings. For example, a variety of technological advancements (e.g., perception, localization, hardware design) are needed for a construction robot to successfully perform construction activities on-site (Lundeen 2019). Instead of attempting to address all these challenges at once, we are focusing on verifying that the proposed I2PL-DT system framework and its associated modules allow human workers to interact with and collaboratively perform construction tasks with the robot; as well as receiving feedback to further improve our system in the future.

The use of virtual simulators such as ROS Gazebo is the first step of evaluating the feasibility of this new method as indicated by several existing studies such as Lin and Berenson (2016) and Murali et al. (2020). Gazebo is a robotics simulator with a robust physical engine that allows rapid prototyping of robotic tasks and direct subsequent transfer of the methods to the corresponding real robotic platforms (Koenig and Howard 2004). When connected with ROS, Gazebo is capable of communicating with and controlling physical robots in the real world with high accuracy. It has been demonstrated that a physical KUKA KR120 robotic arm can be synchronized with its emulator in Gazebo with average errors of each joint angle less than 2.4e-05 in radians (Liang et al. 2020a). In addition, Gazebo allows emulation of unstructured and dynamic construction site conditions such as generating dynamic objects, which would be especially useful for offline system testing before physically deploying the system for field construction. Therefore, the case study utilizes a 6DOF Kuka industrial robotic arm emulator mounted on a tracked mobile robotic platform, which is capable of construction work. The ROE, including the construction site, sensors, and the robot, is emulated in Gazebo.

With the focus on demonstrating the interaction framework between the human worker and the robot, three assumptions are made and considered reasonable because they have already been

extensively studied in the literature. First, the case study assumes accurate registration of the IVE and ROE (Feng et al. 2015). Second, it is assumed that the construction materials are firmly placed (i.e., no sliding) and their placement locations are known (Son et al. 2010). Third, we assume the robot can accurately localize itself on-site (Lundeen et al. 2017; Xu et al. 2020).

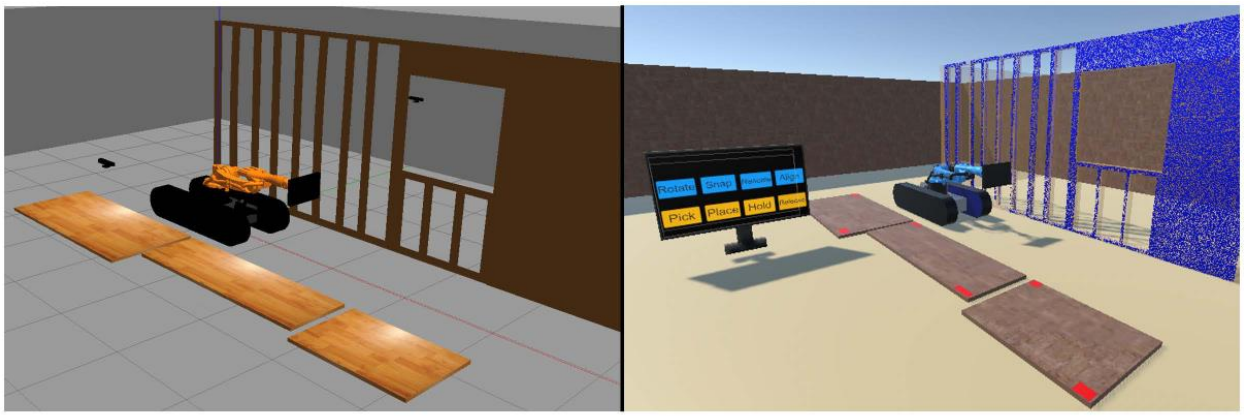
Verification is defined as the "process of evaluating a system or components to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase" (ISO/IEC/IEEE 2017). It involves special tests to model a subsystem (e.g., developing scenarios as proof-of-concept implementation) or using repeating tests to ensure the system meets initial design requirements. For an interfacing system like the one proposed in this study, proof-of-concept implementation is used as verification to confirm that all the modules of the proposed system can work well with each other to reach the goal (Ge and Kuester 2014; Kim et al. 2012, 2021a; Kurien et al. 2018). Some studies also conducted user tests as the preliminary usability study (Akanmu et al. 2020; Chen et al. 2016; Mantha et al. 2020; Quintero et al. 2015). In this chapter, we presented three scenarios from the drywall installation case study as a proof-of-concept implementation. The drywall installation system setup and the technical details of the three scenarios are discussed in depth. A human-in-the-loop study with 20 subjects is also conducted as the preliminary usability study of the system, in which the subject guides the robot to pick up different types of drywall panels stacked on the ground and place the panels on a wall frame that is built with deviations from design. Feedback and suggestions from human subjects are used for system evaluation and to propose future improvements.

2.4.1 Digital Twin Environment Setup

We emulated the ROE in Gazebo (Figure 2.10a). An imperfect wall frame with a window opening has already been constructed. A few pieces of drywall panels in three different sizes are

stacked near the wall frame. A robotics arm emulator on a tracked mobile robotic base is ready for conducting the work. The environment also contains a few Microsoft Kinect camera emulators, which are fixed at certain locations, facing the wall frame and the robot.

The VR digital twin of the ROE is created in Unity (Figure 2.10b). Some stacked drywall panels of the same size and position as the ones in the ROE are created as interactive construction materials, which will be used for high-level task planning and robot guidance. Only the pieces sitting on the top of each stack are activated to be interactable. As the top one is removed, the interactivity of the piece below is activated.



(a) Robot Operational Environment (b) Immersive VR environment

Figure 2.10 Digital Twin Environment Settings

An interactive billboard is developed as an integration of the display media and the internal user interface. The interactive billboard can be separated into three functional zones, as shown in Figure 2.11. The upper zone is used to display augmented robot status messages (e.g., robot planning) and instructions to human workers (e.g., robot needs to pick first). The middle zone is the function panel. It provides some functions to facilitate human co-workers' interaction with the VR scene. The function provided by each button in this panel is summarized in Table 2.3. The bottom zone is the command panel for the human co-worker to send instructions to the robot. It

consists of four buttons, “Pick”, “Place”, “Hold”, and “Release”. The detailed usage of these buttons will be introduced along with the scenarios in the following subsections.

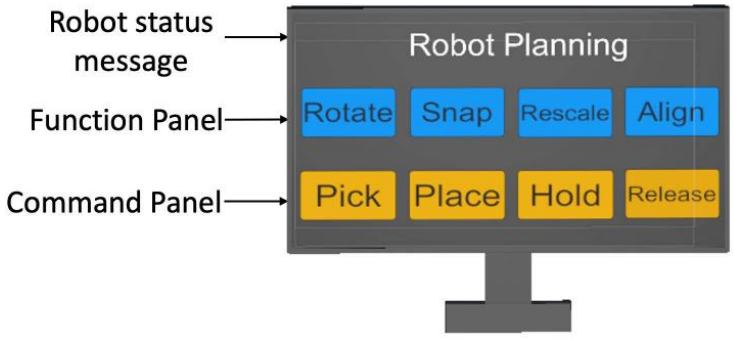


Figure 2.11 Interactive Billboard Functional Zones

Table 2.3 Buttons on the Functional Panel

Button	Function
Rotate	Open a new page with options to snap the target panel orientation along the X, Y, and Z axis to certain angles quickly and accurately.
Snap	Snap the target panel to be side by side and at the same orientation with a designated previously-installed panel.
Align	Show highlighted vertical lines from each corner of the drywall to the ground panel for users to check panel alignment with the wall frame.
Rescale	Quickly rescale the VR scene back to 1: 1 scale.

2.4.2 Sensor Fusion

The type and number of sensors to use and their placement should be decided according to the environment and the type of work the robot will conduct. The sensors should be able to provide sufficient information to support human co-workers’ high-level planning, robot trajectory planning and collision avoidance, and any customized functions to achieve the specific goal of the system. In our case study, four Microsoft Kinect depth cameras are used to visually capture the drywall installation workspace in Gazebo. We chose Kinect since the work is done indoors and Kinect offers acceptable performance under such conditions. They are fixed in the construction environment instead of being installed on-board the robot because the views of cameras mounted

on the robot can be easily occluded when the robot is carrying large construction workpieces. In addition, fixed cameras have lower noise compared to cameras mounted on robots. While depth cameras such as Kinect have relatively lower costs, their performance is limited in outdoor environments because they use infrared sensors to capture depth data and have limited measurement ranges (Liu et al. 2019). Therefore, for outdoor construction tasks or large robot workspaces, 3D laser scanners or stereo cameras that have larger measurement ranges and better outdoor performance should be considered for point cloud capture (Wang et al. 2020a). The process of sensor data processing is shown in Figure 2.12.

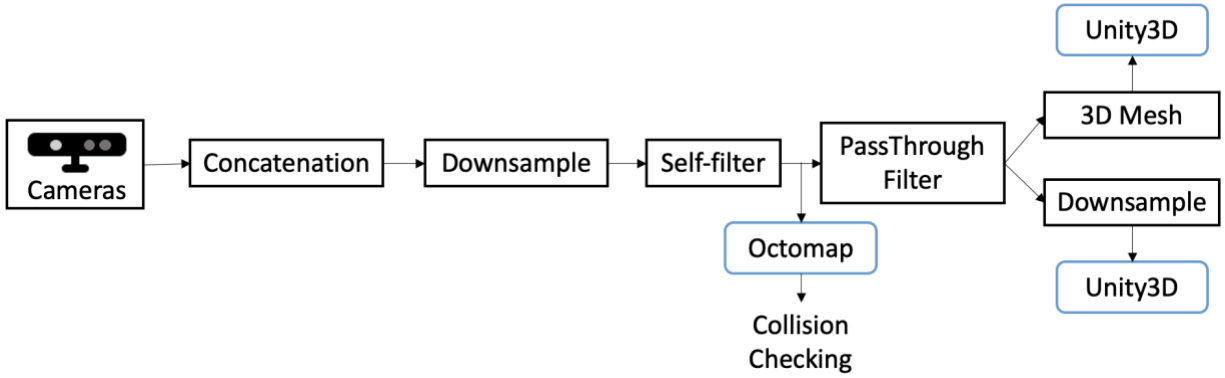


Figure 2.12 Sensor Data Processing

First, the RGB and depth images captured by each Kinect camera are converted into point clouds and concatenated into a single point cloud. The point cloud is downsampled with the Voxel Grid Filter (Figure 2.13a). The downsampled point cloud is then sent to MoveIt and goes through the self-filtering process. Self-filtering removes the points that represent the robot itself (Figure 2.13b). The ground plane, stacked drywall panels, and Kinect cameras installed on-site are added as collision objects in MoveIt. As a result, they are considered for collision checking but their point clouds are removed by self-filtering.

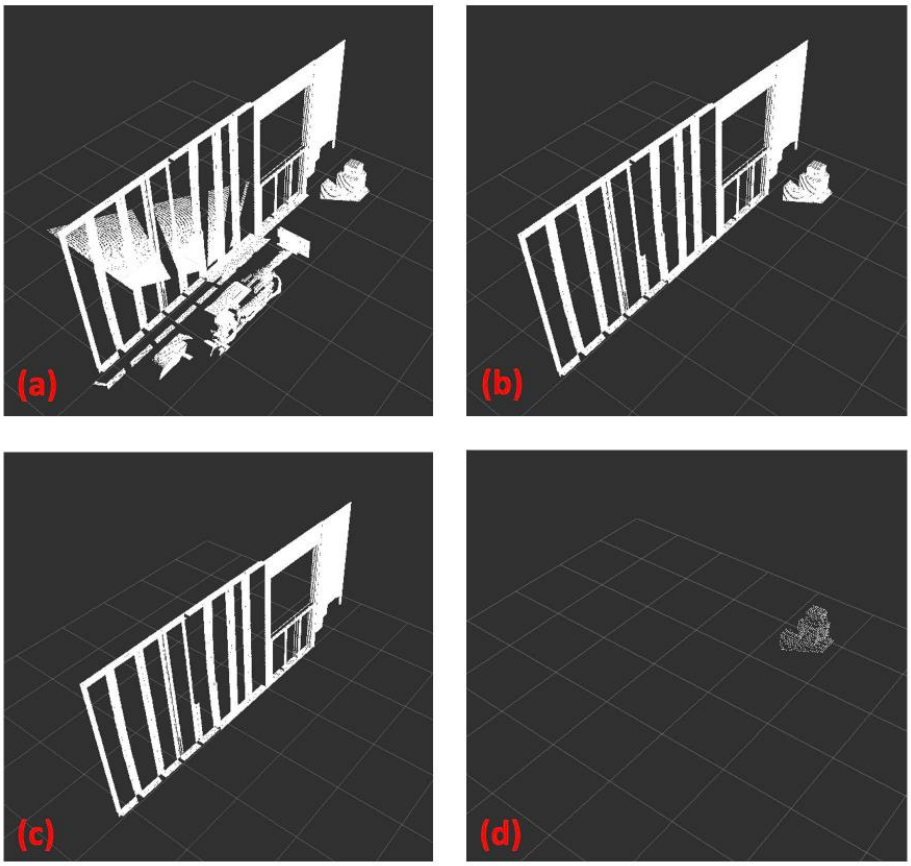


Figure 2.13 Point Cloud Processing Procedures (a) Concatenated Point Cloud (b) Self-Filtered Point Cloud (c) Already-Built Structures (d) Dynamic Objects

The point cloud after self-filtering is converted into a 3D occupancy grid map using Octomap for collision avoidance during motion planning (Hornung et al. 2013). After that, a PassThrough filter is used to separate the point cloud of the already-built structure, which is then converted into a 3D mesh with the Greedy Projection Triangulation algorithm (Marton et al. 2009) and sent to Unity (Figure 2.13c), and dynamic objects (Figure 2.13d), which is sent to Unity for visualization after further downsampling and updates in real-time. The point cloud for dynamic objects is also used to detect workspace clearance.

2.4.3 Experimental Verification Scenarios

2.4.3.1 Working Sequence Guidance

Most construction tasks involve the installation of materials in a certain order (e.g., top to bottom, large to small). According to the wall frame condition, the human co-worker controls the work sequence, including determining the order of conducting tasks and selecting the specific workpiece to manipulate.

Human co-workers can aim the controller at the panel they want to install next and grab it. Then, they can place the panel onto the wall frame, at their preferred position and orientation. The pose of the panel can be fine-tuned several times at different scene scales until they are satisfied. As the human co-worker confirms the task plan, the position ($\boldsymbol{P'}$) and orientation ($\boldsymbol{Q'}$) of the target panel are sent to the middleware. After receiving the task goal, the middleware processes the target panel pose into the target end-effector pose. The target panel pose is first converted from the VR world coordinate ($\boldsymbol{P'}, \boldsymbol{Q'}$), to the ROS world coordinate ($\boldsymbol{P}, \boldsymbol{Q}$) with Eq. 1.1.

$$\begin{aligned} \left[P_x, P_y, P_z\right] &= \left[-P'_x, -P'_z, P'_y\right] \\ \left[Q_x, Q_y, Q_z, Q_w\right] &= \left[Q'_x, Q'_z, -Q'_y, Q'_w\right] \end{aligned} \tag{1.1}$$

It will then be subsequently converted into the end-effector pose ($\boldsymbol{P_E}, \boldsymbol{Q_E}$) with Eq.2.

$$\Delta \boldsymbol{P} = \begin{bmatrix} 1-2Q_yQ_y-2Q_zQ_z & 2Q_xQ_y-2Q_zQ_w & 2Q_xQ_z+2Q_yQ_w \\ 2Q_xQ_y+2Q_zQ_w & 1-2Q_xQ_x-2Q_zQ_z & 2Q_yQ_z-2Q_xQ_w \\ 2Q_xQ_z-2Q_yQ_w & 2Q_yQ_z+2Q_xQ_w & 1-2Q_xQ_x-2Q_yQ_y \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dfrac{T_a+T_b}{2} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{P_E} &= \boldsymbol{P} + \Delta \boldsymbol{P} \\ \boldsymbol{Q_E} &= \boldsymbol{Q} \end{aligned} \tag{1.2}$$

where T_a is the thickness of the drywall panel and T_b is the thickness of the gripper.

There are four types of instructions the human co-worker can send through the command panel on the interactive billboard, “Pick”, “Place”, “Hold”, and “Release”. Because the robot is using a vacuum gripper, it does not directly reach the target end-effector pose while picking or placing workpieces with the “Pick” and “Place’ commands. Instead, the gripper needs to first pause at an intermediate pre-pick or pre-place pose, which is 10 cm before it reaches the target. The intermediate end-effector pose can also be calculated with Eq. 1.2, by replacing $\frac{T_a+T_b}{2}$ with 10 cm. Then, the robot end-effector follows the cartesian path to move from the intermediate pose to the target. The cartesian motion is divided into several small steps at a resolution of 1 cm. If a collision is detected before the target position is reached, the robot will stop at the step before the collision occurs.

For the “Pick” command, the robot picks up the drywall panel as soon as it reaches the target or the collision point. For the “Place” command, the robot will wait until the human co-worker presses the “Release” button to release the drywall panel from the end-effector, which indicates that the human co-worker confirmed the drywall was secured (e.g., screwed or nailed) and is safe to release. For the “Hold” command, the robot directly moves to the target (without pausing at the intermediate pose) and waits for another command before taking any action.

By repeatedly specifying target panels and installation positions and guiding the robot through the pick-place or pick-hold-place installation processes, the human co-worker can collaboratively work with the robot to complete a series of construction activities in a specific work sequence. The snapshot graphs in Figure 2.14 show the work sequence guidance process of installing four drywall panels onto the wall frame. The yellow arrows point to the human-specified targets, which provide information in terms of types of panels and the target installation positions

at this step. The four figures in each step show the actual robot in ROE and the virtual robots that picked up the corresponding panel and placed it at the target position respectively.

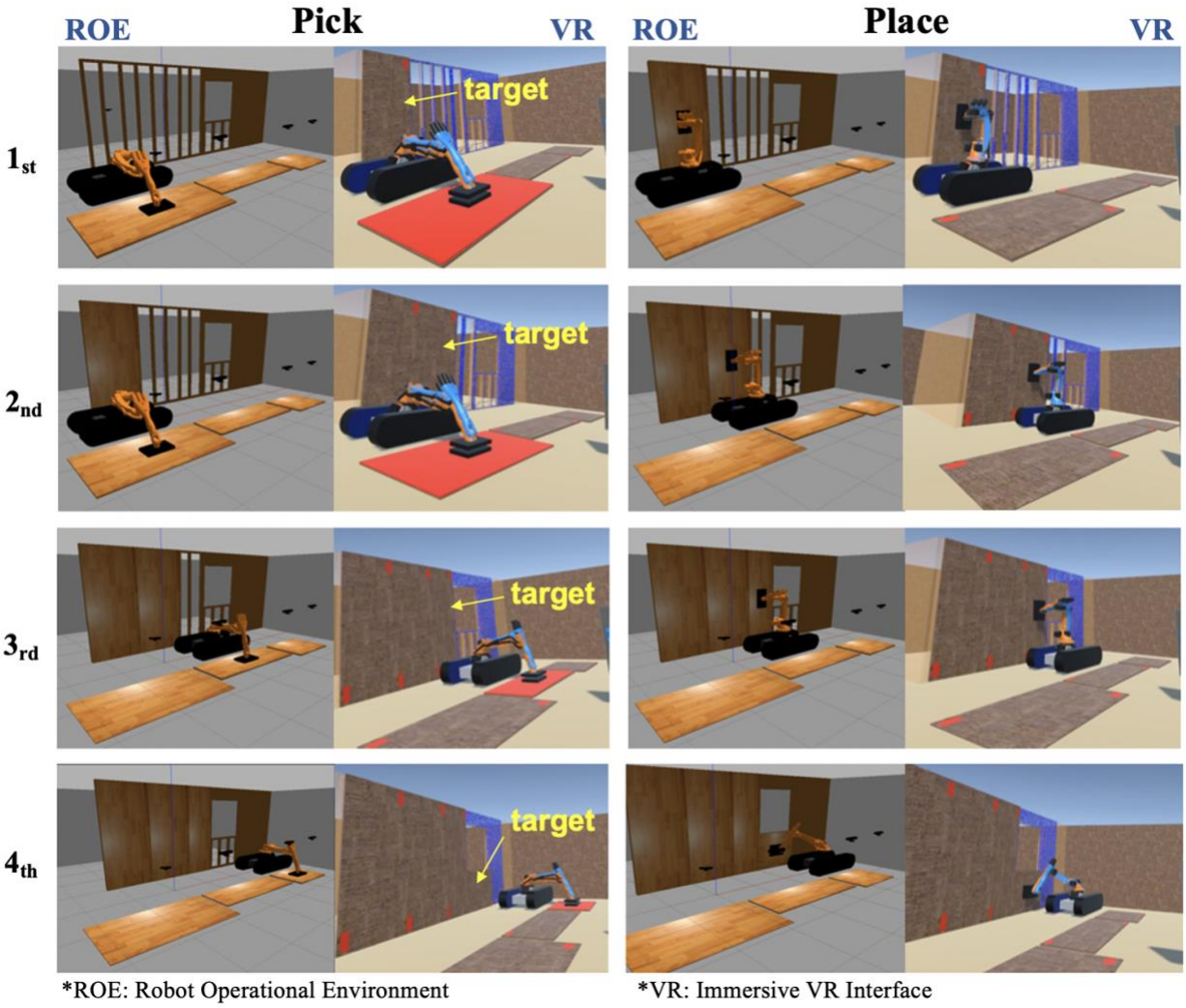


Figure 2.14 Work Sequence Guidance Process Demonstration

2.4.3.2 Optimal Motion Plan Selection

Although OMPL can plan trajectories for the robot to reach the target, it does not guarantee that the trajectory is optimal. Therefore, the proposed system requires the middleware to generate multiple motion plans and allows the human co-worker to select the most desirable plan after viewing generated plans in VR.

The robot motion consists of several planning stages. Pre-processing is needed so that entire planned motions can be viewed by the human co-worker before the actual robot can take any action. This depends on the type of command the human worker gives and whether the robot base movement is needed, as shown in Table 2.4.

Table 2.4 Motion Plan Preprocessing for Visualization in Different Situations

Base	**Command**	**Planning Stages**
Move Base	Pick	*ArmToCarryPose + MoveBase + ArmToPrePick + CartesianMotion*
	Place	*ArmToCarryPose + MoveBase + ArmToPrePlace + CartesianMotion*
	Hold	*ArmToCarryPose + MoveBase + ArmToHold*
Not Move Base	Pick	*ArmToPrePick + CartesianMotion*
	Place	*ArmToPrePlace + CartesianMotion*
	Hold	*ArmToHold*

Since the processes for *MoveBase* and *CartesianMotion* are relatively monotonous in this case study, multiple motion plans are developed for *ArmToCarryPose* and *ArmToPrePick/Place/Hold* stages only. For each of these stages, five stage-level motion plans are generated. After concatenating stage-level plans into entire motion plans that cover all needed stages, four entire motion plans with the shortest time durations are saved. The entire motion plan with the shortest time duration is demonstrated to the human co-worker first. If the human co-worker is satisfied with the plan, they approve it by pressing a controller button and the robot will execute the plan. Otherwise, the next plan, which is the one with the second shortest time duration is demonstrated, and so on. Figure 2.15 shows the snapshots of robot execution supervision processes for the different stages of picking operation with the temporal order from left to right. The description of each stage is given at the bottom. The “supervising” robot the human co-worker sees in VR is synchronized with the actual robot in ROE.

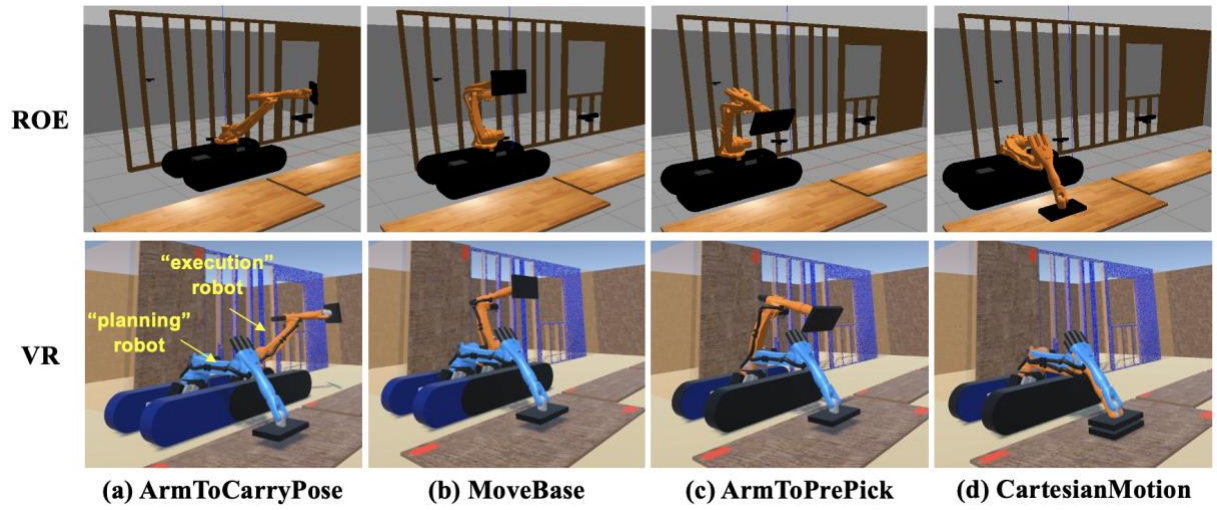


Figure 2.15 Robot Execution Processes Demonstration

2.4.4 Trajectory Guidance with Intermediate Object Poses

The construction environment presents more challenges to the robot motion planning process than ordinary robot working environments because of its complexity. Therefore, the motion planner might fail to develop a motion plan or the robot might be stuck at some locations even if a valid trajectory exists. In some situations, even though the robot can find valid motion plans, the human co-worker may have preferences for the robot to perform the task in a specific way.

Some existing studies allow users to guide the robot by specifying end-effector paths or waypoints (Fang et al. 2012; Ong et al. 2020). However, it is very challenging for users to specify collision-free paths or waypoints when the manipulated workpiece is large and the workspace is complex. In addition, paths and waypoints only possess the end-effector position information. When the workpiece is large, its orientation on the trajectory is also important and can make a notable difference. Therefore, in the proposed system, the human co-worker guides the robot by specifying the intermediate object poses on the trajectory. The human co-worker sets the poses by

placing the interactive drywall panels at desired positions and pressing the “Hold” button. The robot carries the panel to the intermediate poses and holds the panel to wait for another command. Multiple intermediate poses can be specified to guide the robot trajectory step by step. Figure 2.16 demonstrates the process that the human co-worker guides the robot trajectory in four steps from top to bottom. In each step, the human co-worker specifies an intermediate object pose (Target 1-3) by placing the virtual panel and finally guides the robot to the final installation target.

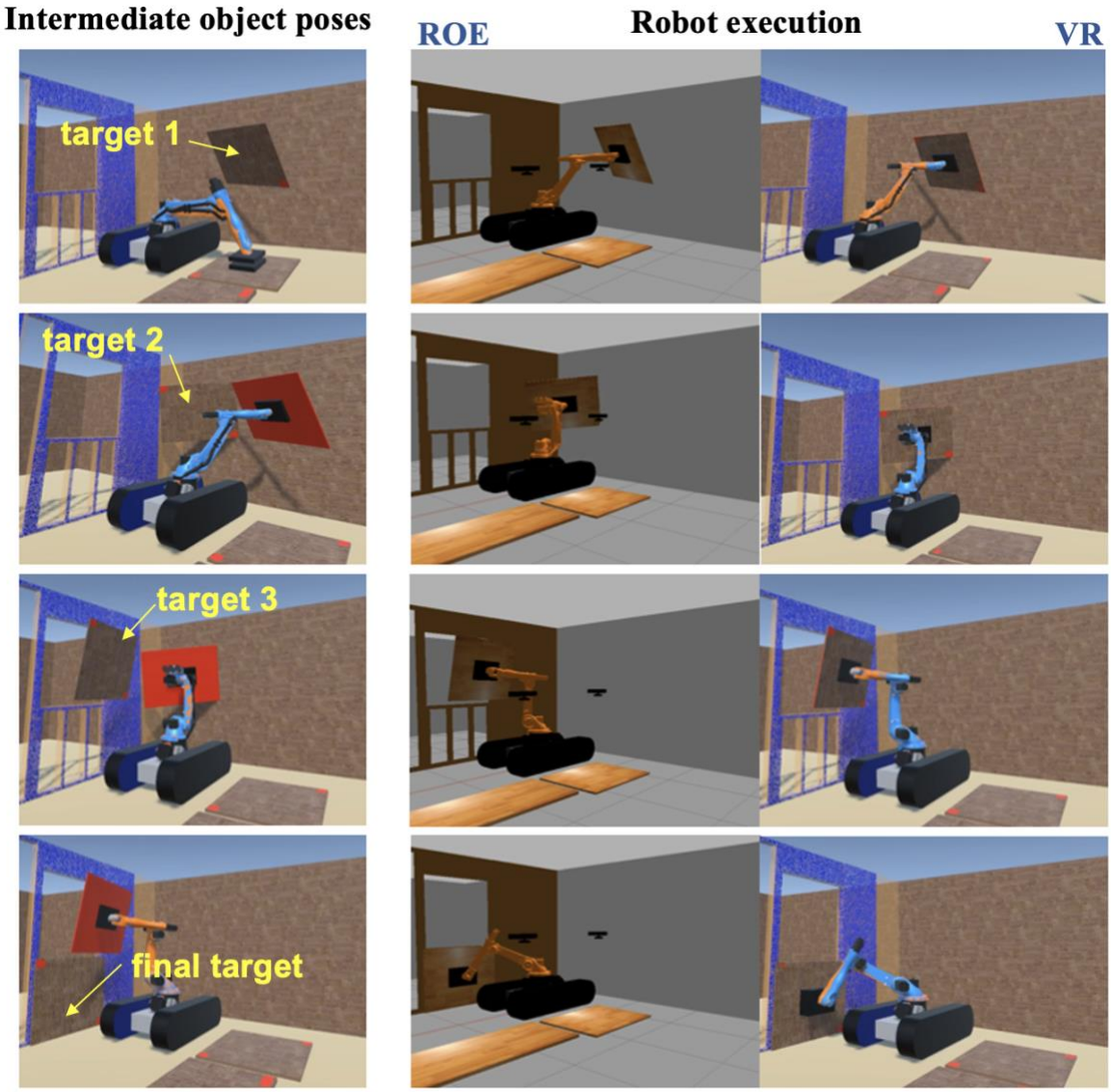


Figure 2.16 Trajectory Guidance with Intermediate Object Poses

2.4.5 Human-in-the-Loop User Study

We conducted a human-in-the-loop user study as a preliminary usability test of the I2PL-DT system on HRCC work and to receive feedback and suggestions for future improvements. Twenty subjects were recruited to perform the drywall installation task with the proposed system. The main objective of the user study is to verify that subjects who were not involved in the system development process and unfamiliar with the system's technical background can successfully use the system to collaborate with the robot and achieve task objectives. Following the drywall installation task, a survey is carried out to collect user feedback for improving the system functions and interaction design. The survey supports three main functions: (1) to evaluate the general usefulness, effectiveness, and understandability of the system; (2) to assess system functions and user experience with the VR interface; and (3), to elicit user feedback and suggestions. In this section, the experimental protocol is introduced and the quantitative ratings from users are analyzed according to their feedback and suggestions.

2.4.5.1 Experimental Protocol

The experiments were conducted one subject at a time in a university research laboratory following all health-informed safety guidelines in place at the time on account of the Covid-19 pandemic. The experimental protocol was approved by the Institutional Review Board at the University of Michigan. Twenty subjects, eleven female and nine male, were recruited and completed the experiment. Several prior studies have invited college students to test a system at its prototype stage to assist with system verification and design (Akanmu et al. 2020; Chen et al. 2016; Mantha et al. 2020; Quintero et al. 2015). Since the VR interface of the proposed system is fundamentally different from the traditional drywall installation approaches, we invited graduate students, who have basic knowledge of visualization and are more familiar with gaming and

computer technologies, as the users for our study at this stage. Most subjects have civil engineering, construction, and/or robotics backgrounds and they were introduced with basic drywall installation knowledge at the start of the experiment. This allows for minimal training time before they can perform the requested task.

The timeline of the experiment can be found in Figure 2.17. As the experiment started, the researcher spent 15 minutes introducing the experiment to the subject and answering their questions. Next, the subject put on the headset and completed a trial session to get familiar with the system. After the trial, the system was reset and the subject was given 30 minutes to perform the main task. The environment settings of the main task are the same as shown in Figure 2.10. The subject was requested to install four drywall panels vertically onto the wall frame. Installations of the first three panels were implemented with the pick-place approach. As the subjects got more familiar with the system while installing the first three, they were requested to use the pick-hold-place approach for the last panel by indicating intermediate object poses on the trajectory. The subject made their own decisions on the working sequence and the type of drywall to install for each operation.

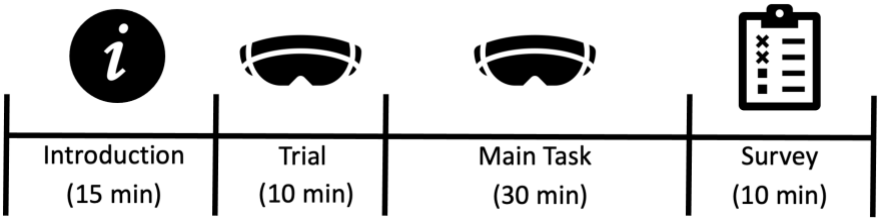


Figure 2.17 Experiment Timeline

The subject was asked to take a survey after the main task. The survey contained 5 different sections. The first section asked about the subjects' basic information and their task completeness. In the second to the fourth sections, subjects evaluated different aspects of the system with a 7-point scale, where 7 represents the most positive evaluation and 1 represents the most negative

evaluation. In the last section, subjects provided written comments on the system and made suggestions for improvements.

2.4.5.2 Results

All subjects were able to use the proposed I2PL-DT system and take advantage of the provided system functions to collaborate with the robot. Out of 20 subjects, 16 completed the installation of all four panels during the 30-minute main task period. In addition, 19 out of 20 subjects successfully noticed and avoided the deviation on the wall frame. An approximate productivity comparison between the subjects using the I2PL-DT system and the standardized data of an experienced carpenter (assuming they will complete the installation by themselves) obtained from RSMeans Data (Mewis 2019) has been performed. Figure 2.18 shows the cumulative time taken to install one to all four panels with the orange line indicating the average time taken for all subjects and the box plot illustrating the time distribution among subjects. The standardized RSMeans data is shown in the dark red line. The RSMeans database uses the area of panels installed as the output to quantify productivity. Therefore, the time required to install a larger panel is proportionally longer than a smaller one. In this comparison, it is assumed that the panels are installed in the order from the larger size to the smaller size when calculating with RSMeans. While a robot performs the task, the time taken for installing a larger and smaller panel is almost the same, and the panels are installed in the individual subject's preferred order. Although it is an approximate comparison, it can clearly show that the proposed system takes significantly less time than traditional methods in addition to reducing the physical stress and increasing the safety of construction workers.

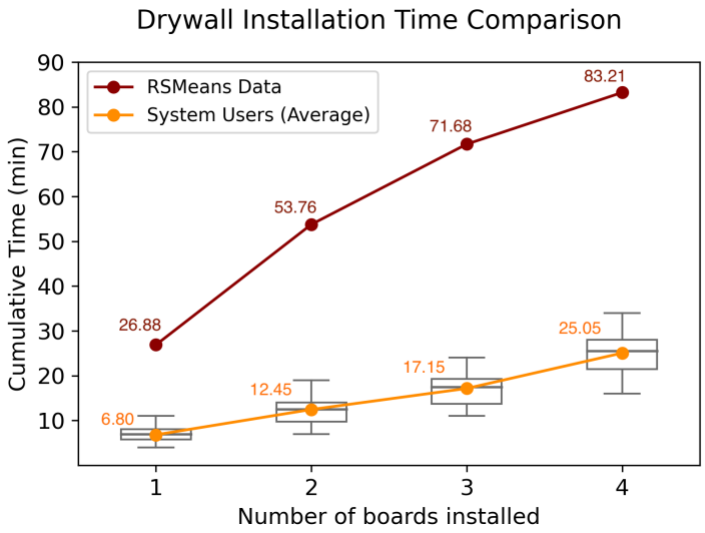


Figure 2.18 Drywall Installation Time Comparison

In the second section of the survey, subjects made a general assessment of the system usefulness, effectiveness, and understandability. The mean and standard deviation of ratings along with the box plots are shown in Figure 2.19. Subjects generally thought that the system is very useful and understandable. However, the ratings of system effectiveness are relatively lower. One of the comments we received is that the plan preview process almost doubles the time needed because the subjects first previewed the motion plan animation and then supervised the robot to execute the same plan. Although safety is ensured, the subjects reflected that the process reduces the overall equipment effectiveness, and thus the job progress is not optimal. Nevertheless, even with the plan preview process, the approximate productivity comparison between the subjects using the I2PL-DT system (9.20 m^2/30 min) and RSMeans Data (6.64 m^2 (71.43 S.F.)/hour) shows that subjects' productivity with the proposed system is highly comparable to that of experienced construction workers (Mewis 2019).

Indicator	Mean	SD	Rating
Usefulness	6.200	0.616	
Effectivenss	5.500	0.827	
Undertandability	6.250	0.910	

Figure 2.19 General Assessment Results

In the third section, subjects evaluated different system functions (Figure 2.20). The functions correspond with the seven useful characteristics of HRCC systems (Table 2.1). Subjects thought the motion plan evaluation process is clear and they could easily and clearly supervise actual robot states with the VR robot models. They can understand the differences between as-designed and as-built geometry. However, they have relatively lower satisfaction with the automatic motion plan generated by the system since there are some unnecessary rotations on the panel that reduce the system efficiency. Some subjects also expect the robot to move faster.

