Reinforcement Learning with Vision-based State Estimation for Generalized Robotic Assembly

Mark Trovinger[1], Leonel Giacobbe[1], Tanner Grantham[1], Chuangchuang Sun[2], Jingdao Chen[1]

[1]Department of Computer Science and Engineering, Mississippi State University, USA
[2]Department of Aerospace Engineering, Mississippi State University, USA

mrt364@msstate.edu, lg1221@msstate.edu, tdg295@msstate.edu, csun@ae.msstate.edu, chenjingdao@cse.msstate.edu

Abstract -

Robotic manipulation is a promising technology that has the potential to further automate construction tasks such as bricklaying, concrete printing, or crane assembly of prefabricated components. With the increasing complexity of modern construction projects, more sophisticated planning algorithms are needed to optimize the efficiency of robotic manipulator operations. Reinforcement learning (RL) has emerged in recent research as a way to solve planning problems in high-dimensional state spaces or where the world model is unknown or noisy. Most RL algorithms for robotic manipulation target low-level control and is not scalable to multi-object or complex assembly tasks. This research proposes a reinforcement learning framework with integration of vision-based state estimation and inverse kinematics for generalized robotic assembly tasks. A contour detection and linear regression algorithm is used to perform initial state estimation and grasp point regression, whereas the reinforcement learning algorithm involves a discrete state space where transitions between states are handled using inverse kinematics. The proposed framework is validated through a physics simulation as well as a lab-based experimental setup with robotic assembly of wooden blocks.

Keywords -

Reinforcement learning; computer vision; robot assembly

1 Introduction

Automation of construction work is of great need in the building industry to allow construction tasks to be carried out more accurately and efficiently, especially in view of the increasing shortage of skilled workers. In the construction industry, roboticized work can come in the form of robotic bricklaying [1], concrete printing [2], or crane assembly of prefabricated components [3]. Still, using robotics technology for construction work remains challenging due to the unstructured and heterogeneous nature of construction environments [4].

Reinforcement learning (RL) has emerged as a promising research area for robotic control and robotic assembly of complex objects [5, 6]. Contrary to prior methods which rely on having a known physical model of an environment, reinforcement learning allows a robotic agent to learn the characteristics of an unknown environment directly through experience and taking actions in that environment.

Existing work in reinforcement learning for robotic assembly tasks [7, 6] usually define the states as end effector pose and actions as end effector displacements, both in continuous Cartesian space. However, the limitation of this approach is that the learning process involves long convergence times and low success rates due to the large state space and action space, making the task extremely difficult to learn. Additionally, robotic movement does not lend itself to discrete timesteps; robots require time to start and stop movement and so additional work is needed to ensure that an agent does not attempt to take actions when the robot is incapable of doing so. These obstacles can largely be overcome in simulation, but additional issues are faced when moving from a simulation of a robot to the real world.

To address these issues, this research proposes to leverage underlying inverse kinematics mechanisms for the robotic manipulator, allowing the reinforcement learning algorithm to act at a high level with a smaller state space, thus simplifying the learning problem. In addition, a vision module is used for state estimation, allowing the robot to adapt to equipment and materials being transported about in a dynamic construction environment.

2 Literature Review

2.1 Robotic assembly tasks

Robots are uniquely suited to handle many assembly tasks that are dull, dirty, or dangerous. As a result, robots are deployed to accomplish basic tasks in the construction industry [8], such as laying bricks [1], concrete 3D printing [2], etc. Other tasks include assembling interior structures, such as flooring installation [9], HVAC and plumbing installation [10], and assembling furniture such as an IKEA chair using multiple collaborative robotic manipulators [11].

2.2 Planning methods in construction robotics

Planning methods find many applications in construction robots, such as for crane lift planning [12]. Work in [13] considers building information modeling in construction robotic task planning. [14] addresses the issue of long-horizon multi-robot planning in construction assembly. Sampling-based planners, such as RRT [15], have been used for crane planning and beyond. However, state-space planners and sampling-based planners mostly assume a known environment with static obstacles. To work in dynamic construction environments, multi-modal perception is key for real-time feedback and situational awareness in planning [16]. In the domain of crane lift planning, relevant work include dynamic 3D modeling in environments with low visibility and dense obstacles [17], and real-time monitoring of crane load [18].

