Bi-Directional Communication Bridge for State Synchronization between Digital Twin Simulations and Physical Construction Robots

C. J. Liang[a], W. McGee[b,c], C. C. Menassa[a,c] and V. R. Kamat[a,c]

[a]Department of Civil and Environmental Engineering, University of Michigan, USA
[b]School of Architecture, University of Michigan, USA
[c]Robotics Institute, University of Michigan, USA

cjliang@umich.edu, wesmcgee@umich.edu, menassa@umich.edu, vkamat@umich.edu

Abstract -

Collaborative robot (co-robots) are being increasingly deployed on construction sites to assist human workers with physically demanding work tasks. However, due to inherent safety and trust-related concerns, human-robot collaborative work is subject to strict safety standards that require robot motion and forces to be sensitive to proximate human workers. Robot simulations in online digital twins can be used to extend designed construction models, such as BIM, to the construction phase for real-time monitoring of robot motion planning and control. Robots plan work tasks and execute them in the digital twin simulations allowing humans to review and approve robot trajectories. Once approved, commands can be sent to the physical robots to perform the tasks. This paper discusses the development of a system to bridge robot simulations and physical robots in construction and digital fabrication. The Robot Operating System (ROS) is leveraged as the primary framework for bi-directional communication and Gazebo is used for robot simulations. The virtual robots in Gazebo receive work tasks from a BIM model to plan their trajectories, and then send the commands to the physical robots for execution. The system is implemented with a digital fabrication case study with a full-scale mobile KUKA KR120 six-degrees-of-freedom robotic arm mounted on a track system for an additional degree-of-freedom, and evaluated by comparing the pose between the physical robot and the virtual robot. The results show a high accuracy of the pose synchronization between two robots, which provide the opportunity for further deploying to real construction sites.

Keywords -

Digital Twin; Co-robots; Robot Operating System; Human-robot Collaboration

1 Introduction

Due to the 3D characteristics of construction work (dull, dirty, and dangerous) [1], construction sites can be hazardous and harmful working environments for human workers. The construction industry ranks the highest in occupational injuries and fatalities across all U.S. industries [2]. Robots deployment on construction sites can help relieve these issues [3]. For instance, the construction robot can group with human workers on job-site to assist with physically demanding tasks, while human workers focus on the work process plan and decision-making [4]. However, such human-robot collaborative work suffers from safety and trust-related concerns [5, 6], and is subject to strict safety standards [7]. For example, the robot must be restricted for speed and force while collaborating with nearby human workers. A real-time human and robot tracking system can ensure safety by providing the information of the robot state to human workers [8].

The Digital Twin (DT) offers opportunities to virtually mimic the conditions of the physical (real) environment in allowing for a cyber-physical system (CPS) [9] where information of the current and forecasted future states of the robot can be displayed [5]. Figure 1 shows the physical robotic arm and its Digital Twin. Madni et al. [10] defined four levels of Digital Twin (Pre-Digital Twin, Digital Twin, Adaptive Digital Twin, and Intelligent Digital Twin) based on the level of intelligence. The Adaptive Digital Twin combines user interface and machine learning with normal DT, whereas the Intelligent Digital Twin further utilizes reinforcement learning to process the state in a partially observed and uncertain environment.

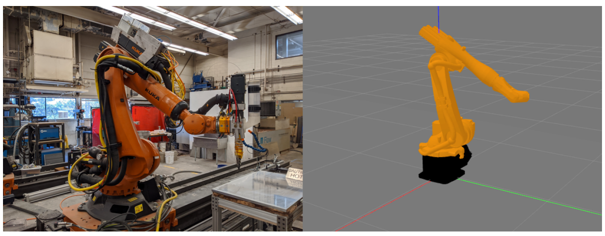

Figure 1. The physical robotic arm (left) and its Digital Twin (right).

One of the major aspects of the DT is the synchronized model [11]. The DT first constructs the virtual model based on the physical environment, then records and tracks the changes in the physical environment and reflects them in the virtual model. The virtual model can be extracted from the designed construction model such as BIM or scanned 3D point cloud of the as-built environment [12, 13]. On the other hand, a communication mechanism is required to synchronize the data between the physical environment and the virtual model [9]. The communication is bi-directional so that the virtual model can reflect the changes of the physical environment, and the user can determine the next steps in the virtual model and send the command to the physical environment.