Indicator	Characteristics	Mean	SD	Rating
Construction site condition visualization	2	5.600	0.598	
As-designed VS. as-built	3	5.750	1.164	
High-level task planning	4	5.750	0.550	
Robot automatic motion plan	4	4.900	1.518	
Motion plan evaluation	5	6.150	0.933	
Real robot states supervision	6	5.850	0.933	
Communicate to robot	7	5.550	0.686	
Information from robot	7	5.400	1.046	

Figure 2.20 System Functions Assessment Results

Some subjects experienced difficulties communicating with the robot because they were not familiar with the usage of the handheld VR controller, especially when their eyes were covered by the VR headset. For the information from the robot, they suggested adding haptic and sound feedback in addition to visual messages. One subject suggested showing messages in front of the

users' view instead of showing them on the TV screen. For the high-level task planning, a subject suggested planning the installation poses of all four panels before the robot starts to develop the motion plan and perform the task instead of planning and installing the panels one by one. In addition, one subject suggested adding a function to indicate whether the intermediate pose the user selected for the robot to temporarily hold the workpiece is valid (i.e., within the robot's reaching range).

In the fourth section of the survey, the subjects were asked to evaluate their VR user experience from eight aspects. The assessment results are shown in Figure 2.21. The questions in this survey are adapted from Presence Questionnaire developed by UQO Cyberpsychology Lab (UQO Cyberpsychology Lab 2004). Subjects were generally satisfied with their VR experience. They indicated that the interaction with the VR environment is natural and they could well-anticipate system responses to their actions. They could quickly adapt to the VR experience and visually search the environment for information they need. However, some subjects reported they experienced some difficulties manipulating the panel. Even though they can scale down the scene, the panel still blocked their vision to some extent because it was very close to their body when they hold it in their hand. Haptic feedback would also be helpful for accurate manipulation of the drywall panel. In addition, several subjects reflected that they experienced motion sickness after working for a while in VR and the handheld controllers were not sensitive enough to show the laser pointer in some situations, which affected their concentration and overall interaction experience. Some subjects reported delays in graphics rendering near the end.

Indicator	Mean	SD	Rating
Involvement	5.900	0.912	
Events control	5.500	0.888	
Visual search	6.000	0.858	
Movement and Manipulation	5.450	1.050	
Interaction (Natural)	6.050	0.826	
Interaction (Overall)	5.650	0.933	
Concentration	5.750	0.910	
Adaption	6.000	1.076	

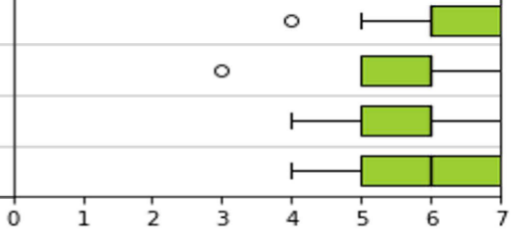


Figure 2.21 VR User Experience Assessment Results

2.5 Discussion

Overall, subjects felt positive about the proposed I2PL-DT system for HRCC. In addition, the valuable feedback from subjects has provided us with remarkable insights for improving the proposed system. To reduce the work progress delay caused by motion plan preview while ensuring safety, the motion plan demonstration speed can be adjusted to be faster than the actual robot's movement speed to save plan preview time. Another possible solution to this problem is to offer human co-workers the option to skip previewing the motion plan with a disclaimer that it is better to preview to ensure safety and prevent unexpected accidents. However, the actual robot's movement speed could not be made faster because of the hardware limitation of the robot.

Several changes could be made to improve the system experience, including adding haptic and audible feedback to facilitate communication and improve object manipulation, showing messages in front of the user's view, making the panel semi-transparent when being grabbed to preserve users' vision while manipulating large-size workpieces, allowing several steps of high-level task planning to be developed at once, validating the intermediate poses for robot "Hold" operation, and optimizing the system for fast rendering. Interface design techniques should be used

to reduce motion sickness. More advanced motion planning algorithms should be developed to reduce unnecessary robot movement. Moreover, systematic training is needed to get users familiar with the system before applying it to real-world construction projects.

In addition to user feedback, the authors identify several limitations to be addressed in future work. First, the robotic arm has a fixed intermediate pose to carry the workpiece while moving its mobile base, which will not change during the moving process. In the future, algorithms will be developed for the robotic arm to dynamically adjust its pose according to the environment geometry while moving the robot base. This will provide more flexibility in collision avoidance for a mobile robotic arm carrying large-size workpieces.

Second, the robot will follow the exact motion plan once the plan is approved. If a new object appears in the robot's workspace during execution, the robot will terminate its execution and wait for instructions from human co-workers for safety reasons. In future work, the robot's autonomy can be enabled out of human supervision to dynamically adapt its path during execution along with the investigation of how to reduce the impact of certain autonomy on system safety.

Third, this chapter assumes the material stacked position is known to the robot. In fact, the materials stacked on-site might be moved from time to time and the position recorded might not be accurate. Computer vision and deep learning-based approaches can be used for the robot to automatically detect materials.

Fourth, there will be time delays caused by point cloud processing and rendering and data exchange, especially when the working environment is large and complex. In the future, more advanced optimization algorithms and computational power could be used for the system to stay closer to real-time for large-scale projects.

Lastly, since the main focus of this chapter is the system framework design and its verification instead of field tests, the experiments are conducted in simulation primarily to ensure subjects' and workspace safety in this first set of experiments. The next step of this research includes gripper hardware design that is capable of large construction workpiece manipulation (e.g., drywall panels). Improving the bi-directional communication for state synchronization between the Gazebo simulation and the physical robot to experiment on real KUKA KR120 robotic arms (Liang et al. 2020a) is also considered as a future step. Future work will include human factor studies and inviting construction workers to use the system to improve system design and investigate the long-term effects of the system on workers' physical workload, mental stress, and job satisfaction.

2.6 Conclusions

This paper proposed an I2PL-DT system for construction workers without robot programming expertise to remotely collaborate with construction robots to perform construction work. The proposed system has several contributions. First, it uses immersive VR and proposes a hybrid approach to create an augmented telepresence experience for human workers while preventing them from being exposed to potential hazards on construction sites. It also allows people with physical disabilities to participate in performing construction activities. Not only can workers access pertinent information they would obtain by physically present on construction sites, but they can also obtain augmented information that they cannot directly perceive on-site, such as the as-designed building model and robot status information.

Second, the system provides an intuitive interface to assist human co-workers to perform high-level task planning. Human co-workers can specify task objectives by interacting with virtual

objects in VR and try different options without exerting substantial physical effort or using up actual resources.

Third, human effort can be notably reduced since the robot is responsible for planning collision-free trajectories after receiving the task objectives. Human co-workers can also guide the robot to perform the task by specifying intermediate object poses on the trajectory.

Fourth, the system enables seamless bi-directional communication between the robot and its human partners and allows real-time robot status supervision. Human co-workers can easily send task objectives or commands to robots by clicking buttons on handheld controllers. It also allows supervision of robots' intentions, actual states, and implicit robot status information in VR.

Overall, the proposed system offers a promising approach for construction workers to collaborate with on-site construction robots from remote locations and demonstrates the potential of transitioning the role of construction workers from physical task performers to robot supervisors, laying the groundwork for future construction work at the collaborative human-robot frontier.

Chapter 3 Closed-Loop BIM-Driven Human-Robot Collaborative Construction System

3.1 Introduction and Motivation

In Chapter 2, an interactive and immersive process-level digital twin (I2PL-DT) system for human-robot collaborative construction (HRCC) has been proposed. However, the system has several limitations. First, the process of creating the digital twin for a construction task takes considerable effort. The immersive VR interface must be manually created by importing BIM data into VR, creating game objects as interactive construction components, and adding interactive functions (e.g., enabling manipulation by human co-workers, sending messages to ROS) to different VR objects. The developer also needs to manually set the regions in the global coordinate frame that mark non-critical components, dynamic objects, and already-built structures. Second, it is assumed that the materials stacking positions are known to the robot. However, the materials can be arbitrarily stacked on an actual construction site and the robot needs the capability to localize these materials. Third, the human co-worker needs to specify the work plan for each component by indicating to the robot the material to pick up and the location to install it. Thus, a construction task that involves a lot of components requires substantial human effort with step-by-step instructions. Lastly, since the human co-worker specifies the high-level task objectives by manipulating and placing virtual objects with controllers in VR, the accuracy is limited.

In order to overcome these critical and practical limitations, this paper integrates BIM with the I2PL-DT system and proposes an automatic approach to create the digital twin for improving system autonomy and work accuracy as well as reducing human workers' mental workload.

BIM is “a digital representation of physical and functional characteristics of a facility” (NIBS 2015). It contains a variety of geometric and semantic information, such as 3D models, schedule, construction methods, and materials, which is used to automate the construction processes (Correa 2016). Although BIM plays an important role in facilitating design, communication, and project management throughout the project life cycle (Han et al. 2021; Zhang et al. 2022), it lacks the interoperability to support construction robot task planning (Correa 2019; Kim et al. 2021b). Currently, robot motions are generated by retrieving geometric data from BIM encoded using Industry Foundation Classes (IFC) (Correa 2016; Ding et al. 2020; Kim et al. 2021b). However, these proposed approaches are limited to specific construction tasks or types of workpieces. Moreover, working environments for field construction involve a lot of uncertainties (e.g., deviations in as-built workpieces) that can cause robot failure when following the program. While BIM can provide information to the robot, the system needs the ability to improvise (i.e., dynamically adjust plans based on encountered situations) to flexibly perform the work during the field construction process.

Compared to robots, humans are more adept at creative and adaptive planning based on their experience (Seong et al. 2019; Sharif et al. 2016). They can adjust the task plan accordingly to ensure the quality and continuity of the work when the robot fails to make correct decisions. Thus, human expertise in improvisation is indispensable for field construction that involves considerable uncertainties. In addition, human workers can supervise the robotic construction process to ensure collision-free safe manipulation in dynamic on-site working environments. Therefore, by enabling human-robot collaboration, the flexibility and robustness of BIM-driven robotic construction systems can be significantly improved.

This research proposes a closed-loop BIM-driven human-robot collaborative construction (HRCC) system (Figure 3.1). The system is built upon an interactive and immersive process-level digital twin (I2PL-DT) system proposed in Chapter 2 (Wang et al. 2021), with an additional BIM module to provide geometric and semantic data to both the human worker through the user interface and the Robot Operating System (ROS). After the robot generates work plans with information from BIM, human co-workers supervise the robot's work status (e.g., preview robot plan and monitor execution status) and make interventions (e.g., adjust installation target or request another trajectory plan) when necessary. In addition, the as-built data collected by the robot during the construction process is sent to the BIM repository to be recorded, thereby closing the loop.

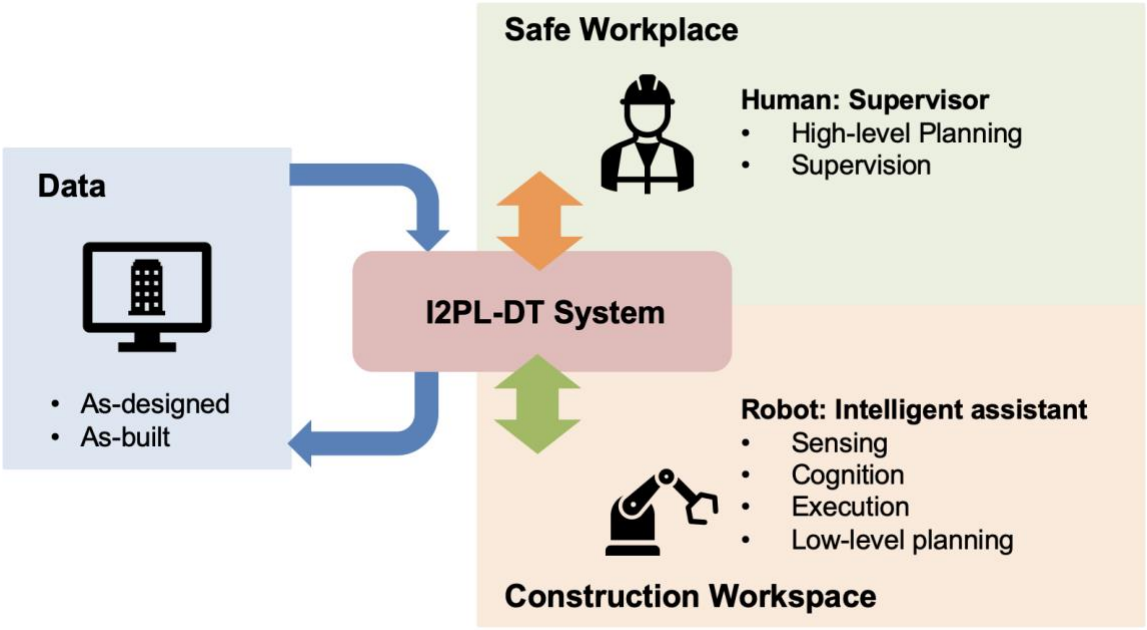


Figure 3.1 Chapter 3 Study Overview

A four-step technical workflow for BIM-driven HRCC along with its technical challenges is discussed in detail, including (1) creation of the BIM repository that supports robotic construction; (2) physical and digital preparation for a construction task; (3) HRCC process with the proposed system, including automatic digital twin generation and resolution of as-built / as-

designed deviations; and (4) updating of the BIM repository with as-built construction data collected by the robot.

A drywall installation case study involving a deviated prefabricated wall frame is presented as a proof-of-concept implementation to explore the physical setup process and for system validation. In addition, three experiments were conducted to evaluate system performance and understand the sources of errors. All the experiments and the case study are conducted with a physical Kuka KR120 industrial robotic arm. The proposed system not only extends the autonomy and accuracy of robotic construction driven by BIM but also has the flexibility to overcome uncertainties in field construction work. The presented framework and workflow can be applied to different types of construction tasks. The overall goal of the proposed system is to improve robot planning capability thereby reducing the planning effort of its human co-workers (Figure 3.2).

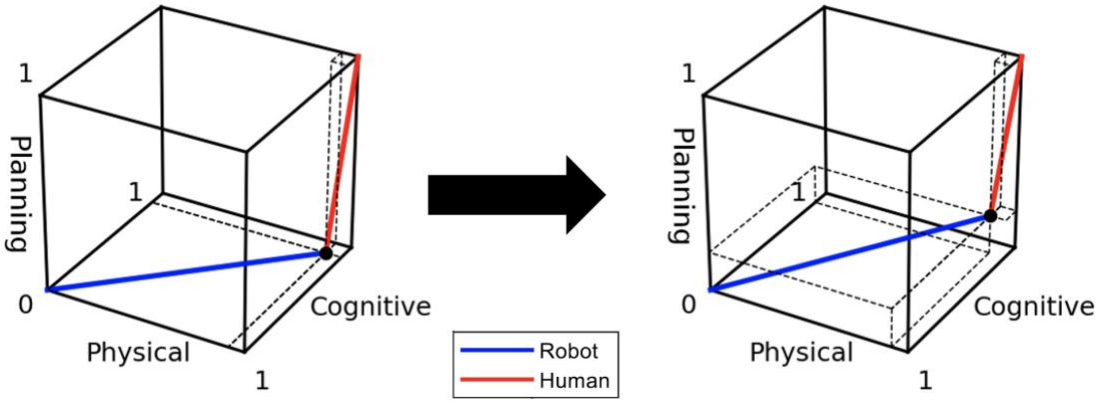


Figure 3.2 System Impact on Human-Robot Effort Distribution (Chapter 3)

3.2 Background and Related Work

3.2.1 BIM in Construction Automation and Robotics

BIM has been widely adopted to promote automation throughout the life cycle of the Architecture, Engineering and Construction, and Facilities Management (AEC/FM) industry

(Deng et al. 2021a). It has been used to facilitate concrete formwork design (Romanovskyi et al. 2019), mechanical, electrical and plumbing (MEP) layout customization (Singh et al. 2018), and fabrication drawings (Deng et al. 2021b) during the design phase. It has also been adopted to automate the construction process (Li et al. 2017; Wong Chong and Zhang 2021) and support building operation and maintenance (Gan et al. 2019; Welle et al. 2011).

With the recent growth of research interest in construction robots, BIM has been used to facilitate robotized construction in several different ways. For example, BIM can provide information to guide the off-site prefabrication process (Abanda et al. 2017; Zhu et al. 2021). It can assist with object recognition for on-site assembly (Dawod and Hanna 2019). Layout information contained in BIM is used to support robot indoor navigation tasks for building maintenance and construction (Follini et al. 2020; Park et al. 2016). Commercial mobile robots have also been introduced to draw layouts on-site based on BIM (Dusty Robotics 2021; ENR 2013). The geometric information contained in BIM data has also been used to facilitate 3D printing in construction (Davtalab et al. 2018; Teizer et al. 2018).

BIM can also provide information to support robot motion planning. For example, in robotic brick assembly tasks, BIM can provide position and orientation data for a robot to pick up and place materials as well as scheduling data to control the robotic workflow (Ding et al. 2020; McClymonds et al. 2022). Kim et al. (2021) proposed an algorithm to convert IFC wall models into Simulation Definition Format (SDF) models, which can be imported into the robotic simulation platform Gazebo and can be used to generate robot navigation and manipulation plans (Kim et al. 2021b). Wong Chong et al. (2022) used BIM to simulate the robotic wall frame assembly process (Wong Chong et al. 2022). In these existing studies, robots fully rely on BIM

for task planning. The construction site needs to meet exact specifications for the robot to successfully perform construction work.

However, considerable uncertainties exist on construction sites. Deviated workpieces, moving workers and equipment, and stacked materials on-site may interrupt the BIM-generated robot motions, causing robots to stall while performing construction work on-site. Lundeen et al. (2019) proposed a Generative Resolution Correlative Scan Matching search algorithm that utilizes a combination of BIM and sensing information to generate adaptive robot motions (Lundeen et al. 2019). Despite such advances, the robot's adaptability is limited to a narrow set of situations and construction tasks (Lundeen et al. 2019).

In summary, existing studies in BIM and robotics have three limitations. First, they lack generality to support various construction tasks. The systems are developed for one type or limited types of construction work by parsing specific information from BIM data. Second, they cannot handle uncertainties for field construction. In field construction, even if the robot has certain adaptability, there are several factors (e.g., workpiece deviation) that might interrupt robot activities generated by BIM. Thus, human intervention is necessary in addition to BIM for the success of robotic field construction. Third, as-built data collected by the robot during the construction process is useful for the subsequent construction, operation, and maintenance phase of the project, however, a closed loop for BIM to both provide and collect information is missing in existing studies. Therefore, a general framework that supports different types of construction tasks and allows human intervention is needed.

3.2.2 Digital Twin Creation in Robotic Applications

Digital twins can be used for visualizing and incorporating information from different resources, and they also support real-time communication and interaction. Therefore, they are a

promising candidate to integrate BIM with HRCC. Based on the demand for different application purposes, digital twins can be created with various approaches. One of the most popular methods in the AEC/FM industry is to use 3D point clouds. The environment is captured with laser scanners or depth cameras as 3D point clouds (Fang et al. 2016; Feng et al. 2015; Xu et al. 2019). Such systems can comprehensively capture the environment in real-time, theoretically allowing for continuous updates of the digital twin to reflect the evolving environment at any timepoint. However, the transmission of large-size point cloud data is computationally expensive and is subject to delays. Furthermore, subsequent steps of processing and registering the point clouds, object detection, or 3D reconstruction are usually needed to achieve the required functions (Stojanovic et al. 2018; Zhou et al. 2020).

Liang et al. (2022) and Roldán et al. (2019) created digital twins of a robot to program, control, and visualize robot motions (Liang et al. 2022; Roldán et al. 2019). Joint state data are exchanged between the physical and virtual robots in real-time. In order to reduce the computation load while allowing real-time visualization of the construction environment, Wang et al. (2021) created a digital twin using a combination of BIM, 3D meshes of as-built structures, and point clouds (Wang et al. 2021). However, these approaches require some manual processes to import models into the digital twin or enable the functionality of the digital twin (e.g., interaction). As a result, it requires human manual input during the digital twin generation process and thus interrupts the automated workflow of HRCC initiation, which can potentially hinder the widespread application of digital twins in HRCC. Therefore, an approach to automatically generate digital twins that is equipped with pertinent functions (e.g., target objects selection, communication between BIM and ROS) is needed to promote and extend the autonomy of the HRCC workflow.

3.2.3 Summary

To address the abovementioned research gaps, a workflow with the following characteristics is needed to effectively support HRCC. First, the enabled HRCC framework should work for a variety of construction tasks. Instead of parsing specific information from BIM repositories to automate one type of construction work, a general framework that supports different types of tasks and workpieces is needed. Second, the workflow should automatically interface with different construction tasks without additional programming or development effort that cannot be performed by construction workers without related expertise. Third, the robots should have the ability to handle significant uncertainties on construction sites while allowing human intervention to resolve cases that extend beyond the robots' capabilities. Lastly, the workflow should record the construction-related data collected by the robot sensors during the HRCC process for future reference to enable loop closure. With these objectives, the following section presents a closed-loop generalizable framework integrating BIM and HRCC that supports automated collaborative workflows and can overcome uncertainties in field construction work.

3.3 Technical Approach

A closed-loop BIM-driven HRCC workflow is proposed in this study. BIM provides data (e.g., workpiece geometry, position, type) to both the robot and the human co-worker by sending messages to ROS and Unity (where the interface is developed). The robot then generates the work plan based on BIM and adapts the plan according to the as-built circumstances detected by its sensors. For example, during drywall installation, the robot receives the name and target position of the next panel to install from BIM; it then adjusts the installation position of the panel based on how the wall frame has actually been built as detected by its camera. Next, human co-workers supervise the robotic construction process (e.g., by evaluating and approving the robot plan) and

intervene to adjust the high-level task plan (e.g., by adjusting the target installation sequence and pose proposed by the robot) when necessary.

An overview of the BIM-driven HRCC workflow is shown in Figure 3.3. First, BIM data that can be used by both human workers and robots needs to be created, which is presented in Section 3.3.2, following a discussion of construction methods (Section 3.3.1). Next, before construction starts, human workers need to prepare physically on the construction site (e.g., staging materials) and digitally in BIM (e.g., indicating targets to install). These steps are followed by the HRCC process. Lastly, after certain construction tasks are finished, the BIM repository is updated with the latest as-built data. The framework and technological approach for integrating BIM to drive HRCC is introduced in detail in Section 3.3.3. Lastly, Section 3.3.4 introduces how the system is deployed onto a physical industrial robot.

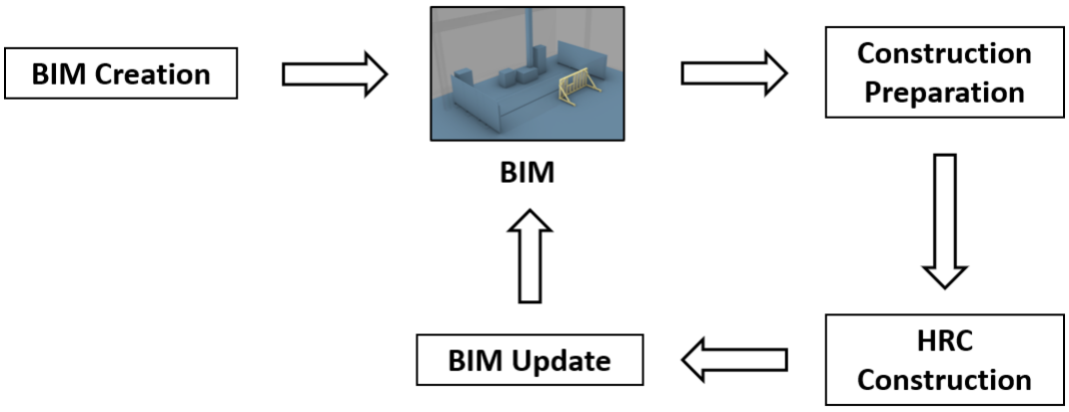


Figure 3.3 Closed-Loop BIM-driven HRCC System Workflow

3.3.1 Construction Methods

The traditional method of building is on-site or in-situ construction, which still dominates the construction industry in the present day. It offers high flexibility in terms of design and operation. However, it also introduces loose tolerances and high uncertainties to constructed facilities that hinder the adoption of automation technology (Martinez et al. 2008). As a result,

construction is one of the most labor-intensive, dangerous, and under-producing industries (BLS 2021; Wang et al. 2020b). In recent years, modular construction has been applied in facilities with repeated units, such as multi-story residential buildings, schools, and hotels (Lawson et al. 2012). Room-size volumetric modules are prefabricated in factories and then assembled on construction sites (Lawson et al. 2012). Compared to on-site construction, off-site construction takes less time to prefabricate room modules and allows for better quality and process control (Thai et al. 2020). Nevertheless, modules restrict the flexibility in architectural design and on-site assembly to some extent. It also increases the material usage, and the large modules are difficult to transport to construction sites (Dörries and Zahradnik 2016; Smith et al. 2018).

In order to overcome these limitations, a hybrid prefabrication mode combining on-site and off-site construction is becoming increasingly popular. Relatively small building components are prefabricated and assembled at off-site locations, and then transported to construction sites to be installed (Boyd et al. 2013). This approach can significantly reduce the amount of tedious work conducted on dangerous construction sites. It maintains the factory-level high quality of the workpieces and the fast production speed while also retaining the flexibility in building design and in resolving uncertainties in the field. For example, wall frames are prefabricated in factories and brought to the construction site to be erected. Then, drywall panels are installed onto the wall frames on-site with some further modification to avoid ducts or collisions.

This hybrid mode also offers the potential for workpieces to be prefabricated by robots in factories or to be 3D printed off-site, and later installed collaboratively by workers and robots on construction sites, thereby increasing the construction automation level. Since the components prefabricated are relatively small, they can be easily transported to construction sites and installed. Therefore, this mode is promising for wide adoption in the near future. However, components

might still be installed with some deviations on construction sites compared to the design, which needs to be considered. As a result, this research assumes that the workpieces are perfectly built but might be installed in deviated poses on construction sites.

3.3.2 BIM for Robotic Construction

BIM serves as "a shared knowledge resource for information about a facility" and has been widely adopted in the AEC/FM industry (NIBS 2015). However, BIM created for traditional construction methods is not compatible with robotic construction (Meschini et al. 2016). Some additional elements are necessary for BIM to better support human-robot collaborative task planning:

- ***Shop drawing-level geometry.*** Shop drawings precede detailed work plans and contain information needed for fabrication, assembly, and erection (Pietroforte and Member 1997). For example, in a robotic drywall installation task, shop drawings for drywall panels are needed so the robot knows where each panel should be located. For a robotic nailing or screw insertion task, shop drawings containing the fastener locations are needed. However, shop drawing-level details are not needed for every component in BIM data and are dependent on the construction plan. For an object that is prefabricated off-site or erected on-site as a whole, it can be represented in BIM as one single object.
- ***Construction sequence:*** Detailed construction sequence information contained in BIM is crucial to automate the construction process. The construction sequence information should be provided in correspondence with the shop drawing-level geometry, such as the sequence showing which panels to install first.
- ***Workpiece relationships.*** Relationships between workpieces can affect robot planning, especially when deviations exist and the robot needs to adapt its plan. Even though

some object arrangements visually look the same, they might indicate different relationships (i.e., topologies) which require different construction plans. For example, objects A and B are placed next to each other in Figure 3.4. They can simply be two separate objects where the deviation of A (the blue block) will not affect the construction of B (the yellow block) unless A collides with B (e.g., two adjacent blocks shown in Figure 3.4a). B might be soft-connected to A where B will only be affected when A deviates vertically (e.g., B sits on A shown in Figure 3.4b). There is also a possibility that B is fully connected to A and needs to follow A in whichever way A deviates (e.g., B is a part of A shown in Figure 3.4c). As a result, clearly defined workpiece relationships in BIM are necessary for the adaptive planning of the human-robot work team.

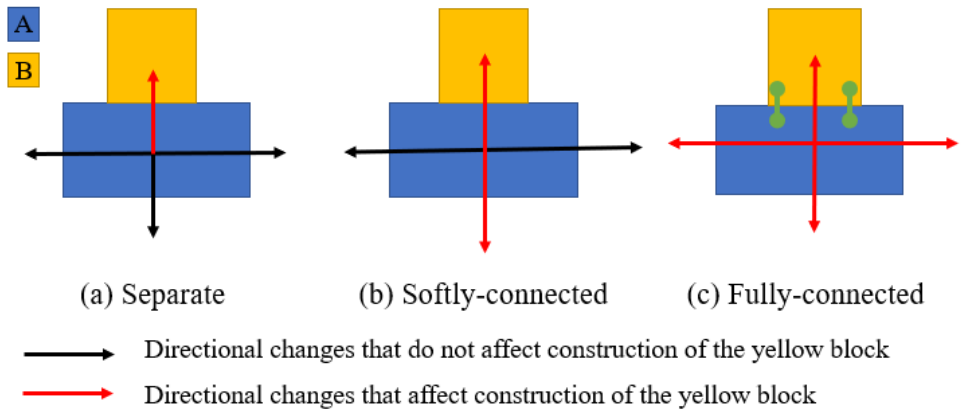


Figure 3.4 Object Inner Relationship Illustration

- ***Object layers and types.*** A unified predefined layer system in BIM not only helps organize objects into different groups for user understandability but also allows the development of interfaces that can quickly connect to different BIM projects and automate related processes in the project life cycle, such as construction. Therefore, a

BIM layer system is proposed to facilitate the automatic generation of interactive digital twins and the HRCC process, as shown in Table 3.1.

Table 3.1 BIM Layer System for Robotic Construction

Layer Name	**Description**
Target	Physical objects (e.g., timber) or virtual indicators (e.g., fastener locations) the robot needs to install or operate for the current construction task
Workspace As-Built	As-built workpieces inside the robot workspace that need to be considered for collision avoidance. They update with the construction progress or as the robot senses the environment. As-built conditions can also affect the subsequent construction process.
Workspace Materials	Construction materials staged inside the robot workspace. They need to be considered for collision avoidance and may update during the construction process.
Task-Related As-Designed	The original design of the already-built components related to the current construction task. The information is also used by human co-workers and robots to understand as-designed versus as-built deviations and develop plans accordingly.
Unrelated As-Built	Built structures that physically exist on-site but are outside the robot workspace and are not related to the current construction task. The information is saved in BIM and the human co-worker can choose to visualize it in the digital twin for the completeness and reality of the virtual environment. However, the information is not considered for robot processing or collision avoidance.
Unrelated As-Designed	The original design of the components both inside and outside the robot workspace that are not related to the current construction task.
Unrelated Materials	Construction materials outside the robot workspace that are not considered for collision avoidance.
Virtual Collision	The space that is not physically occupied but the robot needs to avoid during the movement (e.g., a safety laser curtain that marks the edge of the work zone). The robot considers it for collision detection during motion planning.

- ***Robot operation support.*** Adding information to support robot operation in BIM can increase the success rate of HRCC work. The information added depends on the task type and robot intelligence level. For example, robots can determine how to pick up objects by visually detecting object geometry in some cases, but for some objects with irregular shapes, robots need external guidance to successfully pick up the object. In this case, the robot gripping pose should be added to BIM.

It should be noted that some of the abovementioned information has a high potential to be automatically generated by the computer or the robot. For example, Kim et al. (2020) proposed an approach for automatically generating steel erection sequences and Levine et al. (2017) used convolutional neural network and large-scale robot grasping experiments to generate robot grasping plans (Kim et al. 2020; Levine et al. 2017). Adel et al. (2018) used computational design to generate the cutting planes, gripping planes, and connections for off-site frame prefabrication (Adel et al. 2018). This research focuses on how to leverage such generated information for HRCC. The detailed technical approach for generating such information is thus excluded from the scope of this paper.

3.3.3 BIM-Driven HRCC System Framework

Five elements are included in the BIM-driven HRCC framework: 1) the BIM components that provide and save data about the construction project; 2) the Graphical User Interface (GUI) that supports both immersive VR and 3D options for human workers to interact with the robot; 3) ROS as the middleware for communication and the central unit for data processing, computing, and construction work process and physical robot control; 4) ROE, which is the construction site that includes robots, sensors, and materials; and 5) human workers who supervise the construction

process and intervene when necessary. In this study, Rhino 7 is used as the BIM platform and the GUI is developed in Unity with Oculus Rift S as the headset for the immersive VR option.

The information flow among system elements at different stages is shown in Figure 3.5. To prepare for construction, human co-workers need to physically set up the construction site and configure the BIM data (Section 3.3.3.1). When construction starts, information is taken from both the BIM and the construction site to generate the I2PL-DT system (Section 3.3.3.2). During the construction process, human co-workers bi-directionally communicate with the robot through ROS, which also integrates information from the BIM and the construction site, to supervise and intervene in the construction process (Section 3.3.3.3). After information fusion and processing, ROS controls the robot to conduct construction work. Occasionally, human co-workers need to physically intervene (e.g., replace a workpiece with one of a different shape) to resolve uncertainties on the construction site. When construction completes or reaches a certain checkpoint, the ROS environment that contains the related construction site information at the time sends the information to update the BIM repository and records data in the BIM for future reference (Section 3.3.3.4). It must be noted that even though this study chooses to update the BIM at certain time points, the framework also allows the BIM repository to be updated in real-time with the construction progress. Next, the technical details of each stage are discussed.

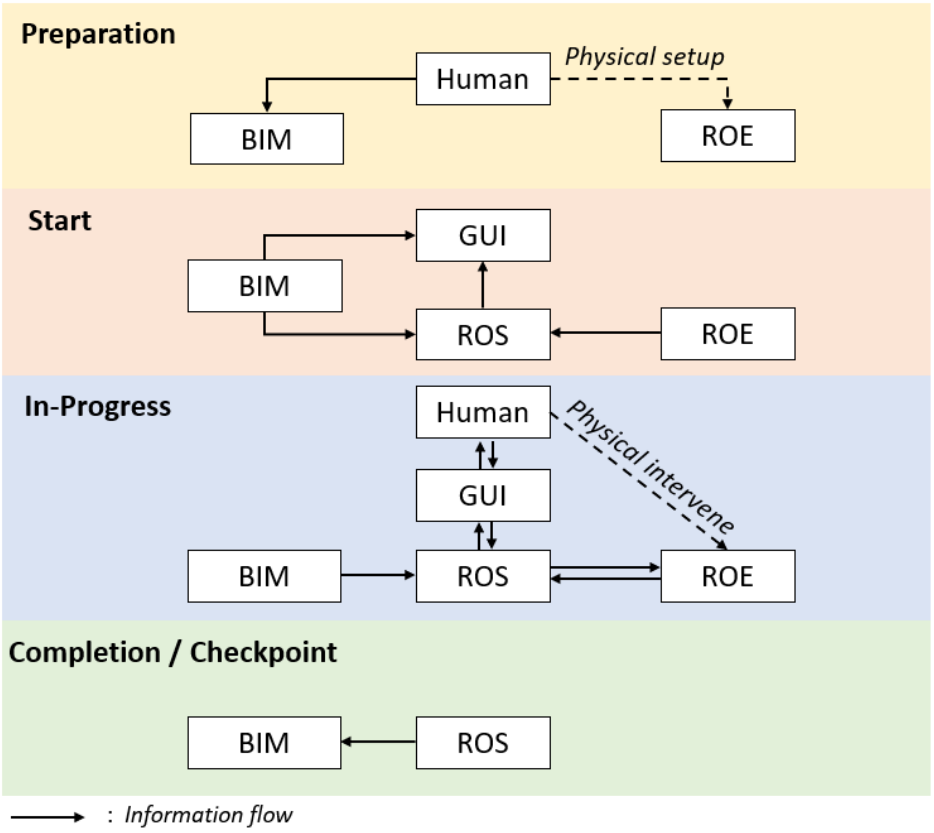


Figure 3.5 System Information Flow at Different Stages

3.3.3.1 Construction Preparation

Setup activities are needed both physically on the construction site and digitally in BIM. In general, workers need to start the sensors on the robot or construction site for the robot to perceive the environment. They also need to place the required construction materials in the robot workspace for the robot to reach and manipulate. Lastly, workers need to instruct the robot on what they want the robot to do in the BIM (e.g., selecting some drywall panels on the frame and placing them onto the "Target" layer). They should also adjust components' layers in the BIM according to the task scope (e.g., moving components that are outside the robot workspace onto the "Unrelated" layers). Considering that the BIM interface is intuitive and generally familiar to

construction personnel, it is typically easier for workers to indicate task scope in BIM rather than directly indicate this to the robot in ROS.

3.3.3.2 Start of Construction

To begin the construction, the digital twin system that integrates visualization, interaction, and computation functions needs to be generated. There are two digital twins in the system. One is the interaction module in the GUI developed in Unity to support visualization, supervision, and intervention. The other is the computation module in ROS that controls the construction workflow (e.g., installation sequence) and generates task plans (e.g., collision-free motion plans). The automated generation process of the digital twin system is shown in Figure 3.6.

The highlight of this process is that instead of manually importing BIM and creating functions in Unity for different construction projects, a template Unity program is developed. It contains template models with functions. When it receives information from other modules (e.g., BIM), Unity can quickly generate interactive game objects using these template models. For example, when it receives an object on the "Target" layer from BIM, it will instantiate a target-type template game object that is attached with functions for user selection and movement, load the mesh geometry received from BIM onto the game object for visualization, and change the game object attributes such as name, layer, and material accordingly. As a result, the template Unity program can automatically generate interactive digital twin interfaces for different BIM projects.