2.3 Learning methods in construction robotics

Recently, learning methods have also been explored in construction robotics [19] to tackle problems of higher complexity and improve operation in uncertain environments. Robots empowered by reinforcement learning, for example, Deep Deterministic Policy Gradient (DDPG), have been used for the robotic assembly of timber joints [20]. Human-robot collaboration in construction [21] can also be achieved via reinforcement learning with safety and efficiency. Another learning paradigm, imitation learning, or Learning from Demonstration (LfD) [22] is also deployed in construction robots, such as performing quasi-repetitive construction tasks through human demonstration [23]. Moreover, [24] developed an imitation learning approach for robot teleoperation in construction. Nevertheless, research gaps remain in integrating these black-box learning-based methods with more established planning or kinematics frameworks, and also in integrating the manipulation system with vision-based feedback. There is also a research need for scalable reinforcement learning algorithms [25] that can work for complex, multi-agent [26] tasks in construction environments.

3 Methodology

3.1 Problem setup

In this research, tasks such as bricklaying, crane assembly etc. are represented as a generalized robotic assembly problem. The generalized robotic assembly problem involves assembling n components from initial states $\{s_1, s_2, s_3, ..., s_n\}$ to a complete structure with target states $\{t_1, t_2, t_3, ..., t_n\}$. The assembly task may involve order constraints such that some components need to be installed before other components are installed, which is specific to the assembly task. The total manipulation cost is the time taken to move each component from its initial state to the final state $\sum_n T(s_n, t_n)$. The key assumptions are that (i) the initial states are unknown, and (ii) the manipulation costs are unknown. Therefore, the optimization process involves finding the least time required for a robotic manipulator to complete the assembly task.

3.2 Vision-based initial state estimation and grasp point regression

The proposed method follows the framework described in Figure 1.The vision module starts by obtaining an image frame from a camera mounted near the end effector of the robotic arm. The robot's position remains consistent during the image acquisition process, ensuring a stable stream of extractable data. The image undergoes several preprocessing steps, including grayscale and denoising filters. Non-local means denoising is chosen due to its high retention of detail. Then, a contour detection algorithm is utilized to obtain the outline of prominent shapes within the frame. In this work, the specific target objects are rectangular wooden blocks, so our implementation uses OpenCV's contour detection algorithm to find the four corner points of any rectangular shape in the image. The preprocessing steps, including the grayscale conversion and denoising, were implemented to overcome the noise limitations induced by the camera included with the Kinova Robotic Arm. The area threshold for contour detection was set to 100 pixels, as that value effectively filtered out noise-induced false positives while maintaining accurate detection for rectangular blocks.

The position of the points in the frame are represented as $(xPixel, yPixel)$ tuples, with the top-left corner of the image serving as the origin $(0, 0)$. At this point, the tuples of the four vertices are known, but the specific object corners they represent (top/ bottom, left/ right) are not. To determine the correct assignment, we treat the 2 points with the highest $xPixel$ values as our right points, and the remaining points are treated as left points. Subsequently, the $yPixel$ values between the two right points and left points are compared. Among the right and left side points, the ones with smaller $yPixel$ values are designated as top points, and the remaining ones are assigned as bottom points. With these four points identified, we can solve for the correct rotation angle about the block's z-axis as well as the coordinates of the center of the block. These coordinates of all blocks detected are stored as the initial state of the assembly task.