To address the issue of human-robot collaboration in construction work, we develop an online Digital Twin system to bridge the virtual robot and physical robot in construction and digital fabrication. We utilize Robot Operating System (ROS) [14] to construct the framework of the system and create a robotic arm model representing the physical robotic arm in Gazebo simulation environment [15]. In terms of bi-directional communication, we use MQTT [16] to connect the virtual robotic arm with the physical robotic arm. The mechanism of checking the synchronization between the physical robotic arm and the virtual twin is also developed. The proposed framework can be adapted to any robotic arm models reflecting physical robots. We implement the system in a fabrication laboratory with a full-scale mobile KUKA KR120 six-degrees-of-freedom robotic arm, and evaluate by comparing the pose of the physical robotic arm with the virtual robotic arm.

2 Related Work

Digital modeling methods, such as 3D visualization or BIM, are used in the construction industry for design, management, and operation throughout the building life cycle [17, 18]. These modeling methods document the project information and provide a platform for stakeholders to record changes, collaborate and resolve conflicts [19, 20]. In order to achieve a high-quality collaboration, the model must be fully synchronized with the physical environment. It is time and cost prohibitive to manually update the model [21]. Thus, existing research focuses on automatically generating and updating the 3D model [22]. Collecting the 3D point cloud is one of the methods for generating the 3D model of the indoor environment [23]. This type of method requires a registration method for obtaining 3D points from camera or laser scanner [24, 25, 26], and then applies segmentation method to separate objects and reconstructs the semantic model [27, 28]. Object recognition algorithms are also applied to identify different objects in the point cloud [29, 30].

A similar approach can be used to integrate a construction robot with digital modeling methods for visualization and task planning [31]. For example, Yang et al. [32] utilized BIM and robot path planner to find and visualize the construction process of the modular construction. However, these types of systems are typically not synchronized between the virtual model and physical robot and require further adaption [33, 1]. The robot Digital Twin (DT) system developed in this work fulfills the demand for real-time data exchange, which is wildly utilized in the manufacturing industry, digital fabrication, and human-robot collaboration assembly [34, 35]. For example, Naboni and Kunic [36] used DT for complex wood structure manufacturing and assembly. Furthermore, by combining with other techniques such as Augmented Reality, the synchronization and communication mechanism of robot DT system can be improved [37].

3 Robot Digital Twin System

The proposed online robot Digital Twin system is shown in 2 and consists of three modules: the physical robot module, the virtual robot module, and the communication module. First, the virtual robot module includes the Digital Twin for visualizing the robot and the motion planner for planning the trajectory and solving the inverse kinematics (IK). Second, the physical robot module includes the real robotic arm and the embedded sensors for measuring joint angles. Finally, the communication module includes the MQTT communication protocol for data exchange and synchronization. The system is developed in Robot Operating System (ROS) since it is the meta-operating system that provides a message exchange mechanism between platforms across a network. For instance, the motion planner in the virtual robot module plans a trajectory and then sends the control commands to the DT robot for execution and visualization. Figure 3 shows the data exchange between each platform. The detailed description of each module is provided in the following subsections.

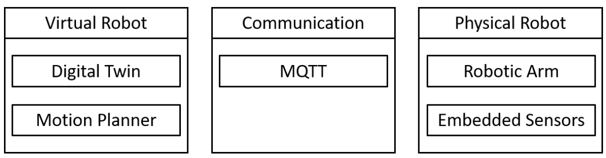

Figure 2. The framework of the online robot Digital Twin system.

3.1 Virtual Robot Module

We use ROS Gazebo and rviz to develop the DT in the virtual robot module on a Linux PC [15, 38]. The Gazebo is a real-world physics simulator that creates a world and