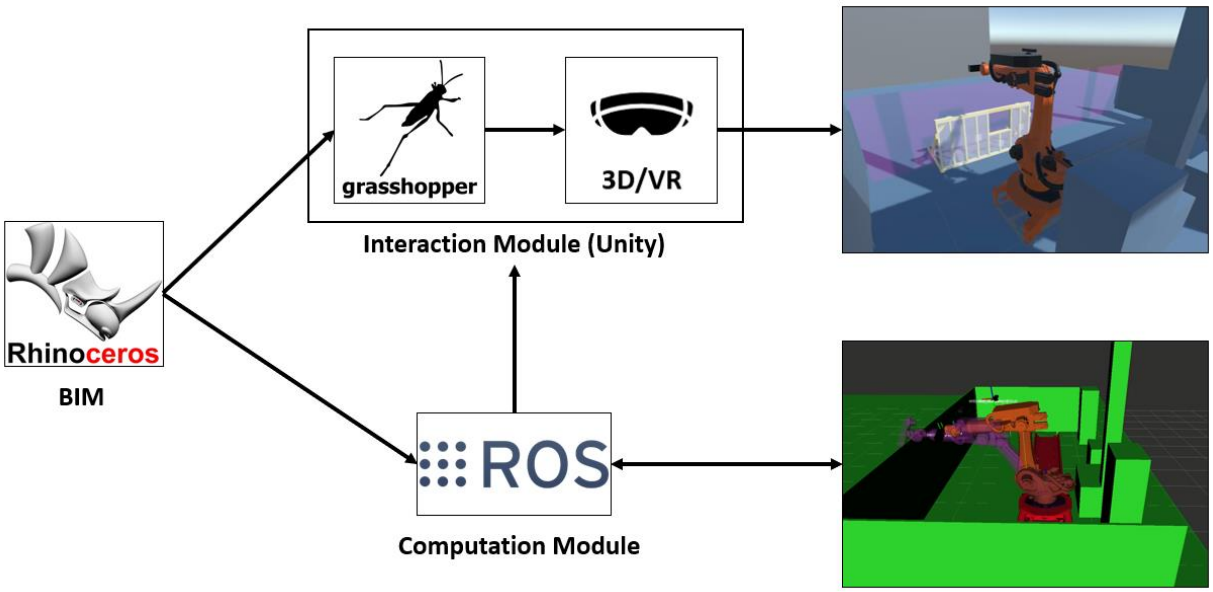


Figure 3.6 Automatic Digital Twin Generation Process

At the initial state, Unity contains 1) light and camera systems for human co-workers to visualize objects in the game interface; 2) event systems to capture user input and enable interaction (e.g., selection, moving); 3) virtual robot models generated from the Unified Robot Description Format (URDF) files and meshes from ROS. One virtual robot is synchronized with the actual robot for supervision. The other one is used to evaluate robot motion plans and only appears when the human workers preview plans; 4) ROS connectors to exchange data with ROS through Rosbridge using the ROS# library (Siemens 2021); 5) Rhino add-in to run Rhino and Grasshopper projects to retrieve BIM geometry and semantic (e.g., layer, name, type) data; 6) model templates that can automatically generate interactive game objects with the information received; 7) interface templates that appear at specific times to prompt and receive user input; and 8) a virtual billboard to show messages.

The workflow is initiated by starting the ROS program, which loads the robot model and related parameters and opens up the listeners and publishers, so the system starts to subscribe and

publish messages (e.g., robot joint states) among different modules. Then, the Unity program is started. As it starts, it builds connections with ROS. At this stage, only a robot model appears as a physical game object in the interface. The robot model is synchronized with the actual robot based on the subscribed joint states data. Since the Unity program also contains Rhino.Inside (McNeel 2022), which is an open-source add-in that enables other applications (e.g., Unity) in the Windows Operating System (OS) to run Rhino and Grasshopper projects, these two applications also start with the Unity program. Grasshopper then loads BIM into Rhino, retrieves information (i.e., geometry, color, name, layer) of objects from BIM, and sends it to Unity. As Unity receives the information, model templates can create game objects with the same names, colors, and geometry as what they received from BIM. The generated game objects are automatically placed onto the corresponding layers in the GUI. Based on the object layer, scripts that contain different functions are attached to the object, thereby creating different interaction patterns.

After the interactive digital twin is generated, BIM information is also sent to ROS. Existing objects on the "Workspace As-Built", "Workspace Materials", "Task-Related As-Designed", and "Virtual Collisions" layers are initially loaded into MoveIt, a motion planning framework integrated into ROS (Chitta et al. 2012), for collision detection while the robot moves to scan the environment. After scanning, as-designed models are removed from the MoveIt planning scene and as-built workpieces scanned by the robot are added to MoveIt as collision objects. As a result, the robot considers the as-built scene for collision avoidance during the construction process. Previously-recorded workspace materials may also update after robot scanning. Lastly, as-built workpieces and materials are sent to Unity to reflect the most up-to-date construction environment to the human co-worker. With these steps accomplished, the digital twin system is completely generated and construction can start.

3.3.3.3 Construction In-Progress

The process of construction is shown in Figure 3.7. Human operations and decisions are shown in orange, and blue elements show processes performed by the robot. The robot and its human co-workers interact through the GUI. The overall workflow is that the robot shows information and decisions intuitively in the GUI for its human co-workers to visualize. Then, the GUI detects the human co-workers' decisions and operations through their input and sends the information to the robot. During the interaction process, corresponding interfaces are generated from the interface templates to prompt the human co-workers on the current process and get inputs. The templates contain corresponding functions (e.g., sending messages to the robot on certain ROS topic channels when the user clicks certain buttons) and are automatically connected to corresponding scene objects when being generated.

After generating the digital twin, the robot retrieves the next target in the construction sequence from the BIM, highlights the target object in the GUI, and asks the human co-worker to confirm the target. If not in agreement, the human co-worker can select another target. After a target is confirmed, the robot checks whether there are deviations in the workspace that affect the operation to achieve the target as originally designed (e.g., a deviated workpiece occupying the target space will collide with the target if following the original design). If there is a deviation, the robot will propose suggestions to resolve the deviation. Otherwise, it will go ahead with the original plan from the BIM. The human co-worker can choose to directly accept the robot's suggestion (e.g., an adjusted pose to install the target). There might be situations where the robot cannot properly improvise a plan to resolve the deviation, so the system also allows the human co-worker to intervene and resolve the deviation (e.g., move the target to indicate the desired

installation pose in GUI). The specific approach for the human-robot work team to resolve deviation depends on the task type, which is introduced in the Case Study (Section 3.4).

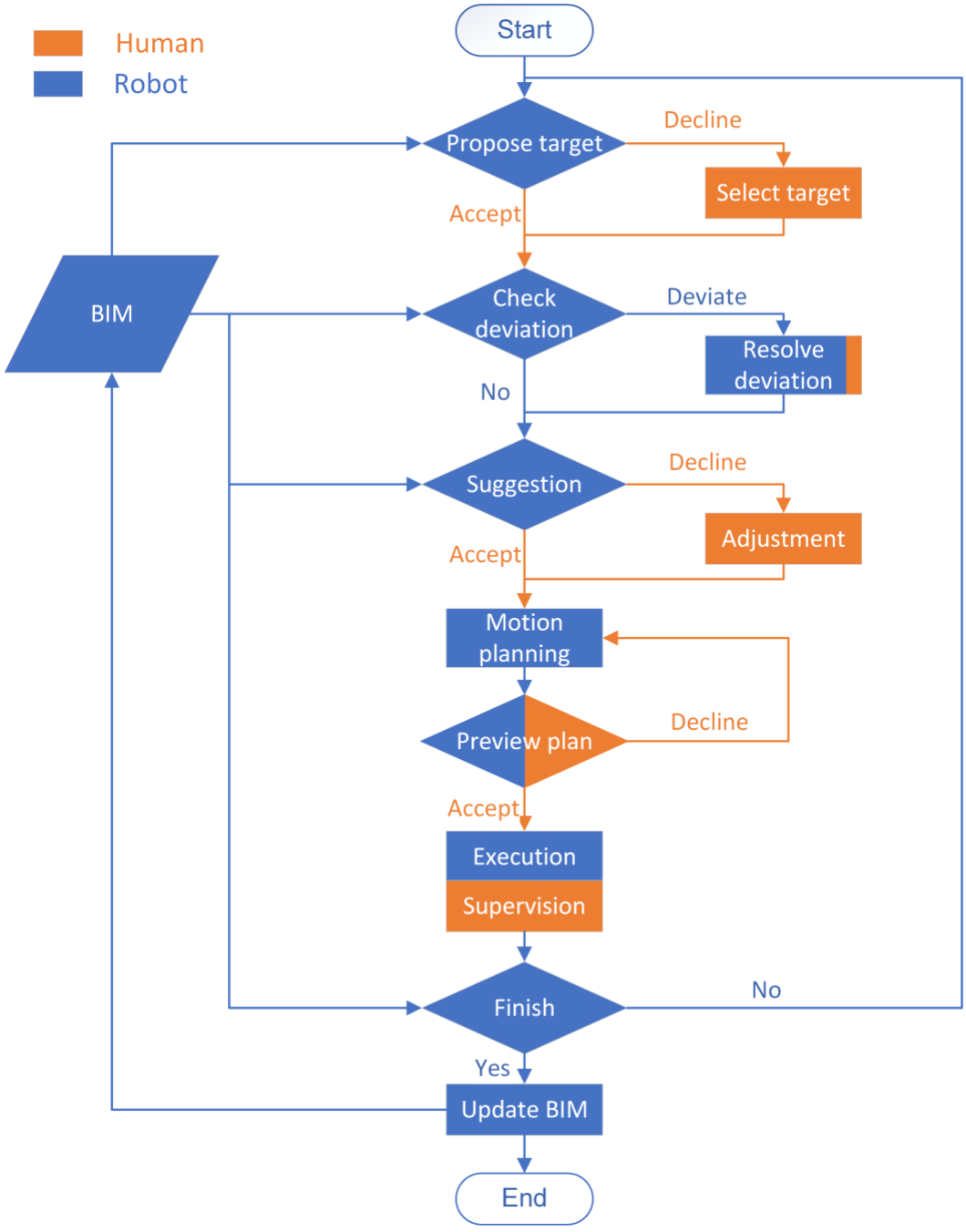


Figure 3.7 Co-Robotic Construction Process Flow Chart

Next, based on the confirmed plan, the robot generates a collision-free motion plan to achieve the target. When the human co-worker requests to preview the plan, a virtual robot will

appear in the GUI and demonstrate the motion plan as an animation. If the human co-worker is not satisfied with the motion plan, the robot will find another plan and demonstrate it until the human co-worker finds an acceptable plan. After a motion plan is accepted, the actual physical robot executes the approved plan. Since a robot model in GUI is synchronized with the actual robot, the human co-worker can supervise the robot execution states through this synchronized robot. They can also understand the robot's cognitive status (e.g., calculating the motion plan) through the messages in the GUI. The technical approaches for establishing connections within the digital twin system and achieving motion planning, plan preview, and execution and supervision functions are discussed in Wang et al. (2021). After a target is achieved, the system will check the next target in the sequence from the BIM and go through the process again. If no target is left in the queue, the assigned task is considered to be finished and the as-built information collected is updated in the BIM.

3.3.3.4 Construction Completion or Reach of Checkpoint

When an assigned construction task finishes or reaches a certain checkpoint, three sources of data are sent from ROS to BIM via Rosbridge using the COMPAS library (COMPAS 2021). The first source of data is workspace sensing data. As construction starts, the robot scans the environment to find out the as-built scenarios of the workspace. The sensing data is saved onto the "Workspace As-Built" layer in BIM. Second, robotic construction data is also saved onto the "Workspace As-Built" layer. Robot states and end-effector poses can accurately reflect how the work is performed, such as how a workpiece has actually been installed. Third, temporary material data is updated onto the "Workspace Materials" layer in the BIM. It reflects the type, number, and places where construction materials are placed on-site. It is inferred from the start state (e.g., materials originally prepared) and the construction process (e.g., how many materials are used).

These three sources of data are saved in the BIM repository for future reference. The repository may also be updated with incoming sensing data and construction work in the future.

3.3.4 Physical System Deployment

The physical portion of the system is deployed on a large-scale Kuka industrial robotic arm on the construction site. The industrial robotic arm is selected because it has a relatively large payload to manipulate heavy construction workpieces and higher flexibility to perform various construction tasks with different tools. The assumption is that the robot has a relatively static workspace (e.g., an area in a room) for each construction task. Workers and equipment conduct construction activities and move outside the robot workspace. For safety reasons, if they get into the robot workspace during the robotic construction process, the robot will stop moving until safety is confirmed by its human co-worker. The robot's human co-workers can enter the workspace to make interventions but upon their entrance, the robot will stop until they finish the intervention and leave the workspace, thereby ensuring their safety.

The system involves several devices connected to a local area network. Devices can communicate with each other through wired connections or wirelessly. The system framework for device communication and robot control is shown in Figure 3.8. The interfaces that the human co-worker directly uses, including the GUI in Unity and the BIM in Rhino, need to run on a computer with the Windows OS. The sensors are connected to portable microcontrollers (e.g., Raspberry Pi). Both the Windows computer and microcontrollers can communicate with ROS wirelessly through a router on the local area network via ROS messages. ROS runs on a computer with Ubuntu OS that is connected to the robot embedded PC through an Ethernet cable. When the computing core in ROS processes data it receives and decides to control the robot, it sends the joint states to the Automation Device Specification (ADS) interface of the programmable logic

controller (PLC) (Liang et al. 2022). ADS is an interface layer of Twincat PLC that allows commands and data exchange between different software modules (Beckoff 2022). The joint states data is received by the PLC and is then sent to the Kuka Robot Sensor Interface (RSI) using the EtherCAT protocol to control the robot (Liang et al. 2022).

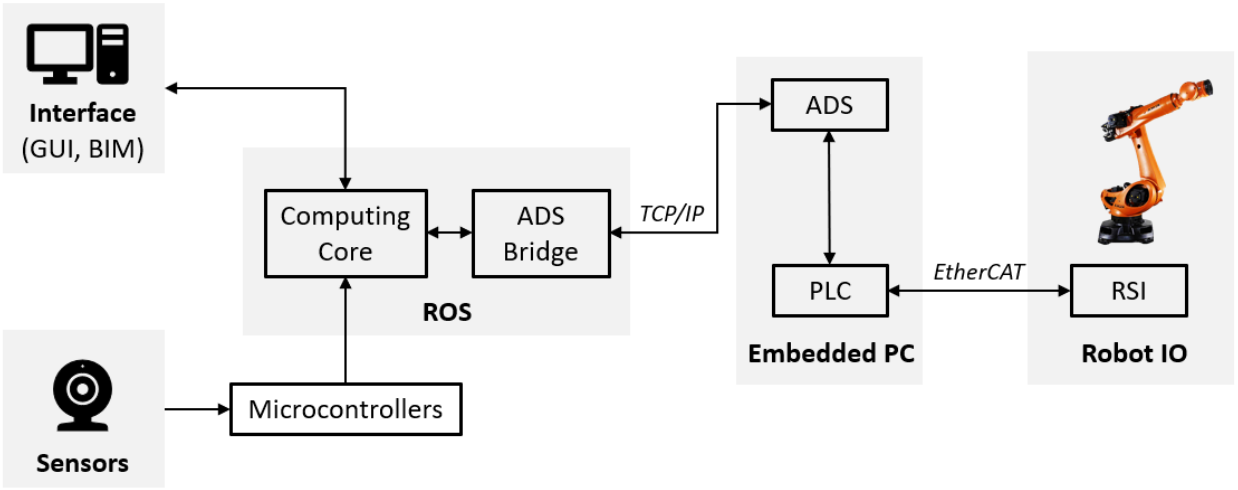
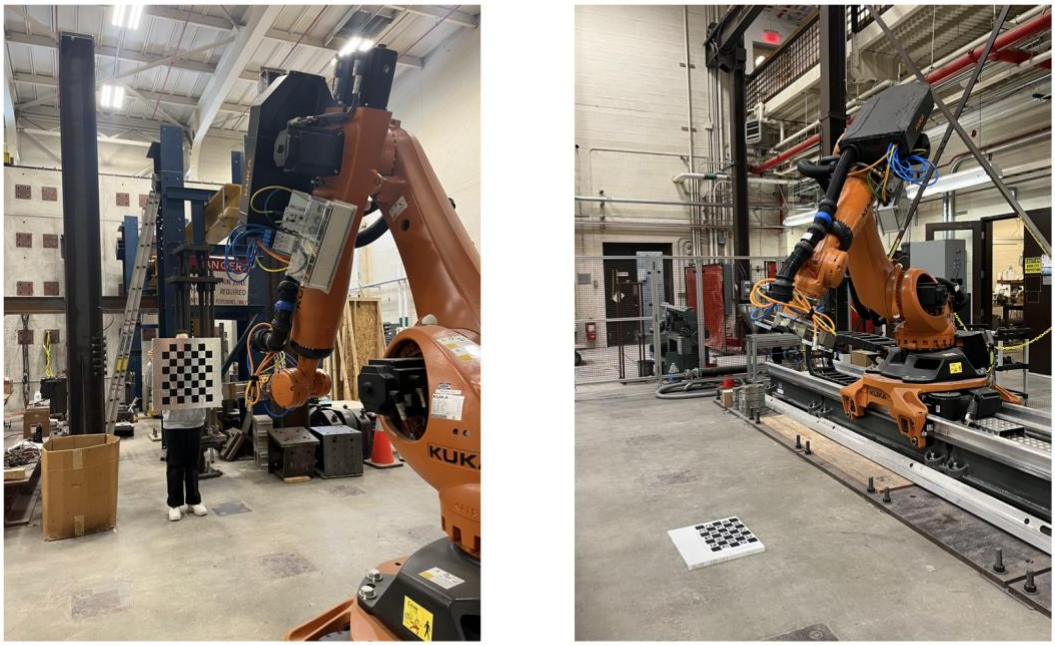


Figure 3.8 Physical Robot System Framework

An RGB camera is installed on the robot end-effector to perceive the workspace environment. The camera's intrinsic matrix is achieved by taking the average of 10 intrinsic calibration measurements, which is used to understand the camera's internal attributes such as focal length, principal point, and distortion (Open Robotics 2020). Since the camera is manually connected to the robot end-effector, its exact position and orientation on the robot are unknown. Therefore, hand-eye calibration is then conducted using MoveIt to understand the camera's transformation relative to the robot end-effector (PickNik Robotics 2022). The intrinsic and hand-eye calibration processes are shown in Figure 3.9. After camera calibration, the robot can localize objects' poses in its own frame to plan and perform construction work.



(a) (b)

Figure 3.9 Camera Calibration (a) Intrinsic Calibration (b) Robot Hand-Eye Calibration

3.4 Case Study

In order to verify the proposed system and explore the setup needed to physically deploy the system, a drywall installation case study is conducted in a research laboratory at the University of Michigan as a proof-of-concept implementation. As discussed in Section 3.3.1, it is assumed that the wall frame is prefabricated and installed on-site as a whole workpiece. In this context, the frame itself is assembled without deviation but is installed at a deviated pose on-site. Four drywall panels in two different shapes need to be installed onto the wall frame. The remaining part of this section introduces the physical and software setup and describes the HRCC process to perform the drywall installation task in detail.

3.4.1 Physical Setup

The robot used for the case study is a 6 DOF Kuka KR120 industrial robotic arm that has 120kg payload and 2.7m reaching range. By mounting it onto Kuka KL4000 Linear Unit, its base can move 4.5m linearly, which adds one DOF to the robot and significantly increases the robot's physical reach. Therefore, the robot has the capability to manipulate a regular-sized drywall panel. The robot workspace is shown in Figure 3.10. A safety gate is used to mark the robot workspace and a laser curtain is installed on the safety gate to prevent other workers and equipment from entering the robot workspace.

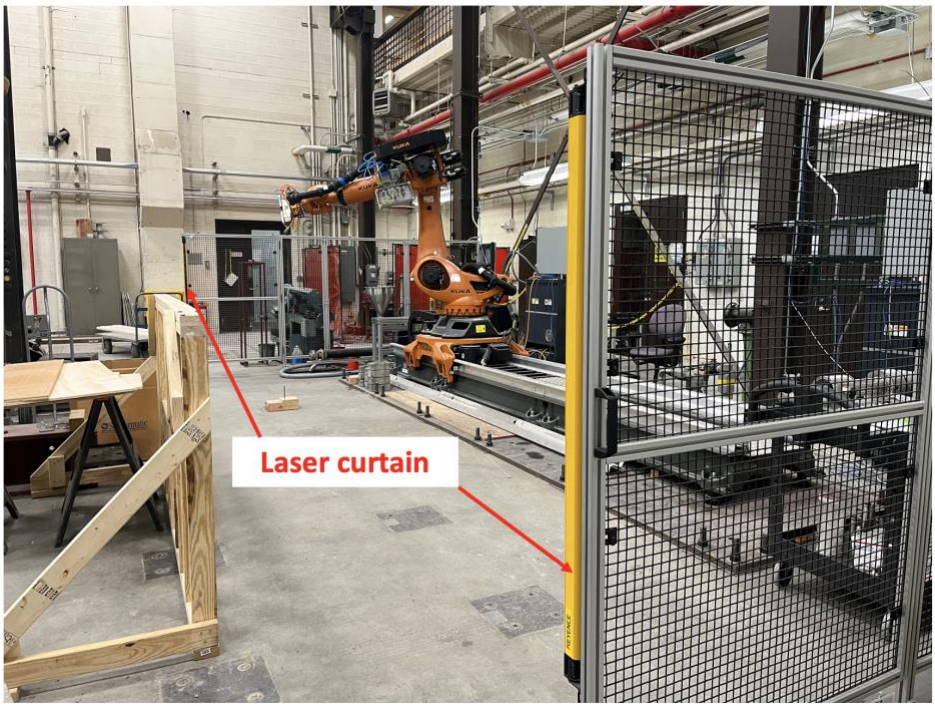


Figure 3.10 Robot Workspace

Since the goal of the case study is to verify the capability of the system framework and functions, the experiment is conducted on a 1:4 scale. A wall frame 4 feet (1.2192m) tall by 8 feet (2.2384m) long with a window area is built with studs on the back side to ensure stability (Figure 3.11). The robot needs to install three larger drywall panels of 2 feet (0.6096m) by 4 feet (1.2192m)

and one smaller panel of 2 feet (0.6096m) by 2 feet (0.6096m) onto the frame. A cubic handle is attached to each panel for the robot to grip. A pneumatic gripper is designed and connected to the robot with a tool changer and a connection plate (Figure 3.12). The jaws of the gripper are made with slopes to clutch the cubic handle on the drywall panel. Stabilizers are installed to ensure that the drywall panel can fully contact the gripper to avoid torque and shaking during manipulation. Rubber pads are used to add friction between the gripper and the cubic handle to prevent slippage.

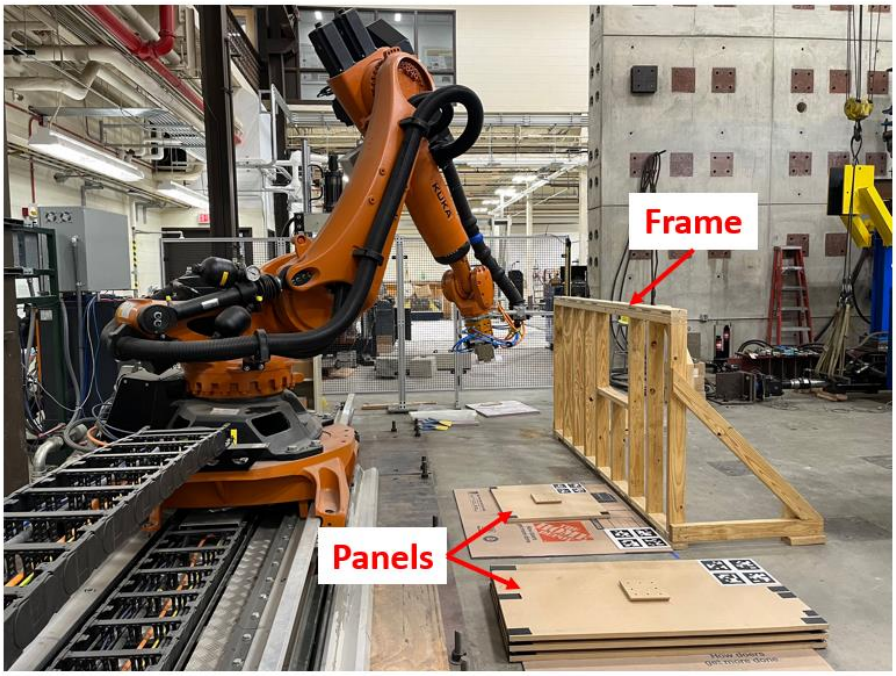


Figure 3.11 Wall Frame and Drywall Panels used in Case Study

Given that only one robotic arm is deployed, for demonstration and experiment repeatability, panels are attached to the wall frame by magnets after the robot releases the panels. For actual construction work, the panels could be fixed onto the wall frame, potentially by a human worker or another robot. AprilTag fiducial markers (Wang and Olson 2016) are installed onto the wall frame and on drywall panels for the robot to localize the 6 DOF pose of the workpieces. A monocular RGB camera is fixed onto the gripper. It is connected to ROS running on a Raspberry

Pi microcontroller powered by a portable battery. Raspberry Pi can send the camera sensing data as ROS messages wirelessly to the ROS master running on the Linux machine.

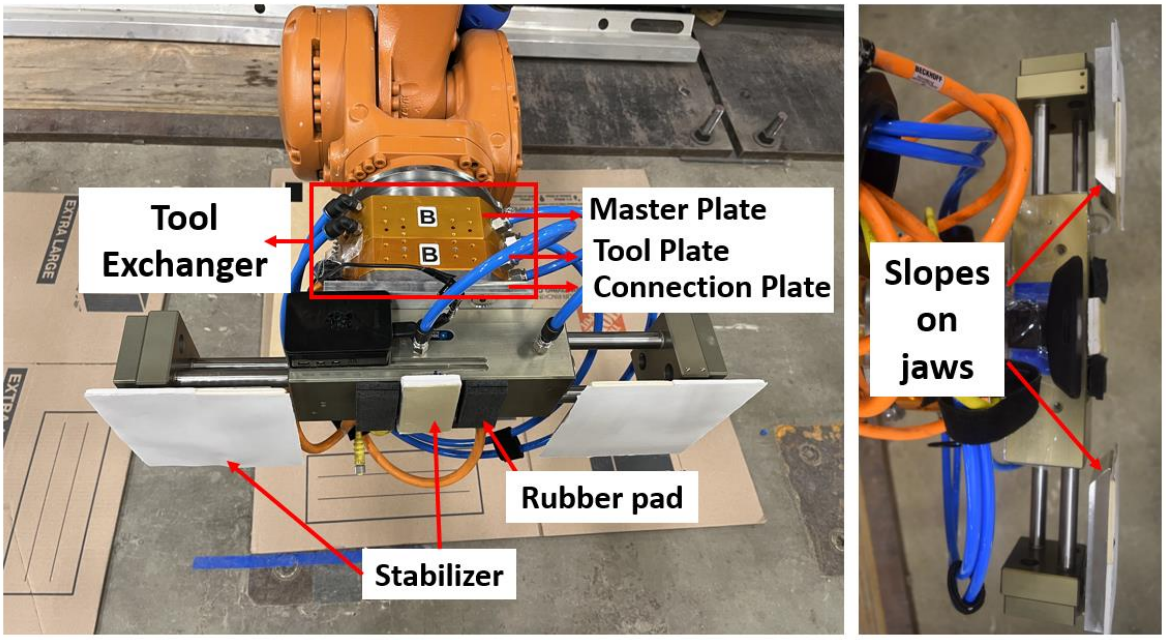


Figure 3.12 Gripper Design

3.4.2 BIM Preparation

Figure 3.13 shows a screenshot of the BIM used for the drywall installation task. The surrounding wall of the laboratory is set to be transparent grey to make it easier to visualize the robot workspace. Shop drawings for drywall installation are shown on the right bottom of the figure. The BIM indicates how the panels are designed to be installed. In this drywall installation task, these panels are the targets. The laser curtain is the plane that lasers come through which does not physically exist. However, interruption of the laser will cause the robot to stop for safety reasons, so it is undesirable for the robot to get into the curtain during operation. Thus, the laser curtain is considered a collision object during robot motion planning. The BIM components and their corresponding layers are shown in Table 3.2. Before construction, the frame has already been

installed and materials are prepared but their poses are unknown and need to be detected by the robot.

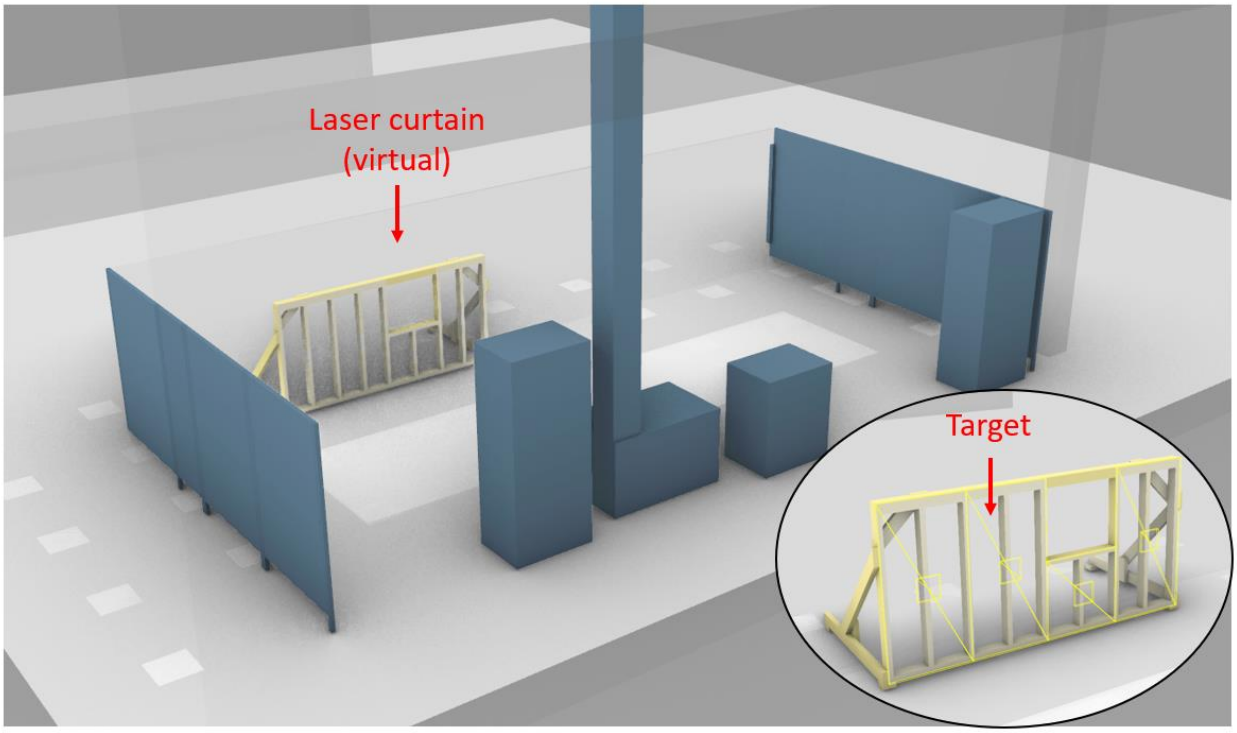


Figure 3.13 Drywall Installation BIM

Table 3.2 BIM Components List

Components	**Layers**
Ground floor	Workspace As-Built
Workspace surroundings (Blue)	Workspace As-Built
Frame (Design) (Yellow)	Task-Related As-Designed
Target panels on frame (Design)	Target
Laboratory walls, breams, and columns	Unrelated As-Built
Laser curtain (Virtual)	Virtual Collision

BIM contains the semantic information needed for robot processing and construction. Take the target panel as an example, it contains its identifier (i.e., name), layer, its relationship with other components (e.g., its parent is Frame), construction sequence, and type (e.g., large or small) (Figure 3.14). It also indicates how the robot should grip it (e.g., at its center with orientation perpendicular to its largest surface). The gripping pose is also used to indicate the object 6 DOF

pose for manipulation. In this study, the program automatically calculates the gripping poses and pose indicators using the centroid of the Rhino object. Otherwise, robot gripping poses and pose indicators can be automatically generated with computational design or human workers can indicate them by selecting points and directions in Rhino.

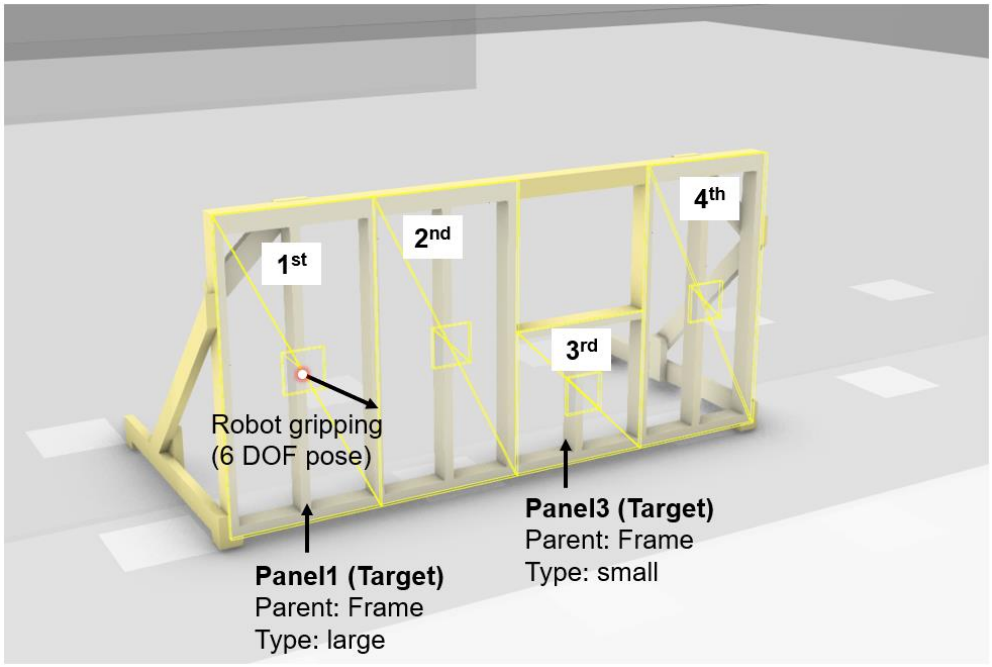


Figure 3.14 Semantic Information in BIM to Drive Robotic Construction

3.4.3 Deviation Resolution

Three types of deviation are considered in this case study, which mainly focuses on the pick-and-place type of tasks:

- ***Parent deviation:*** The parent of the target deviates. Since the target needs to be connected to its parent, the installation pose of the target should be adjusted accordingly. The robot first calculates the design-built deviation $\boldsymbol{T_D^B}$ of the parent using Eq. 1

$$\boldsymbol{T_D^B = T_D^{-1} T_B} \quad (1)$$

where $\boldsymbol{T_D}$ is the as-designed transformation of the parent the robot received from BIM and $\boldsymbol{T_B}$ is the as-built transformation of the parent the robot detected with its camera. Next, the robot calculates the suggested installation transformation $\boldsymbol{T_t}$ to install a target t using Eq. 2

$$\boldsymbol{T_t = D_t T_D^B} \tag{2}$$

where $\boldsymbol{D_t}$ is the as-designed transformation of the target. Lastly, the installation transformation matrix $\boldsymbol{T_t}$ is converted into an installation pose $\boldsymbol{P_t}$ for robot planning and operation.

- ***Seat deviation:*** The seat that supports the target by gravity deviates. The target does not have any hard connection to the seat. In this case, only the height of the target pose will change with its seat (Eq. 3)

$$\boldsymbol{P_{BZ} = P_{DZ} + (S_{BZ} - S_{DZ})} \tag{3}$$

where $\boldsymbol{P_{BZ}}$ and $\boldsymbol{P_{DZ}}$ are as-built and as-designed positions of the target on the Z-axis, and $\boldsymbol{S_{BZ}}$ and $\boldsymbol{S_{DZ}}$ are as-built and as-designed positions of the seat on the Z-axis. In this situation, the orientation of the target is not affected.

- ***Nearby object deviation:*** When some objects near the target are built with deviation, they might occupy the originally planned space of the target. The solution to this type of deviation is shown in Figure 3.15. While deciding the target installation pose, the robot checks whether the planned installation place collides with any as-built objects. If no collision is detected, the robot will go ahead with the original plan upon the human co-worker's confirmation (Figure 3.15a). Otherwise, two options are provided to the human co-worker. The human co-worker can choose to take the robot's suggestion to offset the target (the red block) installation pose (Figure 3.15b). However, it may affect

the installation of subsequent targets (the blue block), causing plan changes for several targets. The human co-worker can also choose to manually resolve the deviation by replacing the target with one in a different shape (the red block) (Figure 3.15c). Human intervention can resolve more complex deviation cases and provide solutions that do not affect the subsequent targets (the blue block).

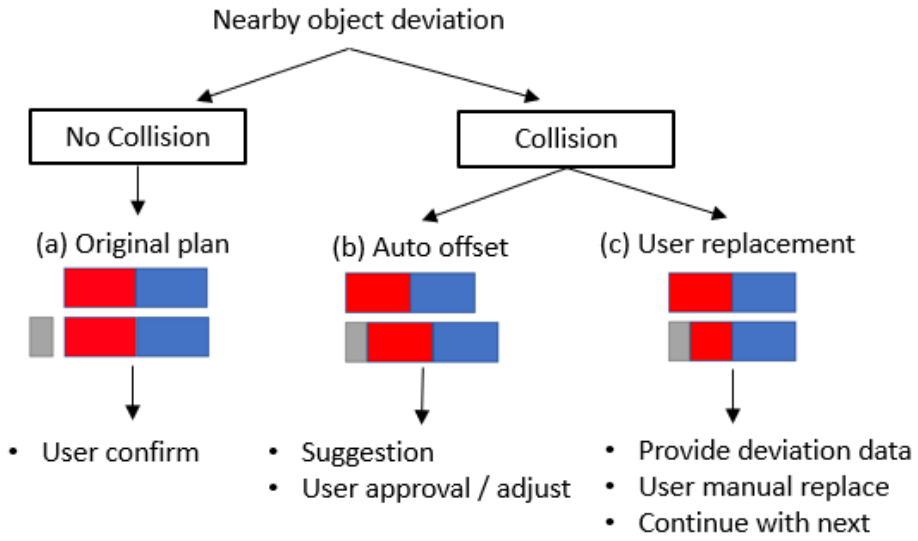


Figure 3.15 Solution to Nearby Object Deviations

3.4.4 HRCC Process

To prepare for the construction work, the drywall panels are placed into two stacks in the robot workspace (Figure 3.11). The camera is calibrated to detect AprilTag fiducial markers in the robot frame. There is a marker on each panel and the wall frame. Offset from the markers to the objects' pose indicator points is recorded and input into the system. The marker is not only used for pose estimation but also provides semantic information (e.g., corresponding object type and quantity).