Then, using the center of the block and its Z-axis rotation, we pass this data to a regression model to calculate the movement required by the robotic arm to

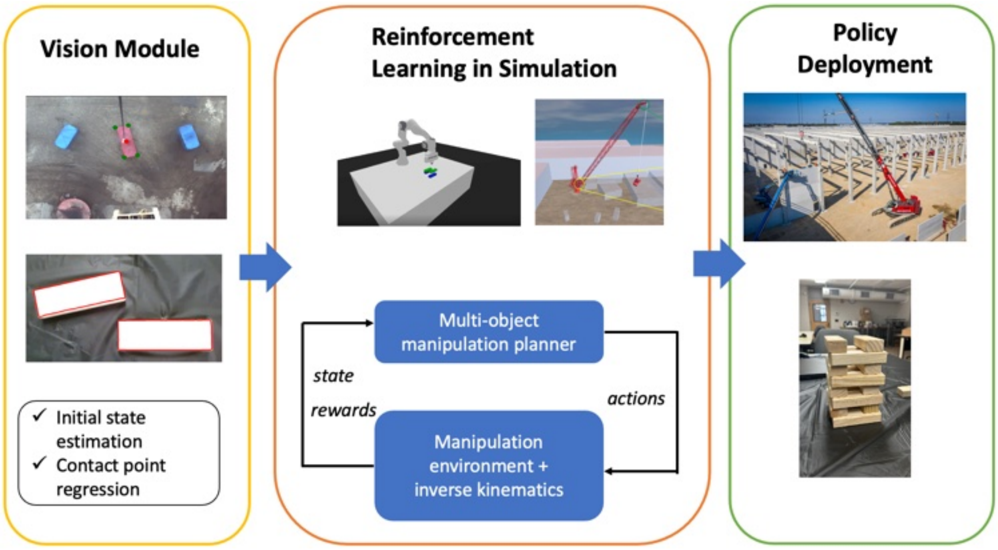

Figure 1. Proposed framework for reinforcement learning with vision-based state estimation for generalized robotic assembly (crane examples are adapted from [27])

successfully grasp the block. Our regression model is a linear model consists of 50 data points of the form $(xPixel, yPixel) \rightarrow (xMovement, yMovement)$, where $xPixel$ and $yPixel$ represent the coordinates of the center of the wooden block, and $xMovement$ and $yMovement$ represent the translational displacement required to position the robotic arm directly above the block to pick up. The training data set was manually annotated through human demonstrations by calculating the block's midpoint and recording the movements needed for the arm to transition from the initial camera position to a pick-up position above the block. The Z movement is not taken into account, as all blocks are assumed to be at the same height when placed on the surface where the arm lies. Figures 2 and 3 show visualizations of the regression model trained to predict the best robot manipulator pose given the visual position of the target object. In our study, the aforementioned linear regression model demonstrated excellent predictive performance, achieving R^2 scores of 0.9865 for the X-coordinates and 0.9985 for the Y-coordinates.

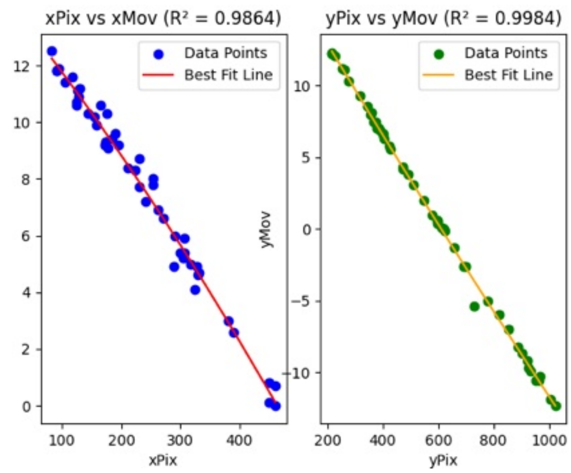

Figure 2. Plot of training data points and line of best fit for regression of desired manipulator pose

3.3 Simulation setup

The simulation consists of a modified Panda Gym environment from [7]. The Panda Gym environment was selected because it allows for simulating robotic manipulation tasks in an RL interface compatible with Stable-Baselines3 [28]. In order to utilize RL algorithms for planning, actions that normally consist of low-level movements in the arm, such as changing joint position and gripper width, are replaced by pre-planned actions. These actions are higher-level, and correspond to the underlying goal within the task; such as moving the arm to a specified location and opening and closing the gripper. In our initial work, we used the simulation of the Franka Emika Panda robot that is present in the Panda Gym environment. This robot was already present in the simulation software, and is similar to the Kinova robot that we were able to utilize it while implementing the Kinova robot. The simulation runs at 25 Hz[7], and each task is given 100 timesteps in order to complete each movement. Using the relationship that a single timestep equals 2ms, we are able to determine how long each subtask takes.

