# Towards a Distributed, Robotically Assisted Construction Framework

## Using Reinforcement Learning to Support Scalable Multi-Drone Construction in Dynamic Environments

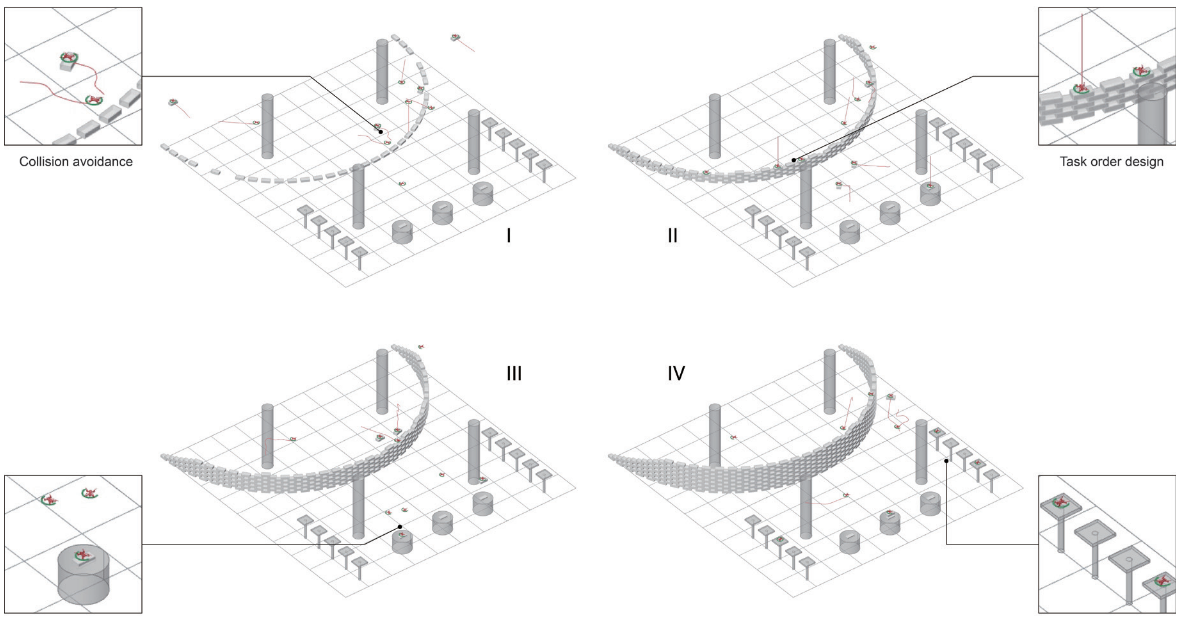
Zhihao Fang
Carnegie Mellon University

Yuning Wu
Carnegie Mellon University

Ammar Hassonjee
Carnegie Mellon University

Ardavan Bidgoli
Carnegie Mellon University

Daniel Cardoso-Llach PhD
Carnegie Mellon University




1

1 Four snapshots from the brick-laying simulation. A video of the entire simulation can be seen in Computational Design Lab (2020).

## ABSTRACT

In this paper we document progress towards an architectural framework for adaptive and distributed robotically assisted construction. Drawing from state-of-the-art reinforcement learning techniques, our framework allows for a variable number of robots to adaptively execute simple construction tasks. The paper describes the framework, demonstrates its potential through simulations of pick-and-place and spray-coating construction tasks conducted by a fleet of drones, and outlines a proof-of-concept experiment. With these elements the paper contributes to current research in architectural and construction robotics, particularly to efforts towards more adaptive and hybrid human-machine construction ecosystems.

The code is available at: https://github.com/c0deLab/RAiC

## INTRODUCTION

Our research investigates how recent advances in deep learning and reinforcement learning (RL) techniques might help improve the performance and adaptivity of robotically assisted construction systems, opening up new possibilities for semiautonomous technologies to support construction teams on- and off-site. Our vision is that networks of smaller, more adaptive robots might more flexibly and effectively assist building construction. We document progress towards a computational framework based on state-of-the-art reinforcement learning techniques that enables users to define simple construction tasks to be executed by a variable number of robots. The framework follows a centralized learning, decentralized execution approach. On the one hand, a server supervises task distribution and monitors the overall progress. On the other, each drone independently runs an RL algorithm for collision-free navigation and follows a series of rule-based macro actions to execute certain tasks. Our experimental scenario comprises a group of drones performing simple construction tasks—pick-and-place, and spray-coating—dynamically. We approach this as both a dynamic task-assigning problem and as a spatial one. We use problem modeling techniques to conceptualize a scenario of coordination involving multiple robots, supply and deployment sites, and a dynamic task list. We further discuss opportunities and limitations for this scenario of coordination in the context of the architecture-engineering-construction industries.