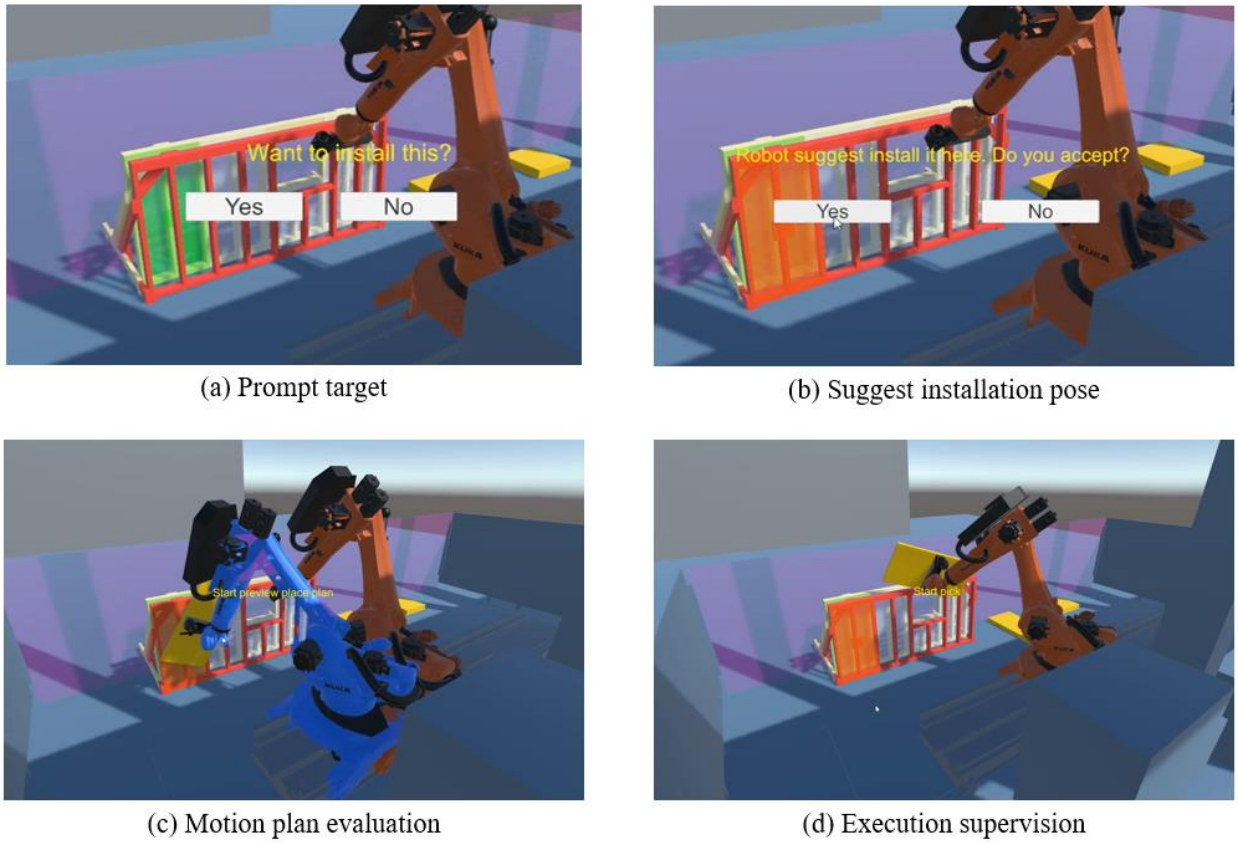
As construction starts, the robot follows a predefined trajectory to scan the environment. After scanning, the robot replaces the as-designed wall frame with the as-built one in the MoveIt planning scene. In the meantime, material poses are inferred from the detected stack location,

quantity, and type data attached to the marker. The materials are also added to the MoveIt planning scene as collision objects. When the robot plans motion with materials in hand, the collision object of the corresponding material is attached to the robot end-effector to ensure that the material held by the robot does not collide with the environment or the robot itself. The as-built wall frame and materials are also sent to Unity to be generated in GUI to support user visualization and decision making.

The robot first highlights the next drywall target in the construction sequence in the GUI and asks for the human co-worker's confirmation or adjustment (Figure 3.16a). After the human co-worker confirms the target, the robot uses the as-built and as-designed deviation of the wall frame to calculate a suggested drywall installation pose. Visualization of the suggested pose is then generated in GUI for the human co-worker's approval or adjustment (Figure 3.16b). Either in VR or in 3D mode, human co-workers can adjust the camera view to inspect the environment from the perspectives they prefer, while VR offers a more natural and intuitive sight of view control.

After the installation pose is approved, the robot generates the motion plan to first pick up a corresponding type of panel (i.e., large or small) from the detected panel stacks and then places it with the approved installation pose onto the wall frame. Upon request from the human co-worker, a virtual "planning" robot manifests in the GUI and demonstrates the robot motion plan and how the panel is manipulated during the installation process for evaluation (Figure 3.16c). If the motion plan is approved, the physical Kuka robot executes the plan, and the human co-worker can supervise the robot execution process with the synchronized robot and understand the robot status from the messages in the GUI (Figure 3.16d). After the robot releases the panel, the panel is attached to the wall frame with magnets. The actual installation pose is recorded by the robot and the robot prompts the next panel in the sequence for installation. The quantity of materials in

the corresponding panel stack is reduced by one, and the position for the robot to reach the next piece of panel in that stack is updated accordingly. These procedures are repeated until all four pieces of drywall panels are installed. The snapshots of the physical robot drywall installation process during the laboratory experiment are shown in Figure 3.17.



(a) Prompt target

(b) Suggest installation pose

(c) Motion plan evaluation

(d) Execution supervision

Figure 3.16 Screenshots of the HRCC Process

After all the panels are installed, the as-built condition of the wall frame is sent from ROS to Rhino through Rosbridge using COMPAS as the workspace sensing data (COMPAS 2021). The recorded installation poses of all panels are also sent to be saved in BIM as the robotic construction data. Both the workspace sensing data and the robotic construction data are saved onto the "Workspace As-Built" layer. Lastly, the up-to-date conditions of the panel stacks are sent to Rhino.

Human co-workers can decide to save it in BIM onto the “Material” layer to reflect the quantity and location of the remaining panels on-site. The updated scene in BIM is shown in Figure 3.18.

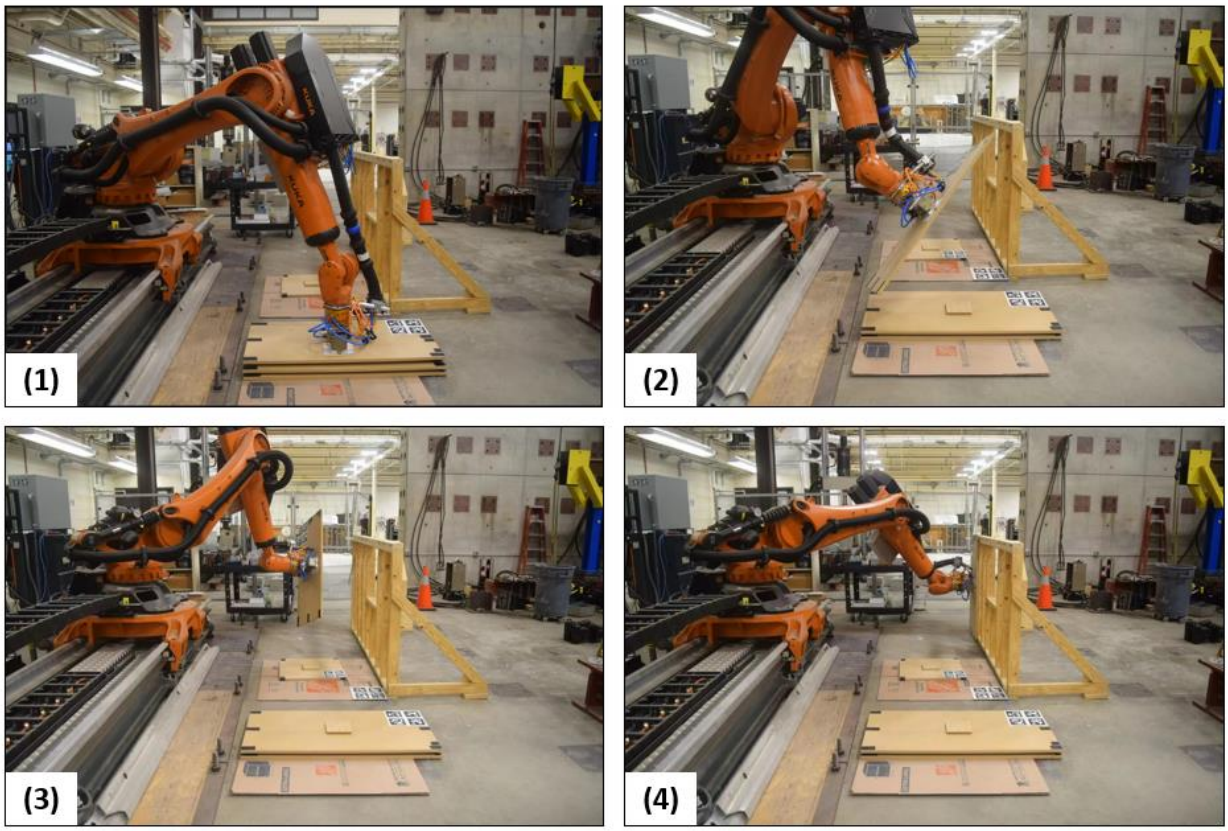


Figure 3.17 Snapshots of Physical Robot Drywall Installation Process

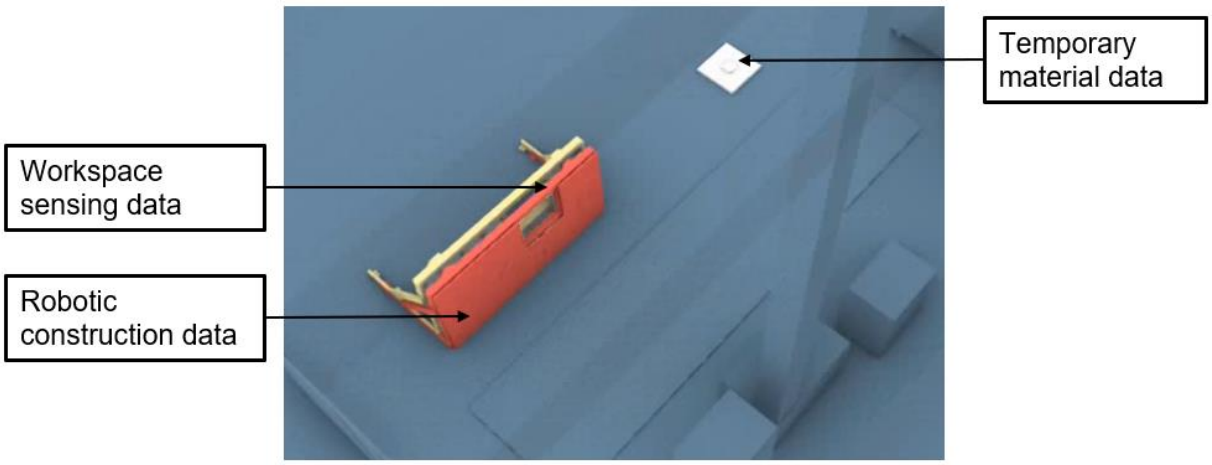


Figure 3.18 Updated BIM

3.5 Experiments and Results

With the proposed BIM-driven HRCC workflow, the robot needs to adequately localize workpieces in the construction environment, make decisions and suggestions to adapt to uncertainties, and accurately reach specific positions and manipulate workpieces. Errors in any of these aspects will disrupt the proposed workflow. Thus, three experiments are conducted to analyze three primary sources of errors and evaluate system performance. First, errors in localizing workpieces are assessed with a visual localization experiment. Error in localizing as-built structures will affect robots' suggestions to install subsequent workpieces. Additionally, errors in detecting materials will lead to inaccurate gripping and placement of workpieces. Secondly analyzed are the robot end-effector pose error. The robot movement is controlled through motors on its joints. Error in each joint position will lead to inaccurate end-effector pose and cause inaccuracy in construction task implementation. Third, since the robots have limited adaptability, there are possibilities that the robot cannot give provide feasible suggestions to adapt the plan, and thus human intervention is needed. Therefore, a block pick-and-place experiment is conducted to have a comprehensive evaluation of the robot's capability in decision making (e.g., adjusting the plan to resolve deviations) together with visual localization and manipulation accuracy.

3.5.1 Visual Localization Accuracy Evaluation

The objective of this experiment is to evaluate the accuracy of the robot onboard camera to localize workpieces through fiducial markers. Coordinates of 200 AprilTag marker bundles in the robot frame are detected by the camera and then compared with the ground truth. Each bundle consists of four markers with a size of 10.4cm. The distance between the robot end-effector and the marker is between 0.6m to 1.8m. Analysis of visual localization errors at each DOF is shown in Table 3.3. The error reflects a cumulative error of camera intrinsic parameters calibration, robot

hand-eye calibration, and how the camera localizes the fiducial marker in its frame. It is also found that the localization accuracy is higher when the camera is facing the fiducial marker, as the localization result of 50 markers shown in Table 3.4.

Table 3.3 Visual Localization Error

	Position (cm)			Rotation (rad)		
	X	Y	Z	Roll	Pitch	Yaw
Mean	0.7277	1.2693	0.0984	0.0224	0.0153	0.0062
Median	0.4077	1.0925	0.6633	0.0141	0.0084	0.0052
Std	1.1108	1.7889	1.1130	0.0270	0.0223	0.0056
Min	0.0026	0.0026	0.0021	0.0056	0.0002	2.22e-05
Max	3.6410	4.8168	5.3753	0.0883	0.0763	0.0212

Table 3.4 Visual Localization Error (Camera Facing Marker)

	Position (cm)			Rotation (rad)		
	X	Y	Z	Roll	Pitch	Yaw
Mean	0.1820	0.4123	0.3288	0.0121	0.0081	0.0050
Median	0.1523	0.2227	0.2483	0.0030	0.0055	0.0050
Std	0.2138	0.6104	0.3765	0.0230	0.0118	0.0022
Min	0.0041	0.0051	0.0021	5.56e-06	4.65e-04	9.87e-04
Max	0.4682	1.7451	1.1205	0.1019	0.5026	0.1232

3.5.2 End-Effector Pose Accuracy Evaluation

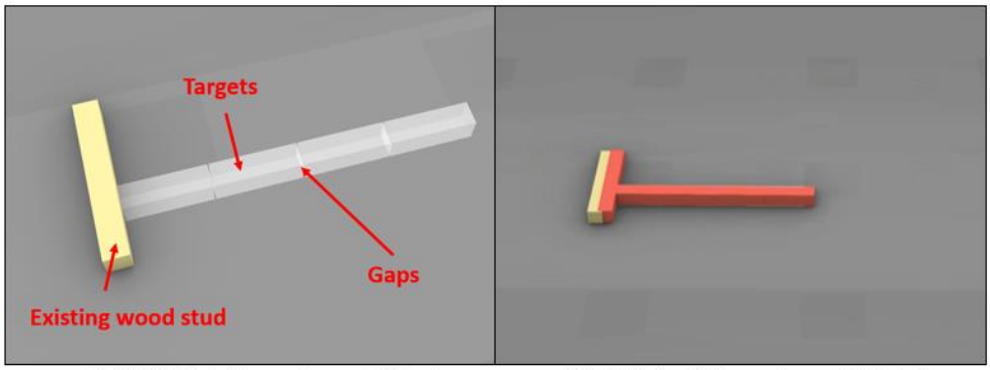
In order to evaluate end-effector pose accuracy, 100 random target poses that can be reached by the robot are selected as the ground truth. The inverse kinematic solver in ROS calculates the motion plan and sends joint states command to the robot to control it to reach the target. After the robot reaches the target, the actual joint states of the physical robot read by the encoders at each joint are sent back to ROS. Then, actual end-effector poses are calculated with the actual joint states through forward kinematics. Errors are calculated by comparing the calculated actual end-effector poses and the ground-truth target poses, which are shown in Table 3.5.

Table 3.5 End-Effector Pose Error

	Position (cm)			Rotation (rad)		
	X	Y	Z	Roll	Pitch	Yaw
Mean	4.907e-03	4.134e-03	3.884e-03	7.220e-04	5.867e-04	6.3970e-04
Median	3.823e-03	3.711e-03	3.884e-03	6.577e-04	5.776e-04	5.919e-04
Std	4.847e-03	4.843e-03	3.884e-03	7.220e-04	5.867e-04	6.397e-04
Min	1.221e-04	2.6090e-05	3.725e-05	2.782e-05	4.317e-06	4.973e-06
Max	1.037e-02	9.750e-03	1.101e-02	2.704e-03	1.238e-03	2.697e-03

3.5.3 Block Pick-and-Place Task

A block pick-and-place task that involves a line of four blocks is conducted as an overall evaluation of the system. The BIM repository of the system and the physical experiment setting are shown in Figure 3.19. Four wood blocks are stacked on the ground floor. A stud that marks the start of the four blocks is considered as a nearby object. The robot first scans the environment to localize the block stack and stud. Then, it needs to first pick up a wood block from the stack of blocks and place it alongside the stud placed on the ground. If the robot finds the stud takes up the space for the planned block placement target, it will suggest offsetting the block placement target to avoid collisions. If the stud is deviated but will not collide with blocks (e.g., deviate outward from the block target position), the robot will follow the original plan and will not make an adjustment. In order to increase system tolerance to the errors and prevent damage to experiment materials and the robot, gaps of different sizes (10mm, 5mm, 3mm, 1mm) are left between blocks. The gripper releases and drops the block when it is about 2mm above the ground floor. For each gap size, 10 trials of picking and placing all four blocks were carried out. The wood stud was deliberately placed in collision with the planned target for 5 out of the 10 trials to test the robot's capability to resolve deviation. The robot was operating at 7% of its full speed.



(a) BIM at Experiment Start (b) BIM at Experiment Finish

(c) Experiment Start (Physical) (d) Experiment Finish (Physical)

Figure 3.19 Experiment Settings

During the experiment, the human co-worker interacts with the robot through the 3D interface. The goal of this experiment is to evaluate the overall performance of the proposed system when the human workers completely rely on robot suggestions. Therefore, the human co-worker agrees with all robot suggestions and does not perform any adjustment or manual intervention during the construction process. However, when the co-worker feels the robot planned manipulation trajectory is not optimal (e.g., taking extra rotations), the co-worker would request the robot to generate a new motion plan for evaluation. The number of replanning requests from the human co-worker is recorded. The trial is counted as a failure if the robot does not place all four blocks successfully. The success rate and the number of replanning requests for different gap sizes are shown in Table 3.6. “Successful placements” means the number of blocks successfully

placed without any collision by the robot. Once a block fails, the trial ends and the rest of the blocks in the four-block line are not placed. On average, a successful pick-and-placement trial of four blocks takes 217.31 seconds. 56.38 seconds are used for the human co-worker's decision-making, such as confirming targets and previewing motion plans. The average time taken by robot computation and execution is 160.93 seconds.

Table 3.6 Experiment Results of Block Pick-and Place Task

Size of gap	Success rate (%)	Replan request / Successful placements	Reason for failure (occurrence)
10mm	100	7 / 40	
5mm	90	2 / 38	Hit ground (1)
3mm	90	6 / 36	Collide with stud (1)
1mm	60	5 / 26	Collide with stud (3) Hit ground (1)

3.6 Discussion

In Section 3.5, three experiments are conducted to analyze the sources of errors that can disrupt the proposed BIM-driven HRCC workflow. From the result of the end-effector pose evaluation (Section 3.5.2), it can be found that the robot can move its end-effector very accurately to the designated pose. Such a high level of precision is one distinguishing characteristic of an industrial robot. During the block pick-and-place experiment, it is observed that once the first block is placed, the rest of the blocks are placed without collision. It indicates that the robot makes good decisions to resolve deviation and find the appropriate pose to place the target, and it executes the plan with high precision.

Most failure cases are caused by the first block colliding with the stud while being placed (Figure 3.20). There are also two cases of failure where the blocks were moved too close to the ground and the robot sensed excessive force on its end-effector and generated errors. The reason for these collisions is that the robot does not localize the block stack or stud accurately. If the robot

does not localize the block stack accurately, it will pick up the block with offsets instead of at its designated pick place. Consequently, when the robot places it, this offset will cause the workpiece to collide with the neighboring objects. For example, if the robot does not localize the stud accurately, it cannot correctly decide the placement pose and thus causes collisions. Such inaccuracy in localizing the wall frame and drywall panels also affected success in the installation of drywall panels.

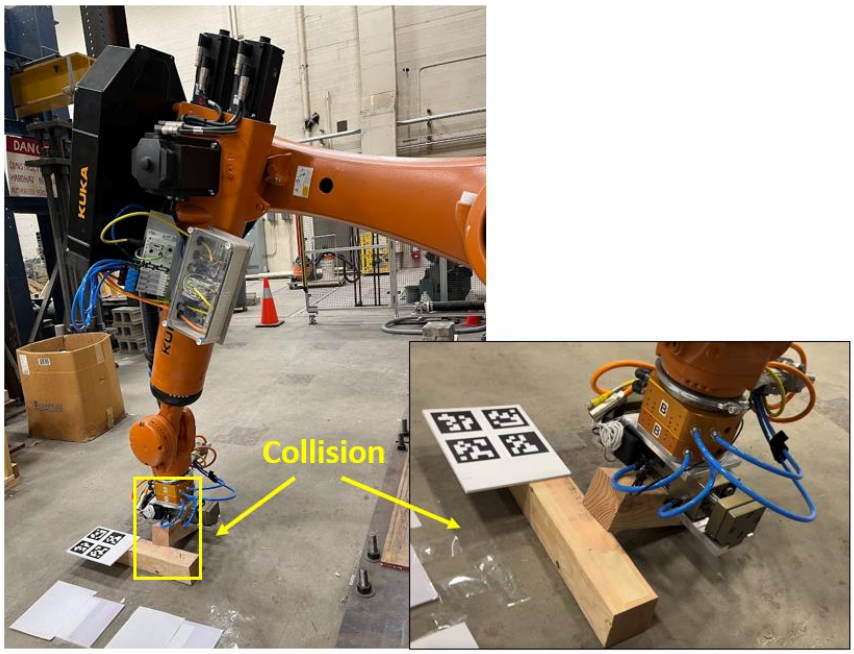


Figure 3.20 Collision During Block Placement

Localization of workpieces is critical for field construction, which differentiates field construction from off-site prefabrication. The factory-level environment for off-site prefabrication allows a structured workflow for workpieces to be accurately transported to a known place where the robot can pick them up (Yang and Kang 2021). However, in-field construction robots need the capability to localize the workpiece to determine the pick-up and install locations. Therefore, accuracy to visually localize objects significantly affects construction work quality. Results in Section 3.5.1 shows that the camera has better localization accuracy when it is facing the fiducial

marker. The comparison of the general marker detection and the cases when the camera is facing the marker is illustrated in the box plot in Figure 3.21. In addition to a smaller median value, the localization error also falls in a smaller range when the camera is facing the marker, indicating a more stable performance. Therefore, in the case study and experiment, the marker is detected with the camera facing the marker. However, it limits the scanning flexibility and efficiency for an actual construction task. Moreover, even though the visual localization error is mostly within 1cm with the orientation error less than 1˚, it is not sufficient for some construction tasks that require high accuracy with the workpieces in close contact with each other. Therefore, a more accurate localization method should be investigated in the future to maximize the success rate of the proposed workflow.

Considering the visual localization accuracy, the success rate of the block pick-and-place experiment is found to be particularly high. Several reasons may lead to this result. First, all the blocks are stacked at one location. Even if the robot localizes the stack inaccurately, it is still picking up each block from the same position. As a result, when the robot places the block, the block maintains the same offset. If the first block is placed successfully, the rest of the blocks will not collide with the previous one. Second, the gripper always grabs the block at its center. Even if there is a limited error in position or orientation, the gripper can correct it when closing the jaws. Third, only the localization error in a certain direction can cause task failure. For example, the block will only collide with the stud if the stud is very close to the block target pose while the robot thinks the stud is still far away and it does not make adjustments to resolve the collision. Fourth, the robot is dropping the blocks onto the ground at a height of about 2mm, which increases the tolerance of the task. Even if the robot drops the block from a slightly higher or lower position, the block can still be placed successfully.

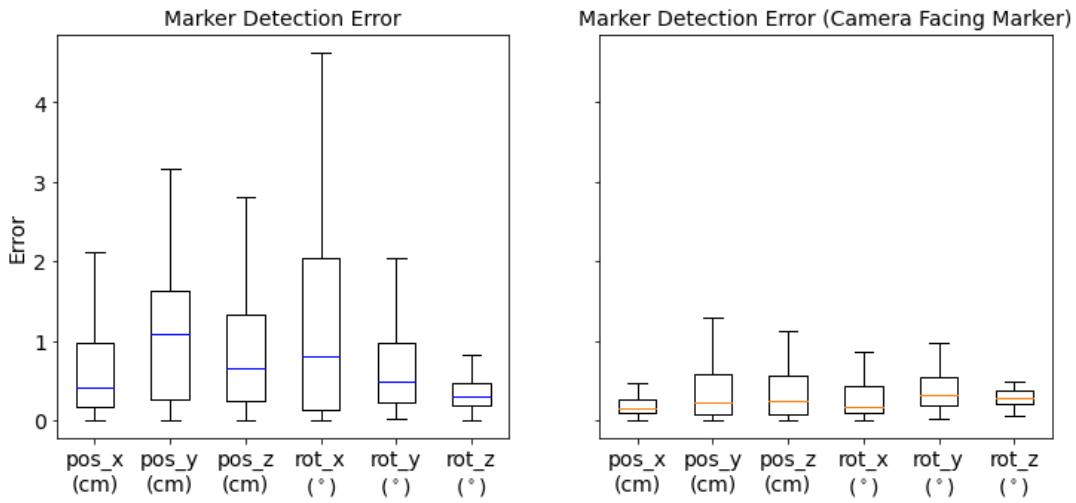


Figure 3.21 Marker Detection Error Comparison

During the physical experiment, several limitations are detected and future research directions are identified. First, as discussed above, although the robot can localize objects with millimeter-level accuracy, it is not sufficient for some construction tasks that require high precision. It also takes time for the robot to scan fiducial markers on all related workpieces in the workspace. A faster and more precise approach for the robot to perceive the environment should be considered in the future. Second, because of the laboratory condition, only one robotic arm is used and the panels are grabbed by a 2-jaw gripper in the drywall installation case study. One more robotic arm can be included to fasten panels onto the wall frame, and a vacuum gripper can be used to grab panels so that cubic handles are not needed. The robotic arm can be seated on the mobile platform to give the robot higher flexibility to move around the construction site. Third, information to support robotic construction, such as robot gripping pose and construction sequence, is manually created in BIM. Future work should integrate computational design into the system, which can automatically generate detailed digital fabrication information, construction sequence, and gripping planes for workpieces (Adel 2020). Lastly, future studies may consider

applying force feedback control and reinforcement learning for more precise and adaptable workpiece manipulation and to avoid collision between workpieces (Liang et al. 2021a).

3.7 Conclusions

This chapter proposes a closed-loop BIM-driven HRCC system that covers the technical solutions from the preparation stage to the end of the construction work. The proposed system has several improvements to the previous I2PL-DT system as well as independent contributions. First, it presents a BIM framework that supports HRCC. BIM contains the semantic and geometric information human workers and robots need for construction, and the defined layer system provides a unified standard to interface BIM with the interactive digital twin, thereby improving BIM interoperability. Second, an automatic approach for generating interactive digital twins for HRCC is proposed, which is enabled by a template-based Unity program and the defined layer system in BIM. Third, this chapter introduces an approach for resolving workpiece placement deviations using a combination of as-designed data from BIM and perceived as-built information. Lastly, the construction site as-built information is sent back to the BIM to be recorded, forming a closed-loop system. The research findings demonstrate the potential for using as-built BIM to support decision-making and automation in subsequent construction, operation, and maintenance of a facility. Physical experiments are conducted to identify the effort needed to enable a physical construction robotic system, evaluate system performance, and recognize limitations for future improvements. Overall, through the integration of BIM, the proposed system not only improves construction work quality but also increases the robot's capability in lower-level task planning thereby reducing human co-workers' planning efforts.

Chapter 4 Enabling Automatic High-Level Motion Sequencing in Robotic Construction Assembly through Interactive Learning from Demonstration

4.1 Introduction and Motivation

In the previous chapters, a BIM-integrated immersive and interactive process-level digital twin (I2PL-DT) system for human-robot collaborative construction (HRCC) work has been proposed. Several technical solutions have been developed to reduce the programming effort to enable the HRCC process. For example, the interactive digital twin for different construction tasks can be automatically generated to reduce the programming effort involved in manually creating the digital twin. BIM and human workers collaboratively provide high-level task objectives (e.g., places to pick up and install a workpiece), which significantly reduces the effort to pre-program robot movements. However, several sequential robot motions are needed to perform a construction task. Take the pick-and-place task as an example, the robot needs to reach the workpiece with its end-effector, activate the gripper to pick it up, manipulate the workpiece to the target location, and then deactivate the gripper to release it.

Two methods have been used to indicate the motion sequence to the robot in the previous chapters. First, the motion sequence is programmed onto the robot as a series of default parameterized movements that take the target poses of each movement as inputs to install a workpiece. For example, in the drywall installation case study of Chapter 3, a sequence of motions has been programmed for the robot to install each panel, as shown in Table 4.1. However, there are a variety of motion sequences for different types of construction work. Even for the same type

of construction task, the motion sequence to perform the work might need adjustment according to different task situations. Thus, it is unrealistic for robot engineers without domain knowledge about construction to pre-program all different motion sequences for a construction robot. Moreover, with pre-programming, the robot always follows the previously determined motion sequence until all the motions in the sequence are completed. As a result, the robot cannot adapt to failures or uncertainties at the intermediate steps (e.g., failure grasping the workpiece).

Table 4.1 Programmed Motion Sequence in Section 3.4

Order	Motion
1	Reach (material pre-pick)
2	Reach (material pick)
3	Grasp ()
4	Reach (material post-pick)
5	Reach (target pre-place)
6	Reach (target place)
7	Release ()
8	Pullback ()

Chapter 2 uses a combination of human instructions and programming to indicate robot sequential motions. Elemental motions of robots are encapsulated into “Pick”, “Place”, “Hold”, and “Release” high-level operations that the human co-worker can select from the interactive virtual billboard during the construction process. However, it takes significant human effort to provide step-by-step instructions to the robot. Most importantly, it is observed that there is only a limited number of elemental robot motions that are being reused and sequenced for different types of construction tasks. Therefore, an approach that allows robots to sequentially determine the next-step motions by themselves with minimal human instructions is necessary to improve HRCC efficiency.

With this vision, this chapter presents an approach for robots to automatically sequence their motions by interactively requesting and learning from human demonstrations, referred to as interactive LfD. More specifically, it focuses on construction assembly tasks. However, it can also

be applied to other construction activities with minor modifications. A Scene Distance Matrix (SDM) based on multi-level scene state representation matrices has been proposed. Robots build probabilistic mappings from SDM to modular skill primitives in their knowledge base from human demonstrations in the digital twin system. The learned knowledge is then used for automatic motion sequencing for different types of construction tasks. Based on the proposed approach, a delivery framework for construction robots is presented.

The proposed method is demonstrated through a case study that contains three types of construction tasks, exterior wall sheathing, drywall installation, and timber frame construction. A shelf construction task is used to verify the proposed system, which integrates LfD module with the digital twin system developed with ROS and Unity. With the proposed approach, construction robots have the capability to learn new construction tasks from construction workers and automatically determine their next-step motion based on the current construction progress. A construction robot delivery framework has been proposed and discussed. The proposed construction robot delivery framework, along with the robot task learning and sequential motion planning capabilities, has the potential to improve the usage and cost efficiency of construction robots, and thus improve the acceptance and deployment of HRCC in the construction industry. The overall objective of this chapter is to further improve robot planning capabilities thereby reducing human workers' planning effort (Figure 4.1)

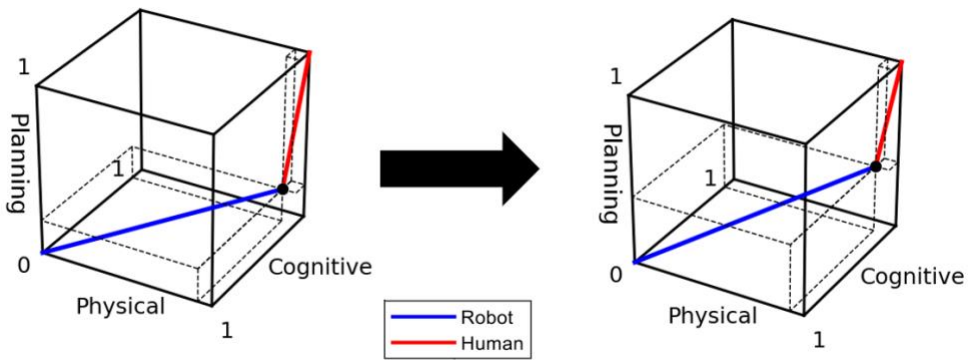


Figure 4.1 System Impact on Human-Robot Effort Distribution (Chapter 4)

4.2 Background

4.2.1 Robot Learning from Demonstration

LfD allows robots to learn skills from the motions or decisions they observe from human demonstrations (Zhu and Hu 2018). It does not require robot programming expertise from the teachers or demonstrators, and thus allows experts with greater domain knowledge, such as construction workers, to teach the robot to perform work in the specific domain (e.g., construction) (Argall et al. 2009). A typical LfD problem usually involves mapping from the environment state to the robot action. Since the environment state usually cannot be fully observed by the robot, most LfD studies also involve an observed state as the connection between the environment state and the robot's action. In this situation, both the mapping from the environment state to the observed state and the mapping from the observed state to the robot action need to be considered in the framework design.

LfD is used by robots to learn skills on different levels. Low-level LfD focuses on learning motor policy for manipulation, referred to as “trajectory encoding”, such as learning peg-in-hole or slide-in-groove (Billard and Calinon 2008; Kramberger et al. 2017; Peternel et al. 2018). Compared to computational-based motion planners, motor control policy learned from

demonstrations has higher adaptability and can solve more complex cases. For example, Liang et al. (2020) taught a construction robot to manipulate ceiling tiles through the ceiling grid and install them at desired locations in the grid. Since the ceiling tile has a bigger surface than the opening of the grid and the manipulation space above the grid is highly restricted by the floor above and the surrounding utilities (e.g., mechanical, electrical, and plumbing systems), the manipulation plan cannot be directly resolved by ordinary motion planners, but it can be learned from human demonstrations. On the other hand, high-level LfD learns the "symbolic encoding" about how to organize predefined motion primitives into a sequence, such as sorting colored blocks into different bowls (Billard and Calinon 2008; Cubek et al. 2015). With the goal of enabling automatic sequencing of high-level motions, this chapter focus on high-level LfD.

There are several demonstration methods for high-level LfD. Humans can demonstrate motion sequences using the kinesthetic approach by physically moving the robot's passive joints (Figueroa et al. 2016). While allowing demonstrations to be identically transferred to the robot, this approach is restricted by the scale of the task and safety considerations. Thus, teleoperation is used instead such that the human movements are captured by vision-based systems (Kulić et al. 2012), motion capture systems (Dang and Allen 2010; Lioutikov et al. 2015), hand-coded controllers (Niekum et al. 2012), VR (Zhang et al. 2021), or a combination of several sources (Pardowitz et al. 2007), and used as demonstrations. Nevertheless, these demonstrations are continuous movements, which need to be segmented into motion primitives for the robot to learn the sequencing of primitives (Ravichandar et al. 2020). Some demonstration approaches can directly use motion primitives to achieve high-level task objectives thereby avoiding errors from the segmentation process. The most widely-used method is to use the graphical user interface (GUI) (French et al. 2019; Mohseni-Kabir et al. 2015). Additionally, language-based demonstrations can

guide the robot to implement certain motion primitives in specific sequences with language instructions (Scheutz et al. 2017; She et al. 2014).

Most of these studies evaluated their proposed approach at a relatively small (e.g., desk-top) scale with a limited number of relatively simple motions involved (e.g., push, pick, place). Since this chapter aims at high-level LfD that can generalize across different types of construction assembly tasks, two characteristics of construction tasks need to be considered. First, construction tasks involve large-size robots and objects that prohibit demonstrations through directly operating the robot, such as motion capture or kinesthetic approach. In addition, it is also challenging to effectively map the environment state to the observed state. While most approaches use vision-based systems to model scene states (Dang and Allen 2010; She et al. 2014), it is hard to identify object relationships and detect scene changes in a construction environment directly from camera images due to the large space and complexity of the environment. Second, construction tasks include various complex operations. It is difficult to accurately segment construction operations from a movement sequence detected by motion sensors or cameras. Therefore, this chapter builds the LfD module upon the proposed BIM-integrated I2PL-DT system. The digital twin allows real-time monitoring of the robotic construction workspace that provides effective observations of the construction workspace scene state. Additionally, it provides an immersive VR interface that allows the human co-workers to clearly observe the scene states and supports intuitive human demonstrations by directly selecting primitives from the interface.

4.2.2 Robot's Motion Sequence Determination

In the previous chapters, the robot motion sequence is determined by programming or a combination of programming and step-by-step instructions from human co-workers. Although sensing and machine learning have greatly improved robot intelligence in recent years, direct

programming is still the most common method to determine the motion sequence for construction robotic manipulators (Dawod and Hanna 2019; Ding et al. 2020; Zhu et al. 2021).

Some studies have used automatic planners such as the Stanford Research Institute Problem Solver (STRIPS) to determine robot motion sequence (Fikes and Nilsson 1971; Zeng et al. 2018). Pre-conditions and post-conditions of each motion are defined. Then, STRIPS can generate the motion sequence with the provided initial scene state and target scene state. However, it is very difficult to define construction motions while considering all the pre-conditions and post-conditions correctly and be consistent with other motions and scene states so that the planner can solve the problem. In addition, with the automatic planners, the robot follows the planned motion sequence during execution but cannot adapt its plan according to the real-time work status (e.g., re-execute the gripping motion if the object has not been successfully picked up).

Hierarchical task modeling organizes different levels of task primitives of a complex task into a hierarchical structure and uses the symbolic representation to identify task primitive relations (Hayes and Scassellati 2016; Sohn et al. 2020). The task primitives and their relationships in the hierarchical model greatly reduce search space and thus improve robot planning efficiency (Hayes and Scassellati 2016). Therefore, it has been used in combination with LfD to represent the knowledge the robot learned from human demonstrations. For example, Mohseni-Kabir et al. (2015) integrated computerized situated dialogue into a GUI to guide users to build the hierarchical task networks for a robotic task with provided primitive tasks. However, the robot is programmed to follow a fixed motion sequence, which is vulnerable to interruptions during the task implementation. French et al. (2019) proposed an approach for the robot to learn a behavior tree from human demonstrations but the learned model lacked generalization to other scene spaces. Hierarchical task networks and And-Or-Graph have been used to represent knowledge achieved

from human demonstrations to facilitate robot planning on the sequencing of motion primitives (Chen et al. 2020; Zhang et al. 2020). These approaches mainly focus on tasks with multiple sequential solutions and emphasize the combination of different demonstrated sequences to provide the most preferred option. As a result, multiple demonstrations are needed for the same type of task for the robot to build the hierarchical model.