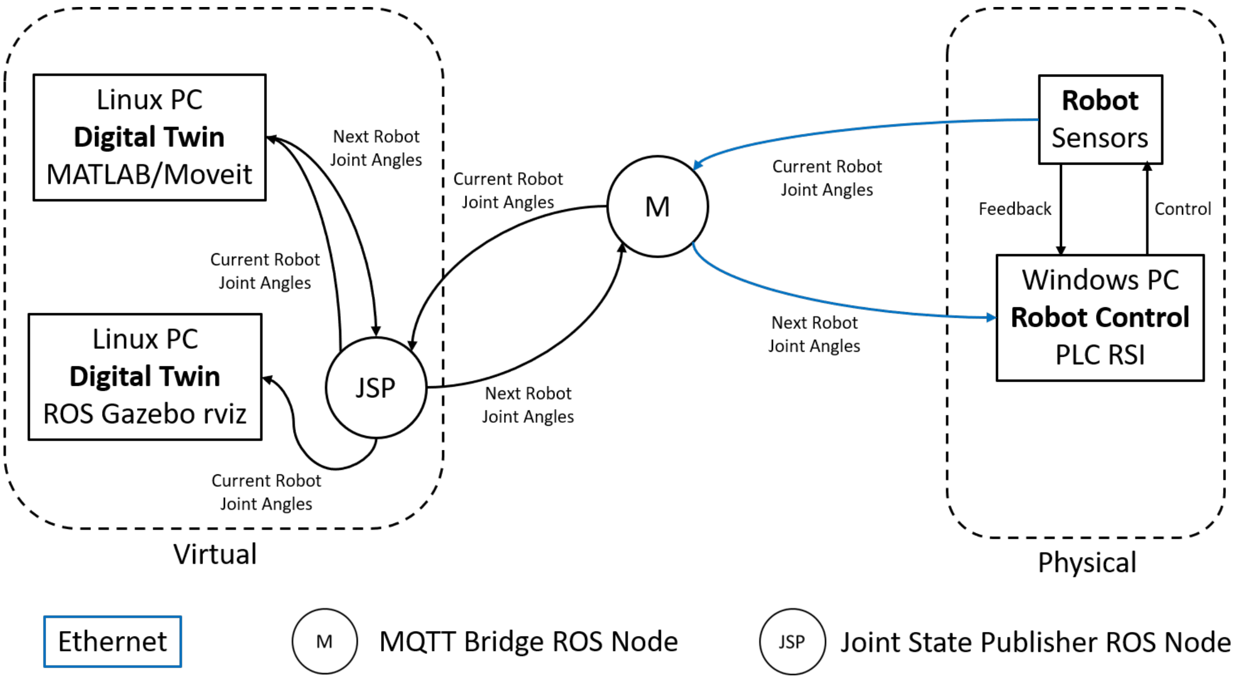

Figure 3. The flowchart of the data exchange and each platform.

simulates the robot, whereas the rviz is visualization software that can read and display the data from Gazebo or real-world sensors. The robotic arm model is imported to the Gazebo and rviz, as shown in Figure 4. The joint angles of the robotic arm are exchanged between the two programs to ensure synchronization.

In order to plan the specific construction task or motion, a motion planner is required in the module. Either MATLAB or MoveIt! can be used as the motion planner to achieve the task [39]. The Robotic System Toolbox in MATLAB can plan the trajectory and solve the inverse kinematics of the robot. However, it suffers from the latency issue and is not fast enough for real-time planning purpose. On the other hand, the MoveIt! is a motion planning package for ROS, which plans the motion inside rviz and sends to Gazebo. Figure 4 top shows the interface of the MoveIt! motion planning in rviz. The start state, goal state, and time parameters can be customized and determined by the user as input to the motion planner. The result of the motion planning will then be demonstrated in rviz and sent back to Gazebo for execution. Both MATLAB and MoveIt! can be run on the same Linux PC as the DT, or run on a different PC and connected through network.

For the data exchange, only the current robot joint angles and the next robot joint angles are displayed within the virtual robot module. Both Gazebo and rviz read the current robot joint angles to visualize the robot state. The MATLAB or MoveIt! package read the robot joint angles, determine the next robot joint angles, and send back to Gazebo and rviz for execution. The joint state publisher (JSP) is the ROS node for publishing the current robot

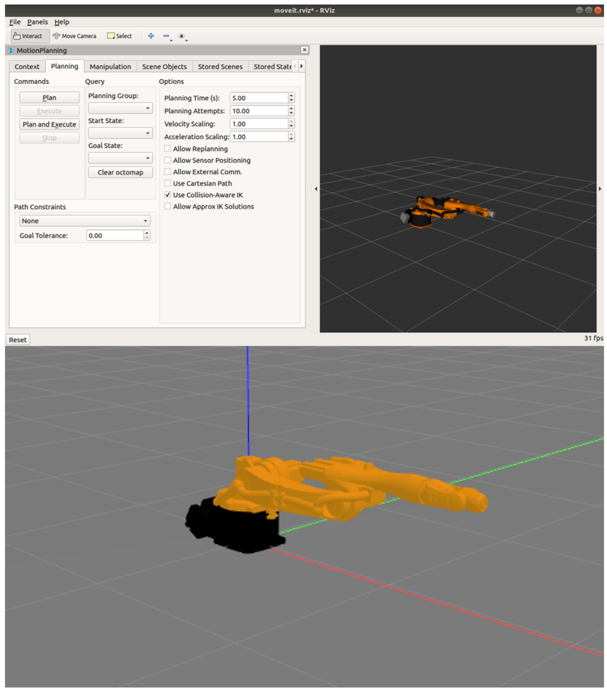

Figure 4. The robotic arm in Gazebo (bottom) and rviz with MoveIt! package (top).

state to different ROS nodes, including the current robot joint angles from the physical robot module.