In the first section of this paper we discuss the state-of-the-art in multi-drone construction, identifying some shortcomings. In particular, we note the lack of a usable general framework for the introduction of these technologies in support of building processes. We then discuss our approach through an experimental scenario comprising both a software prototype, simulations, and progress towards a hardware prototype. Through simulations we test our software's capacity to coordinate a small group of drones in two kinds of construction-related activities: pick-and-place tasks and spray-coating. This section is followed by a discussion on the limitations of our approach, next steps, and notes on our broader vision on human-robot ecosystems of construction.

### Background

Despite their original and ongoing applications in warfare, unmanned aerial vehicles (UAV), or drones, have recently been utilized in a variety of fields, including scientific research (Marris 2013), agriculture (Maes and Steppe 2019; Zhang and Kovacs 2012), and environmental monitoring (Lucieer et al. 2014; Nishar et al. 2016), and have gained popularity among flight and aerial photography/cinematography enthusiasts and specialists (Mademlis et al. 2018; Nägeli et al. 2017). A limitation of drones compared with industrial robots and rovers is their lower payload capacity, larger margins of error (Goessens, Mueller, and Latteur 2018), and shorter battery life.

On the positive side, drones are more agile, can cover larger distances, and can reach greater heights (Chaltiel, Maite, and Abdullah 2018). In addition, when equipped with sensors (Dackiw et al. 2019) and robust path-planning algorithms, they can operate in a wider range of conditions. Moreover, as recent works have shown, groups of drones can be programmed to execute tasks synchronously and "collaborate" in scenarios such as large-scale public displays (Intel n.d.). Drones' agility and flexibility can thus offer important advantages in architectural and construction-related tasks and, over the last decade or so, have attracted the attention of architectural and construction researchers.

Recent research in the field of autonomous construction has explored drones' potential to support construction tasks including, but not limited to, frame structure assembly (Lindsey, Mellinger, and Kumar 2011), bricklaying (Augugliaro et al. 2014), 3D printing (Hunt et al. 2014), tensile structure weaving (Ammar 2016), modular canopy structures (Wood et al. 2018), real-scale masonry construction (Goessens, Mueller, and Latteur 2018), roofing (Romano et al. 2019), and spraying mortar (Chaltiel et al. 2018). The majority of these efforts have employed a single drone, rather than a fleet of multiple drones. Among the exceptions is a 2011 project developed at the University of Pennsylvania comprising an array of quadcopters assembling truss-like structures consisting of "beams" and "columns" with magnets embedded at joints with a gripper (Lindsey, Mellinger, and Kumar 2011). The drones are controlled by a turn-taking algorithm that coordinates a pick-and-place process wherein drones take materials from a supply station to the construction, and assemble the structure following predetermined routes.

Another example of multirobot construction, exhibited live at the Fonds Régional d'Art Contemporain du Centre in Orléans, France, is the flight-assembled architecture, an installation comprising a 6 m tall tower composed of 1,500 foam modules deployed by four quadcopters (Augugliaro et al. 2014). Here, the researchers designed a state machine for additive drone-based construction: four drones worked collaboratively, each picking up a foam brick, transporting it to the building area, placing the brick at the designated target, and charging when needed. For planning the drones' trajectories, this project relies on a space reservation system preventing collisions. To avoid creating deadlocks when two drones try to swap positions, the team created separate "freeways" at different altitudes for the drones. Some researchers go beyond the homogeneous array of robots and combine multiple types of robots to address the limitations of drones. Felbrich et al. (2017) combined the accuracy of the industrial robotic arms with the reachability of drones to fabricate long-span structures.

These projects usefully demonstrate some potential applications of multiple drones in construction tasks. However, these approaches lack generalizability and scalability. For example, while working with predefined trajectories may suit certain construction types in strictly controlled environments, it would not be useful in dynamic and less structured ones—such as those characteristic of construction sites where multiple builders and tradespeople participate. Thus, it is important to consider how automated systems may interface with human teams at the design, operation, and maintenance stages. The literature in architectural and construction robotics is also sparse in examples of dynamic human-robot interactions. An exception is (Wood et al. 2018), who propose a model of interaction between the users and their UAV configurable architectural system where users could directly manipulate the system behavior by defining the growth patterns. They also propose an indirect interface between the users and the system through a learning mode, where the system could potentially collect data to study the patterns of interactions between users, environment, and their system. However, these researchers do not take concrete steps towards implementation.