Reinforcement learning is also a popular approach for robot sequential decisions in its motions (Argall et al. 2009). It allows the robot to conduct exploration by itself and provides feedback to the robot with rewards to guide the robot to find the solution (Van Otterlo and Wiering 2012). Chitnis et al. (2020) and Nasiriany et al. (2022) used two separate policy models to decide which motion primitive to select and to decide the parameters of the selected motion with reinforcement learning. The policy is trained in simulation and is then transferred to the physical world for execution. In order to further reduce the robot exploration load and accelerate the reinforcement learning process, some studies take advantage of the robot's prior experience (Pertsch et al. 2020; Singh et al. 2020). Demonstration is also a popular approach to improving reinforcement learning (Argall et al. 2009). Higher rewards are given to demonstrated actions or observed transitions between actions (Kent et al. 2019; Rosenstein and Barto 2004). Nevertheless, it is difficult to define a reward function to effectively guide the robot to find solutions, especially in the context of construction when there are several types of motions and the search space is large. The reward function is usually task-specific and cannot be generalized to a different task (Xu et al. 2018; Zhu and Hu 2018).

These approaches have some limitations in generating motion sequences for construction robots. First, construction work is usually at a large scale with heavy construction components being manipulated. Therefore, it takes considerable effort and consumes significant resources for

the human co-workers to demonstrate one type of task repeatedly or letting the robot explore different task implementation options. Second, construction involves many different types of tasks. Therefore, it is critical that the previously learned sequential motion planning ability can be transferred to other tasks with minimal additional demonstrations required. Although some studies transferred previously learned skills to different tasks, the differences between the tasks are limited, such as picking up a tray versus picking up a stud (Chitnis et al. 2020; Nasiriany et al. 2022).

Third, construction work involves a variety of operations that are extremely challenging for the robot to solve through exploration or automatic planners. Fourth, construction assembly tasks are quasi-repetitive, each workpiece needs to be picked up from and installed to different locations. It means the motions are parameterized (i.e., need to take input from the environment), which increases the difficulties to explore solutions through reinforcement learning. Lastly, construction has a lot of uncertainties. Instead of consistently following a previously generated or learned fixed sequence, the robot needs to adapt its plan according to the work status so as to robustly perform construction tasks. Therefore, the system should also allow human supervision (e.g., preview robot plan before execution) and intervention (e.g., change robot plan) to ensure construction safety and quality. To summarize, a system for construction robots to develop motion sequences that is robust to uncertainties, generalizable across different types of construction assembly tasks, supports parameterized motion primitives, and only requires a minimal number of prior exploration trials or human demonstrations is necessary for efficiency in HRCC.

4.3 Technical Approach

4.3.1 Construction Robotics Skill Primitives

Motion primitives, which are defined as “modular and re-usable robot movement generators”, have been used in previous studies for robot programming and learning (Kulić et al.

2012; Paraschos et al. 2013; Strudel et al. 2020). This study expands the concept of motion primitives into construction robotics skill primitives. Skill primitives are previously defined, programmed, or learned skills the robot can directly utilize to perform tasks. There are three types of skill primitives.

The first type is motion primitives, which are the basic operation skills to perform construction work. The idea of modularization of construction operations has been explored by previous researchers. Warszawski and Sangrey (1985) identified ten basic activities (i.e., placing, connecting, attaching, finishing, coating, concrete building, inlaying, covering, and jointing) for robotics building construction. Everett (1991) developed a taxonomy for construction automation and robotics covering eleven "Basic Tasks" that can be performed by both humans and machines (i.e., connect, cover, dig, inspect, measure, place, plan, position, spray, spread, finish), and Feng (2015) built upon this and proposed a construction basic task automation methodology by adding information needed for each type of basic task. Nevertheless, the elements proposed in these studies are not detailed enough to program robot motions. For example, one of the basic tasks proposed by Everett (1991) is "connect", defined as "join or fasten together", such as "nail". However, to program a robot to nail, we need to first let the robot reach the desired position, then activate the end-effector for driving a nail. This process contains a sequence of two motions that can be modularized and reused in other tasks. Even though Everett (1991) suggests that basic tasks can be further broken down into elemental motions, no such example for construction machines or robots is given. Therefore, a collection of basic construction motions that can be modularized and reused by the robot to the broadest extent is needed.

From the perspective of facilitating robot learning and programming, a dynamic system that adapts "Basic Tasks" and "Elemental Motions" is used. Some "Basic Task" such as "place"

is broken down into reusable modules of "reach", "grasp" and "release" that can be reused in many different construction activities. However, for "Basic Task" like "spray", even though it involves two elemental motions of moving end-effector and spraying, it is usually programmed as one robot motion as these two motions occur at the same time. Thus, it is considered as one motion primitive. The motion primitive collection is dynamic and subject to change based on the robot and task properties. For example, manipulating a ceiling tile through the grid is a "reach" motion but cannot be directly solved by some robot motion planners. In this case, human workers can teach the robot the manipulation process through low-level LfD (Liang et al. 2021a), and it is treated as a separated motion primitive that can be reused as a module.

While motion primitives are closely related to construction activities, the other two types of skill primitives are general robotic skills. One of them is sensing primitives. These are skills for the robot to understand the world through seeing, hearing, touching, etc. Examples of sensing primitives are environment mapping, object localization, force detection, and speech recognition. The other type is reasoning primitives that function as the brain of the robot. Some of them are embedded as a system function module of the robot, such as machine learning, motion planning, and collision avoidance. Others are encapsulated modular computation and processing units that can carry out certain functions which have substantial effects on the system (e.g., $Start$: perform all system internal preparation work to start an operation). All three types of primitives are needed for robots to successfully perform construction tasks.

Previous studies do not have a unified taxonomy to describe construction work and cannot meet the needs for modular programming and teaching of construction robots (Everett 1991; Kisi et al. 2017; Warszawski and Sangrey 1985). Therefore, in order to avoid ambiguity, a hierarchical taxonomy—activity, task, operation, and elemental motions—adapted from previous studies is

used to describe robotic construction assembly bottom-up in this chapter. Elemental motions are the basic movements that the robot can be programmed to reuse as modules (e.g., reaching with the end-effector, opening the gripper, cutting along a path). An operation is made up of a series of elemental motions and other operations (e.g., the cutting operation needs the robot to reach the start point, perform the cutting motion, and pull back its end-effector from the workpiece). A construction task consists of several operations and elemental motions that result in a completed piece of work, such as the installation of a drywall panel. Activities are at the highest level of this taxonomy, which are subdivisions of a construction project that are usually used as the basic units of typical construction scheduling problems, such as drywall installation (Harris 1978). An example of hierarchical taxonomy for modular robotic programming is shown in Figure 4.2. This chapter mainly focuses on sequencing elemental motions, including the ones that fall under operations, to perform a construction task. As a result, by constantly querying the next task target from BIM, robots can sequence elemental motions to perform a construction activity with supervision from their human co-workers.

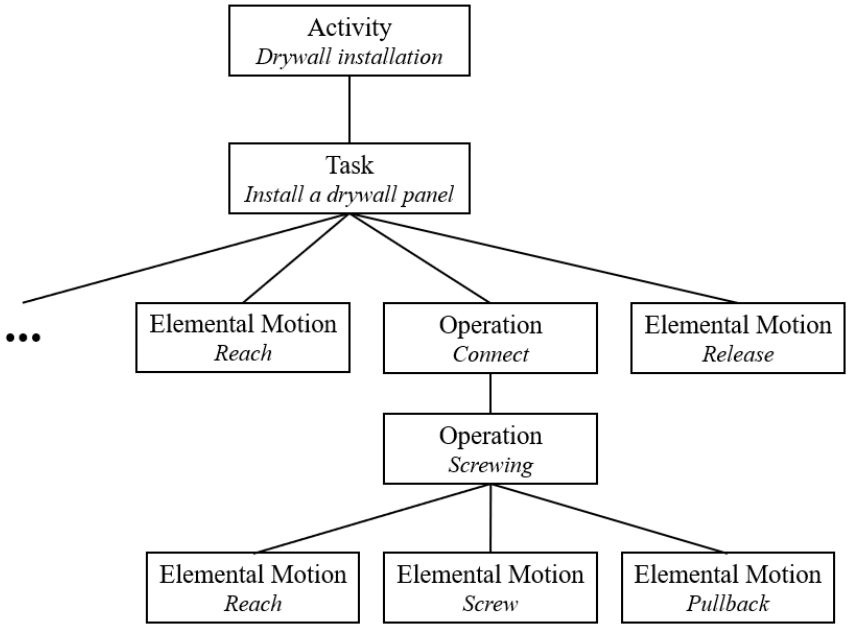


Figure 4.2 Example of Proposed Hierarchical Taxonomy

4.3.2 System Overview

Based on the concept of modular skill primitives, a commercial construction robot delivery framework is conceived. Construction robots are programmed with motion, sensing, and reasoning primitives by the robot engineers, and equipped with the proposed digital twin system with the LfD module. After a robot is transported to a construction site, construction workers teach the robot how to sequence these primitives to perform different construction tasks. The benefits of this delivery framework are two folds. First, the programming work is performed by robot engineers who have higher expertise in robot programming, and the high-level instructions for the robot to use these skill primitives to perform construction work are given by construction workers who have more domain knowledge in construction. It significantly improves the quality and rationality by reducing the knowledge gap between the two domains. Second, construction robots can learn different types of new construction tasks from workers. Compared to single-purpose construction robots that perform one type of task and stay idle for most time of a construction project, it allows robots to transfer to various types of jobs at different construction stages, making it more cost-efficient. In this chapter, we present an efficient method for robot engineers to prepare the digital twin system and skill primitives for interactive LfD for construction assembly tasks.

It should be noted that construction assembly tasks usually involve several tools and coordination steps, and thus to be performed by a robotic automation system that includes several robots or equipment. In this chapter, it is assumed that different operations or elemental motions have already been assigned to different robots or automation devices like Computer Numerical Control (CNC) machines, When the system decides to take a certain motion primitive, the corresponding device that is responsible for the motion will be used. To make it concise and clear,

in the remaining parts of this paper, the term “robot” means the robotic automation system that includes several possible devices.

An overview of the system is shown in Figure 4.3, which is built upon the BIM-driven HRCC system proposed in Chapter 3. The robot and its human partners interact through the digital twin system while BIM provides information to both of them to drive the construction process and save construction data, which is discussed in detail in Chapter 3. An additional interactive LfD module is integrated into the digital twin system, which is the focus of this chapter. In interactive LfD, robots learn by actively requesting demonstrations and processing them into applicable knowledge (Chernova and Veloso 2010). The interaction is enabled by the digital twin system the LfD module is embedded in, which supports bi-directional communication with the human co-workers, controls the robot movement, and extracts related information from the BIM. The mapping from the environment state to the observed scene state is also enabled by the digital twin, which tracks the object positions and has functions to update object relationships according to the demonstration. When the robot is delivered, the motion primitives are saved as reusable modules the human co-worker can select from the GUI of the digital twin system. Sensing and some reasoning primitives are embedded into the digital twin system and are initiated when certain conditions are met, while some reasoning primitives need to be invoked by the human co-worker as part of the demonstrated sequence.

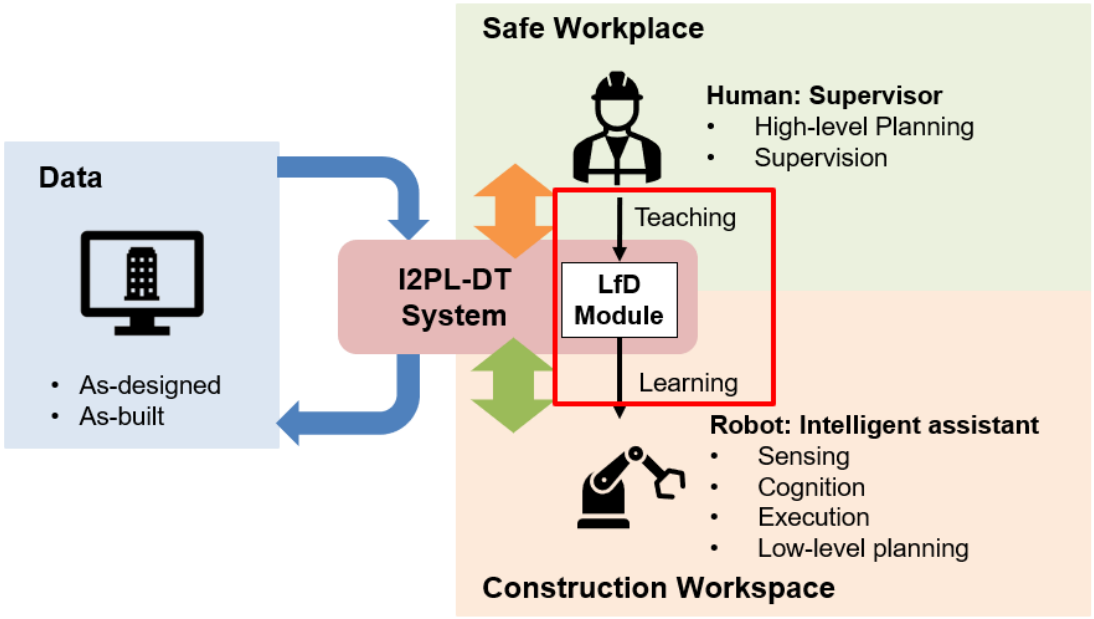


Figure 4.3 Chapter 4 System Overview

The outcome of LfD is that the robot can sequentially decide the motions step-by-step to perform a construction task. One target object is considered at each time following the construction sequence saved in BIM or human co-workers' decisions. However, the learned skills can be applied to other targets in the current and future construction tasks or shared with other robots. Motivated by this objective, an LfD workflow is proposed, as shown in Figure 4.4. The robot has a knowledge base that contains the probabilistic mapping from SDM to skill primitives, which is discussed in detail in Section 4.3.3. When the assembly of a target starts, the BIM proposes the next target in the construction sequence and highlights it in the GUI of the digital twin system. After the human co-worker confirms the target, an initial SDM is built accordingly. The robot searches in its knowledge base whether the SDM is known. If the SDM is known, the robot will use the mapping to find the corresponding skill primitive of the SDM. Otherwise, the robot will prompt its human co-worker to demonstrate the case. The human co-worker inspects the scene from the GUI and provides demonstrations to the robot by selecting the primitive the robot should

take. Then, the robot saves the mapping from the SDM to the demonstrated primitive into its knowledge base so that it can automatically make a decision without requesting a demonstration the next time the SDM is encountered.

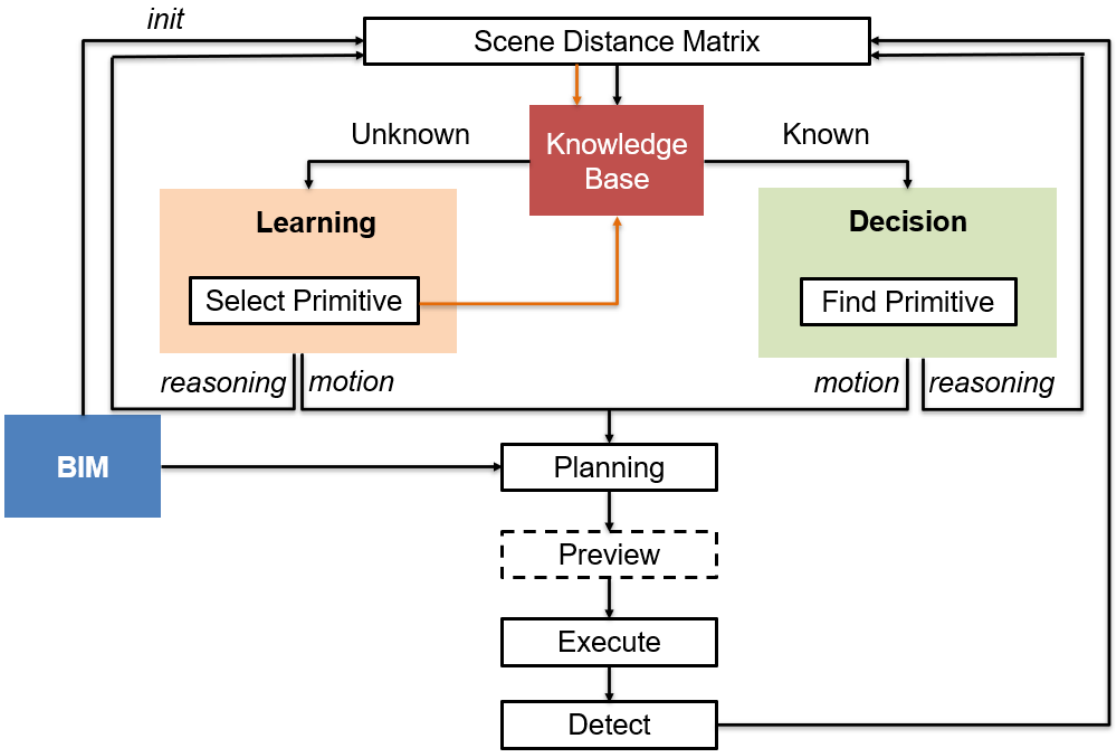


Figure 4.4 LfD Workflow

If a motion primitive is decided, the robot requests related parameters (e.g., target pose to reach) from the BIM for motion planning. Then, the human-robot work team follows the methods in Chapter 2 and Chapter 3 for planning, optional preview, and task execution and supervision. After task execution, the SDM is re-detected and updated. Then, the system will go through the process again until all the procedures related to this task are finished.

The robot or its human co-worker decides the next primitive at each step based on the latest (i.e., most current) SDM instead of developing a whole assembly sequence to follow. Therefore, the system is robust to interruptions during the construction process, such as the robot's failure in

picking up the workpiece. In this case, the SDM will not be updated and the robot will attempt to grasp the workpiece again.

4.3.3 Scene Distance Matrix

As shown in the LfD workflow in Figure 4.4, the robot learns mappings from SDM to skill primitives and saves them in the knowledge base. Therefore, SDM is the key to deciding robot motions. The proposed process to determine motion from the scene also follows the human decision-making process. For example, a human worker who wants to perform some work with a wrench needs to hold the wrench in their hand. However, the worker finds that the wrench is on the table. Therefore, the worker decides to pick up the wrench from the table. In this process, two scenes are considered by the human worker to make the decision. One is the target scene where the wrench is in the hand, the other is the current scene where the wrench is on the table. Therefore, SDM needs to take both the goal and the current scene states into consideration.

In addition, SDM needs to meet several requirements considering the characteristics of construction tasks. First, construction involves a wide variety of assembly tasks with different motion sequences. Therefore, the SDM needs to have the flexibility to represent different circumstances in construction assembly. Moreover, the robot needs human demonstration for each SDM to build the mapping relationship in its knowledge base. Too many different types of SDMs will significantly increase human workers' burden of demonstration. As a consequence, SDM needs to be general so that fewer demonstrations are required for different types of tasks. Additionally, SDM needs to be designed in a way that robots can automatically build it during construction, with the information from its sensors, BIM, and the digital twin system. It should be understandable to the robot and be computational-efficient to be saved by the robot in its knowledge base.

Based on these considerations, an SDM representation is proposed. A multi-level matrix is first used to reflect the scene state (Figure 4.5). Two multi-level matrices are needed at each decision step. One is used to reflect the current scene state, and the other one is used to reflect the goal state. The main-level matrix has four rows and columns, with each row and column representing the material (M), target place (T), robot (R), and connection (C). Values in the cells represent the corresponding state or relationship. Values in Cell (M, T), (R, M), and (R, T) represent the "At" relation, which is calculated using Eq. 4.1. For example, when the distance between the material and the target place is below a certain threshold (i.e., very close to zero), we can consider that material is at the target place so the value of Cell (M, T) is 1. Cell (M, T) represents the "On" relation, showing whether the material is being grabbed by the robot. Cell (M, M) and Cell (T, T) represent material and target processing respectively, indicating the percentage of the processing work that has been completed. Cell (C, C) represents whether a connection is needed. Lastly, after the robot places the workpiece and releases the gripper, it will typically pull its end-effector back for a certain distance instead of remaining in contact with the workpiece. Therefore, we use Cell (R, R) to represent whether the pullback motion has been performed. For the goal state matrix, Cell (R, R) is set to one. For the current state matrix, Cell (R, R) starts with 0 and changes to one after the pullback is conducted.

	M	T	R	C
M	Processing	At	On	
T		Processing		
R	At	At	Pullback	
C				Need

Figure 4.5 Multi-Level Scene State Matrix

$$At[X,Y] = \begin{cases} 1, & \text{distance (X, Y)} \leq \text{threshold} \\ 0, & \text{else} \end{cases} \quad (4.1)$$

SDM is generated by subtracting the current state matrix from the goal state matrix. It indicates what is left to reach the goal, which leads to the decision of the motion at the next step. If the SDM has been encountered by the robot, the robot will follow the mapping it previously learned to automatically select the corresponding skill primitive, generating the motion plan or updating the SDM. However, the human co-worker can choose to let the robot take another primitive after evaluating the plan. It has a low possibility but one SDM may correspond to multiple motion primitives. Therefore, probabilities of the correspondence are also saved in the mapping. The robot takes the primitive with the highest correspondence probability with the SDM by default. Each time the human co-worker approves the default primitive or makes a new selection, the probabilistic mapping is updated to provide better suggestions the next time.

In order to reduce the number of demonstrations required for different tasks, transition-level and second-level matrices are introduced. The motivation for introducing multiple levels emerges from the fact that many construction assembly tasks follow similar sequences but with differences only in a few steps. Take connection operations as an example, there are several different ways of connection in construction assembly, but they may occur at the same stage in the assembly sequences. In one case, the robot might need to pick and place a workpiece, nail it, and release it. In another case, the robot might need to pick and place a workpiece, screw it, and release it. It is a redundant human effort to require them to demonstrate the whole assembly sequence to the robot. The solution is to consider connection as a high-level operation. Only the percentage is has been completed appears on the main-level matrix. If such an operation is needed, the system will enter the transition level that further leads to the generation of second-level scene state representation matrices. At each level, SDM based on the current and goal state is calculated.

Figure 4.6 shows the structures of current and goal state matrices on the transition level. The transition level decides the sequence of lower-level operations (e.g., nailing, screwing, drilling) needed for a high-level operation (e.g., connection). Each row represents a type of operation required by certain high-level operations in the main-level matrix. For example, if the connection requires two types of operations, nailing and gluing, the transition-level matrices for connection will be 2 by 2 matrices. Each type of operation is assigned a unique integer identification number. The number of the nailing and drilling operations that are required and finished is recorded respectively in the current and goal state matrices. SDM at the transition level is calculated by subtracting the two state matrices. The number of non-null rows in the SDM indicates how many types of operations are left.

ID	Num
$\begin{cases} 0, & not\ all\ finished \\ id1, & all\ finished \end{cases}$	Numbers finished
$\begin{cases} 0, & not\ all\ finished \\ id2, & all\ finished \end{cases}$	Numbers finished
•••	•••

(a) Current State

ID	Num
id1	Numbers needed
id2	Numbers needed
•••	•••

(b) Goal State

Figure 4.6 Transition-Level Scene State Matrix Structure

However, there is one issue with transition-level SDM. Figure 4.7 shows two transition-level SDM examples. Let us place it in the context of material processing for better understanding. Task A requires 2 cutting operations (id: 1) and 6 drilling operations (id: 2) while Task B requires 2 cutting operations and 4 drilling operations. These material processing operations are very

similar. However, these are two different SDMs to the robot, which means knowledge to select primitive based on Task A SDM cannot be transferred to Task B. This will result in unnecessary human demonstrations. Therefore, fuzzy search is used to match transition-level SDMs. If there is an SDM with the same number of rows that contains the same ID, the mapping of that SDM will be used to decide the primitive for the transition-level SDM of the new task even if the numbers of operations (right column) are different. As a result, if human workers taught the robot to first $start\_cutting()$ in Task A, the robot will automatically select $start\_cutting()$ when performing Task B. After cutting is finished, there is only one type of operation left so the robot can start drilling. If multiple SDMs containing the same types of operation are included in the knowledge base, the robot will select the one with the nearest distance D calculated with Eq. 4.1, where $Num(A_i)$ is the number of operations of ID $= i$.

ID	Num
1	2
2	6

Task A

ID	Num
1	2
2	4

Task B

Figure 4.7 Examples of Transition-Level SDM

$$D = \sum_i |Num(A_i) - Num(B_i)| \tag{4.1}$$

An example of a second-level matrix structure is shown in Figure 4.8, which decides the elemental motions (e.g., reach, nail, pullback) to perform a low-level operation (e.g., nailing). Cell (T, T) shows whether the motion has been completed. It is represented by the assigned motion id in the corresponding category (e.g., in connection: nail is 1 and screw is 2). Cell (R, T) represents

whether the robot is at the start point of the specified motion being ready for the work. Cell (R, R) is the same as the main level, indicating whether pullback has been performed. The probabilistic mappings for matrices on different levels are saved separately in the robot knowledge base.

In the proposed main-level matrix, Cell (C, C), (M, M), and (T, T) represents high-level operations of connection, material processing, and target processing respectively. The system decision-making process to find out the elemental motion sequences for these high-level operations is shown in Figure 4.9. Reasoning primitives are used to enable the transitions between different levels. When the robot or the human workers decide to start an operation, they can select the corresponding reasoning primitives from the knowledge base or GUI of the digital twin system to trigger the next level. The reasoning primitives are responsible for retrieving related information from the BIM (i.e., the number and type of operations) and generating the corresponding SDM at the lower level or returning to the upper level.

	T	R
T	Finish (motion id)	
R	At (start point)	Pullback

Figure 4.8 General Template of Second-Layer Matrix

Six default transitions are embedded into the system and will be triggered when certain conditions are met. First, when the second-level or transition-level SDM is null, indicating the low-level or high-level operations are all finished, the system will return to the upper level (i.e., transition or main level). Second, when there is only one row in the transition-level SDM, indicating only one type of operation is needed, the system will directly start the operation. Third,

no matter how many operations of the same type (i.e., 4 nails) are required in the transition-level SDM, only the 2-dimensional matrix for one operation (i.e., 1 nail) will be generated each time at the second level. After one operation is finished, the system returns to the transition level and checks whether all operations of the same type have been completed. If not, the system by default will continue with the same type of operation until all operations of this type are completed. The motion sequencing skills learned from previous operations can be used to automatically decide the motion sequence for subsequent operations of the same type. This approach can avoid the repetitive demonstration effort when several operations of the same type are needed to assemble a workpiece. Lastly, the start and finish transitions respectively create the initial main-level SDM and close up the SDM to proceed to the next target.

The matrices and the SDM, including how they are generated from the digital twin system and the transition between different levels, are embedded into the digital twin system that can directly be used by the construction workers with minor training. Customized second-level matrices might be needed to support specific operations. When programming the motion primitives, three aspects need to be considered: (1) which robot or machine should be used to perform the work; (2) what BIM and sensing information is needed as the input to perform the motion, and how to request the information; and (3) how to control the robot to perform the motion with the provided tools and information. With the encapsulated skill primitives and LfD module, human workers can teach the robot the motion sequencing skills to perform different construction tasks through demonstration in the digital twin system. After initial demonstrations, the robot has the capability to determine the motion sequence by itself under human supervision without requiring programming for each type of task or step-by-step human instructions.

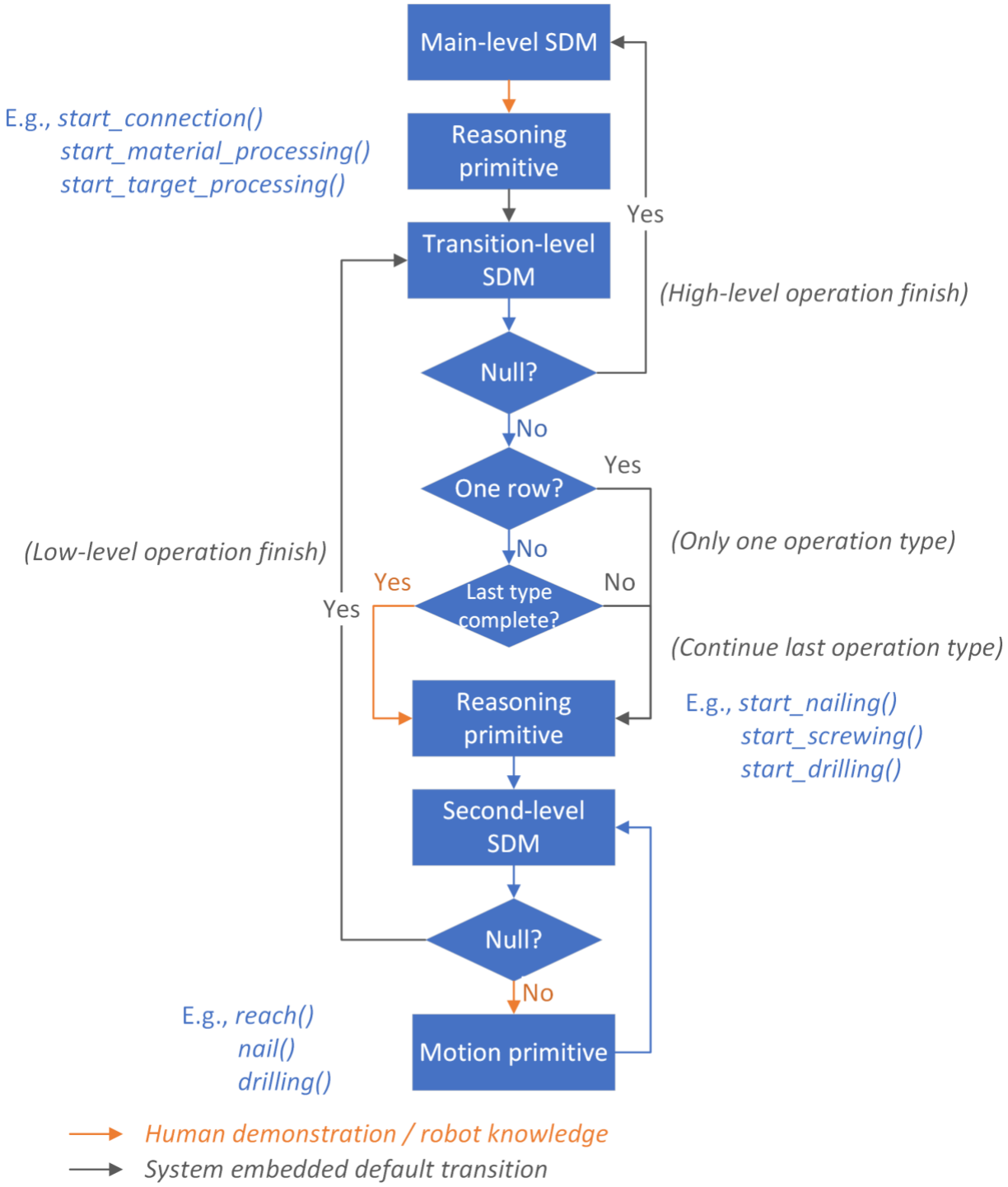


Figure 4.9 Decision Process for Motion Sequences of High-Level Operations

4.4 Case Study

A case study with three construction assembly scenarios, including exterior wall sheathing, drywall installation, and timber frame construction, is demonstrated to verify the proposed interactive LfD approach. The number of human demonstrations required and robot automatic decisions during the three assembly processes are analyzed. It should be noted that after integrating the LfD module, the digital twin system still follows the same workflow and has the capabilities discussed in Chapter 2 and Chapter 3, such as plan evaluation and deviation resolution. However, since this chapter focuses on the additional LfD module, only the processes and information related to the LfD and motion sequencing processes are discussed in this case study.

The robotic system in the case study contains two robots and a CNC machine. One robot is responsible for workpiece manipulation, and the other robot performs motions to assist connection, such as nailing, screwing, and drilling (with different tools installed on the end-effector). The CNC machine can cut the workpiece into different shapes following the predefined cutting plane. The skill primitives provided to enable the assembly motion sequence are shown in Table 4.2. Robots are equipped with MoveIt and the OMPL motion planner for them to reach specified poses without collision (Chitta et al. 2012; Sucan et al. 2012). They also have a predefined vector for it to pull back its end-effector. Additionally, it is assumed that the robot has already detected the materials' locations.

Detailed construction information, such as the sequential pose to pick up and place the target, connection and drilling positions, and cutting planes are provided by the BIM. The information can be generated by computational design (Adel 2020). For the verification purpose of this dissertation, the finishing processes (e.g., taping, smoothing, painting) are not included but they are within the system's capability. It is also assumed that all robot motions are executed

successfully and changed the scene state. However, in actual construction work, the scene state needs to be detected at each step to understand whether the motion has been executed successfully.

Table 4.2 List of Skill Primitives

Motion Primitives	Reasoning Primitives
• Reach (P) • Grasp • Release • Pullback • Nail (N) • Screw (S) • Cut (C) • Drill (D)	• start_connection () • start_material_processing () • start_nailing () • start_screwing () • start_cutting () • start_drilling () • return_trans_level() • return_main_level()

4.4.1 Exterior Wall Sheathing

The first scenario is exterior wall sheathing. At this stage, the robots are newly delivered and do not have any knowledge about the construction assembly sequence. When the task starts, the next sheathing target in the construction sequence is retrieved from the BIM. The material gripping poses sequence $G(g_0, g_1)$, target placing pose sequence $P(p_0, p_1, p_2)$ and nailing points $N(n_0, n_1, n_2, n_3)$ are saved in the BIM (Figure 4.10). A sequence of targeted poses is included in material gripping and target placing, which is consistent with robot movement convention and ensures the wall panel firmly stays in contact with its neighbor. The goal of the task is to have the wall panel placed at the target and secured by nails to the frame, which is represented by the goal state matrix in Table 4.3.

Initial state: When the task starts, the current scene state matrix is all zeros. The SDM is calculated by subtracting the current state from the goal state (Table 4.4).

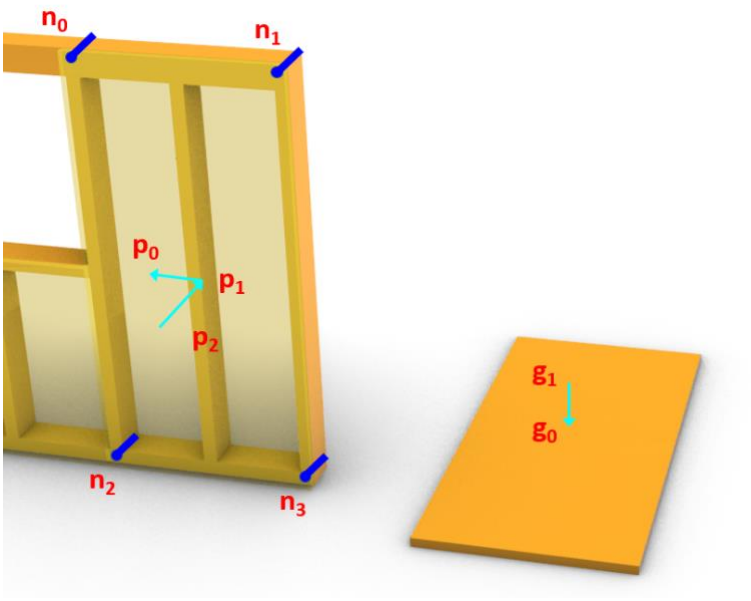


Figure 4.10 Illustration of BIM information for Robot Planning

Table 4.3 Scenario 1 - Goal State

	Goal State			
	M	T	R	C
M	0	1	0	0
T	0	0	0	0
R	0	0	1	0
C	0	0	0	1

Table 4.4 Scenario 1 - Initial State

Step 0 (Robot Default): $Start()$									
Current State					**SDM (1-0)**				
	M	T	R	C		M	T	R	C
M	0	0	0	0	M	0	1	0	0
T	0	0	0	0	T	0	0	0	0
R	0	0	0	0	R	0	0	1	0
C	0	0	0	0	C	0	0	0	1

Step 1: SDM 1-0 is unknown to the robot. Therefore, a human demonstration is requested. The human co-worker selects the $Reach()$ motion primitive from the primitive list in the GUI and selects the material as the target for the $Reach()$ motion upon system prompt. Next, the robot retrieves material gripping information G from the BIM, develops the motion plan, and successfully executes $Reach(G)$ upon human worker's confirmation. Mapping from SDM 1-0 in the initial state to the $Reach(G)$ motion primitive is saved in the robot knowledge base as main-level mapping (M1). After robot execution, the robot end-effector is at the material location. Therefore, Cell (R, M) updates to 1, and a new SDM is generated (Table 4.5).

Table 4.5 Scenario 1 - Step 1

Step 1 (Human Demonstration – M1): $Reach(G)$									
Current State					**SDM (1-1)**				
	M	T	R	C		M	T	R	C
M	0	0	0	0	M	0	1	0	0
T	0	0	0	0	T	0	0	0	0
R	**1**	0	0	0	R	**-1**	0	1	0
C	0	0	0	0	C	0	0	0	1

Step 2: The human co-worker decides to let the robot "Grasp" to pick up the object. After grasping, Cell (M, R) changes to 1 because the material is on the robot (Table 4.6). Mapping from SDM 1-1 to the $Grasp()$ motion is saved in the robot knowledge base as main-level mapping (M2).

Table 4.6 Scenario 1 - Step 2

Step 2 (Human Demonstration – M2): $Grasp()$									
Current State					**SDM (1-2)**				
	M	T	R	C		M	T	R	C
M	0	0	**1**	0	M	0	1	**-1**	0
T	0	0	0	0	T	0	0	0	0
R	1	0	0	0	R	-1	0	1	0
C	0	0	0	0	C	0	0	0	1

Step 3: The human co-worker decides to let the robot reach the target. The robot follows the sequential placing targeted poses $P(p_0, p_1, p_2)$ from BIM to execute $Reach(P)$. Both the material and the robot are at their corresponding target now so both Cell (M, T) and Cell (R, T) update to 1 (Table 4.7).