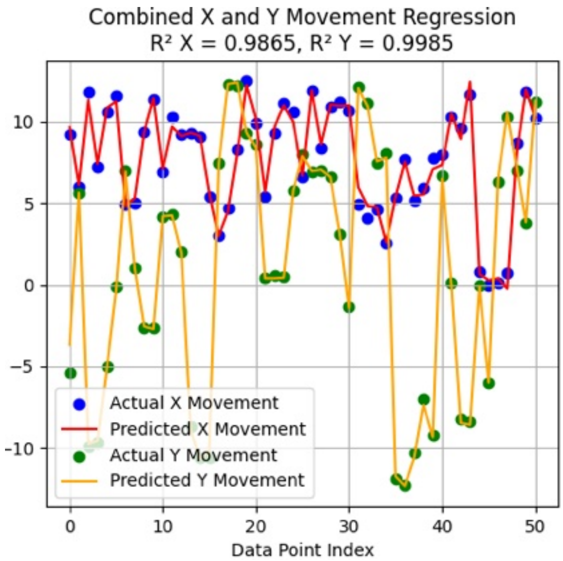

Figure 3. Graph of regression line between the desired robot manipulator pose and the visual position of the target object to be manipulated

3.4 Reinforcement learning

In typical robotic manipulator simulations, the state of the simulation is a set of continuous variables, that define the position of the components of the robot in simulated space. For example, in environments where the agent controls the end-effector's position, the coordinates of the end effector are present in the state or observation. These observations are used by the agent to select actions from a policy π, in order to maximize the returns (or sum of rewards) over the course of a training episode. These rewards are typically "sparse" in manipulator environments, meaning that the agent only receives a positive reward at the completion of the task.

Whereas, in our proposed RL framework, a discrete state space is used where transitions between discrete states are handled using inverse kinematics. The learning process thus assumes that there is an underlying inverse kinematics function which is an unknown model. The cost (i.e. manipulation time) of applying this unknown inverse kinematics model has to be learned through interactions with the environment. In this research, we also assume that this cost of manipulation can be approximated through a physics simulation. This way, the process of moving a single object can be viewed as "deterministic" since it is controlled by inverse kinematics and not varied by the RL agent. The RL agent then handles the process of optimizing the movement of multiple objects to complete an assembly task.

3.5 Implementation Details

All code and algorithms are deployed using a software stack consisting of the Kinova Gen3 API, OpenCV, Stable-Baselines3, PyBullet, and Panda-Gym's learning environment. OpenCV is used to estimate the real-world coordinates of wooden blocks positioned on a lab bench. Whereas, the Kinova API is used for manipulation planning, where our algorithm computes the desired goal state of the robot's end effector, and the underlying API will calculate the required joint angles to reach that position using inverse kinematics for the Kinova robot. The simulation environment is built upon Panda Gym, which allows for modular integration between robot simulations and reinforcement learning algorithms. The simulation environment uses PyBullet, a physics engine that models collisions, soft body dynamics, and rigid body dynamics to replicate real world environmental constraints as closely as possible. The Kinova Gen3 7DOF robot arm is loaded into simulation using the Unified Robot Description Format (URDF). The URDF contains information regarding mass, moments of inertia and force limits for each of the arm's 7 joints. Finally, the reinforcement learning algorithms used are obtained from the Stable Baselines3 library.

4 Results

4.1 Evaluation metrics in simulation

To evaluate the proposed framework, we design two assembly tasks which are the 3-layer tower structure (6 components) and the 3-layer wall structure (8 components). The wall structure is designed to simulate the brick laying task in construction. The tasks created in the simulator were designed to match, as closely as possible, the tasks performed in the lab setting. The metrics used for evaluating the performance of the agents are the assembly time and success rate, where a successful run is defined as one in which the blocks are detected, picked up, and deposited at the correct target positions accurately.