## HYPOTHESIS

We hypothesize that reinforcement learning techniques under a "centralized learning, decentralized execution" paradigm can enable a more flexible and generalizable software framework for robotically assisted construction that is more resilient to the dynamic nature of actual construction environments. Our technical approach relies on a server for task scheduling, progress monitoring, and drone management where each drone is equipped with reinforcement learning-based navigation algorithms for collision avoidance, and a library of rule-based macro actions, which are sequences of steps for accomplishing discrete subtasks such as building, resupplying, charging, and deviation handling. In addition, we use proximal policy optimization (PPO), a policy-based reinforcement learning algorithm (Schulman et al. 2017), to train a variable number of drones to navigate without collision. Coordination can thus be achieved in task execution time through specific task order and waiting mechanisms. Combining these technical strategies, a software framework can open up opportunities for more adaptive and efficient systems for robotically assisted construction and to interface dynamically with construction teams on site.

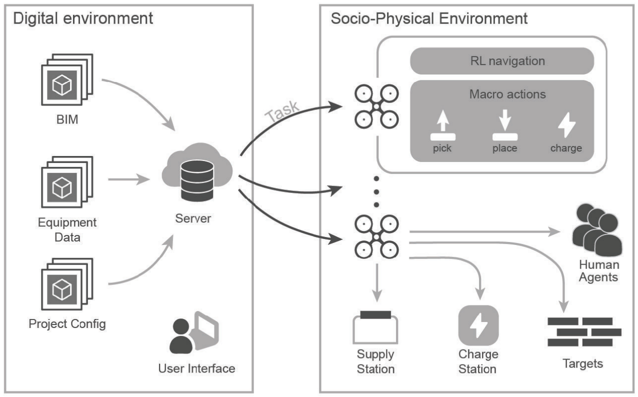
## METHODS

### Technical Framework

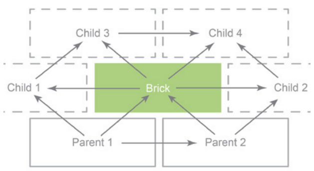
Our framework has a server for task distribution, progress monitoring, and drone management, allowing each drone to operate in a decentralized fashion. Each drone runs an RL-based navigation and collision avoidance algorithm and relies on rule-based macro actions to accomplish specific subtasks such as charging, resupplying, and placing objects (Fang 2020). Following the state machine proposed by Augugliaro et al. (2014), each drone follows the following states: (1) moving to the supply station for resupply; (2) navigating to the designated target position to install the blocks; and then, depending on its remaining battery, (3) issuing another task request; or (4) moving to the charging station to charge or change its battery.

Traditionally, centralized methods may suffer from combinatorial complexity when the number of robots grows, whereas decentralized methods may suffer from an incomplete solution with limited quality guarantees. Despite methods such as MAPP (Wang and Botea 2011), FAR (Wang and Botea 2008), and WHCA* (Silver 2005), the aforementioned limitations are inherent in the problem context. Given these limitations, reinforcement learning (RL) has emerged as an alternative approach in recent years. Through experiments, we are able to observe some key caveats that may improve generalizability and scalability concurrently, combining the advantages of both sides.

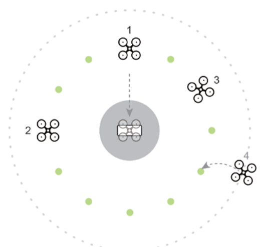
Our proposed system is more scalable, since introducing new drones to the system will not add significant load to the central server (Fig. 2). Given enough resources, the framework supports a virtually unlimited number of drones as well as various scales of construction tasks. RL navigation and collision avoidance makes the framework adaptable to a variety of environments, and is thus generalizable as long as tasks are structured as macro actions, following the state machine. Although this research is focused on scenarios with multiple drones, the problem itself only includes the simplest setting of multi-agent system (MAS), i.e., homogeneous agents without inter-agent communication or collaboration. Therefore at this stage, this research does not consider other topics featured in common MAS research, i.e., communication,



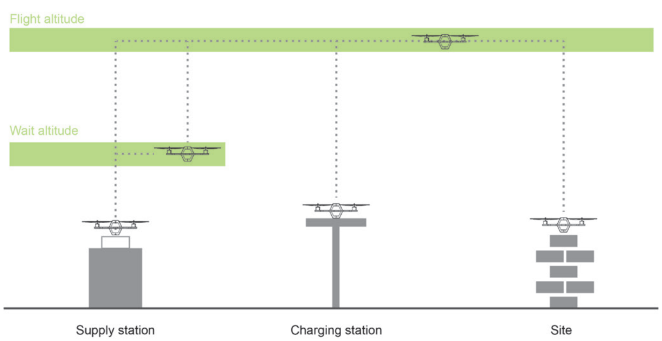

2




3




4




5

2 The proposed framework and its components.

3 Brick dependency concept.

4 Waiting mechanism.

5 Drone altitude allocation.

interaction, fault tolerance, adaptivity, cooperative and competitive environment.