Table 4.7 Scenario 1 - Step 3

Step 3 (Human Demonstration – M3): $Reach(P)$									
Current State					**SDM (1-3)**				
	M	T	R	C		M	T	R	C
M	0	**1**	1	0	M	0	**0**	-1	0
T	0	0	0	0	T	0	0	0	0
R	1	**1**	0	0	R	-1	**-1**	1	0
C	0	0	0	0	C	0	0	0	1

Step 4: At this step, the human co-worker decides to start the connection operation by selecting the $start\_connection()$ reasoning primitive. The robot then executes the reasoning primitive by retrieving connection information from the BIM and generating the transition-level SDM for the connection operation. The SDM contains one row, indicating that 4 connection operations with ID=1 (i.e., nailing) are needed (Table 4.8). Following the same procedure of

motion primitives, mapping from SDM 1-3 to the corresponding reasoning primitive is also saved in the robot knowledge base as M4.

Table 4.8 Scenario 1 - Step 4

Step 4 (Human Demonstration – M4): $start\_connection()$			
Current State		**SDM (1-4)**	
ID	Num	ID	Num
0	0	1	4

Step 5: According to the embedded default function, when there is only one type of operation needed, the robot will directly select the corresponding reasoning primitive to start the operation. In the current SDM, nailing is the only type of connection operation needed, thus the robot directly takes the $start\_nailing()$ primitive and generates the current state matrix and SDM on the second level (Table 4.9).

Table 4.9 Scenario 1 - Step 5

Step 5 (Robot Default): one operation type - $start\_nailing()$					
Current State			**SDM (1-5)**		
	T	R		T	R
T	0	0	T	1	0
R	0	0	R	0	1

Steps 6-10: The robot is guided to conduct the first nailing operation by reaching the first nailing position n_0 to prepare for nailing, shoot the nailing gun, and pull back the end-effector after nailing (Table 4.10). The second-level SDMs (1-5, 1-6, 1-7) are updated accordingly with these steps. Mapping from the SDMs to corresponding primitives is saved in the robot knowledge base as connection mapping (C1, C2, C3). After the $Pullback()$ operation, SDM 1-8 is detected as a null matrix, which means one nailing operation is completed. Thus, the robot returns to the

transition level at Step 9 following the system default embedded transitions. Then, the robot follows the default setting to start another nailing operation at Step 10.

Table 4.10 Scenario 1 - Step 6-8

Step 6 (Human Demonstration – C1): $Reach(n_0)$

Current State			**SDM (1-6)**		
	T	R		T	R
T	0	0	T	1	0
R	**1**	0	R	**-1**	1

Step 7 (Human Demonstration – C2): $Nail()$

Current State			**SDM (1-7)**		
	T	R		T	R
T	**1**	0	T	**0**	0
R	1	0	R	-1	1

Step 8 (Human Demonstration – C3): $Pullback()$

Current State			**SDM (1-8)**		
	T	R		T	R
T	1	0	T	0	0
R	**0**	**1**	R	**0**	**0**

Step 9 (Robot Default):
second-level SDM null - $return\_trans\_level()$

Current State		**SDM (1-9)**	
ID	Num	ID	Num
0	**1**	1	**3**

Step 10 (Robot Default):
continue same type operation - $start\_nailing()$

Current State			**SDM (1-10)**		
	T	R		T	R
T	0	0	T	1	0
R	0	0	R	0	1

Steps 11 - 25: The robot uses the C1, C2, and C3 mappings learned from human demonstrations to repeat Steps 6 – 10 until the other three nails are all completed. After Step 24, all the nails are installed. The SDM at the transition level is a null matrix, which means all the connection operations are completed (Table 4.11). Therefore, the system returns to the main level at Step 25.

Table 4.11 Scenario 1 - Step 24-25

Step 24 (Robot Default):
one operation finish - $return\_trans\_level()$

Current State		**SDM (1-24)**	
ID	Num	ID	Num
1	**4**	**0**	**0**

Step 25 (Robot Default):
transition-level SDM null - $return\_main\_level()$

Current State					**SDM (1-25)**				
	M	T	R	C		M	T	R	C
M	0	1	1	0	M	0	0	-1	0
T	0	0	0	0	T	0	0	0	0
R	1	1	0	0	R	-1	-1	1	0
C	0	0	0	**1**	C	0	0	0	**0**

Steps 26-28: The robot releases the panel and pulls back its end effector according to human demonstration (Table 4.12). The demonstrations and corresponding SDMs are saved in the robot knowledge base (M5 and M6). After Step 27, the main-level SDM is null, which means the assembly sequence of the current workpiece is finished and the assembly of the next target in the construction sequence can start.

Table 4.12 Scenario 1 – Step 26 – 28

Step 26 (Human Demonstration – M5): $Release$ ()

Current State					**SDM (1-26)**				
	M	T	R	C		M	T	R	C
M	0	1	**0**	0	M	0	0	**0**	0
T	0	0	0	0	T	0	0	0	0
R	1	1	0	0	R	-1	-1	1	0
C	0	0	0	1	C	0	0	0	0

Step 27 (Human Demonstration – M6): $Pullback$ ()

Current State					**SDM (1-27)**				
	M	T	R	C		M	T	R	C
M	0	1	0	0	M	0	0	0	0
T	0	0	0	0	T	0	0	0	0
R	**0**	**0**	**1**	0	R	**0**	**0**	**0**	0
C	0	0	0	1	C	0	0	0	0

Step 28 (Robot Default): $Finish$ ()

4.4.2 Drywall Installation

The second scenario demonstrated in the case study is drywall installation. The drywall installation process is similar to exterior wall sheathing. However, instead of nailing the wall panel, drywall panels are screwed onto the wall frame. Since robots have already learned exterior wall sheathing, the skills they learned are transferred to the drywall installation task to automate the motion sequencing process. The BIM contains the material gripping target poses sequence $G(g_0, g_1)$, target placing poses sequence $P(p_0, p_1, p_2)$ and screw locations $S(s_0, s_1, s_2, s_3)$.

The initial state of drywall installation is the same as the one of exterior wall sheathing. The robot follows the mapping relation from its knowledge base to perform the first four steps, which are also the same as the wall sheathing (Table 4.13). Since screwing operations are needed for connection, an unencountered transition-level SDM with a different ID is generated after Step

4. Consequently, since only one type of connection (i.e., screwing) is needed, the system starts the screwing operation directly and generates the second-level SDM (Table 4.14).

Table 4.13 Scenario 2 - Step 1-4

Step 1 (Robot Learned – M1): $Reach(G)$	
Step 2 (Robot Learned – M2): $Grasp()$	
Step 3 (Robot Learned – M3): $Reach(P)$	
Step 4 (Robot Learned – M4): $start\_connection()$	
Current State	**SDM (2-4)**
ID Num	ID Num
0 0	2 4

Table 4.14 Scenario 2 - Step 5

Step 5 (Robot Default): one operation type - $start\_screwing()$					
Current State			**SDM (2-5)**		
	T	R		T	R
T	0	0	T	2	0
R	0	0	R	0	1

SDM 2-5 is new to the robots, thus the human co-worker demonstrates Step 6 and Step 7 to guide the robots. The new demonstrations are saved as connection mappings C4 and C5 in the robot knowledge base (Table 4.15). After Step 7, SDM 2-7 has already been known as C3 mapping by the robot so the robot can automatically make the decisions in the subsequent steps until all four screws are installed and the system returns to the main level. Lastly, the robots follow previously learned mapping M5 and M6 to release and pull back the end-effector and complete the task.

Table 4.15 Scenario 2 - Step 6-7

Step 6 (Human Demonstration – C4): $Reach(s_0)$						
Current State				**SDM (2-6)**		
	T	R			T	R
T	0	0		T	2	0
R	**1**	0		R	**-1**	1
Step 7 (Human Demonstration – C5): $Screw()$						
Current State				**SDM (2-7)**		
	T	R			T	R
T	**2**	0		T	**0**	0
R	1	0		R	-1	1

4.4.3 Timber Frame Construction

Compared to exterior wall sheathing and drywall installation that have high similarity, timber frame construction has a more complicated assembly sequence. The workflow in this scenario is shown in Figure 4.11. The workpiece manipulation robot first needs to pick up the targeted timber material following the material gripping pose sequence $G(g_0, g_1)$ (Figure 4.11a). Next, the material is brought to the CNC machine to be cut following the cutting plane C and drilled by another robot with drilling poses $D(d_0, d_1, d_2, d_3)$ while the manipulation robot firmly holds the workpiece at c_h (Figure 4.11b). Then, it is placed at the installation location onto the wall frame with placing pose sequence $P(p_0, p_1, p_2)$ (Figure 4.11c). Lastly, screws are installed onto two of the drilled holes $S(s_0, s_1)$ for connection (Figure 4.11d).

The goal state matrix is the same as the two previous scenarios (Table 4.3). However, since material processing (i.e., cutting and drilling) are expected, the Cell (M, M) of the initial scene state matrix equals 1 (i.e., material processing needed). Therefore, the initial state SDM is different from the two previous scenarios and thus human demonstrations are needed to start the assembly sequence (Table 4.16). Table 4.17 shows the step-by-step sequential motion planning processes

for Scenario 3. Most SDM updating processes are similar to the previous scenarios and thus are not included in the table. However, details for the steps that are different from previous scenarios are presented and discussed.

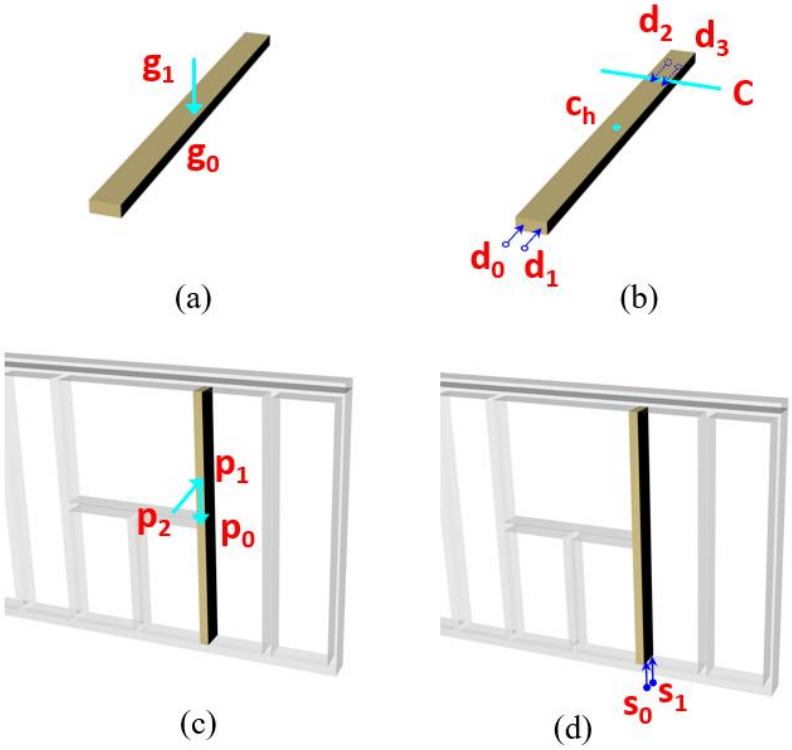


Figure 4.11 Timber Frame Construction Workflow

Table 4.16 Scenario 3 - Initial State

Step 0 (Robot Default): $Start()$									
Current State					**SDM (3-0)**				
	M	T	R	C		M	T	R	C
M	1	0	0	0	M	-1	1	0	0
T	0	0	0	0	T	0	0	0	0
R	0	0	0	0	R	0	0	1	0
C	0	0	0	0	C	0	0	0	1

Table 4.17 Sequential Motion Planning Processes for Scenario 3

Step 1 (Human Demonstration – M7): $Reach(G)$			
Step 2 (Human Demonstration – M8): $Grasp()$			
Step 3 (Human Demonstration – M9): $start\_material\_processing()$			
Current State		**SDM (3-3)**	
ID	Num	ID	Num
0	0	1	1
0	0	2	4
Step 4 (Human Demonstration – T1): $start\_cutting()$			
Current State			**SDM (3-4)**
	T	R		T	R
T	0	0	T	1	0
R	0	0	R	0	1
Step 5 (Human Demonstration – MP1): $Reach(c_h)$			
Step 6 (Human Demonstration – MP2): $Cut(C)$			
Step 7 (Human Demonstration – MP3): $Pullback()$			
Step 8 (Robot Default): second-level SDM null - $return\_trans\_level()$			
Current State		**SDM (3-3)**	
ID	Num	ID	Num
1	1	0	0
0	0	2	4
Step 9 (Robot Default): one operation type - $start\_drilling()$			
Step 10 (Human Demonstration – MP4): $Reach(d_0)$			
Step 11 (Human Demonstration – MP5): $Drill(d_0)$			
Step 12 (Robot Learned – MP3): $Pullback()$			
Steps 13 - 27 (Robot Default, Robot Learned – MP4, MP5, MP3): Repeat Steps 8 – 12 for the other three drilling			
Step 28 (Robot Default): second-level SDM null - $return\_trans\_level()$			
Step 29 (Robot Default): transition-level SDM null - $return\_main\_level()$			
Step 30 (Robot Learned – M3): $Reach(P)$			
Step 31 (Robot Learned – M4): $start\_connection()$			

Steps 32 - 41 (Robot Default, Robot Learned – C4, C5, C3): Repeat $start\_screwing()$, $Reach(S)$, $Screw()$, $Pullback()$, $return\_trans\_level()$ twice to install screws.
Step 42 (Robot Default): transition-level SDM null - $return\_main\_level()$
Step 43 (Robot Learned – M5): $Release()$
Step 44 (Robot Learned – M6): $Pullback()$
Step 45 (Robot Default): $Finish\ ()$

In Step 3, the human co-worker indicates the start of the material processing operation through demonstration. Since two low-level processing operations, cutting (id: 1) and drilling (id: 2), are required for material processing, two rows are included in the transition-level SDM generated. Each row represents an operation mode. As a consequence, the human co-worker needs to select one operation from the two as a demonstration. After the cutting operation is completed at Step 8, the corresponding row in the transition-level SDM becomes null. Therefore, there is only one operation left in the matrix and the robot will start the drilling operation (id: 2) following the default transition. It should be noted that the mapping for different high-level operations is saved separately to avoid mismatching between different types of operations. As a result, even though the initial SDM for cutting in the material processing operation at Step 4 is the same as the nailing in the connection operation, demonstrations are still required. The robot knowledge base after three operations is shown in Appendix A.

4.5 Proof-of-Concept Implementation

In order to verify the proposed system, a proof-of-concept implementation has been conducted in Gazebo simulation. An industrial robot has been used to build a wooden shelf with timbers. The robot end-effector is a combination of a vacuum gripper and a nailing gun to perform both the gripping and nailing functions. Following the BIM-driven HRCC system proposed in Chapter 3, the BIM that contains the related information is created in Rhino, and the digital twin

used by human co-workers' to interact with the robot is developed with ROS and Unity. Figure 4.12 shows the BIM used for the shelf construction task. The timber materials to be used have already been detected by the robot and saved in BIM. As defined in the BIM, the robot will first install two base studs on the bottom. Then, it starts to install the three studs on the top. After each top stud is placed, the robot needs to use 8 nails to fix the top stud onto the bases, with 4 nails on each base. It is assumed that the material poses can be detected and tracked in the robot workspace during the experiment process.

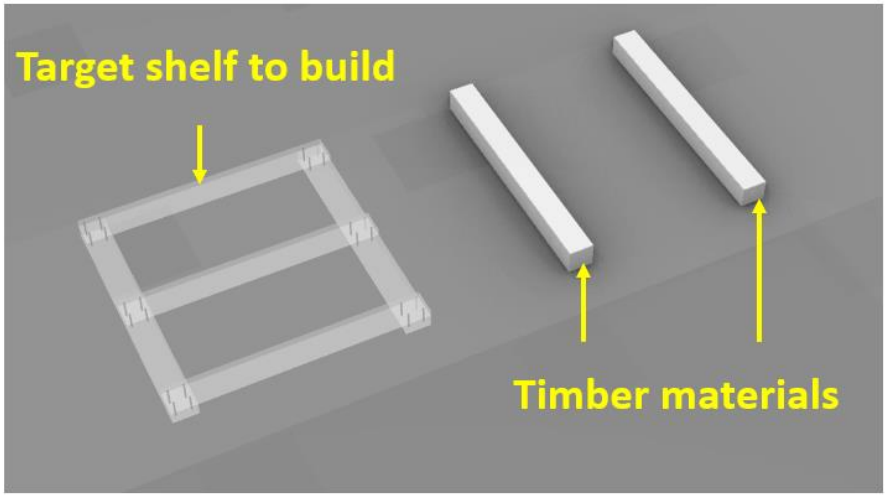


Figure 4.12 BIM of Shelf Construction task

The BIM framework for HRCC proposed in Chapter 3 primarily focuses on the physical components for construction tasks. In this chapter, an improved BIM data structure, which can represent more complex construction tasks with various operations, has been developed to meet the needs of interactive LfD. The proposed data structure for BIM elements with explanations is shown in Table 4.18.

During the HRCC process, the human co-worker observes construction site conditions and interacts with the robot through the digital twin interface. The digital twin is provided with both the 3D version for more convenient headset-free interaction and the VR version for higher fidelity.

Similar to the HRCC process described in Chapter 3, the human co-worker first needs to confirm the target and the target pose from BIM and robot suggestions (Figure 4.13a). After the target and its pose is confirmed, the robot generates the SDM for the target and starts the LfD process. When the SDM is unknown to the robot, the robot will request demonstrations from its human co-workers. The human co-workers can select corresponding primitives from the interactive menu from the digital twin interface (Figure 4.13b). If parameters are needed for the demonstrated primitive (e.g., which object to reach), the robot will prompt the human co-worker to indicate related information through the digital twin (Figure 4.13c). When the SDM is within the robot's knowledge base, the robot will show its decision to its human co-workers, as shown in Figure 4.13d. If the selected primitive is a motion primitive that can be demonstrated, such as manipulation, the robot will demonstrate the detailed motion plan to the human co-worker for approval before execution. Human co-workers can request the robot to develop another motion plan if they are not satisfied with the demonstrated one. Otherwise, if they are not satisfied with the selection of the motion primitive, they can select another primitive before the robot takes any action. If the selected primitive is a reasoning primitive, the robot will directly update the SDM. However, the human co-worker has the option to return to the previous step because the robot does not take any actual movement with the reasoning primitive.

Table 4.18 BIM Element Data Structure

Name	Name of the element
Family	The element belongs to one of the three families, Component (e.g., a stud), Connection (e.g., a nailing point indicator), Processing (e.g., a cutting plane indicator)
Parents	If the element is a component, it may or may not have parents that the component fully connected to. Otherwise, if the element is in the Connection or Processing family, its parent indicates the component the operation directly associated with (e.g., the stud to cut is the parent of the cutting plane indicator).

Type	Indicating the type of material the robot should obtain for the corresponding operation.
Position	Coordinates the target points for the robot to reach an element. It should be a list that contains all the intermediate and the final positions in the pose sequence to reach the element.
Orientation	List of orientations for the robot end-effector to reach certain points in the pose sequence
Order	Construction sequence
Connection	For Component family members. It is a list of connection operations associated with a component (e.g., [drilling, screwing]).
Processing_M	For Component family members. It is a list of material processing operations associated with a component (e.g., [cutting]).
Processing_T	For Component family members. It is a list of target processing operations associated with a component (e.g., [painting]).
Methods	For Connection or Processing family members. It indicates the operation method of the element. For example, if the element is a nailing point indicator, the method should be "nailing".

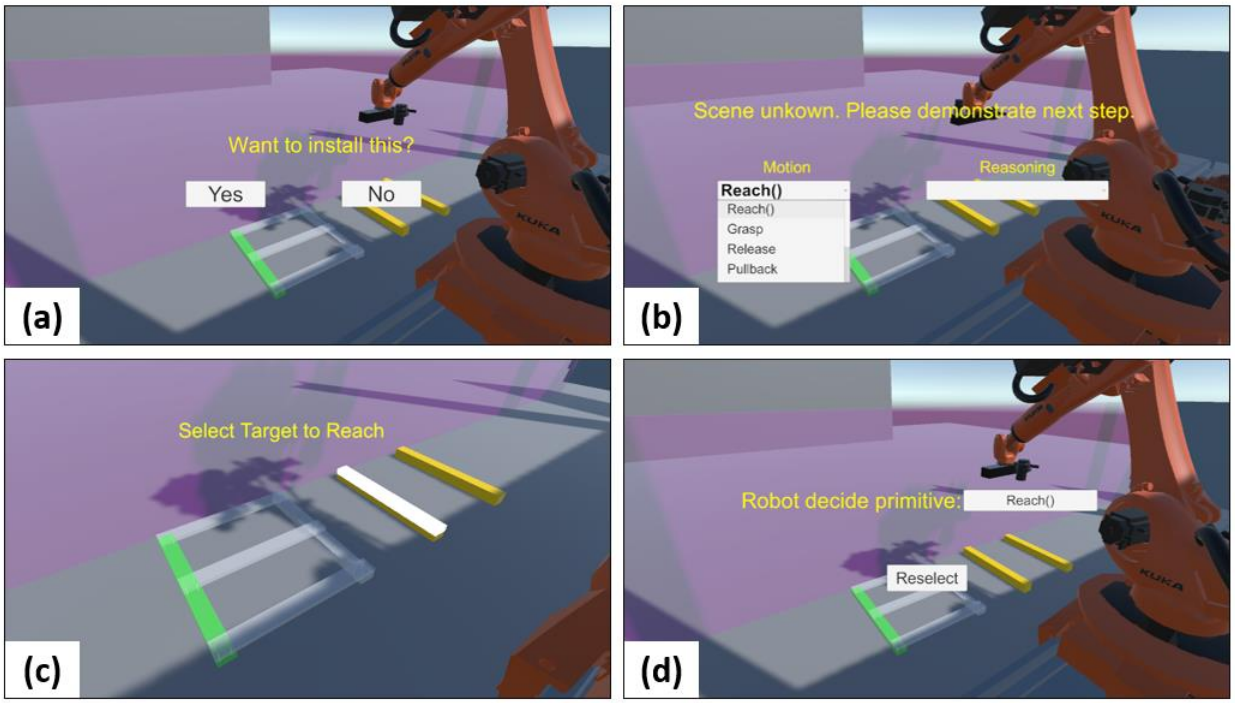


Figure 4.13 Digital Twin LfD Interface

4.6 Discussion

In the case study, with the provided 8 parameterized motion primitives, 8 reasoning primitives, and 6 system embedded default transitions, the robot learns three types of construction tasks from human demonstrations. Table 4.19 shows the number of decisions made through human demonstrations, robot learned knowledge, and robot embedded default transitions during the motion sequencing process. Even though only 6 default transitions, including start and finish, are embedded into the system, they make up a substantial proportion of decisions to facilitate the transition among different levels of SDM.

Table 4.19 Sources of Decision in Different Scenarios

	Scenario 1	Scenario 2	Scenario 3
Human Demonstrations	9	2	9
Robot Learned	9	16	20
Robot Default	11	11	17
Total	29	29	46

From the distribution graph (Figure 4.14), it can be found that there is a significant reduction in the percentage of human demonstrations from Scenario 1 (exterior wall sheathing) to Scenario 2 (drywall installation). The reason is that these two scenarios are similar, with considerable overlap in the assembly sequence. Most of the knowledge required to support decision-making in Scenario 2 has already been achieved from Scenario 1. Scenario 3 (timber frame construction) witnesses a slight increase in the percentage of human demonstrations because it requires some material processing work such as cutting and drilling, making it more complex and varied from the two previous scenarios. Nevertheless, there is still a huge reduction in the percentage of human demonstrations because the robots become more knowledgeable compared to Scenario 1.

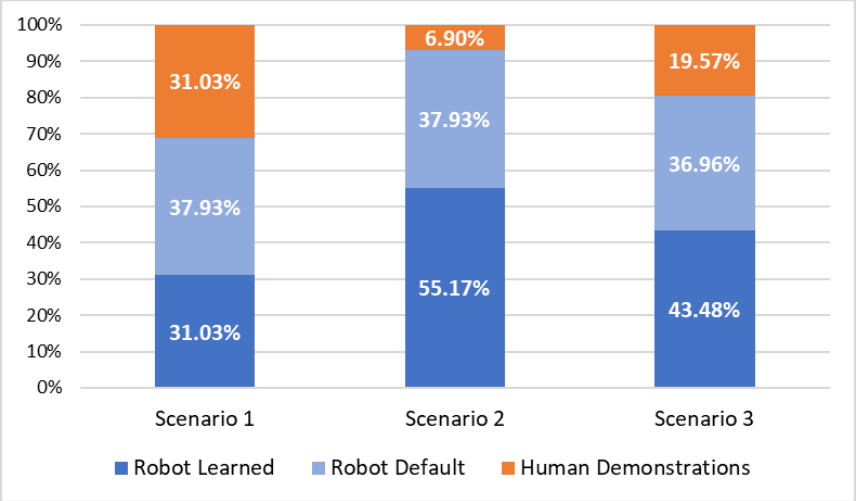


Figure 4.14 Human and Robot Decision Distribution for Case Studies

The proposed SDM can not only be used by robot LfD processes but can also support robot automatic sequential motion planning for future tasks. Even though the case study demonstrates the capability of the robots to interactively learn from human demonstrations and transfer the learned knowledge to different types of construction work, the most common cases in construction are those where the robots apply the learned skills to repetitively perform the same types of construction tasks. In these cases, the robots will be able to automatically decide their motion sequences after going through the initial learning processes with humans. For example, in the proof-of-concept implementation, the motion sequence to install the two base studs and three top studs are identical respectively. Therefore, no human demonstration is needed for the second or third workpiece of each type because the robot can autonomously follow the learned mapping to develop the motion sequence (Figure 4.15). As a result, for the overall shelf construction process, only 8.7% of decisions request human demonstrations and 91.3% of decisions are autonomously made by the robot.

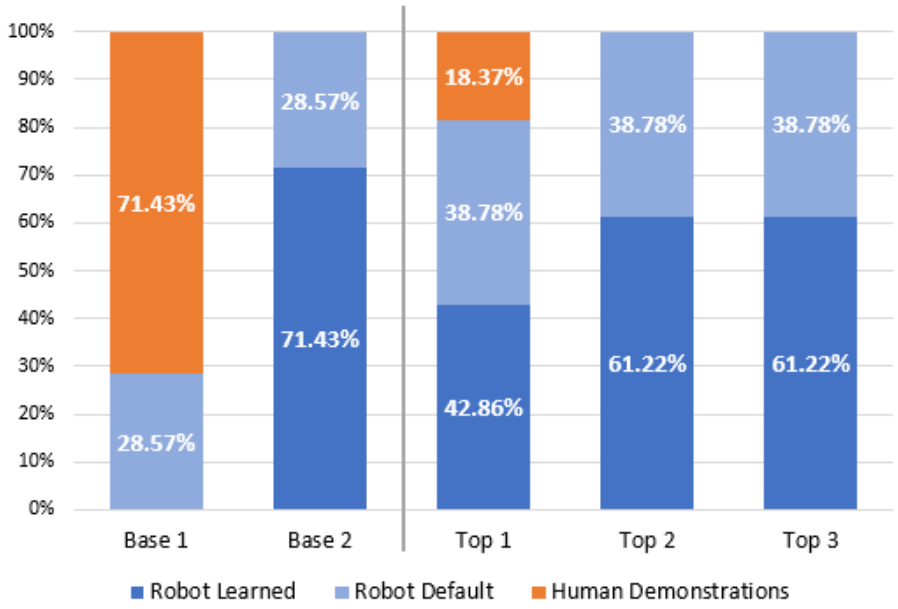


Figure 4.15 Human and Robot Distribution for Shelf Construction

4.7 Conclusions

The objective of the chapter is to explore how to enable robots to automatically decide their sequential motions to conduct construction assembly tasks. An interactive LfD framework has been proposed for the robot to learn motion sequencing skills from human demonstrations with the digital twin system. The learned knowledge is not only applied to the same types of construction tasks but is also transferred to other task types to reduce the teaching effort. This research has several contributions. First, it formalizes a four-level terminology—activities, tasks, operations, elemental motions—to describe robotic construction work at different levels. Three types of construction robotics skill primitives are introduced to enable modularized construction robot "programming", making robots accessible for construction workers without programming expertise.

Second, SDM based on multi-level scene state representation matrices is proposed. The probabilistic mapping from SDMs to corresponding skill primitives is learned by the robot from

human demonstrations and recorded in the robot's knowledge base. A case study with 8 motion primitives, 8 reasoning primitives, and 6 system-embedded default functions for SDM level transitions is presented. Human demonstrations and robot decisions required during the interactive LfD process are analyzed in detail. A shelf construction task is used to verify the combination of the motion sequencing LfD module with the digital twin HRCC system as a proof-of-concept implementation. It is found that as more types of construction tasks are learned, robots can increasingly and automatically decide sequential motions with fewer human demonstrations required.

Third, a technical and social framework for construction robot delivery based on the proposed modularized skill primitives and the digital twin HRCC system. The proposed delivery framework takes advantage of the domain knowledge of both robot engineers and construction workers to improve system reliability and work quality. It allows construction robots to be involved in different tasks at different construction stages thereby improving their cost efficiency. Therefore, it has the potential to enable the widespread deployment of HRCC in the construction industry.

Instead of generating the whole motion sequence for a construction task upfront, the robot decides its motions step by step and requests human workers' confirmation before execution. This complies with the workflow of the proposed digital twin system that human workers confirm whether the robot adapts its plan successfully according to the as-built conditions and intervene if necessary. It also improves the system's robustness to uncertainties and failures during the task execution process. Future work will investigate the robot's confidence in its decisions and skip the confirmation process when the robot is highly confident.

Chapter 5 Conclusions

5.1 Significance of the Research

Human-robot partnership in digitally-driven field construction can take advantage of both human and robot intelligence as well as robots' physical operational capabilities to overcome uncertainties and successfully perform construction work on-site. This research presents a human-robot collaborative construction (HRCC) framework based on the integration of building information models (BIM), interactive virtual reality (VR), and process-level digital twins. Human workers play the role of supervisors that perform high-level decision-making and supervision, while the robots serve as their intelligent assistants that perform lower-level task planning and execute construction work. As robots interact with their human partners, they can learn new skills such as optimal motion sequencing thereby further reducing human workload. This research also explores the integration of BIM to drive the construction process and the deployment of the physical robotic system and construction site configuration to enable field construction work.

Importantly, this research is not a combination of independent disconnected studies, but it rather proposes a scalable and extensible framework that provides considerable adoption and expansion opportunities to the research community. The developed platform has the flexibility for scholars with different research focuses to integrate their own work as a module in the framework, such as developing robot motion planners that work better for construction robotics. Researchers can also build upon this collaboration platform to investigate other directions of their interest, such as studying the impact of human-robot partnerships on construction project delivery.

In addition, this research has several broader impacts. First, it has the potential to transition the role of construction workers from physical laborers performing strenuous construction work in dangerous working environments to that of robot supervisors who digitally teach and supervise the robots from safe workplaces. This has significant potential to attract the younger generation into the construction industry thereby mitigating the prevailing aging workforce issues.

Second, with the help of the robot for physical work execution, the physical demand on construction workers is significantly reduced. As a result, the population who are currently underrepresented in construction because of their physical capabilities, such as women and people with physical disabilities, can get more opportunities to become part of the construction workforce.

Third, by enabling robots to learn new construction tasks on-site, a new construction robot delivery framework is proposed. The robots delivered to the construction site are embedded with LfD modules and programmed with modular skill primitives for construction workers to intuitively teach robots construction tasks on-site. It can eliminate misunderstandings caused by the knowledge gap between the robotic programming and construction domains, thereby improving the rationality, flexibility, and quality of robotic construction work.

Lastly, with more construction tasks learned by the robot, the robot involvement is extended to more construction activities at different construction stages. Compared to single-task construction robots that perform one type of construction task and remain idle for the remainder of a construction project, multi-task robots have much higher potential usage and cost efficiency. Therefore, it has the potential to enhance the motivation of construction stakeholders to deploy robots in their projects.

5.2 Research Contributions

This research contributes to promoting construction automation and robotics in field construction work by creating a pathway to enable human-robot partnerships in digitally-driven construction work by connecting human workers, robots, BIM, and construction sites. The specific contributions of this research are as follows:

- An immersive and interactive process-level digital twin system that allows human workers to guide and supervise robotic construction work through real-time visualization and bi-directional communication.
- A closed-loop system that integrates BIM to drive the HRCC and save as-built construction data.
- An automatic approach to generate interactive human-in-the-loop digital twins of robotic construction workfaces using BIM and sensing data.
- A physical setup to deploy the digitally-driven construction system on a physical industrial robotic arm and laboratory-simulated construction sites.
- Interactive Learning from Demonstration workflow for human workers to teach robots to automatically determine their motion sequences.
- A Scene Distance Matrix framework to guide robot sequential decisions on its motions in the performance of multi-step construction assembly tasks.
- A multi-task construction robot delivery framework based on modular construction robot skill primitives and robot Learning from Demonstration capability.

5.3 Future Directions

5.3.1 System Evaluation through In-Field Construction Experiments

This research evaluated the proposed system through simulated construction tasks in the research laboratory. However, there are unexpected situations and external factors on construction sites that might affect system performance, such as power supply to robots, the ability to provide relatively static workspaces for robots, environmental conditions (e.g., lighting, temperature, humidity, dust), and workers' acceptance of robots. Future work should seek opportunities to test the system with full-scale construction tasks on-site.

5.3.2 Human-in-the-Loop Multi-Robot Collaboration

Construction work has a collaborative nature. Many construction tasks are collaboratively performed by a crew of construction workers and equipment. For example, in the RSMeans Building Construction Cost Data, beam concrete forming cost is calculated based on a standard workgroup setting of Crew C-2 that includes 1 carpenter foreman, 4 carpenters, and 1 laborer (Mewis 2019). In Chapter 4, an automation system that consists of two robots and a CNC machine is used as the case study. However, in this case study, different robot or equipment has already been assigned with different elemental motions. At each step, after the motion is decided, the corresponding robot is applied to complete the motion, thus only the motion of one robot is involved at each step.

In the future, more flexible cases of multi-robot collaboration in construction should be explored, such as multi-robot tandem manipulation of large workpieces. Another typical collaboration pattern in construction is that multiple workers perform different types of operations simultaneously on a project. In correspondence with this pattern, the collaboration of

heterogeneous multi-robot-human construction teams is another promising area to explore. Research directions in this area include optimized task allocation among robots and multi-robot team level LfD. It is also worth exploring efficient ways to handle the quasi-repetitive nature of construction tasks considering transition cases (e.g., intersection of two walls).

5.3.3 Improving Robot and Infrastructure Intelligence

This research mainly focuses on taking advantage of the robot's intelligence in computation and decision-making for previously encountered, learned situations. With the fast development of machine learning, artificial agents can make decisions for unexperienced situations based on historical data. Future work will take advantage of machine learning to improve robot and infrastructure intelligence. First, future work will study how to automatically prepare BIM for robotic construction and to improve robot adaptability to uncertainties to improve system autonomy. Second, technical approaches to raise robots' and infrastructures' awareness of human workers and to allow natural interaction with human workers will be investigated. Third, the current system requires human co-workers' confirmation of the target selected, the high-level task plan, and the motion plan at each step to ensure construction safety. In the future, the robot's confidence in its decisions will be integrated to optimize human supervision workflow and allow one human worker to supervise multiple robots at the same time.

5.3.4 Understanding Human Factors during HRCC

The ultimate goal of this research is to make human workers' lives easier with the proposed robotic automation technology. While this dissertation mainly focuses on discussing the technological aspects to enable the HRCC work, future studies can potentially integrate the work on understanding human factors, including their physiological and mental status and work

performance (e.g., (Lee et al. 2021; Wang et al. 2019b)), with different system settings during the collaboration process. The concept of human-centric monitoring and control (Deng et al. 2022) should be explored to adapt to the personal needs of different co-workers while they are working with robots. Approaches to help human workers establish appropriate trust in their robot partners will also be investigated.

Appendices

Appendix A Robot Knowledge Base Learned from Case Study Scenarios (Chapter 4)

Main-Level Mapping:

Index	SDM				Primitive	Probability
M1	0	1	0	0	$Reach(G)$	1.0
	0	0	0	0		
	0	0	1	0		
	0	0	0	1		
M2	0	1	0	0	$Grasp()$	1.0
	0	0	0	0		
	-1	0	1	0		
	0	0	0	1		
M3	0	1	-1	0	$Reach(P)$	1.0
	0	0	0	0		
	-1	0	1	0		
	0	0	0	1		
M4	0	0	-1	0	$start\_connection()$	1.0
	0	0	0	0		
	-1	-1	1	0		
	0	0	0	1		
M5	0	0	-1	0	$Release\ ()$	1.0
	0	0	0	0		
	-1	-1	1	0		
	0	0	0	0		

M6	0	0	0	0	$Pullback\ ()$	1.0
	0	0	0	0		
	-1	-1	1	0		
	0	0	0	0		
M7	-1	1	0	0	$Reach(G)$	1.0
	0	0	0	0		
	0	0	1	0		
	0	0	0	1		
M8	-1	1	0	0	$Grasp()$	1.0
	0	0	0	0		
	-1	0	1	0		
	0	0	0	1		
M9	-1	1	-1	0	$start\_material\_processing()$	1.0
	0	0	0	0		
	-1	0	1	0		
	0	0	0	1		

Transition-Level Mapping:

Index	SDM	Primitive	Probability
T1	1 1 2 4	$start\_cutting()$	1.0

Connection Mapping:

Index	SDM	Primitive	Probability
C1	$\begin{matrix}1 & 0\\ 0 & 1\end{matrix}$	$Reach(n_0)$	1.0
C2	$\begin{matrix}1 & 0\\ -1 & 1\end{matrix}$	$Nail()$	1.0
C3	$\begin{matrix}0 & 0\\ -1 & 1\end{matrix}$	$Pullback()$	1.0
C4	$\begin{matrix}2 & 0\\ 0 & 1\end{matrix}$	$Reach(s_0)$	1.0
C5	$\begin{matrix}2 & 0\\ -1 & 1\end{matrix}$	$Screw()$	1.0

Material Processing Mapping:

Index	SDM	Primitive	Probability
MP1	$\begin{matrix}1 & 0\\ 0 & 1\end{matrix}$	$Reach(c_h)$	1.0
MP2	$\begin{matrix}1 & 0\\ -1 & 1\end{matrix}$	$Cut(C)$	1.0
MP3	$\begin{matrix}0 & 0\\ -1 & 1\end{matrix}$	$Pullback()$	1.0
MP4	$\begin{matrix}2 & 0\\ 0 & 1\end{matrix}$	$Reach(d_0)$	1.0
MP5	$\begin{matrix}2 & 0\\ -1 & 1\end{matrix}$	$Drill(d_0)$	1.0

Appendix B Setup Raspberry Pi for Remote Connection with ROS Melodic

B1. Install ROS and Ubuntu 18 on Raspberry PI (RPI)

Step 1: Insert a freshly formatted SD card onto an Ubuntu computer

Step 2: In Terminal of the Ubuntu computer:

```
sudo apt-get update

snap install rpi-imager
```

Step 3: Open RPI Imager

(1) Choose OS – Use Custom – ubuntu-18.04.5-preinstalled-server-arm64+raspi4.img

(2) Choose Storage – SD Card

(3) Write – Yes

Notes:

- If RPI Imager does not work, consider using Belena Etcher (https://www.balena.io/etcher/)
- For the latest version of Ubuntu image for RPI, it may directly available in RPI Imager

Step 4: Start RPI

(1) Insert SD card into RPI

(2) Connect HDMI, mouse, and keyboard

(3) Connect power to start RPI

(4) User name: ubuntu

Password: ubuntu

Then, follow instructions to reset the password.