3.2 Physical Robot Module

We utilize the KUKA KR120 robotic arm on the track system as the physical robot for the DT system, as shown in Figure 5. In the current version of the online DT system, the track system is not included in the virtual robot module. The programmable logic controller (PLC) and robot sensor interface (RSI) are running on a Windows PC to control the robotic arm and retrieve the sensor data. The embedded encoders on the robotic arm are used to measure the joint angles and read by the RSI. After activating the robotic arm, the system first records the current robot joint angles as the origin of the robot for robot controlling purpose. Once the physical robot receives the next joint angles from the virtual robot, it will calculate the differences of the joint angles and then uses the recorded origin to control the robotic arm in the relative mode. The robot control command and the sensor measurement are two data exchanges inside the physical robot module, as shown in Figure 3 right side.

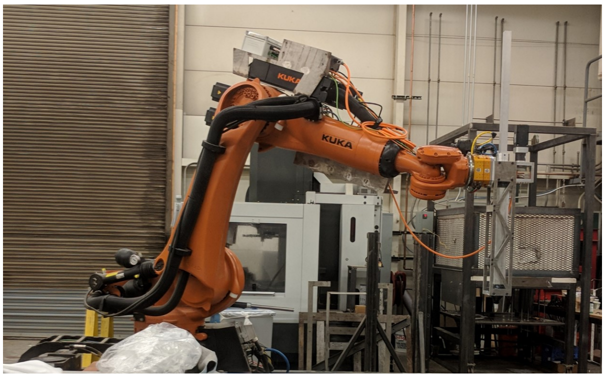

Figure 5. The KUKA KR120 robotic arm for the physical robot module.

3.3 Communication Module

Finally, the communication module links the virtual robot module and the physical robot module. We use MQTT communication protocol for data exchange between ROS system in the virtual robot module and the PLC in the physical robot module. The MQTT communication protocol is capable of real-time communication and thus is suitable for smooth robotic control. We develop an MQTT Bridge ROS node (M) to connect the MQTT to the ROS system, as shown in the middle of Figure 3. The MQTT Bridge node is run on the same Linux PC as the DT system to exchange the joint angles with JSP node and connect with PLC in the physical robot module through Ethernet. The data exchange frequency in the MQTT Bridge is set to be 250 Hz to ensure the transmission speed and avoid jitter effects on the robotic arm.

The joint angles of the robotic arm are the main data stream exchanged in the MQTT bridge ROS node. Figure 6 illustrates the data structure and exchange process in the MQTT bridge ROS node. The data stream concatenates the robot joint angles from A1 to A6 with a plus-minus sign and comma. Each joint angle is rounded to three decimal places and pads zeros to the left. Thus, the length of the data is consistent and easily retrieved by PLC. After receiving the joint angles data from the virtual robot module through the ROS topic, the system first converts the data to python string for easy storage and access. Next, the data is converted to the MQTT string type and sent to the physical robot module. This process can also avoid the garbled text issue when directly converting from the ROS topic to the MQTT string type. The data stream from the physical robot module is also processed with the same procedure and data structure and sent to the virtual robot module.

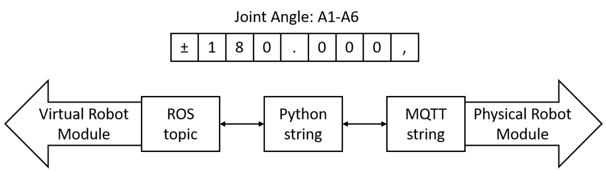

Figure 6. The data structure and exchange in the MQTT Bridge ROS node.

When exchanging the data between the virtual robot module and the physical robot module, the system must ensure the control commands are executed completely and the pose of the physical and virtual robot is synchronized. We develop a robot pose checking algorithm to confirm the synchronization between the two robotic arms. Algorithm 1 shows the pseudo-code of the pose checking algorithm (PCA). The algorithm takes the current virtual robot pose $\theta_{virtual}$, current physical robot pose $\theta_{physical}$, and the next robot pose θ_{next} as input. First, the PCA calculates the difference of $\theta_{virtual}$ and $\theta_{physical}$. If the difference exceeds the pre-defined threshold, the next joint angle θ_{next} will be assigned with the current joint angles $\theta_{virtual}$ to ensure the physical robot can reach the desired joint angles. The trajectory also needs to be re-planned to reflect the new current joint angles. On the other hand, if the difference does not exceed the threshold, the robot will simply execute the next joint angles.