### Multi-Drone Coordination

In our framework, three auxiliary procedures support the coordination of multiple drones: (1) task order, (2) waiting mechanisms, and (3) altitude allocation.

A *task order algorithm* determines the execution order of tasks. For instance, in a bricklaying process, the task order algorithm determines which bricks should be installed first and which ones should be followed. We use a directed acyclic graph (Fig. 3) to encode bricklaying order as a series of dependencies. Bricks at the bottom are considered as "parent" items that should be laid before the "children" at the top. In addition, the drone's dimensions play a key role in defining such dependencies, as the system needs to constantly determine whether there is enough space for the drone to place the next brick. During construction, the server will only schedule the placement of bricks whose "parent" bricks are successfully installed. This approach eliminates the possibility of collision between the drone and other bricks during the construction process.

This task is presented to illustrate how simple construction tasks can be defined within the framework and is not meant to advocate drone bricklaying in particular as a construction methodology.

A *waiting mechanism* is designed to let drones take turns to resupply (Fig. 4). We define each supply station surrounded by a circular waiting zone. If a drone needs to resupply at an already occupied station, it can hover at the nearest available waiting position until the system summons it in a first in, first out (FIFO) order. Every time a drone finishes resupplying at the station, the server will signal the first drone waiting in the queue to initiate the resupplying process.

To simplify the path-planning process and to reduce it to a 2D problem, we assign a specific *altitude* to each drone. This assignment minimizes the down-wash effect among the drones. Additionally, a waiting altitude is defined below the flight altitude for drones to wait at the supply station, reducing the risk of waiting drones being approached by other active drones (Fig. 5).

### Reinforcement Learning for Navigation

An RL model controls drones' navigation. Its objective is to learn an optimal policy such that drones are able to navigate from a starting to a target position without collision, including other drones, humans, walls, obstacles, etc., which are detected by the sensors at runtime. We use a proximal policy optimization (PPO) reinforcement learning model adapted from (Long et al. 2018), to learn the optimal policy. Compared with other uses of multi-agent reinforcement learning models such as Multi-Agent Deep Deterministic Policy Gradient (MADDPG), for example (Qie et al. 2019), we focus on the scalability of drone numbers, where communication among peer drones is not a major concern. Therefore, all the drones share the same RL model.

We modeled the problem in a software simulation environment based on the implementation of Multi-Agent Particle Environment (Lowe et al. 2017), an OpenAi Gym environment featuring a multi-agent particle world with a continuous observation and action space along with some basic simulated physics. The environment is defined as a 2.5D world, where the drone is allowed to fly only on either a horizontal plane during navigation or along a vertical axis to take off or land.

During the training phase, observations from different drones are collected into a common buffer in a decentralized manner before feeding into the neural network. The network input contains three types of observations: (1) simulated lidar ray data; (2) drone velocity; and (3) relative position of the target. Lidar data is specifically encoded with two CNN layers before concatenating with velocity and position. The network is updated using a composite loss that takes into account generalized advantage estimation (GAE) in a clipped manner (Schulman et al. 2017). The network outputs an acceleration vector for the drone's next movement. For better generalization, we also added Gaussian sampling.

The drone is rewarded if it reaches its target or makes an approaching step towards it, and it is penalized for getting away from the target or collision. For the criteria of reaching the target, the distance between the drone and its target should be within a tolerance distance and the velocity of the drone should be lower than a threshold. This ensures drones do not overpass the target because of inertia.

### RL Training

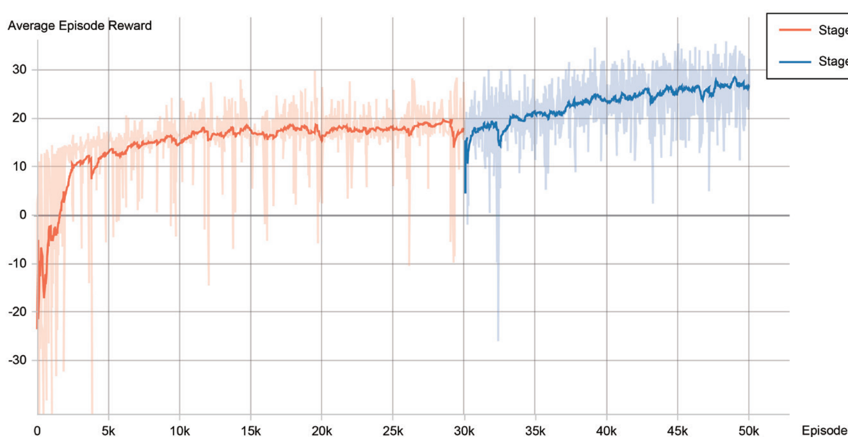
We use a two-stage training method to learn the policy. The first stage comprises five agents in a randomly generated scenario. In every episode, five random targets, each associated with an independent drone, are generated on a 2D square arena. Each drone is expected to reach its target within a maximum limit of time steps. Later, 10 agents were trained in a scenario with a number of sparsely distributed threats. We trained the model for a total of 50 thousand episodes. We tested the model on 20 agents on a 20 × 20 m square arena with six randomly generated obstacles (Fig. 6). The results show that the model is robust to scale up to let more agents reach targets without collision. Compared with other methods (i.e., MADDPG, DDPG), the increase in the number of agents will not have a drastic effect on the training time.