Step 5: Connect RPI to Wifi.

(1) In Terminal:

```
ls /sys/class/net
```

You should see wlan0 as an option in the output.

(2) Then, in Terminal:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

(3) Type the following content into the file. Always use spaces but not “Tab” during this process.

```
wifis:
    wlan0:
        dhcp4: true
        optional: true
        access-points:
            "SSID_name":
                password: "WiFi_password"
```

Example:

```
wifis:
    wlan0:
        dhcp4: true
        optional: true
        access-points:
            "TP-LINK_1A11_5G":
                password: "12345678"
```

(4) Save and go back to Terminal. Restart network:

```
sudo netplan generate

sudo netplan apply
```

(5) Test network connection

```
ping www.google.com
```

Example output showing connection success:

```
PING www.google.com (142.250.191.164) 56(84) bytes of data.
64 bytes from ord38s30-in-f4.1e100.net (142.250.191.164): icmp_seq=1
ttl=54 time=159 ms
64 bytes from ord38s30-in-f4.1e100.net (142.250.191.164): icmp_seq=2
ttl=54 time=164 ms
64 bytes from ord38s30-in-f4.1e100.net (142.250.191.164): icmp_seq=3
ttl=54 time=159 ms
64 bytes from ord38s30-in-f4.1e100.net (142.250.191.164): icmp_seq=4
ttl=54 time=164 ms
64 bytes from ord38s30-in-f4.1e100.net (142.250.191.164): icmp_seq=5
ttl=54 time=173 ms
64 bytes from ord38s30-in-f4.1e100.net (142.250.191.164): icmp_seq=6
ttl=54 time=157 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5188ms
rtt min/avg/max/mdev = 157.776/163.361/173.681/5.363 ms
```

Figure B.1 Output showing Connection Success

If connected, reboot the computer.

```
sudo reboot
```

(6) Install Desktop GUI

```
sudo apt-get update

sudo apt install xubuntu-desktop //optionally: ubuntu-desktop

sudo reboot
```

Step 6: Fix Wifi Connection

(1) Now the Ubuntu Desktop interface shows up but the GUI for scanning Wifi and finding the network does not work. To fix it:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

(2) Change the document to:

```
network:
    version: 2
    renderer: NetworkManager
```

(3) Save the file and go back to Terminal to restart the network setting.

```
sudo netplan generate

sudo netplan apply

sudo reboot
```

(4) After reboot, click the network (arrow) logo on the top-right corner and now the Wifi can be connected.

(5) Optional: since RPI has limited disk space, some applications can be removed to save disk space:

```
sudo apt-get remove libreoffice-common

sudo apt-get -y purge thunderbird* simple-scan hplip* printer-
  driver* libreoffice* onboard gnome-sudoku gnome-mines
  dictionaries-common atril-common* mate-calc* parole sgt-puzzles
  orage

sudo apt autoremove
```

Step 7: Change Hostname and Username.

(1) The default user name is “ubuntu”, which conflicts with the system name. It may cause problems later and is hard to identify thus it is strongly recommended to change the default user and host name.

```
sudo su

hostnamectl set-hostname rpiA

sed -i "s/\bubuntu\b/xirpi/g" `grep ubuntu -rl /etc/passwd`

sed -i "s/\bUbuntu\b/xirpi/g" `grep ubuntu -rl /etc/passwd`

sed -i "s/\bubuntu\b/xirpi/g" `grep ubuntu -rl /etc/shadow`

mv /home/ubuntu /home/xirpi

sed -i "s/\bubuntu\b/xirpi/g" `grep ubuntu -rl /etc/group`

sudo reboot
```

Notes:

- Replace **rpiA** with your preferred host name. Same for the following contents.
- Replace **xirpi** with your preferred user name. Same for the following contents.

(2) After reboot, an error “Configured directory for incoming files does not exist” may appear. In this case:

```
gsettings get org.blueman.transfer shared-path
```

It will return:

```
/home/ubuntu/Downloads
```

Change it to the new path and reboot:

```
gesttings set org.blueman.transfer shared-path
  '/home/xirpi/Downloads'

sudo reboot
```

Step 8: Install ROS following instructions on http://wiki.ros.org/melodic/Installation/Ubuntu.

Notes:

- Choose the ROS version that corresponds to the Ubuntu version.
- The ROS and Ubuntu version should be the same as other computers that communicate with RPI.

B2. Config for Remote ROS Connection

All the bold font in the following content should be replaced based on your own computer and network settings.

Step 1: Setup static ip address for RPI and master ROS computer.

(1) Settings – Network – Setting for the corresponding network – IPv4

Address: 191.168.1.**22** (Change to a different number that is unique on the local network)

Netmask: 255.255.255.0

Gateway: 192.168.1.1

Step 2: Reboot the computer and test the SSH connection

(1) After reboot and confirm ip address of your computers:

```
ifconfig
```

It should show 192.168.1.**22** at a certain line.

(2) Test connection with ssh from the Terminal of the master ROS computer

```
ssh xirpi@192.168.1.22
```

If it cannot connect, refer to https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/ for troubleshooting.

Step 3: Configure the master ROS computer

(1) Find out hostname

```
hostname
```

It should return your hostname. Example: xi-master

(2) Configure host file:

```
sudo nano /etc/hosts
```

Add the following content to the top of the file:

```
127.0.0.1 localhost

127.0.1.1 xi-master //master computer hostname

192.168.1.20 xi-master //master computer hostname

192.168.1.22 rpiA //other (RPI) hostname
```

(3) Save the file and go back to Terminal:

```
nano ~/.bashrc
```

(4) Add the following content to the bottom:

```
export ROS_HOSTNAME=xi-master //master computer hostname

export ROS_MASTER_URI=http://xi-master:11311
```

(5) Save and go back to Terminal.

```
source ~/.bashrc
```

Step 4: Configure the RPI computer

(1) Find out hostname

```
hostname
```

It should return your hostname. Example: xi-master

(2) Configure host file:

```
sudo nano /etc/hosts
```

Add the following content to the top of the file:

```
127.0.0.1 localhost

127.0.1.1 rpiA //RPI hostname

192.168.1.22 rpiA //RPI hostname

192.168.1.20 xi-master //main computer hostname
```

(3) Save the file and go back to Terminal:

```
nano ~/.bashrc
```

(4) Add the following content to the bottom:

```
export ROS_HOSTNAME=rpiA //master computer hostname

export ROS_MASTER_URI=http://xi-master:11311
```

(5) Save and go back to Terminal.

```
source ~/.bashrc
```

Appendix C Setup Static IP for Computers on Local Area Network

C1. Setup Static IP on the Windows Operating System

Step 1: Click on the Internet button on the bottom-right corner from the menu bar, then select "Network & Internet settings". This operation will open the network status settings window.

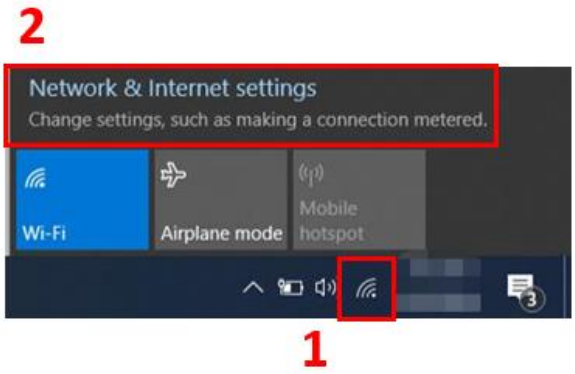


Figure C.1 Open Network Settings

Step 2: Click "Change adapter options" under "Advanced network settings". This operation will open the "Network Connection" window.

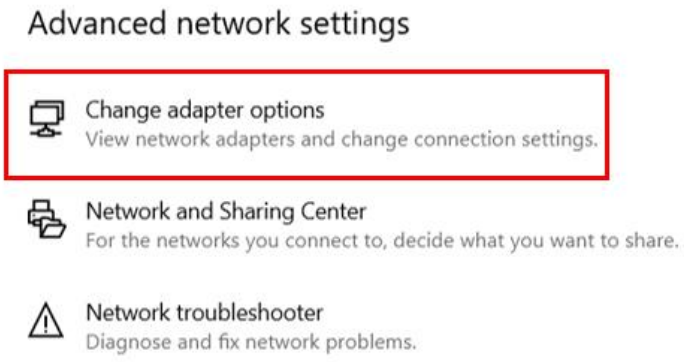


Figure C.2 Change Adapter Options

Step 3: Right-click “Wi-Fi” in “Network Connection” and select “Properties”. It will require administrator access. The “Wi-Fi Properties” window will open.

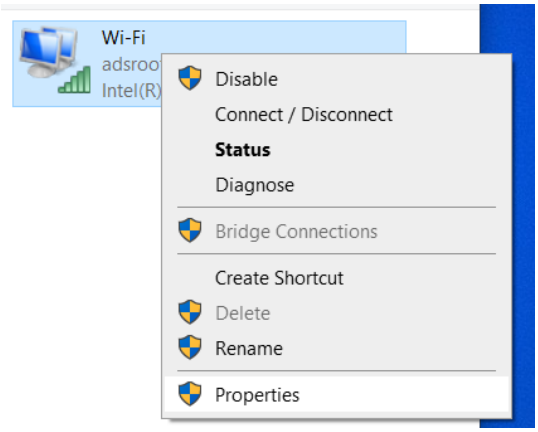


Figure C.3 Open Wi-Fi Properties

Step 4: In the “Wi-Fi Properties” window, under the “Networking” tab, click on “Internet Protocol Version 4 (TCP/IPv4)” and then click on “Properties”.

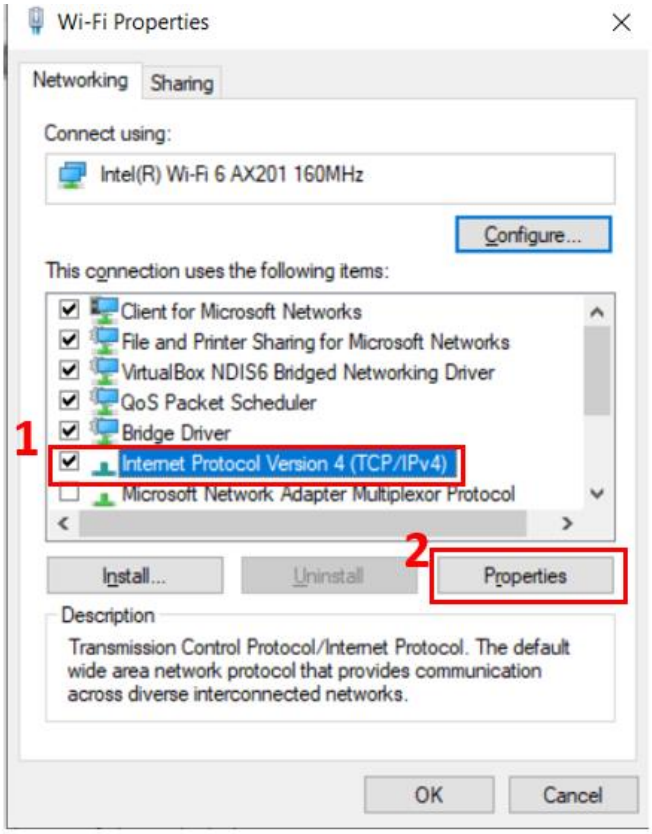


Figure C.4 Open IPv4 Settings

Step 5: In the IPv4 properties window, instead of letting the computer automatically obtain IP address and DNS server address, select the options to manually setting address. Assign an unoccupied IP address for the computer on the Local Area Network (LAN). In our case, the address for the robot embedded computer is 192.168.1.11. Therefore, an IP address on the same subnet is assigned to this computer. After all the address is set up, click “OK” to confirm.

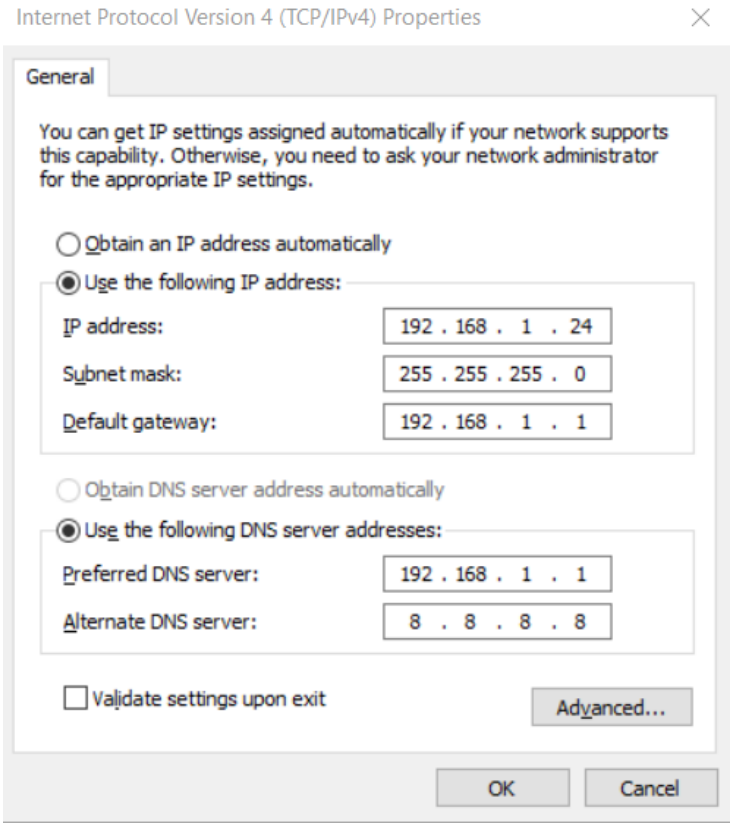


Figure C.5 Manually Input Static IP Address

Step 6: Wait for a few seconds for the configuration to apply. Make sure the computer is connected to the correct Wi-Fi. Open the Command Prompt window and ping another computer on the LAN to verify the connection.

C2. Setup Static IP on the Linux Operating System

Step 1: Click on the Internet button on the top-right corner from the menu bar. Then, select the connected network and click the setting button on its right.

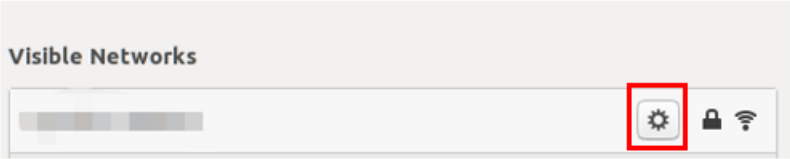


Figure C.6 Open Network Settings (Linux)

Step 2: In the setting window, navigate to the IPv4 tab. Select “Manual” as the IPv4 method and manually input the address, netmask, and gateway.

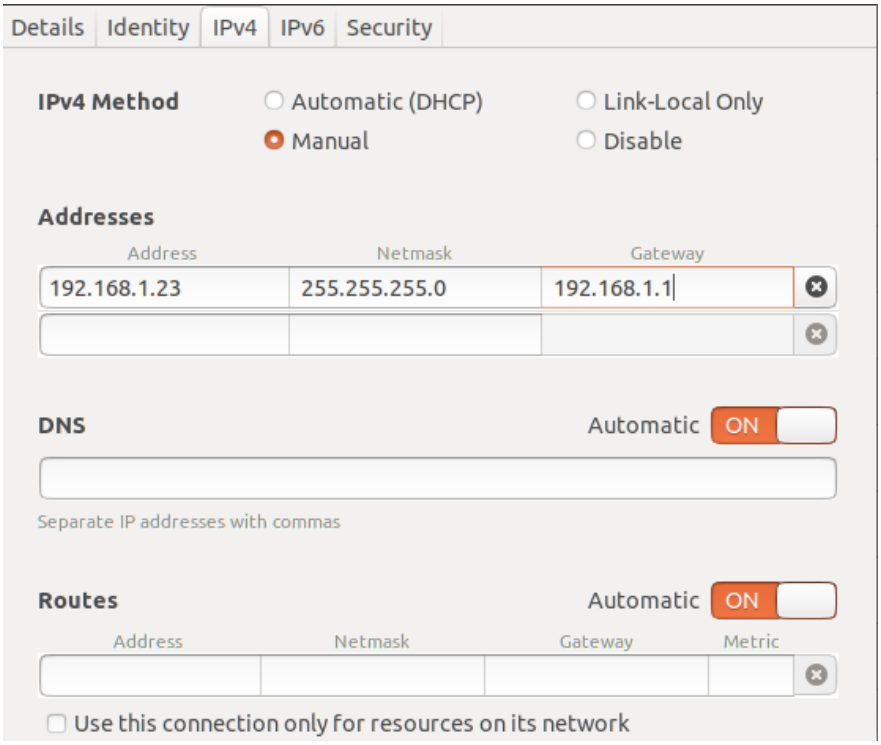


Figure C.7 Manually Input Static IP Address (Linux)

Step 3: After setting is finished, reboot the system for the new IP address to apply. To verify the connection, ping another computer on the LAN in the Terminal.

Appendix D Setup Rhino for Connection with ROS

Step 1: On a Windows OS computer with Rhino installed, install Compas and Compas_rhino libraries. Open an Anaconda Prompt terminal:

```
conda create -n rhinoenv

conda activate rhinoenv

conda install -c conda-forge compas

pip install roslibpy

python -m compas_rhino.install -v 7.0

conda install -c conda-forge compas_fab

python -m compas_rhino.install -v 7.0
```

References:

- https://compas.dev/compas/latest/installation.html
- https://compas.dev/compas/latest/gettingstarted/rhino.html

Notes:

- If the following error is encountered:

  ```
  Exception: The scripts folder does not exist in this location: C:\Users\xiwang\AppData\Roaming\McNeel\Rhinoceros\7.0\scripts
  ```

  Start Rhino and run EditPythonScript. Then continue with the terminal commands above.

- If the following error is encountered:

  ```
  One or more errors occurred:

     - Exception('The grasshopper folder does not exist in this location: C:\\Users\\xiwang\\AppData\\Roaming\\Grasshopper')
  ```

  Start Grasshopper and then continue with the terminal commands above.

- If `python -m compas_rhino.install -v 7.0` continuously has error, manually copy the `compas_fab` folder in `C:\Users\xiwang\Anaconda3\envs\rhinoenv\Lib\site-packages` to `C:\Users\xiwang\AppData\Roaming\McNeel\Rhinoceros\7.0\scripts`

  After installation, open a new python script test.py in Rhino. Test whether the following program can run successfully:

```
import compas

import compas_rhino
```

Step 2: Connect with ROS

(1) Find out the ip address of the ROS computer to connect to (e.g., 192.168.1.20)

(2) Change Line 31 in `C:\Users\xiwang\Anaconda3\envs\rhinoenv\Lib\site-packages\roslibpy\ros.py` and `C:\Users\xiwang\AppData\Roaming\McNeel\Rhinoceros\7.0\scripts\roslibpy\ros.py` to

```
url = RosBridgeClientFactory.create_url('192.168.1.20', port,
is_secure)
```

Appendix E Commonly Encountered Errors

E1. Robot Intrude Safety Curtain

Solution 1: Workspace Override

On the teaching pendant, click the "Robot" button. On the screen, click Configuration – Miscellaneous – Workspace Monitoring – Override. Then, manually jog the robot out of the safety curtain. After moving the robot out of the curtain, the override mode should automatically exit.

Solution 2: Start-up Mode

If the workspace override mentioned above does not work, the robot needs to be jogged out of the safety curtain in the start-up mode.

Before entering the start-up mode, the Twincat PLC needs to run in Config mode. After opening the remote desktop at 192.168.1.11, right-click the Twincat button on the bottom-right corner of the screen and select "Config" mode. Wait for a moment until the PLC runs properly. Then, on the teaching pendant, log in with "Administrator". Select Start-up – Service – Start-up mode. If the "Start-up" button is grey, make sure the pendant is in T1 mode and the running program is stopped. Then, manually jog the robot out of the curtain in the Start-up mode. Finally, close the Start-up mode on the pendant and restart the PLC on the remote desktop. Reset the robot by pressing the white button on the robot control box before running any program.

E2. External Safety Stop 2

Error message “External Safety Stop 2” is usually related to the interruption of the safety curtain. When this error appears, first check to make sure the safety curtain is clear with all green lights on, then press the white button on the robot control box to reset. If only one red light appears on the top of the laser bar and all other lights are off, try to realign the two laser bars by adjusting the safety gate. If all the lights are red and blinking, it means the curtain needs to be reset. To reset the curtain, remove the two screws and disconnect the power (**Error! Reference source not found.**), wait for a few minutes, and then reconnect the power. After all the lights are green, press the reset button on the robot control box and the error should disappear.

Figure E.1 Reset Safety Gate Power Connection

E3. Unresponsive Pendant with White Screen

The pendant has a white screen with only the message “RDP Session; Connected; Successful connected to 172.17.0.1” on the top. After turning the switching key, the pendant responds properly for a moment. Then, it becomes unresponsive with the white screen again.

In this case, the robot needs to be completely restarted. First, turn the handle on the robot control box to the OFF position (**Error! Reference source not found.**). Wait for several minutes u ntil the pendant screen turns black, which means the robot has been completely shut down. Then, turn it to the ON position. The robot takes several minutes to completely start. Then, check the Twincat status and reactivate or restart Twincat if needed.

Figure E.2 Handle to Restart the Robot

E4. Resolving Errors in the Status Bar

In the normal status, the status bar on the robot teaching pendant should show ”S” and “I” in green. Otherwise, errors such as “Active commands inhibited” may occur when operating the robot.

When the “S” button is red, it means no submit interpreter is active. To active the submit interpreter, switch the pendant into “T1” mode, log in as “Expert” and click on “S”. Then, select “SYS” and press the “Select/Start” button on the right (**Error! Reference source not found.**). A fter this operation, the “S” button should turn green.

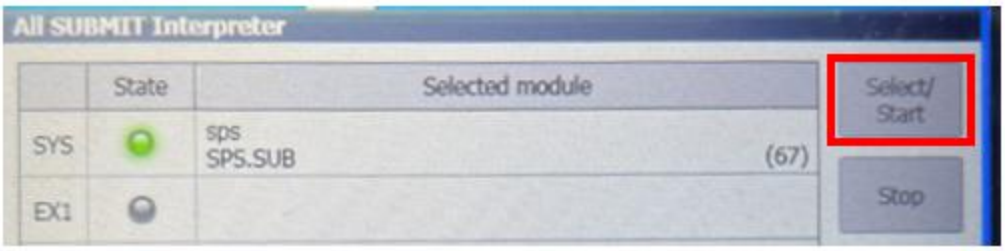


Figure E.3 Activate Submit Interpreter

When the second button shows “O” with a grey background, it means the drives are off. To turn the drives on, press the “O” button, then press “I” under “Drives” (**Error! Reference s ource not found.**). After that, the button should become “I” and turns green.

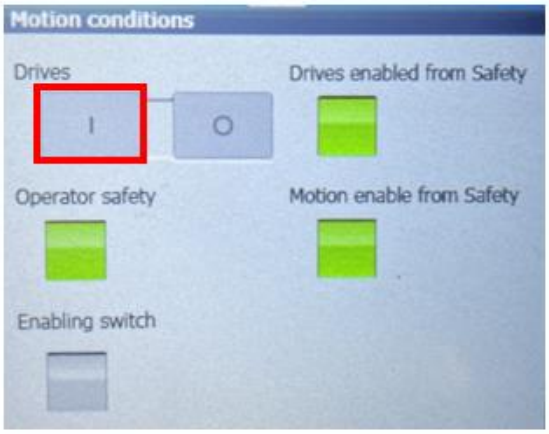


Figure E.4 Turn on Drives

When the pendant shows the error that the tool is not detected, check whether the tool and its frame have been set properly. Press the tool button in the menu bar and select the corresponding tool and its frame.

Figure E.5 Tool and Frame Setting

Bibliography

Abanda, F. H., J. H. M. Tah, and F. K. T. Cheung. 2017. "BIM in off-site manufacturing for buildings." *J. Build. Eng.,* 14: 89–102. Elsevier. https://doi.org/10.1016/J.JOBE.2017.10.002.

Adel, A. 2020. "Computational Design for Cooperative Robotic Assembly of Nonstandard Timber Frame Buildings." ETH Zurich.

Adel, A., A. Thoma, M. Helmreich, F. Gramazio, and M. Kohler. 2018. "Design of Robotically Fabricated Timber Frame Structures." 394–403.

AGC. 2020. "2020 Construction Outlook Survey Results National Results." Accessed March 22, 2022. https://www.agc.org/sites/default/files/Files/Communications/2020_Outlook_Survey_National.pdf.

Akanmu, A. A., J. Olayiwola, O. Ogunseiju, and D. McFeeters. 2020. "Cyber-physical postural training system for construction workers." *Autom. Constr.,* 117: 103272. Elsevier B.V. https://doi.org/10.1016/j.autcon.2020.103272.

Argall, B. D., S. Chernova, M. Veloso, and B. Browning. 2009. "A survey of robot learning from demonstration." *Rob. Auton. Syst.,* 57 (5): 469–483. https://doi.org/10.1016/J.ROBOT.2008.10.024.

Aulin, R., and M. Jingmond. 2011. "Issues confronting women participation in the construction industry." *Adv. Eng. Technol. Sci. Res. Dev.,* 312–318.

Barbash, G. I., and S. A. Glied. 2010. "New Technology and Health Care Costs — The Case of Robot-Assisted Surgery." *N. Engl. J. Med.,* 363 (8): 701–704. Massachusetts Medical Society. https://doi.org/10.1056/NEJMP1006602/SUPPL_FILE/NEJMP1006602_DISCLOSURES.PDF.

Bassier, M., B. Van Genechten, and M. Vergauwen. 2019. "Classification of sensor independent point cloud data of building objects using random forests." *J. Build. Eng.,* 21: 468–477. Elsevier Ltd. https://doi.org/10.1016/j.jobe.2018.04.027.

Beckoff. 2022. "Beckoff Information System - ADS Introduction." Accessed March 22, 2022. https://infosys.beckhoff.com/english.php?content=../content/1033/tcadscommon/html/tcads

common_intro.htm&id=.

Bilberg, A., and A. A. Malik. 2019. “Digital twin driven human–robot collaborative assembly.” *CIRP Ann.*, 68 (1): 499–502. CIRP. https://doi.org/10.1016/j.cirp.2019.04.011.

Billard, A., and S. Calinon. 2008. “Robot Programming by Demonstration.” *Handb. Robot.*, 1371–1394. Springer.

BLS. 2021. “Employed persons by detailed occupation, sex, race, and Hispanic or Latino ethnicity.” Accessed March 22, 2022. https://www.bls.gov/cps/cpsaat11.htm.

Bock, T. 2007. “Construction robotics.” *Auton. Robots*, 22 (3): 201–209. https://doi.org/10.1007/s10514-006-9008-5.

Boyd, N., M. M. A. Khalfan, and T. Maqsood. 2013. “Off-Site Construction of Apartment Buildings.” *J. Archit. Eng.*, 19 (1): 51–57. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000091.

Brosque, C., G. Skeie, J. Orn, J. Jacobson, T. Lau, and M. Fischer. 2020. “Comparison of Construction Robots and Traditional Methods for Drilling, Drywall, and Layout Tasks.” *HORA 2020 - 2nd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc.* https://doi.org/10.1109/HORA49412.2020.9152871.

Calinon, S. 2009. *Robot programming by demonstration : a probabilistic approach*.

Chen, J. Y. C., E. C. Haas, and M. J. Barnes. 2007. “Human performance issues and user interface design for teleoperated robots.” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, 37 (6): 1231–1245. https://doi.org/10.1109/TSMCC.2007.905819.

Chen, K., S. Chernova, and D. Kent. 2020. “Learning Hierarchical Task Networks with Preferences from Unannotated Demonstrations.” (CoRL).

Chen, Y.-C., H.-L. Chi, S.-C. Kang, and S.-H. Hsieh. 2016. “Attention-Based User Interface Design for a Tele-Operated Crane.” *J. Comput. Civ. Eng.*, 30 (3): 04015030. https://doi.org/10.1061/(asce)cp.1943-5487.0000489.

Chernova, S., and M. Veloso. 2010. “Confidence-based multi-robot learning from demonstration.” *Int. J. Soc. Robot.*, 2 (2): 195–215. https://doi.org/10.1007/s12369-010-0060-0.

Chitnis, R., S. Tulsiani, S. Gupta, and A. Gupta. 2020. “Efficient Bimanual Manipulation Using Learned Task Schemas.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 1149–1155. https://doi.org/10.1109/ICRA40945.2020.9196958.

Chitta, S., E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernandez Perdomo. 2017. “ros_control: A generic and simple control framework for ROS.” *J. Open Source Softw.*, 2 (20): 456. The Open Journal. https://doi.org/10.21105/joss.00456.

Chitta, S., I. Sucan, and S. Cousins. 2012. “MoveIt!” *IEEE Robot. Autom. Mag.*, 18–19.

Chotiprayanakul, P., D. K. Liu, and G. Dissanayake. 2012. “Human-robot-environment interaction interface for robotic grit-blasting of complex steel bridges.” *Autom. Constr.*, 27: 11–23. Elsevier B.V. https://doi.org/10.1016/j.autcon.2012.04.014.

Chung, J., S. H. Lee, B. J. Yi, and W. K. Kim. 2010. “Implementation of a foldable 3-DOF master device to a glass window panel fitting task.” *Autom. Constr.*, 19 (7): 855–866. Elsevier B.V. https://doi.org/10.1016/j.autcon.2010.05.004.

COMPAS. 2021. “COMPAS.” Accessed March 22, 2022. https://compas.dev/.

Construction Robotics. 2022. “MULE (Material Unit Lift Enhancer).” Accessed March 22, 2022. https://www.construction-robotics.com/mule/.

Correa, F. 2019. “Simulating wood-framing wall panel’s production with timed coloured petri nets.” *Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019*, (Isarc): 1026–1033.

Correa, F. R. 2016. “Robot-oriented design for production in the context of building information modeling.” *ISARC 2016 - 33rd Int. Symp. Autom. Robot. Constr.*, (February): 853–861. https://doi.org/10.22260/isarc2016/0103.

CPWR. 2018. “The U.S. Construction Industry and Its Workers.” Accessed March 22, 2022. https://www.cpwr.com/sites/default/files/publications/The_6th_Edition_Construction_eChart_Book.pdf.

Cubek, R., W. Ertel, and G. Palm. 2015. “High-level learning from demonstration with conceptual spaces and subspace clustering.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 2015-June (June): 2592–2597. https://doi.org/10.1109/ICRA.2015.7139548.

Dang, H., and P. K. Allen. 2010. “Robot learning of everyday object manipulations via human demonstration.” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, 1284–1289. IEEE. https://doi.org/10.1109/IROS.2010.5651244.

David, O., F. X. Russotto, M. Da Silva Simoes, and Y. Measson. 2014. “Collision avoidance, virtual guides and advanced supervisory control teleoperation techniques for high-tech construction: Framework design.” *Autom. Constr.*, 44: 63–72. Elsevier B.V. https://doi.org/10.1016/j.autcon.2014.03.020.

Davids, A. 2002. “Urban Search and Rescue Robots: From Tragedy to Technology.” *IEEE Intell. Syst.*, 17 (2): 81–83. https://doi.org/10.1109/MIS.2002.999224.

Davila Delgado, J. M., L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, and H. Owolabi. 2019. “Robotics and automated systems in construction: Understanding industry-specific challenges for adoption.” *J. Build. Eng.*, 26: 100868. https://doi.org/10.1016/j.jobe.2019.100868.

Davtalab, O., A. Kazemian, and B. Khoshnevis. 2018. “Perspectives on a BIM-integrated

software platform for robotic construction through Contour Crafting." *Autom. Constr.*, 89 (September 2017): 13–23. Elsevier. https://doi.org/10.1016/j.autcon.2018.01.006.

Dawod, M., and S. Hanna. 2019. "BIM-assisted object recognition for the on-site autonomous robotic assembly of discrete structures." *Constr. Robot. 2019 31*, 3 (1): 69–81. Springer. https://doi.org/10.1007/S41693-019-00021-9.

Deng, M., C. C. Menassa, and V. R. Kamat. 2021a. "From BIM to digital twins: A systematic review of the evolution of intelligent building representations in the AEC-FM industry." *J. Inf. Technol. Constr.*, 26: 58–83. https://doi.org/10.36680/J.ITCON.2021.005.

Deng, M., Y. Tan, J. Singh, A. Joneja, and J. C. P. Cheng. 2021b. "A BIM-based framework for automated generation of fabrication drawings for façade panels." *Comput. Ind.*, 126: 103395. Elsevier. https://doi.org/10.1016/J.COMPIND.2021.103395.

Deng, M., X. Wang, D. Li, and C. C. Menassa. 2022. "Digital ID framework for human-centric monitoring and control of smart buildings." *Build. Simul. 2022 1510*, 15 (10): 1709–1728. Springer. https://doi.org/10.1007/S12273-022-0902-3.

Devadass, P., S. Stumm, and S. Brell-Cokcan. 2019. "Adaptive haptically informed assembly with mobile robots in unstructured environments." *Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019*, (Isarc): 469–476. https://doi.org/10.22260/isarc2019/0063.

Dijkstra, E. W. 1959. "A note on two problems in connexion with graphs." *Numer. Math. 1959 11*, 1 (1): 269–271. Springer. https://doi.org/10.1007/BF01386390.

Ding, L., W. Jiang, Y. Zhou, C. Zhou, and S. Liu. 2020. "BIM-based task-level planning for robotic brick assembly through image-based 3D modeling." *Adv. Eng. Informatics*, 43: 100993. Elsevier. https://doi.org/10.1016/J.AEI.2019.100993.

Dörries, C., and S. Zahradnik. 2016. *Container and Modular Buildings: Construction and Design Manual*. Berlin: DOM Publishers.

Dröder, K., P. Bobka, T. Germann, F. Gabriel, and F. Dietrich. 2018. "A machine learning-enhanced digital twin approach for human-robot-collaboration." *Procedia CIRP*, 76: 187–192. Elsevier B.V. https://doi.org/10.1016/j.procir.2018.02.010.

Du, G., P. Zhang, J. Mai, and Z. Li. 2012. "Markerless Kinect-based hand tracking for robot teleoperation." *Int. J. Adv. Robot. Syst.*, 9: 1–10. https://doi.org/10.5772/50093.

Du, J., Z. Zou, Y. Shi, and D. Zhao. 2018. "Zero latency: Real-time synchronization of BIM data in virtual reality for collaborative decision-making." *Autom. Constr.*, 85: 51–64. Elsevier B.V. https://doi.org/10.1016/j.autcon.2017.10.009.

Dusty Robotics. 2021. "Build Better with BIM-Driven Layout." Accessed March 22, 2022. https://www.dustyrobotics.com/.

ENR. 2013. "Jobsite Robot Shows Promise in Speed and Accuracy." Accessed March 22, 2022.

https://www.enr.com/articles/8591-jobsite-robot-shows-promise-in-speed-and-accuracy.

ENR. 2020. “Construction Loses 975,000 Jobs in April, Due to COVID-19 Impacts.” Accessed March 22, 2022. https://www.enr.com/articles/49333-construction-loses-975000-jobs-in-april-due-to-covid-19-impacts.

Everett, J. G. 1991. “Construction Automation: Basic Task Selection an Development of the Cranium.” Massachusetts Institute of Technology.

Eversmann, P., F. Gramazio, and M. Kohler. 2017. “Robotic prefabrication of timber structures: towards automated large-scale spatial assembly.” *Constr. Robot.*, 1 (1–4): 49–60. Springer International Publishing. https://doi.org/10.1007/s41693-017-0006-2.

Fang, H. C., S. K. Ong, and A. Y. C. Nee. 2012. “Interactive robot trajectory planning and simulation using augmented reality.” *Robot. Comput. Integr. Manuf.*, 28 (2): 227–237. Elsevier. https://doi.org/10.1016/j.rcim.2011.09.003.

Fang, Y., Y. K. Cho, and J. Chen. 2016. “A framework for real-time pro-active safety assistance for mobile crane lifting operations.” *Autom. Constr.*, 72: 367–379. Elsevier B.V. https://doi.org/10.1016/j.autcon.2016.08.025.

Feng, C. 2015. “Camera marker networks for pose estimation and scene understanding in construction automation and robotics.”

Feng, C., Y. Xiao, A. Willette, W. McGee, and V. R. Kamat. 2015. “Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites.” *Autom. Constr.*, 59: 128–138. Elsevier B.V. https://doi.org/10.1016/j.autcon.2015.06.002.

Figueroa, N., A. L. P. Ureche, and A. Billard. 2016. “Learning complex sequential tasks from demonstration: A pizza dough rolling case study.” *ACM/IEEE Int. Conf. Human-Robot Interact.*, 2016-April: 611–612. IEEE. https://doi.org/10.1109/HRI.2016.7451881.

Fikes, R. E., and N. J. Nilsson. 1971. “Strips: A new approach to the application of theorem proving to problem solving.” *Artif. Intell.*, 2 (3–4): 189–208. Elsevier. https://doi.org/10.1016/0004-3702(71)90010-5.