Algorithm 1 Pose Checking Algorithm

procedure NEXT POSE($\theta_{virtual}, \theta_{physical}, \theta_{next}$)
  $diff(\theta) \leftarrow |\theta_{virtual} - \theta_{physical}|$
  if $diff(\theta) > threshold$ **then**
    $\theta_{next} \leftarrow \theta_{virtual}$
    Re-plan the trajectory based on θ_{next}
  else
    $\theta_{next} \leftarrow \theta_{next}$
  end if
  return θ_{next}
end procedure

4 Experiment and Results

4.1 Experiment

The online robot Digital Twin system is implemented and deployed in the Digital Fabrication Laboratory at the Taubman College of Architecture and Urban Planning at the University of Michigan. Two KUKA KR120 robotic arms are the target physical robots, as shown in Figure 1 and Figure 5. To evaluate the proposed system, we conduct an experiment to verify the pose between the physical robot and its DT are synchronized during trajectory execution. Figure 7 shows the procedure of the online robot Digital Twin system experiment. One reaching task trajectory is prepared and executed in MATLAB and Gazebo DT, then the joint angles are sent to the physical robot. Figure 8 shows the planned reaching task trajectory (pink line) in MATLAB. We use the embedded encoders on the KUKA robotic arm to measure and record the joint angles of the physical robot.

4.2 Results

The joint angles of the physical robot and the virtual robot are recorded and compared with each other. Figure 9 shows the results of the virtual and physical robot joint angles. Each line represents the angle of each joint (A1, A2, A3, A4, A5, and A6) in radians. The trajectory from the virtual robot consists of 1,500 waypoints and the measurement from the physical robot includes 18,802 data points. The result showed that the line of each joint angle had the same trend in two robots, which demonstrated the consistency of the synchronization between the two robots.

To further evaluate the accuracy of the synchronization, we calculate the average error and the maximum error of each joint angle between the two robots. Table 1 lists the result of the average and the maximum joint angle error. The average errors of each joint angle are less than 2.4e-05 in radians and the maximum errors of each joint angle are less than 2.1e-05 in radians. These results indicate that the synchronization of the virtual and the physical robot demonstrated high accuracy. The proposed pose checking

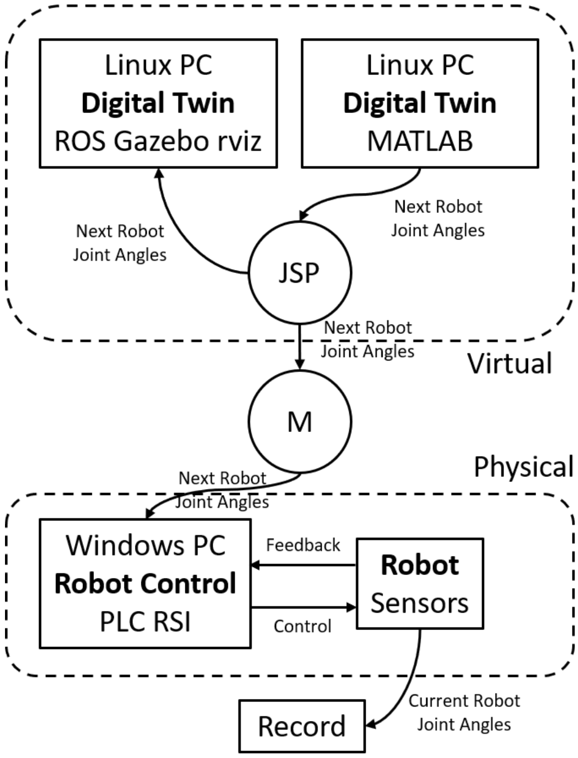

Figure 7. The procedure of the online robot Digital Twin system experiment.

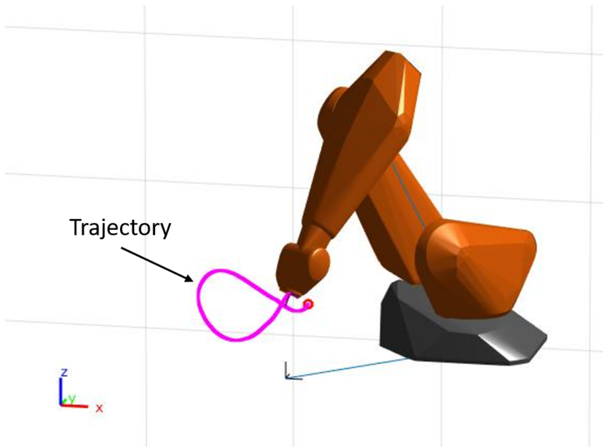

Figure 8. The planned reaching task trajectory (pink line) in MATLAB.

algorithm (PCA) also helped minimize the latency during the transmission.