In order to evaluate our proposed "deterministic" approach compared to other RL algorithms, we evaluated three current state of the art algorithms used in robotic manipulation, Deep Deterministic Policy Gradients (DDPG)[29], Twin Delayed Deep Deterministic policy gradient (TD3)[30], and Soft Actor Critic (SAC)[31]. In addition, these algorithms make use of Hindsight Experience Replay (HER), which allows for a more sample efficient agent training process. HER was used due to the sparse nature of the rewards given in each task; with the agent only receiving a reward at end of the training loop, and even then, only when the agent has successfully completed the task. Hyperparameters used for the baseline testing match those found in HER[32].

For the RL agents, the training regime is 50M timesteps, and the hyperparameters used match those used in Panda Gym. The higher number of timesteps for training was possible due to training the agents in parallel. The reward each agent receives is -1 for each timestep taken for each subtask (moving the arm to the object, opening and closing the gripper, etc.), with a maximum number of timesteps allowed per objective task of 350 for the 3-layer tower, and 500 for the 3-layer wall and the total return and is represented as:

$$G = -\sum_{0}^{T}\sum_{0}^{\infty} t_{subtask} \tag{1}$$

Where G represents the total episodic return, T represents the total number of allowed timesteps and $t_{subtask}$ is the number of timesteps per deterministic subtask. Hyperparameters used in the deterministic approach are referenced from work by Da Costa[33].

Quantitative results for assembling the 3-layer tower and 3-layer wall are shown in Tables 1 and 2 respectively. Results show that the proposed algorithm achieves higher rewards (and therefore shorter number of steps to complete the assembly task) compared to other algorithms. The proposed algorithm also achieves a much higher success rate: since the algorithm solves the state transitions in a "deterministic" way using inverse kinematics, it is almost guaranteed to finish the assembly task. Whereas, other RL algorithms struggle to solve the task because they have to search for a policy in a very large state space and action space. Qualitative results of the completed assembly tasks in simulation are shown in Figure 4.

Table 1. Performance comparison for 3-layer tower assembly in simulation

Method	Reward	Success Rate
DDPG	-350	0.01
SAC	-349	0.02
TD3	-350	0.01
Proposed deterministic	-315*	1.00

Table 2. Performance comparison for 3-layer wall assembly in simulation

Method	Reward	Success Rate
DDPG	-498	0.015
SAC	-500	0.010
TD3	-500	0.010
Proposed deterministic	-494*	1.000

4.2 Qualitative results for lab testbed

Figure 5 shows proof-of-concept results for physically assembling a 3-layer hollow tower, and a 3-layer wall structure out of wooden blocks using a Kinova Gen3 L53 7 degree-of-freedom (DOF) robotic arm. The implementation makes use of the inverse kinematics module in the Kinova API, and the geometry of the structure to be assembled is pre-programmed in a Python script and deployed on the robot.

Table 3 shows a comparison of the total assembly time of the 3-layer tower as well as the 3-layer wall from the lab testbed. The time taken to place each component is highly dependent on the component's target position and orientation, as well as the specific implementation of the underlying inverse kinematics algorithms (the same end effector pose can be reached through many different possible joint angles). Reinforcement learning provides an opportunity to learn these interactions and arrive at an optimized policy to minimize assembly time.

Table 3. Comparison of total assembly time from the lab testbed

Structure	#Components	Assembly Time (s)
3-layer tower	6	195
3-layer wall	8	180

5 Limitations and Discussion

Currently, the vertical (Z-axis) distance between the arm's end effector and the target object to be manipulated is assumed to be fixed. Future work will look to overcome this limitation by implementing triangle similarity or using depth images, allowing for more generalized behavior that more closely resembles real-life construction environments. Moreover, the current model only accounts for rotation about the Z axis and not about the X and Y axes. Moving forward, the regression model can be adjusted to consider these additional rotation angles as parameters, and allow for more robust grasping and manipulation of objects. The current implementation only considers vision input in the initial stage for state estimation and grasp point regression. Ideally, continuous vision feedback will be important to correct state estimation errors or motion noise. Theoretically, our approach can scale to larger construction environments, as long as we can represent the scene accurately in simulation and train the agent in simulation. Our approach relies on having access to inverse kinematics and this is a key assumption. For complex construction machinery where an inverse kinematics solution is not available, probabilistic motion planning algorithms may be a viable alternative [34].