Follini, C., V. Magnago, K. Freitag, M. Terzer, C. Marcher, M. Riedl, A. Giusti, and D. T. Matt. 2020. “BIM-Integrated Collaborative Robotics for Application in Building Construction and Maintenance.” *Robot. 2021, Vol. 10, Page 2*, 10 (1): 2. Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/ROBOTICS10010002.

Frank, J. A., A. Brill, and V. Kapila. 2016. “Interactive mobile interface with augmented reality for learning digital control concepts.” *2016 Indian Control Conf. ICC 2016 - Proc.*, 85–92. IEEE.

French, K., S. Wu, T. Pan, Z. Zhou, and O. C. Jenkins. 2019. “Learning behavior trees from demonstration.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 2019-May: 7791–7797. https://doi.org/10.1109/ICRA.2019.8794104.

Gan, V. J. L., M. Deng, Y. Tan, W. Chen, and J. C. P. Cheng. 2019. “BIM-based framework to analyze the effect of natural ventilation on thermal comfort and energy performance in buildings.” *Energy Procedia*, 158: 3319–3324. Elsevier. https://doi.org/10.1016/J.EGYPRO.2019.01.971.

García de Soto, B., I. Agustí-Juan, J. Hunhevicz, S. Joss, K. Graser, G. Habert, and B. T. Adey. 2018. “Productivity of digital fabrication in construction: Cost and time analysis of a robotically built wall.” *Autom. Constr.*, 92 (March): 297–311. Elsevier. https://doi.org/10.1016/j.autcon.2018.04.004.

Ge, L., and F. Kuester. 2014. “Integrative Simulation Environment for Conceptual Structural Analysis.” *J. Comput. Civ. Eng.*, 29 (4): B4014004. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000405.

Gil, M. S., M. S. Kang, S. Lee, H. D. Lee, K. S. Shin, J. Y. Lee, and C. S. Han. 2013. “Installation of heavy duty glass using an intuitive manipulation device.” *Autom. Constr.*, 35: 579–586. Elsevier B.V. https://doi.org/10.1016/j.autcon.2013.01.008.

Grieves, M. 2014. “Digital Twin : Manufacturing Excellence through Virtual Factory Replication.” *A Whitepaper*, (November): 9.

El Hafi, L., S. Isobe, Y. Tabuchi, Y. Katsumata, H. Nakamura, T. Fukui, T. Matsuo, G. A. Garcia Ricardez, M. Yamamoto, A. Taniguchi, Y. Hagiwara, and T. Taniguchi. 2020. “System for augmented human–robot interaction through mixed reality and robot training by non-experts in customer service environments.” *Adv. Robot.*, 34 (3–4): 157–172. Robotics Society of Japan. https://doi.org/10.1080/01691864.2019.1694068.

Han, B., S. M. Asce, F. Leite, and M. Asce. 2021. “Measuring the Impact of Immersive Virtual Reality on Construction Design Review Applications: Head-Mounted Display versus Desktop Monitor.” *J. Constr. Eng. Manag.*, 147 (6): 04021042. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002056.

Harris, R. B. (Robert B. 1978. *Precedence and arrow networking techniques for construction*. Wiley.

Hashimoto, S., A. Ishida, M. Inami, and T. Igarash. 2011. “TouchMe: An Augmented Reality Based Remote Robot Manipulation.” *21st Int. Conf. Artif. Real. Telexistence*, 1–6.

Hayes, B., and B. Scassellati. 2016. “Autonomously constructing hierarchical task networks for planning and human-robot collaboration.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 2016-June: 5469–5476. https://doi.org/10.1109/ICRA.2016.7487760.

Hirabayashi, T., J. Akizono, T. Yamamoto, H. Sakai, and H. Yano. 2006. “Teleoperation of construction machines with haptic information for underwater applications.” *Autom. Constr.*, 15 (5): 563–570. Elsevier. https://doi.org/10.1016/J.AUTCON.2005.07.008.

Hornung, A., K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. 2013. “OctoMap: An efficient probabilistic 3D mapping framework based on octrees.” *Auton. Robots*, 34 (3):

189–206. Springer. https://doi.org/10.1007/s10514-012-9321-0.

ISO/IEC/IEEE. 2017. “15939-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering--Measurement process.” *IEEE*, 2017 (IEEE). IEEE.

Jen, Y. H., Z. Taha, and L. J. Vui. 2008. “VR-Based robot programming and simulation system for an industrial robot.” *Int. J. Ind. Eng. Theory Appl. Pract.*, 15 (3): 314–322.

Jung, K., B. Chu, S. Park, and D. Hong. 2013. “An implementation of a teleoperation system for robotic beam assembly in construction.” *Int. J. Precis. Eng. Manuf.*, 14 (3): 351–358. https://doi.org/10.1007/s12541-013-0049-3.

Kent, D., S. Banerjee, and S. Chernova. 2019. “Learning Sequential Decision Tasks for Robot Manipulation with Abstract Markov Decision Processes and Demonstration-Guided Exploration.” *IEEE-RAS Int. Conf. Humanoid Robot.*, 2018-Novem: 958–965. https://doi.org/10.1109/HUMANOIDS.2018.8624949.

Kim, B., C. Kim, and H. Kim. 2012. “Interactive Modeler for Construction Equipment Operation Using Augmented Reality.” *J. Comput. Civ. Eng.*, 26 (3): 331–341. https://doi.org/10.1061/(asce)cp.1943-5487.0000137.

Kim, D., J. Kim, K. Lee, C. Park, J. Song, and D. Kang. 2009. “Excavator tele-operation system using a human arm.” *Autom. Constr.*, 18 (2): 173–182. Elsevier B.V. https://doi.org/10.1016/j.autcon.2008.07.002.

Kim, J., S. You, S. Lee, V. R. Kamat, and L. P. Robert. 2015. “Evaluation of Human Robot Collaboration in Masonry Work Using Immersive Virtual Environments Need for IVE for Advancing HRC in construction.” *Int. Conf. Constr. Appl. Virtual Real.*, 1–8.

Kim, K., J. W. Park, and C. Cho. 2020. “Framework for Automated Generation of Constructible Steel Erection Sequences Using Structural Information of Static Indeterminacy Variation in BIM.” *KSCE J. Civ. Eng. 2020 2411*, 24 (11): 3169–3178. Springer. https://doi.org/10.1007/S12205-020-0163-6.

Kim, S., M. Peavy, P.-C. C. Huang, and K. Kim. 2021a. “Development of BIM-integrated construction robot task planning and simulation system.” *Autom. Constr.*, 127 (April): 103720. Elsevier B.V. https://doi.org/10.1016/j.autcon.2021.103720.

Kim, S., M. Peavy, P. C. Huang, and K. Kim. 2021b. “Development of BIM-integrated construction robot task planning and simulation system.” *Autom. Constr.*, 127: 103720. Elsevier. https://doi.org/10.1016/J.AUTCON.2021.103720.

Kisi, K. P., N. Mani, E. M. Rojas, and E. T. Foster. 2017. “Optimal Productivity in Labor-Intensive Construction Operations: Pilot Study.” *J. Constr. Eng. Manag.*, 143 (3): 04016107. American Society of Civil Engineers (ASCE). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001257.

Koenig, N., and A. Howard. 2004. “Design and use paradigms for Gazebo, an open-source

multi-robot simulator.” *2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2149–2154.

Kramberger, A., A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, and A. Ude. 2017. “Generalization of orientation trajectories and force-torque profiles for robotic assembly.” *Rob. Auton. Syst.*, 98: 333–346. North-Holland. https://doi.org/10.1016/J.ROBOT.2017.09.019.

Kubik, T., and M. Sugisaka. 2001. “Use of a cellular phone in mobile robot voice control.” *Proc. SICE Annu. Conf.*, 106–111. https://doi.org/10.1109/sice.2001.977815.

Kulić, D., C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. 2012. “Incremental learning of full body motion primitives and their sequencing through human motion observation.” *Int. J. Rob. Res.*, 31 (3): 330–345. https://doi.org/10.1177/0278364911426178.

Kurien, M., M. K. Kim, M. Kopsida, and I. Brilakis. 2018. “Real-time simulation of construction workers using combined human body and hand tracking for robotic construction worker system.” *Autom. Constr.*, 86 (September 2017): 125–137. Elsevier. https://doi.org/10.1016/j.autcon.2017.11.005.

Kuts, V., T. Otto, T. Tahemaa, Y. Bondarenko, and E. Equilibrium. 2019. “Digital Twin Based Synchronised Control and Simulation of the Industrial Robotic Cell Using Virtual Reality.” *J. Mach. Eng.*, 19 (1): 128–145. https://doi.org/10.13799/j.cnki.mdjyxyxb.2015.03.003.

Kyjanek, O., B. Al Bahar, L. Vasey, B. Wannemacher, and A. Menges. 2019. “Implementation of an augmented reality AR workflow for human robot collaboration in timber prefabrication.” *Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019*, (Isarc): 1223–1230. https://doi.org/10.22260/isarc2019/0164.

Lawson, R. M., R. G. Ogden, and R. Bergin. 2012. “Application of Modular Construction in High-Rise Buildings.” *J. Archit. Eng.*, 18 (2): 148–154. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000057.

Lee, B. G., B. Choi, H. Jebelli, and S. H. Lee. 2021. “Assessment of construction workers’ perceived risk using physiological data from wearable sensors: A machine learning approach.” *J. Build. Eng.*, 42: 102824. Elsevier. https://doi.org/10.1016/J.JOBE.2021.102824.

Lee, S., S. Lee, S. Yu, and C. Han. 2007. “Intuitive OCU (Operator Control Unit) of MFR (Multipurpose Field Robot) on construction site.” *Autom. Robot. Constr. - Proc. 24th Int. Symp. Autom. Robot. Constr.*, 259–264. https://doi.org/10.22260/isarc2007/0046.

Lee, S., and J. Il Moon. 2014. “Introduction of human-robot cooperation technology at construction sites.” *31st Int. Symp. Autom. Robot. Constr. Mining, ISARC 2014 - Proc.*, (Isarc): 978–983. https://doi.org/10.22260/isarc2014/0134.

Levine, S., P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. 2017. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection:” *Int. J. Robot. Res.*, 37 (4–5): 421–436. SAGE PublicationsSage UK: London, England.

https://doi.org/10.1177/0278364917710318.

Li, X., A. Qureshi, and M. Al-Hussein. 2017. “Developing a BIM-Based Integrated Model for CAD to CAM Production Automation.” *34th Int. Symp. Autom. Robot. Constr.*, 51–58. Waterloo: IAARC.

Li, X., W. Yi, H. L. Chi, X. Wang, and A. P. C. Chan. 2018. “A critical review of virtual and augmented reality (VR/AR) applications in construction safety.” *Autom. Constr.*, 86: 150–162. Elsevier B.V. https://doi.org/10.1016/j.autcon.2017.11.003.

Liang, C.-J., V. R. Kamat, C. C. Menassa, and W. Mcgee. 2021a. “Trajectory-Based Skill Learning for Overhead Construction Robots Using Generalized Cylinders with Orientation.” *J. Comput. Civ. Eng.*, 36 (2): 04021036. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001004.

Liang, C.-J., W. Mcgee, C. C. Menassa, and V. R. Kamat. 2022. “Real-time state synchronization between physical construction robots and process-level digital twins.” *Constr. Robot.* https://doi.org/10.1007/s41693-022-00068-1.

Liang, C.-J., W. McGee, C. Menassa, and V. Kamat. 2020a. “Bi-Directional Communication Bridge for State Synchronization between Digital Twin Simulations and Physical Construction Robots.” *Proc. 37th Int. Symp. Autom. Robot. Constr.*, 1480–1487. International Association for Automation and Robotics in Construction (IAARC).

Liang, C.-J., ; Xi Wang, V. R. Kamat, and C. C. Menassa. 2021b. “Human–Robot Collaboration in Construction: Classification and Research Trends.” *J. Constr. Eng. Manag.*, 147 (10): 03121006. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002154.

Liang, C. J., V. R. Kamat, and C. C. Menassa. 2019. “Teaching robots to perform construction tasks via learning from demonstration.” *Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019*, 1305–1311. International Association for Automation and Robotics in Construction I.A.A.R.C).

Liang, C. J., V. R. Kamat, and C. C. Menassa. 2020b. “Teaching robots to perform quasi-repetitive construction tasks through human demonstration.” *Autom. Constr.*, 120: 103370. Elsevier B.V. https://doi.org/10.1016/j.autcon.2020.103370.

Lin, Y. C., and D. Berenson. 2016. “Using previous experience for humanoid navigation planning.” *IEEE-RAS Int. Conf. Humanoid Robot.*, 794–801. IEEE Computer Society.

Lioutikov, R., G. Neumann, G. Maeda, and J. Peters. 2015. “Probabilistic segmentation applied to an assembly task.” *IEEE-RAS Int. Conf. Humanoid Robot.*, 2015-Decem: 533–540. IEEE. https://doi.org/10.1109/HUMANOIDS.2015.7363584.

Liu, C., S. Shirowzhan, S. M. E. Sepasgozar, and A. Kaboli. 2019. “Evaluation of Classical Operators and Fuzzy Logic Algorithms for Edge Detection of Panels at Exterior Cladding of Buildings.” *Buildings*, 9 (2): 40. MDPI AG. https://doi.org/10.3390/buildings9020040.

Liu, M., S. Han, and S. Lee. 2017. “Potential of Convolutional Neural Network-Based 2D Human Pose Estimation for On-Site Activity Analysis of Construction Workers.” *Comput. Civ. Eng. 2017*, 141–149. American Society of Civil Engineers (ASCE).

Liu, Y., F. Castronovo, J. Messner, and R. Leicht. 2020. “Evaluating the Impact of Virtual Reality on Design Review Meetings.” *J. Comput. Civ. Eng.*, 34 (1): 04019045. American Society of Civil Engineers (ASCE). https://doi.org/10.1061/(asce)cp.1943-5487.0000856.

Lundeen, K. M. 2019. “Autonomous Scene Understanding, Motion Planning, and Task Execution for Geometrically Adaptive Robotized Construction Work.” University of Michigan.

Lundeen, K. M., V. R. Kamat, C. C. Menassa, and W. McGee. 2017. “Scene understanding for adaptive manipulation in robotized construction work.” *Autom. Constr.*, 82 (November 2016): 16–30. Elsevier. https://doi.org/10.1016/j.autcon.2017.06.022.

Lundeen, K. M., V. R. Kamat, C. C. Menassa, and W. McGee. 2019. “Autonomous motion planning and task execution in geometrically adaptive robotized construction work.” *Autom. Constr.*, 100 (October 2018): 24–45. Elsevier. https://doi.org/10.1016/j.autcon.2018.12.020.

Malik, A. A., and A. Brem. 2021. “Digital twins for collaborative robots: A case study in human-robot interaction.” *Robot. Comput. Integr. Manuf.*, 68 (September 2019): 102092. Elsevier Ltd. https://doi.org/10.1016/j.rcim.2020.102092.

Mantha, B. R. K., C. C. Menassa, V. R. Kamat, and C. R. D’Souza. 2020. “Evaluation of Preference- and Constraint-Sensitive Path Planning for Assisted Navigation in Indoor Building Environments.” *J. Comput. Civ. Eng.*, 34 (1): 04019050. https://doi.org/10.1061/(asce)cp.1943-5487.0000865.

Maragkos, C., G. C. Vosniakos, and E. Matsas. 2019. “Virtual reality assisted robot programming for human collaboration.” *Procedia Manuf.*, 38 (2019): 1697–1704. Elsevier B.V. https://doi.org/10.1016/j.promfg.2020.01.109.

Martinez, S., A. Jardon, J. M. Maria, and P. Gonzalez. 2008. “Building industrialization: Robotized assembly of modular products.” *Assem. Autom.*, 28 (2): 134–142. Emerald Group Publishing Limited. https://doi.org/10.1108/01445150810863716/FULL/PDF.

Marton, Z. C., R. B. Rusu, and M. Beetz. 2009. “On fast surface reconstruction methods for large and noisy point clouds.” *2009 IEEE Int. Conf. Robot. Autom.*, 3218–3223. Institute of Electrical and Electronics Engineers (IEEE).

McClymonds, A., S. Asadi, A. Wagner, and R. M. Leicht. 2022. “Information Exchange for Supporting BIM to Robotic Construction.” *Constr. Res. Congr. 2022*, 839–848. Reston, VA: American Society of Civil Engineers.

Mckinsey Global Institute. 2017. “Reinventing construction through a productivity revolution | McKinsey.” Accessed March 22, 2022. https://www.mckinsey.com/business-functions/operations/our-insights/reinventing-construction-through-a-productivity-

revolution.

McNeel. 2022. “Rhino - Rhino.Inside.” Accessed March 22, 2022. https://www.rhino3d.com/features/rhino-inside/.

Meschini, S., K. Iturralde, T. Linner, and T. Bock. 2016. “Novel applications offered by Integration of Robotic Tools in BIM-based Design Workflow for Automation in Construction Processes.” *CIB* IAARC W199 CIC 2016 Work.*

Mewis, R. W. 2019. *Building construction costs with RSMeans data, 2020*. Gordian.

Milberg, C. T. 2006. “Application of Tolerance Management to Civil Systems.”

Mitsi, S., K. D. Bouzakis, G. Mansour, D. Sagris, and G. Maliaris. 2005. “Off-line programming of an industrial robot for manufacturing.” *Int. J. Adv. Manuf. Technol.*, 26 (3): 262–267. Springer. https://doi.org/10.1007/S00170-003-1728-5/FIGURES/9.

Mohseni-Kabir, A., C. Rich, S. Chernova, C. L. Sidner, and D. Miller. 2015. “Interactive Hierarchical Task Learning from a Single Demonstration.” *ACM/IEEE Int. Conf. Human-Robot Interact.*, 2015-March: 205–212. https://doi.org/10.1145/2696454.2696474.

Murali, P. K., K. Darvish, and F. Mastrogiovanni. 2020. “Deployment and evaluation of a flexible human–robot collaboration model based on AND/OR graphs in a manufacturing environment.” *Intell. Serv. Robot.*, 13 (4): 439–457. Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/s11370-020-00332-9.

NAHB. 2011. *Residential construction performance guidelines for professional builders & remodelers*. NAHB.

Nasiriany, S., H. Liu, and Y. Zhu. 2022. “Augmenting Reinforcement Learning with Behavior Primitives for Diverse Manipulation Tasks.” *IEEE Int. Conf. Robot. Autom.*

Navon, R. 2000. “Automated Quality Assurance for a Floor-Tiling Robot.” *Proc. 4th Int. Conf. Expo. Robot. Challenging Situations Environ. - Robot. 2000*, 299: 320–327. American Society of Civil Engineers. https://doi.org/10.1061/40476(299)42.

Navon, R., and A. Retik. 1997. “Programming construction robots using virtual reality techniques.” *Autom. Constr.*, 5 (5): 393–406. https://doi.org/10.1016/S0926-5805(96)00162-8.

NIBS. 2015. “National BIM Standard-United States® Version 3 | National BIM Standard - United States.” Accessed March 22, 2022. https://www.nationalbimstandard.org/nbims-us.

Nicholas Flann, C. S., R. W. Gunderson, K. L. Moore, and C. G. Wood. 2000. “Intelligent mobility through omnidirectional vehicles: a research program.” *https://doi.org/10.1117/12.391633*, 4024 (10): 228–240. SPIE. https://doi.org/10.1117/12.391633.

Niekum, S., S. Osentoski, G. Konidaris, and A. G. Barto. 2012. “Learning and generalization of complex tasks from unstructured demonstrations.” *IEEE Int. Conf. Intell. Robot. Syst.*, 5239–5246. IEEE. https://doi.org/10.1109/IROS.2012.6386006.

Ong, S. K., A. W. W. Yew, N. K. Thanigaivel, and A. Y. C. Nee. 2020. “Augmented reality-assisted robot programming system for industrial applications.” *Robot. Comput. Integr. Manuf.*, 61: 101820. Elsevier Ltd. https://doi.org/10.1016/j.rcim.2019.101820.

Open Robotics. 2020. “camera_calibration - ROS Wiki.” Accessed March 22, 2022. http://wiki.ros.org/camera_calibration.

Open Robotics. 2021. “gazebo_ros_pkgs - ROS Wiki.” Accessed March 22, 2022. http://wiki.ros.org/gazebo_ros_pkgs.

OSHA. 2021. “OSHA Technical Manual (OTM) - Section IV: Chapter 4 Industrial Robot Systems and Industrial Robot System Safety.” Accessed March 22, 2022. https://www.osha.gov/otm/section-4-safety-hazards/chapter-4#applications.

Van Otterlo, M., and M. Wiering. 2012. “Reinforcement Learning and Markov Decision Processes.” *Adapt. Learn. Optim.*, 12: 3–42. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27645-3_1.

Pan, J., S. Chitta, and D. Manocha. 2012. “FCL: A general purpose library for collision and proximity queries.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 3859–3866. Institute of Electrical and Electronics Engineers Inc.

Paraschos, A., C. Daniel, J. Peters, and G. Neumann. 2013. “Probabilistic movement primitives.” *Adv. Neural Inf. Process. Syst.*, 1–9.

Pardowitz, M., S. Knoop, R. Dillmann, and R. D. Zollner. 2007. “Incremental learning of tasks from user demonstrations, past experiences, and vocal comments.” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, 37 (2): 322–332. https://doi.org/10.1109/TSMCB.2006.886951.

Park, J., Y. K. Cho, and D. Martinez. 2016. “A BIM and UWB integrated Mobile Robot Navigation System for Indoor Position Tracking Applications.” *J. Constr. Eng. Proj. Manag.*, 6 (2): 30–39. Korea Institute of Construction Engineering and Management. https://doi.org/10.6106/JCEPM.2016.6.2.030.

Patra, A. K., and S. J. Ray. 2007. “Guiding robots using mobile phone.” *ISARC- Proc. 24th Int. Symp. Autom. Robot. Constr.*, 339–344.

Pertsch, K., Y. Lee, and J. J. Lim. 2020. “Accelerating Reinforcement Learning with Learned Skill Priors.” *4th Conf. Robot Learn.* Cambridge MA, USA.

Peternel, L., T. Petrič, and J. Babič. 2018. “Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation.” *Auton. Robots*, 42 (1): 1–17. Springer Science and Business Media, LLC. https://doi.org/10.1007/S10514-017-9635-Z/FIGURES/11.

PickNik Robotics. 2022. “MoveIt Tutorial - Hand-Eye Calibration.” Accessed March 22, 2022. https://ros-planning.github.io/moveit_tutorials/doc/hand_eye_calibration/hand_eye_calibration_tutorial.html.

Pietroforte, R., and A. Member. 1997. “Shop Drawing Process of Stone Veneered Cladding Systems.” *J. Archit. Eng.*, 3 (2): 70–79. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)1076-0431(1997)3:2(70).

Qian, K., A. Song, J. Bao, and H. Zhang. 2012. “Small Teleoperated Robot for Nuclear Radiation and Chemical Leak Detection:” *https://doi.org/10.5772/50720*, 9. SAGE PublicationsSage UK: London, England. https://doi.org/10.5772/50720.

Quigley, M., B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. 2009. “ROS: an open-source Robot Operating System.” *ICRA Work. open source Softw.*

Quintero, C. P., R. Tatsambon, M. Gridseth, and M. Jagersand. 2015. “Visual pointing gestures for bi-directional human robot interaction in a pick-and-place task.” *Proc. - IEEE Int. Work. Robot Hum. Interact. Commun.*, 349–354. Institute of Electrical and Electronics Engineers Inc.

Ravichandar, H., A. S. Polydoros, S. Chernova, and A. Billard. 2020. “Recent Advances in Robot Learning from Demonstration.” *Annu. Rev. Control. Robot. Auton. Syst.*, 3: 297–330. https://doi.org/10.1146/annurev-control-100819-063206.

Roldán, J. J., E. Peña-Tapia, D. Garzón-Ramos, J. de León, M. Garzón, J. del Cerro, and A. Barrientos. 2019. “Multi-robot Systems, Virtual Reality and ROS: Developing a New Generation of Operator Interfaces.” *Robot Oper. Syst. (ROS). Springer, Cham.*, 29–64. Springer Verlag.

Roldán, J. J., E. Peña-Tapia, A. Martín-Barrio, M. A. Olivares-Méndez, J. Del Cerro, and A. Barrientos. 2017. “Multi-Robot Interfaces and Operator Situational Awareness: Study of the Impact of Immersion and Prediction.” *Sensors*, 17 (8): 1720. https://doi.org/10.3390/s17081720.

Romanovskyi, R., L. Sanabria Mejia, and E. Rezazadeh Azar. 2019. “BIM-based decision support system for concrete formwork design.” *Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019*, 1129–1135. International Association for Automation and Robotics in Construction I.A.A.R.C). https://doi.org/10.22260/ISARC2019/0150.

Rosenstein, M. T., and A. G. Barto. 2004. *Supervised Actor-Critic Reinforcement Learning*. *Learn. Approx. Dyn. Program. Scaling Up to Real World*.

Ruiz, J. J., A. Viguria, J. R. Martinez-De-Dios, and A. Ollero. 2015. “Immersive displays for building spatial knowledge in multi-UAV operations.” *2015 Int. Conf. Unmanned Aircr. Syst.*, 1043–1048. Institute of Electrical and Electronics Engineers Inc.

Scheutz, M., E. Krause, B. Oosterveld, T. Frasca, and R. Platt. 2017. “Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture.” *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, 3 (1): 1378–1386.

Seong, S., C. Lee, and J. Kim. 2019. “Intelligent Multi-Fingered Dexterous Hand Using Virtual Reality (VR) and Robot Operating System (ROS).” *Adv. Intell. Syst. Comput.*, 751 (January): 35–41. https://doi.org/10.1007/978-3-319-78452-6.

Sharif, M.-M., M. Nahangi, C. Haas, and J. West. 2017. “Automated Model-Based Finding of 3D Objects in Cluttered Construction Point Cloud Models.” *Comput. Civ. Infrastruct. Eng.*, 32 (11): 893–908. Blackwell Publishing Inc. https://doi.org/10.1111/mice.12306.

Sharif, S., T. R. Gentry, and L. M. Sweet. 2016. “Human-robot collaboration for creative and integrated design and fabrication processes.” *ISARC 2016 - 33rd Int. Symp. Autom. Robot. Constr.*, 596–604.

She, L., S. Yang, Y. Cheng, Y. Jia, J. Y. Chai, and N. Xi. 2014. “Back to the blocks world: Learning new actions through situated human-robot dialogue.” *SIGDIAL 2014 - 15th Annu. Meet. Spec. Interes. Gr. Discourse Dialogue, Proc. Conf.*, (June): 89–97. https://doi.org/10.3115/v1/w14-4313.

Shum, H. P. H., E. S. L. Ho, Y. Jiang, and S. Takagi. 2013. “Real-time posture reconstruction for Microsoft Kinect.” *IEEE Trans. Cybern.*, 43 (5): 1357–1369. https://doi.org/10.1109/TCYB.2013.2275945.

Siemens. 2021. “GitHub - ROS#.” Accessed March 22, 2022. https://github.com/siemens/ros-sharp.

Singh, A., H. Liu, G. Zhou, A. Yu, N. Rhinehart, and S. Levine. 2020. “Parrot: Data-Driven Behavioral Priors for Reinforcement Learning.” *Int. Conf. Learn. Represent.*, 1–19.

Singh, J., M. Deng, and J. C. P. Cheng. 2018. “Implementation of mass customization for MEP layout design to reduce manufacturing cost in one-off projects.” *IGLC 2018 - Proc. 26th Annu. Conf. Int. Gr. Lean Constr. Evol. Lean Constr. Towar. Matur. Prod. Manag. Across Cult. Front.*, 1: 625–635. The International Group for Lean Construction. https://doi.org/10.24928/2018/0519.

Smith, R. E., M. N. Hamedani, and G. Griffin. 2018. “Developing Timber Volume Calculators Through a Comparative Case Study Analysis of Wood Utilization in On-Site and Off-Site Construction Methods.” *https://doi.org/10.1080/24751448.2018.1420965*, 2 (1): 55–67. Routledge. https://doi.org/10.1080/24751448.2018.1420965.

Sohn, S., H. Woo, J. Choi, and H. Lee. 2020. “Meta Reinforcement Learning with Autonomous Inference of Subtask Dependencies.” 1–29.

Son, H., C. Kim, and K. Choi. 2010. “Rapid 3D object detection and modeling using range data from 3D range imaging camera for heavy equipment operation.” *Autom. Constr.*, 19 (7): 898–906. Elsevier. https://doi.org/10.1016/j.autcon.2010.06.003.

Stojanovic, V., M. Trapp, R. Richter, B. Hagedorn, and J. Döllner. 2018. “Towards the generation of digital twins for facility management based on 3D point clouds.” *Proceeding 34th Annu. ARCOM Conf. ARCOM 2018*, (June 2019): 270–279.

Strudel, R., A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, and C. Schmid. 2020. “Learning to combine primitive skills: A step towards versatile robotic manipulation §.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 4637–4643. https://doi.org/10.1109/ICRA40945.2020.9196619.

Sucan, I. A., M. Moll, and L. E. Kavraki. 2012. “The Open Motion Planning Library.” *IEEE Robot. Autom. Mag.*, 19 (4): 72–82. https://doi.org/10.1109/MRA.2012.2205651.

Sukumar, D. K., S. Lee, C. Georgoulas, and T. Bock. 2015. “Augmented Reality-based Tele-robotic System Architecture for On-site Construction.” *ISARC. Proc. Int. Symp. Autom. Robot. Constr.*, 1–8.

Tavares, P., C. M. Costa, L. Rocha, P. Malaca, P. Costa, A. P. Moreira, A. Sousa, and G. Veiga. 2019. “Collaborative Welding System using BIM for Robotic Reprogramming and Spatial Augmented Reality.” *Autom. Constr.*, 106 (March): 102825. Elsevier. https://doi.org/10.1016/j.autcon.2019.04.020.

Teizer, J., A. Blickle, T. King, O. Leitzbach, D. Guenther, H. Mattern, M. König, J. Teizer, A. Blickle Ed, T. King, D. Guenther, O. Leitzbach, H. Mattern, and M. König. 2018. “BIM for 3D Printing in Construction.” *Build. Inf. Model. Technol. Found. Ind. Pract.*, 421–446. Springer, Cham. https://doi.org/10.1007/978-3-319-92862-3_26.

Thai, H. T., T. Ngo, and B. Uy. 2020. “A review on modular construction for high-rise buildings.” *Structures*, 28: 1265–1290. Elsevier. https://doi.org/10.1016/J.ISTRUC.2020.09.070.

UQO Cyberpsychology Lab. 2004. “Revised Presence Questionnaire 3.0.” Accessed August 8, 2020. http://w3.uqo.ca/cyberpsy/docs/qaires/pres /PQ_va.pdf.

Victores, J. G., S. Martínez, A. Jardón, and C. Balaguer. 2011. “Robot-aided tunnel inspection and maintenance system by vision and proximity sensor integration.” *Autom. Constr.*, 20 (5): 629–636. Elsevier B.V. https://doi.org/10.1016/j.autcon.2010.12.005.

Walker, M., H. Hedayati, J. Lee, and D. Szafir. 2018. “Communicating Robot Motion Intent with Augmented Reality.” *ACM/IEEE Int. Conf. Human-Robot Interact.*, 316–324. New York, NY, USA: IEEE Computer Society.

Wallén, J. 2008. *The history of the industrial robot*.

Wang, J., and E. Olson. 2016. “AprilTag 2: Efficient and robust fiducial detection.” *IEEE Int. Conf. Intell. Robot. Syst.*, 2016-November: 4193–4198. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IROS.2016.7759617.

Wang, Q., J. Guo, and M. K. Kim. 2019a. “An application oriented scan-to-bim framework.” *Remote Sens.*, 11 (3). https://doi.org/10.3390/rs11030365.

Wang, Q., Y. Tan, and Z. Mei. 2020a. “Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications.” *Arch. Comput. Methods Eng.*, 27 (2): 479–499. Springer. https://doi.org/10.1007/s11831-019-09320-4.

Wang, X., D. Li, C. C. Menassa, and V. R. Kamat. 2019b. “Investigating the effect of indoor thermal environment on occupants’ mental workload and task performance using electroencephalogram.” *Build. Environ.*, 158: 120–132. Pergamon. https://doi.org/10.1016/J.BUILDENV.2019.05.012.

Wang, X., C.-J. Liang, C. C. Menassa, and V. R. Kamat. 2021. “Interactive and Immersive Process-Level Digital Twin for Collaborative Human–Robot Construction Work.” *J. Comput. Civ. Eng.*, 35 (6): 04021023. American Society of Civil Engineers. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000988.

Wang, X., C.-J. Liang, C. Menassa, and V. Kamat. 2020b. “Real-Time Process-Level Digital Twin for Collaborative Human-Robot Construction Work.” *Proc. 37th Int. Symp. Autom. Robot. Constr.* International Association for Automation and Robotics in Construction (IAARC).

Wang, X. V., Z. Kemény, J. Váncza, and L. Wang. 2017. “Human–robot collaborative assembly in cyber-physical production: Classification framework and implementation.” *CIRP Ann. - Manuf. Technol.*, 66 (1): 5–8. https://doi.org/10.1016/j.cirp.2017.04.101.

Warszawski, A., and D. A. Sangrey. 1985. “Robotics in Building Construction.” *J. Constr. Eng. Manag.*, 111 (3): 260–280. https://doi.org/10.1061/(asce)0733-9364(1985)111:3(260).

Welle, B., J. Haymaker, and Z. Rogers. 2011. “ThermalOpt: A methodology for automated BIM-based multidisciplinary thermal simulation for use in optimization environments.” *Build. Simul. 2011 44*, 4 (4): 293–313. Springer. https://doi.org/10.1007/S12273-011-0052-5.

Whitney, D., E. Rosen, D. Ullman, E. Phillips, and S. Tellex. 2018. “ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots.” *IEEE Int. Conf. Intell. Robot. Syst.*, 5018–5025.

Wong Chong, O., and J. Zhang. 2021. “Logic representation and reasoning for automated BIM analysis to support automation in offsite construction.” *Autom. Constr.*, 129: 103756. Elsevier. https://doi.org/10.1016/J.AUTCON.2021.103756.

Wong Chong, O., J. Zhang, R. M. Voyles, and B.-C. Min. 2022. “BIM-based simulation of construction robotics in the assembly process of wood frames.” *Autom. Constr.*, 137: 104194. https://doi.org/10.1016/J.AUTCON.2022.104194.

Xu, D., S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese. 2018. “Neural Task Programming: Learning to Generalize Across Hierarchical Tasks.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 3795–3802. IEEE. https://doi.org/10.1109/ICRA.2018.8460689.

Xu, L., C. Feng, V. R. Kamat, and C. C. Menassa. 2019. “An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied

environments.” *Autom. Constr.*, 104: 230–245. Elsevier B.V. https://doi.org/10.1016/j.autcon.2019.04.011.

Xu, L., C. Feng, V. R. Kamat, and C. C. Menassa. 2020. “A scene-adaptive descriptor for visual SLAM-based locating applications in built environments.” *Autom. Constr.*, 112: 103067. Elsevier B.V. https://doi.org/10.1016/j.autcon.2019.103067.

Xu, X., and B. Garcia de Soto. 2020. “On-site Autonomous Construction Robots: A review of Research Areas, Technologies, and Suggestions for Advancement.” *Proc. 37th Int. Symp. Autom. Robot. Constr.* International Association for Automation and Robotics in Construction (IAARC).

Yang, C. H., and S. C. Kang. 2021. “Collision avoidance method for robotic modular home prefabrication.” *Autom. Constr.*, 130 (July): 103853. Elsevier B.V. https://doi.org/10.1016/j.autcon.2021.103853.

You, S., J. H. Kim, S. H. Lee, V. Kamat, and L. P. Robert. 2018. “Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments.” *Autom. Constr.*, 96 (September): 161–170. Elsevier. https://doi.org/10.1016/j.autcon.2018.09.008.

Yousefizadeh, S., J. de D. Flores Mendez, and T. Bak. 2019. “Trajectory adaptation for an impedance controlled cooperative robot according to an operator’s force.” *Autom. Constr.*, 103 (January): 213–220. Elsevier. https://doi.org/10.1016/j.autcon.2019.01.006.

Yu, Y. H., C. H. Yeh, T. T. Lee, P. Y. Chen, and Y. H. Shiau. 2014. “Chip-based real-time gesture tracking for construction robot’s guidance.” *31st Int. Symp. Autom. Robot. Constr. Mining, ISARC 2014 - Proc.*, 821–828.

Zeng, Z., Z. Zhou, Z. Sui, and O. C. Jenkins. 2018. “Semantic Robot Programming for Goal-Directed Manipulation in Cluttered Scenes.” *Proc. - IEEE Int. Conf. Robot. Autom.*, 7462–7469. Institute of Electrical and Electronics Engineers Inc.

Zhang, J., H. Luo, and J. Xu. 2022. “Towards fully BIM-enabled building automation and robotics: A perspective of lifecycle information flow.” *Comput. Ind.*, 135: 103570. Elsevier. https://doi.org/10.1016/J.COMPIND.2021.103570.

Zhang, N., T. Qi, and Y. Zhao. 2021. “Real-time learning and recognition of assembly activities based on virtual reality demonstration.” *Sensors*, 21 (18): 1–15. https://doi.org/10.3390/s21186201.

Zhang, Z., Y. Zhu, and S.-C. Zhu. 2020. “Graph-based Hierarchical Knowledge Representation for Robot Task Transfer from Virtual to Physical World.” *2020 IEEE/RSJ Int. Conf. Intell. Robot. Syst.* Las Vegas, NV, USA.

Zhou, T., Q. Zhu, and J. Du. 2020. “Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction.” *Adv. Eng. Informatics*, 46: 101170. Elsevier Ltd. https://doi.org/10.1016/j.aei.2020.101170.

Zhu, A., P. Pauwels, and B. De Vries. 2021. “Smart component-oriented method of construction robot coordination for prefabricated housing.” *Autom. Constr.*, 129: 103778. Elsevier. https://doi.org/10.1016/J.AUTCON.2021.103778.

Zhu, Z., and H. Hu. 2018. “Robot learning from demonstration in robotic assembly: A survey.” *Robotics*, 7 (2). https://doi.org/10.3390/robotics7020017.