5 Conclusion

This paper presented the initial development of the online robot Digital Twin system for human-robot collabo-

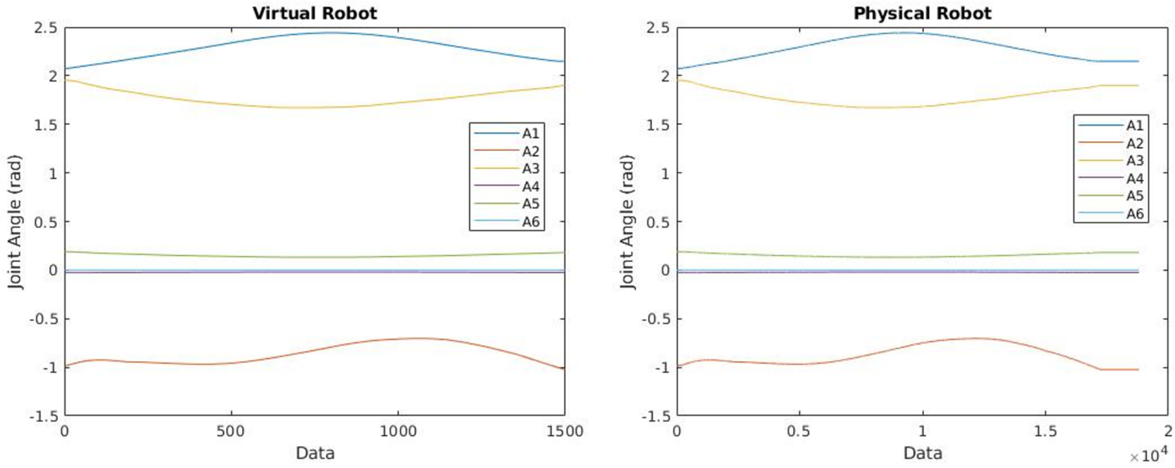

Figure 9. The results of the virtual and physical robot joint angles.

Table 1. The average and the maximum joint angle error between the virtual and the physical robot.

Joint	Average Error (rad)	Maximum Error (rad)
A1	1.5269e-06	1.7232e-06
A2	1.9848e-06	2.2391e-06
A3	2.2334e-05	2.0717e-05
A4	2.2993e-07	2.8146e-07
A5	6.2252e-06	7.0128e-06
A6	4.3442e-06	3.9486e-06

ration in the construction and digital fabrication. The system includes the virtual robot module, the physical robot module, and the communication module. We leveraged ROS Gazebo and rviz to develop the virtual robot module, i.e., Digital Twin of the physical robot, and connect to the physical robot module through MQTT Bridge in the communication module. The joint angles of the robotic arm are exchanged and synchronized between two robots. We also utilized MATLAB or MoveIt! package to plan and control the robotic arm in the virtual robot module, then send the command to the physical robot module for execution. In addition, we developed a pose checking algorithm (PCA) to ensure the pose of the two robots were synchronized.

The system was implemented and deployed on a KUKA KR120 robotic arm in the digital fabrication laboratory. Although we developed the system for the specific KUKA robotic arm, it can be easily adapted to other robot models. We evaluated the system by comparing the joint angles between the virtual and physical robot in a planned trajectory, and calculated the average and maximum errors. The results showed that the proposed online robot Digital Twin system could plan the robot trajectory inside the virtual environment and execute it in the physical environment with high accuracy. In ongoing work, we are designing the user interface for displaying the information of the physical robot in Digital Twin. We are also developing the robot planning mechanism such that the robot can first demonstrate the planned trajectory inside Digital Twin before executing by the physical robot. The human can thus expect the movement of the robot in advance and approve the task. Finally, we are designing and conducting more case studies for evaluating the proposed online robot Digital Twin system.

References

[1] Kurt M. Lundeen, Vineet R. Kamat, Carol C. Menassa, and Wes McGee. Autonomous motion planning and task execution in geometrically adaptive robotized construction work. *Automation in Construction*, 100:24–45, 2019. doi:10.1016/j.autcon.2018.12.020.

[2] BLS. Census of fatal occupational injuries (CFOI) - current and revised data, 2018. URL `https://www.bls.gov/iif/oshcfoi1.htm`.

[3] Ci-Jyun Liang, Vineet Kamat, and Carol Menassa. Teaching robots to perform construction tasks via learning from demonstration. In *Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC)*, pages 1305–1311, Banff, Canada, May 2019. IAARC. doi:10.22260/ISARC2019/0175.