6 Conclusion

This research proposes a reinforcement learning framework with integration of vision-based state estimation and

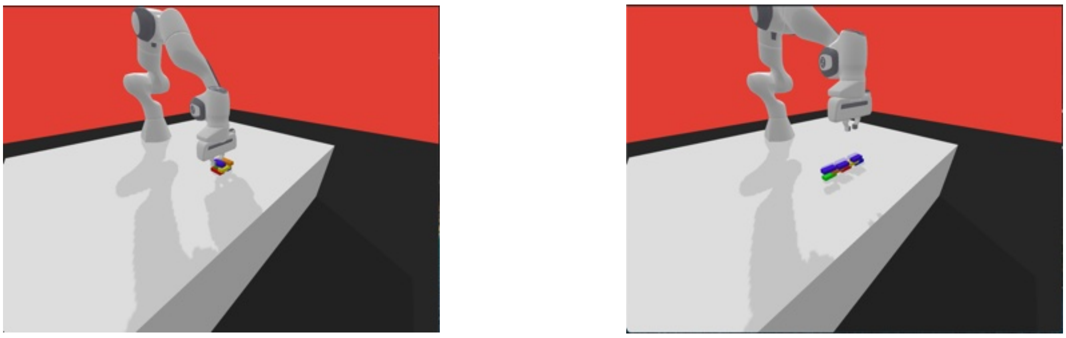

Figure 4. Results for 3-layer tower assembly and 3-layer wall assembly in the Panda Gym simulation

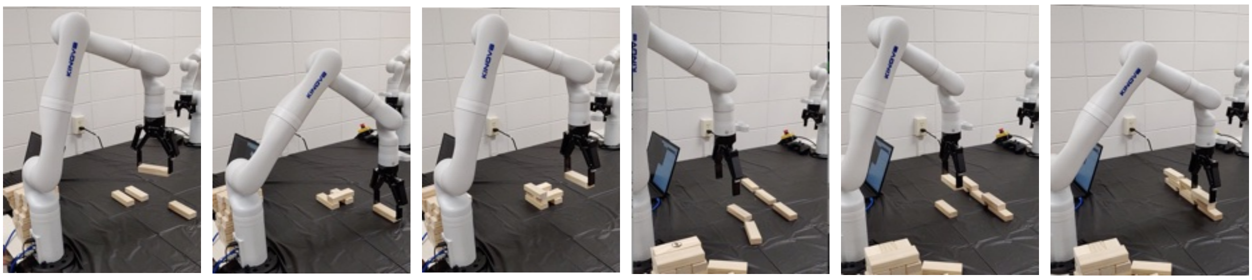

Figure 5. Qualitative results of a 7-DOF Kinova robot arm assembling a hollow tower (left 3 images) and a wall structure (right 3 images) using wooden blocks

inverse kinematics for generalized robotic assembly tasks. The advantage of the proposed framework is a generalized formulation of robotic assembly tasks for reinforcement learning algorithms that can also be adapted to arbitrary initial states. Future work in this area involves exploring multi-agent reinforcement learning algorithms that can allow the assembly tasks to be completed by two or more robotic manipulators that can learn to coordinate with one another. Human-robot collaboration [35] will also be explored to investigate how robotic agents can collaborate with human workers to complete assembly tasks.

7 Acknowledgements

The work reported herein was supported by the National Science Foundation (NSF) (Award #IIS-2153101). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Zakaria Dakhli and Zoubeir Lafhaj. Robotic mechanical design for brick-laying automation. *Cogent Engineering*, 4(1):1361600, 2017.