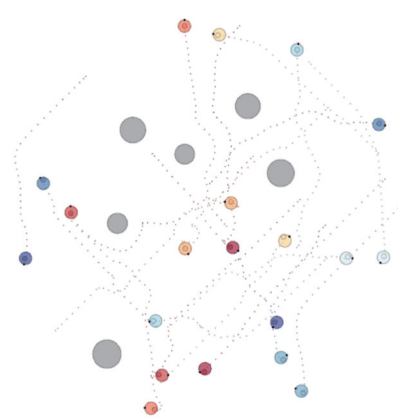
## EXPERIMENTS

We tested our framework in simulations of two hypothetical construction activities: bricklaying and spray-coating. The simulations were designed to test the RL algorithm's capacity to control each drone autonomously and independently from the server after being assigned to a task.

### Bricklaying Simulation

A curved brick wall modeled in Rhinoceros and Grasshopper holds metadata (i.e., position, orientation, type, and dependency) for each brick, which are entered as inputs into the control framework. The test site comprises charging and supply stations, each with designated waiting areas. In the simulation, drones are deployed into charging stations prior to assembly. Four "threats" are defined in the working area to account for common obstacles in a construction site such as columns, stacks of materials, walls, and humans. Based on multiple tests, the optimal number of drones in this scenario was set to 10. We observed that despite the robustness of the RL algorithm to control the drones, crowded environments will result in resource competition between the drones and eventually lead to a significant waste of time in the resupply or charging queues.



6 Average episode reward during the training model (left); evaluation of 20 agents reaching their targets (right).

6

In this experiment (Fig. 1), the framework was able to successfully decompose the task of building the curved wall into subtasks based on the units, distribute these tasks among multiple drones, and execute them without collisions in an environment that the drones had not "seen" before. Notably, the experiment showed an improvement in scalability over Augugliaro et al. (2014), which incorporates only four drones, and only allows two to fly at the same time. Our framework allows all 10 drones to operate efficiently and without collision at the same time. The experiment thus demonstrates the scalability and extensibility of the system.

### Facade Coating Simulation

We also tested the system in a facade coating simulation. Following the outline of the pick-and-place state machine, drones are supposed to refill spray material and spray at designated locations. We use six drones to coat a dome-like object with another color (Fig. 7). Though different from bricklaying, the simulation is smooth and it shows how the system can be utilized in different scenarios.