[4] Ci-Jyun Liang, Vineet R. Kamat, and Carol C. Menassa. Teaching robots to perform quasi-repetitive construction tasks through human demonstration. *Automation in Construction*, 120:103370, 2020. ISSN 0926-5805. doi:10.1016/j.autcon.2020.103370.

[5] Amos Freedy, Ewart DeVisser, Gershon Weltman, and Nicole Coeyman. Measurement of trust in human-robot collaboration. In *Proceedings of the International Symposium on Collaborative Technologies and Systems*, pages 106–114, Orlando, FL, USA, May 2007. IEEE. doi:10.1109/CTS.2007.4621745.

[6] Sangseok You, Jeong-Hwan Kim, SangHyun Lee, Vineet Kamat, and Lionel P. Robert. Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments. *Automation in Construction*, 96:161–170, 2018. doi:10.1016/j.autcon.2018.09.008.

[7] Timo Salmi, Jari M. Ahola, Tapio Heikkilä, Pekka Kilpeläinen, and Timo Malm. Human-robot collaboration and sensor-based robots in industrial applications and construction. In Henriette Bier, editor, *Robotic Building*, pages 25–52. Springer International Publishing, 2018. ISBN 978-3-319-70866-9.

[8] Carlos Morato, Krishnanand N. Kaipa, Boxuan Zhao, and Satyandra K. Gupta. Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *Journal of Computing and Information Science in Engineering*, 14(1):011006, 2014. doi:10.1115/1.4025810.

[9] Shohin Aheleroff, Jan Polzer, Huiyue Huang, Zexuan Zhu, David Tomzik, Yuqian Lu, Yuan Lin, and Xun Xu. Smart manufacturing based on digital twin technologies. In Carolina Machado and J. Paulo Davim, editors, *Industry 4.0: Challenges, Trends, and Solutions in Management and Engineering*, page 77. CRC Press, 2020. ISBN 978-0-8153-5440-6.

[10] Azad M. Madni, Carla C. Madni, and Scott D. Lucero. Leveraging digital twin technology in model-based systems engineering. *Systems*, 7(1):7, 2019. doi:10.3390/systems7010007.

[11] Yuqian Lu and Xun Xu. Resource virtualization: A core technology for developing cyber-physical production systems. *Journal of Manufacturing Systems*, 47:128–140, 2018. doi:10.1016/j.jmsy.2018.05.003.

[12] Tim Delbrügger, Lisa Theresa Lenz, Daniel Losch, and Jürgen Roßmann. A navigation framework for digital twins of factories based on building information modeling. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, Limassol, Cyprus, September 2017. IEEE. doi:10.1109/ETFA.2017.8247712.

[13] Matthew Q. Marshall and Cameron Redovian. An application of a digital twin to robotic system design for an unstructured environment. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, page V02BT02A010, Salt Lake City, UT, USA, November 2019. ASME. doi:10.1115/IMECE2019-11337.

[14] Open Robotics. Robot Operating System, 2018. URL `https://www.ros.org/`.

[15] Open Robotics. Gazebo, 2019. URL `http://gazebosim.org/`.

[16] Roger A. Light. Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software*, 2(13):265, 2017. doi:10.21105/joss.00265.

[17] Vineet R. Kamat and Julio C. Martinez. Large-scale dynamic terrain in three-dimensional construction process visualizations. *Journal of Computing in Civil Engineering*, 19(2):160–171, 2005. doi:10.1061/(ASCE)0887-3801(2005)19:2(160).

[18] Robert Eadie, Mike Browne, Henry Odeyinka, Clare McKeown, and Sean McNiff. BIM implementation throughout the UK construction project lifecycle: An analysis. *Automation in Construction*, 36:145–151, 2013. doi:10.1016/j.autcon.2013.09.001.

[19] Alcinia Z. Sampaio and Edgar Berdeja. Collaborative BIM environment as a support to conflict analysis in building design. In *Proceedings of the Experiment@International Conference (exp.at'17)*, pages 77–82, Faro, Portugal, June 2017. IEEE. doi:10.1109/EXPAT.2017.7984348.

[20] Tzong-Hann Wu, Feng Wu, Ci-Jyun Liang, Yi-Fen Li, Ching-Mei Tseng, and Shih-Chung Kang. A virtual reality tool for training in global engineering collaboration. *Universal Access in the Information Society*, pages 1–13, 2017. doi:10.1007/s10209-017-0594-0.

[21] Sebastian Ochmann, Richard Vock, Raoul Wessel, and Reinhard Klein. Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics*, 54:94–103, 2016. doi:10.1016/j.cag.2015.07.008.