[2] Daniel Kajzr, Tomáš Myslivec, and Josef Černohorskỳ. Modelling, analysis and comparison of robot energy consumption for three-dimensional concrete printing technology. *Robotics*, 13(5):78, 2024.

[3] Aiyu Zhu, Tianhong Dai, Gangyan Xu, Pieter Pauwels, Bauke De Vries, and Meng Fang. Deep reinforcement learning for real-time assembly planning in robot-based prefabricated construction. *IEEE Transactions on Automation Science and Engineering*, 20(3):1515–1526, 2023. doi:10.1109/TASE.2023.3236805.

[4] Iro Armeni, Jingdao Chen, Chen Feng, Inbae Jeong, Ci-Jyun Liang, Kristian Morin, Nikos Tsagarakis, Xi Wang, and Liangjun Zhang. Construction robotics and automation [tc spot light]. *IEEE Robotics and Automation Magazine*, 31(4):186–192, 2024. doi:10.1109/MRA.2024.3481769.

[5] Lei Huang and Zhengbo Zou. Deep reinforcement learning-based construction robots collaboration for sequential tasks. In *Proceedings of the 1st Future of Construction Workshop at the International Conference on Robotics and Automation (ICRA 2022)*, pages 48–51, Philadelphia, PA, USA, May 2022. International Association for Automation and Robotics in Construction (IAARC). doi:10.22260/ICRA2022/0015.

Figure 6. The robot camera's top view of two blocks and an assembled hollow tower. Red lines indicate detection results from the vision-based state estimation algorithm.

[6] Weijia Cai and Zhengbo Zou. A reinforcement learning based approach for conducting multiple tasks using robots in virtual construction environments. In *Proceedings of the 1st Future of Construction Workshop at the International Conference on Robotics and Automation (ICRA 2022)*, pages 44–47, Philadelphia, PA, USA, May 2022. International Association for Automation and Robotics in Construction (IAARC). doi:10.22260/ICRA2022/0014.

[7] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.

[8] Thomas Bock. Construction robotics. *Autonomous Robots*, 22:201–209, 2007.

[9] Kepa Iturralde, Malte Feucht, Daniel Illner, Rongbo Hu, Wen Pan, Thomas Linner, Thomas Bock, Ibon Eskudero, Mariola Rodriguez, Jose Gorrotxategi, et al. Cable-driven parallel robot for curtain wall module installation. *Automation in Construction*, 138:104235, 2022.

[10] Kangkang Duan, Christine Wun Ki Suen, and Zhengbo Zou. Robot morphology evolution for automated hvac system inspections using graph heuristic search and reinforcement learning. *Automation in Construction*, 153:104956, 2023.

[11] Francisco Suárez-Ruiz, Xian Zhou, and Quang-Cuong Pham. Can robots assemble an ikea chair? *Science Robotics*, 3(17): eaat6385, 2018. doi:10.1126/scirobotics.aat6385. URL `https://www.science.org/doi/abs/10.1126/scirobotics.aat6385`.

[12] Songbo Hu, Yihai Fang, and Yu Bai. Automation and optimization in crane lift planning: A critical review. *Advanced Engineering Informatics*, 49: 101346, 2021.

[13] Sungjin Kim, Matthew Peavy, Pei-Chi Huang, and Kyungki Kim. Development of bim-integrated construction robot task planning and simulation system. *Automation in Construction*, 127:103720, 2021.

[14] Valentin N Hartmann, Andreas Orthey, Danny Driess, Ozgur S Oguz, and Marc Toussaint. Long-horizon multi-robot rearrangement planning for construction assembly. *IEEE Transactions on Robotics*, 39(1):239–252, 2022.

[15] Ying Zhou, Endong Zhang, Hongling Guo, Yihai Fang, and Heng Li. Lifting path planning of mobile cranes based on an improved rrt algorithm. *Advanced Engineering Informatics*, 50:101376, 2021.