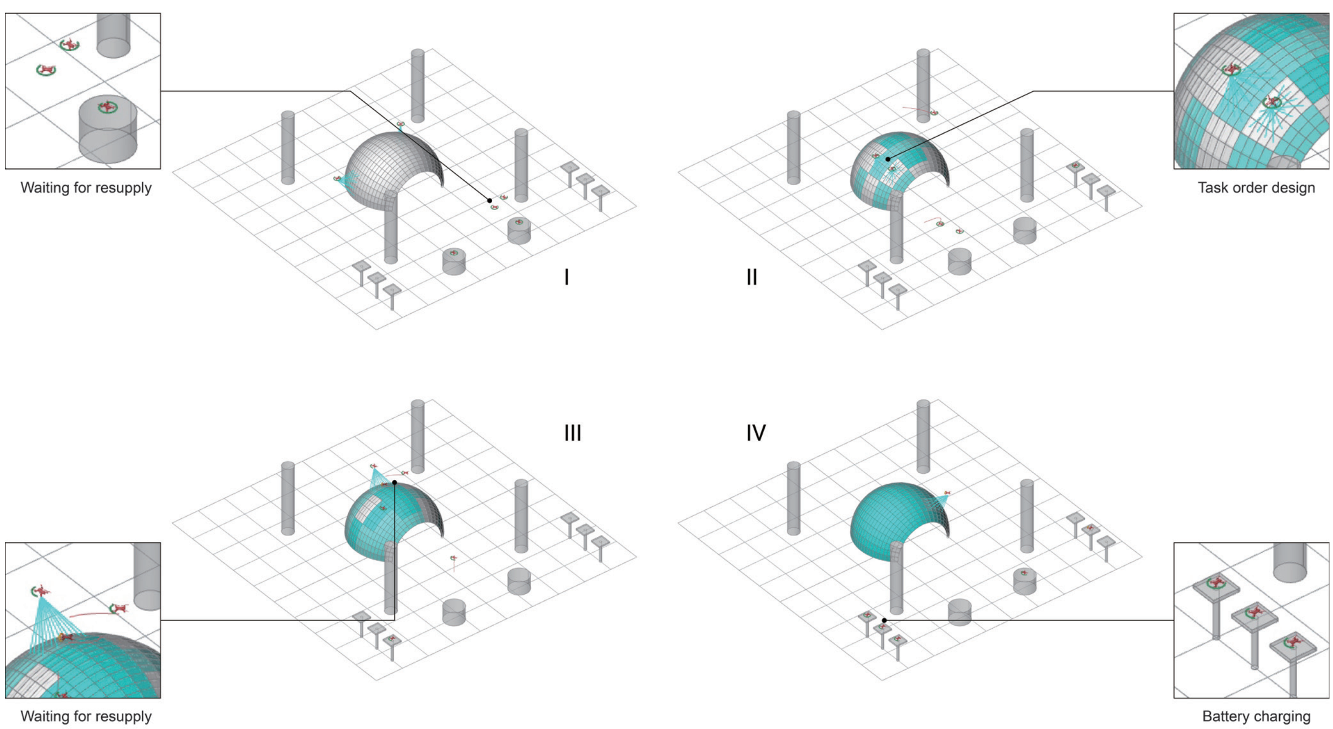
The simulation demonstrates that the proposed navigation algorithm can be efficiently scaled up to control a flock of drones until they face a logistics bottleneck, e.g., charging and resupply limitations. Accordingly, we expect the algorithm to be efficiently scalable until the drones exhaust the logistical resources or hit the physical limitations of the work environment. Regarding the generalizability of the algorithm, we are actively working to adapt the algorithm to control unmanned ground vehicles (UGV) in a construction manufacturing scenario, which will be reported in a separate publication.

### Progress Towards a Proof-of-Concept Implementation

We are currently developing the hardware setup to test the framework in a physical proof-of-concept implementation. The proposed hardware setup is designed solely to test the path-planning algorithm. However, deploying drones in indoor and unknown environments in close proximity to human users requires the use of adequate safety features for indoor and unpredictable environments (Shahmoradi et al. 2020), such as safety boxes and collision avoidance algorithms.

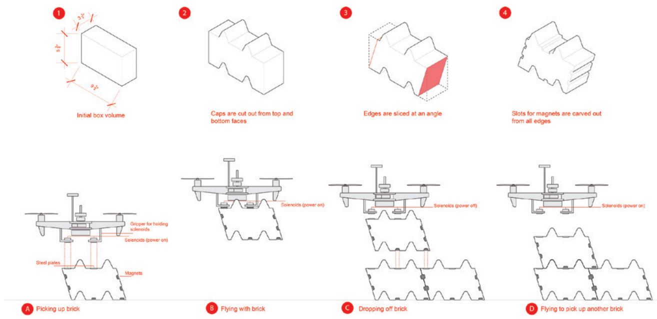
We developed a quadcopter drone that meets the specific functional and performance requirements that cannot be achieved by off-the-shelf products. Such requirements include a 500-gram payload capacity, integration with RL high-level flight commands, customizable flight control, telemetry hardware integration, and expansion points for attaching a gripper arm.

The drone is built on a lightweight F450-family frame with various attachment points for custom hardware integration. Motors are chosen to provide sufficient power to lift the 1,500 grams of drone empty weight as well as an additional 500 grams of payload. The selected battery pack can provide approximately 15 minutes of fly time at this




7 Four snapshots of the facade coating simulation, representing another use case scenario of the multi-drone framework.

7




8

9

8 Design development of drone-compatible building components and illustration of magnetic-based drone gripping mechanism.

9 An illustration of the long-term vision for a distributed robotically assisted construction framework.

payload. A GPS/compass module is used to maintain real-time positioning in open spaces. To improve accuracy in real-time localization, this positioning system is combined with motion capture markers. A Pixhawk flight controller connected to an onboard radio transmitter for the flight control enables manual override through a handheld radio remote controller unit. In addition to the flight controller, a Raspberry Pi 4 board is used to relay high-level flight control commands between the RL algorithm running on a remote computer and the Pixhawk board controller via the Wi-Fi network. To monitor real-time flight data, a radio telemetry device connected to the Pixhawk controller collects and relays flight data to a ground-based computer. In addition, the drone is equipped with an underside custom arm to accurately pick up, transport, and release blocks. The arm is designed for holding solenoid electromagnet modules for picking blocks. Two electromagnets are attached to the two ends of the custom arm.