[22] Hesam Hamledari, Brenda McCabe, Shakiba Davari, and Arash Shahi. Automated schedule and progress updating of IFC-based 4D BIMs. *Journal of Computing in Civil Engineering*, 31(4):04017012, 2017. doi:10.1061/(ASCE)CP.1943-5487.0000660.

[23] Yong Xiao, Yuichi Taguchi, and Vineet R. Kamat. Coupling point cloud completion and surface connectivity relation inference for 3D modeling of indoor building environments. *Journal of Computing in Civil Engineering*, 32(5):04018033, 2018. doi:10.1061/(ASCE)CP.1943-5487.0000776.

[24] Lichao Xu, Chen Feng, Vineet R. Kamat, and Carol C. Menassa. An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments. *Automation in Construction*, 104:230–245, 2019. doi:10.1016/j.autcon.2019.04.011.

[25] Chen Feng, Yong Xiao, Aaron Willette, Wes McGee, and Vineet R. Kamat. Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites. *Automation in Construction*, 59:128–138, 2015. doi:10.1016/j.autcon.2015.06.002.

[26] Frédéric Bosché, Mahmoud Ahmed, Yelda Turkan, Carl T. Haas, and Ralph Haas. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction*, 49:201–213, 2015. doi:10.1016/j.autcon.2014.05.014.

[27] Andrey Dimitrov and Mani Golparvar-Fard. Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. *Automation in Construction*, 51:32–45, 2015. doi:10.1016/j.autcon.2014.12.015.

[28] Vladeta Stojanovic, Matthias Trapp, Rico Richter, Benjamin Hagedorn, and Jürgen Döllner. Towards the generation of digital twins for facility management based on 3D point clouds. In *Proceedings of the ARCOM 34th Annual Conference*, pages 270–279, Belfast, UK, September 2018.

[29] Chao Wang and Yong K. Cho. Smart scanning and near real-time 3D surface modeling of dynamic construction equipment from a point cloud. *Automation in Construction*, 49:239–249, January 2015. doi:10.1016/j.autcon.2014.06.003.

[30] Jacob J. Lin, Jae Yong Lee, and Mani Golparvar-Fard. Exploring the potential of image-based 3D geometry and appearance reasoning for automated construction progress monitoring. In *Proceedings of the ASCE International Conference on Computing in Civil Engineering (i3CE)*, pages 162–170, Atlanta, GA, USA, June 2019. ASCE. doi:10.1061/9780784482438.021.

[31] Vineet R. Kamat and Julio C. Martinez. Dynamic 3d visualization of articulated construction equipment. *Journal of Computing in Civil Engineering*, 19(4):356–368, 2005. doi:10.1061/(ASCE)0887-3801(2005)19:4(356).

[32] Cheng-Hsuan Yang, Tzong-Hann Wu, Bo Xiao, and Shih-Chung Kang. Design of a robotic software package for modular home builder. In *Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC)*, pages 1217–1222, Banff, AB, Canada, May 2019. IAARC. doi:10.22260/ISARC2019/0163.

[33] Kurt M. Lundeen, Vineet R. Kamat, Carol C. Menassa, and Wes McGee. Scene understanding for adaptive manipulation in robotized construction work. *Automation in Construction*, 82:16–30, 2017. doi:10.1016/j.autcon.2017.06.022.

[34] Cunbo Zhuang, Jianhua Liu, and Hui Xiong. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. *The International Journal of Advanced Manufacturing Technology*, 96(1):1149–1163, 2018. doi:10.1007/s00170-018-1617-6.

[35] Arne Bilberg and Ali Ahmad Malik. Digital twin driven human–robot collaborative assembly. *CIRP Annals*, 68(1):499–502, 2019. doi:10.1016/j.cirp.2019.04.011.

[36] Roberto Naboni and Anja Kunic. A computational framework for the design and robotic manufacturing of complex wood structures. In *Proceedings of the Education and Research in Computer Aided Architectural Design in Europe and Iberoamerican Society of Digital Graphics, Joint Conference*, volume 7, pages 189–196, Porto, Portugal, September 2019. doi:10.5151/proceedings-ecaadesigradi2019_488.

[37] Yi Cai, Yi Wang, and Morice Burnett. Using augmented reality to build digital twin for reconfigurable additive manufacturing system. *Journal of Manufacturing Systems*, In Press, May 2020. doi:10.1016/j.jmsy.2020.04.005.

[38] ros-visualization. ROS 3D Robot Visualizer, 2020. URL `https://github.com/ros-visualization/rviz`.

[39] David T. Coleman, Ioan A. Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a MoveIt! case study. *Journal of Software Engineering for Robotics*, 5(1): 3–16, 2014. doi:10.6092/JOSER_2014_05_01_p3.