[16] Jingdao Chen, Pileun Kim, Dong-Ik Sun, Chang-Soo Han, Yong-Han Ahn, Jun Ueda, and Yong K. Cho. Workspace modeling: Visualization and pose estimation of teleoperated construction equipment from point clouds. In *37th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 781–788, October 2020.

[17] Jingdao Chen, Yihai Fang, and Yong K. Cho. Real-time 3d crane workspace update using a hybrid visualization approach. *Journal of Computing in Civil Engineering*, 31(5):04017049, 2017. doi:10.1061/(ASCE)CP.1943-5487.0000698.

[18] Leon C. Price, Jingdao Chen, and Yong K. Cho. Dynamic crane workspace update for collision avoidance during blind lift operations. In Eduardo Toledo Santos and Sergio Scheer, editors, *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*, pages 959–970, Cham, 2020. Springer International Publishing. ISBN 978-3-030-51295-8.

[19] Juan Manuel Davila Delgado and Lukumon Oyedele. Robotics in construction: A critical review of the reinforcement learning and imitation learning

paradigms. *Advanced Engineering Informatics*, 54: 101787, 2022.

[20] Aleksandra Anna Apolinarska, Matteo Pacher, Hui Li, Nicholas Cote, Rafael Pastrana, Fabio Gramazio, and Matthias Kohler. Robotic assembly of timber joints using reinforcement learning. *Automation in Construction*, 125:103569, 2021.

[21] Jiannan Cai, Ao Du, Xiaoyun Liang, and Shuai Li. Prediction-based path planning for safe and efficient human–robot collaboration in construction via deep reinforcement learning. *Journal of Computing in Civil Engineering*, 37(1):04022046, 2023.

[22] Zuyuan Zhu and Huosheng Hu. Robot learning from demonstration in robotic assembly: A survey. *Robotics*, 7(2), 2018. ISSN 2218-6581. doi:10.3390/robotics7020017. URL https://www.mdpi.com/2218-6581/7/2/17.

[23] 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.

[24] Yan Li, Songyang Liu, Mengjun Wang, Shuai Li, and Jindong Tan. Teleoperation-driven and keyframe-based generalizable imitation learning for construction robots. *Journal of Computing in Civil Engineering*, 38(6):04024031, 2024.

[25] Chuangchuang Sun, Macheng Shen, and Jonathan P How. Scaling up multiagent reinforcement learning for robotic systems: Learn an adaptive sparse communication graph. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 11755–11762. IEEE, 2020.

[26] Dong Ki Kim, Miao Liu, Matthew D Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan How. A policy gradient algorithm for learning to learn in multiagent reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5541–5550. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21g.html.

[27] Leon C. Price, Jingdao Chen, Jisoo Park, and Yong K. Cho. Multisensor-driven real-time crane monitoring system for blind lift operations: Lessons learned from a case study. *Automation in Construction*, 124:103552, 2021. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2021.103552. URL https://www.sciencedirect.com/science/article/pii/S0926580521000030.

[28] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[29] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. URL https://arxiv.org/abs/1509.02971.

[30] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018. URL https://arxiv.org/abs/1802.09477.

[31] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

[32] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay, 2018. URL https://arxiv.org/abs/1707.01495.

[33] Théo Alves Da Costa. Solving the Traveling Salesman Problem with Reinforcement Learning — Eki.Lab — ekimetrics.github.io. https://ekimetrics.github.io/blog/2021/11/03/tsp/, 2021. [Accessed 14-03-2025].

[34] Seied Mohammad Langari, Faridaddin Vahdatikhaki, and Amin Hammad. Improving the performance of rrt path planning of excavators by embedding heuristic rules. *Advanced Engineering Informatics*, 62:102724, 2024.

[35] Ali Garshasbi, Jun Wang, Jingdao Chen, and Junfeng Ma. Human-robot collaboration in the construction industry: A mini-review. In *Proceedings of the 2nd Future of Construction Workshop at the International Conference on Robotics and Automation (ICRA 2023)*, pages 1–4, London, UK, July 2023. International Association for Automation and Robotics in Construction (IAARC). ISBN -. doi:10.22260/ICRA2023/0003.