Custom-designed bricks have been designed and fabricated for use in this experiment. Each brick is equipped with a 1 × 1 in steel plate on its pickup point. A solenoid actuator controls the behavior of electromagnets on the arm to attract and release the steel plate on the designated pick and place points, respectively. In order to account for discrepancies between the drones' simulated location and their real location during the operation, these customized bricks have slanted faces, pointed caps, and embedded magnets on all faces so that the blocks can self-align into position when placed (Fig. 8).

## DISCUSSION

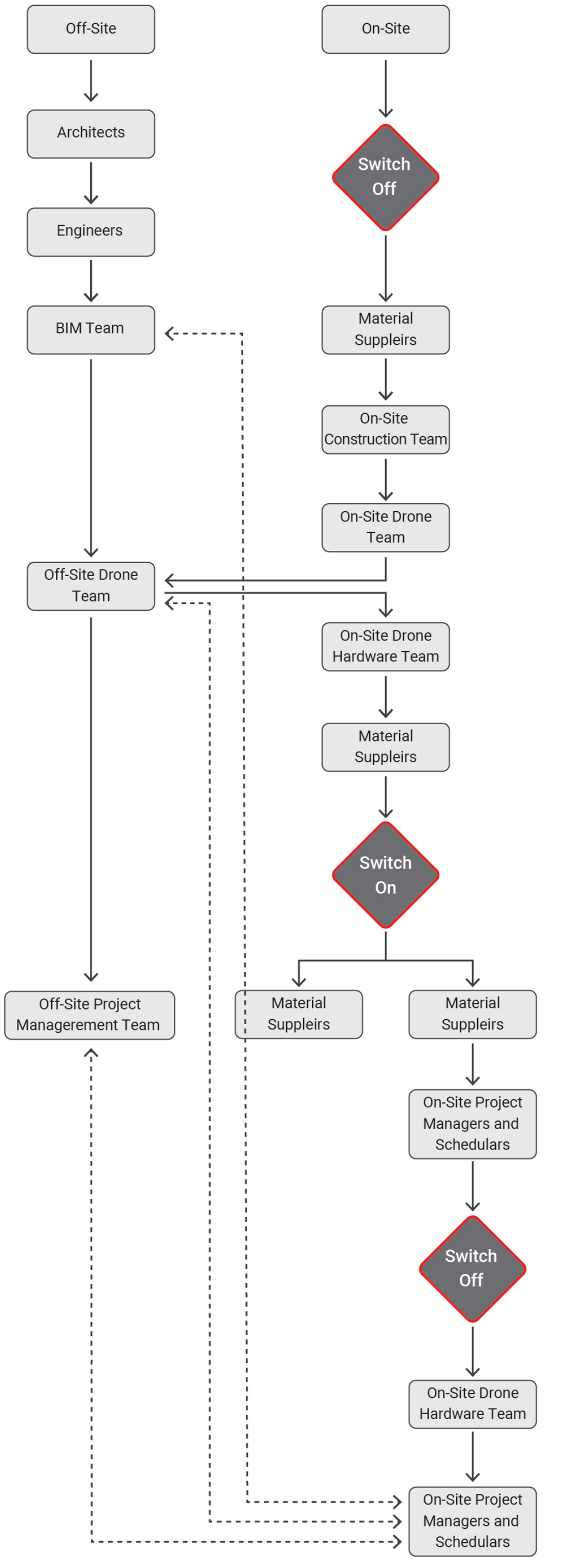
### Overview of Contributions

This paper presented a framework based on reinforcement learning techniques under a "centralized learning, decentralized execution" paradigm that enables a more flexible and generalizable approach to robotically assisted construction. We developed and trained a reinforcement algorithm for semiautonomous path planning for UAVs in dynamic environments, showing through simulations the successful, collision-free execution of two construction activities: bricklaying and spray-coating. Improving on previous literature, the RL model serving as the framework's back-end proved to be successful with a variable number of drones, and without prior knowledge of the obstacles in the environment. In addition to these simulations, we discussed progress towards a hardware prototype for testing the framework in practice.

### Limitations and Next Steps

While the real-world testing of the framework has been delayed by the pandemic, we expect to deploy it and document results in the spring of 2021. While the RL algorithm proved to be successful in the simulations, we are not expecting to find one universal trained model to address all construction scenarios and environments. Instead, our framework entails the need to develop a growing library of RL models accounting for the specific characteristics of different construction environments and tasks. This will require close studies of such environments. At a technical level, we intend to optimize the framework for on-the-edge computing using the Raspberry Pi board in order to generate high-level flight commands onboard. In addition, we intend to replace the temporary motion capture-based localization method with lidar scanners to accurately scan the environment and detect peer drones, obstacles, and human agents in real time.

While our interest is in creating systems that support "real" activities on site, the tasks we accomplished in this paper are quite simple compared to those taking place on actual construction sites. However, we believe that our RL framework provides the flexibility to incorporate a greater




10

10 Long-term construction workflow pipeline showing different human roles involved and the process for implementing an advanced future framework in an unstructured environment.

degree of complexity, including changing environmental conditions, human workers, machinery, and scaffolding. Further work needs to be done to be able to test for these contingencies.

As discussed before, the limited payload capacity of drones limits their applicability to actual construction scenarios. This could be addressed by including new macro actions that entail the collaboration of multiple drones in carrying a single component. While this scenario is interesting, our framework is flexible, and we are currently in the process of adapting it to other types of robots and autonomous vehicles with greater carrying capacity.

### Human-Robot Ecosystems of Construction: A Broader Vision for the Future

The framework described in this paper contributes to current efforts towards robotically assisted construction, and towards a longer-term vision of human-machine construction ecosystems where human builders are supported by semiautonomous technologies. The pipeline diagram in Figure 10 conceptualizes such a vision by speculating on how roles might be redistributed across human experts and nonhuman actors in a design and construction context. Off-site teams consisting of component manufacturers, architects, engineers, and the BIM team—all of whom collectively design and model the structure to be built—might be joined by simulation and robotics experts analyzing the building model and using it to identify and define tasks apt for robotic execution. On-site teams including conventional construction roles such as construction laborers, project supervisors, and managers, material suppliers, as well as a dedicated team of drone hardware experts, might be joined by robotically assisted construction experts in charge of programming, supervising, and maintaining robotic systems on site. Shaping these emergent roles and contributing to hybrid and humane construction ecosystems is an important task for researchers at the intersection of architecture and computing today.

## ACKNOWLEDGMENTS

The authors would like to thank Yanwen Dong, Michael Hasey, and Willa Yang from the MS Computational Design program at CMU School of Architecture who contributed to this project. We would also like to express our gratitude towards the Design Fabrication Lab (DFab) at the School of Architecture at Carnegie Mellon University for their assistance with fabrication.

## REFERENCES


Ammar, Mirjan. 2016. "Aerial Construction: Robotic Fabrication of Tensile Structures with Flying Machines." PhD dissertation, ETH Zurich.

Augugliaro, Federico, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W. Mueller, Jan Sebastian Willmann, et al. 2014. "The Flight Assembled Architecture Installation: Cooperative Construction with Flying Machines." *IEEE Control Systems Magazine* 34 (4): 46–64.

Chaltiel, Stephanie, Maite Bravo, Sebastien Goessens, Pierre Latteur, Masoumeh Mansouri, and Ismail Ahmad. 2018. "Dry and Liquid Clay Mix Drone Spraying for Bioshotcrete." In *Creativity in Structural Design [Proceedings of the IASS Symposium 2018]*, Boston, MA, 16–20 July 2018, edited by Caitlin Mueller and Sigrid Adriaenssens, 1–8. International Association for Shell and Spatial Structures (IASS).

Chaltiel, Stephanie, Maite Bravo, and Ibrahim Abdullah. 2018. "Adaptive Strategies for Mud Shell Robotic Fabrication." *International Journal of Environmental Science & Sustainable Development* 3 (2): 64–74.

Computational Design Lab. 2020. "Reinforcement Learning to Support Scalable Multi-Drone Construction in Dynamic Env. Simulation." https://youtu.be/oe1T1j5nVqM.

Dackiw, Jean-Nicolas, Andrzej Foltman, Soroush Garivani, Keith Kaseman, and Aldo Sollazzo. 2019. "Cyber-Physical UAV Navigation and Operation 1 Waypoint Navigation Interface for Drones through Augmented Reality." In *ACADIA 19: Ubiquity and Autonomy [Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)]*, Austin, TX, 21–26 October 2019, edited by K. Bieg, D. Briscoe, and C. Odom, 360–367. CUMINCAD.

Fang, Zhihao. 2020. "Towards Multi-Drone Autonomous Construction via Deep Reinforcement Learning." Master's thesis. Carnegie Mellon University.

Felbrich, Benjamin, Nikolas Frueh, Marshall Prado, Saman Saffarian, James Solly, Lauren Vasey, Jan Knippers, and Achim Menges. 2017. "Multi-Machine Fabrication: An Integrative Design Process Utilising an Autonomous UAV and Industrial Robots for the Fabrication of Long Span Composite Structures." In A*CADIA 2017: Disciplines and Disruption [Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)]*, Cambridge, MA, 2–4 November 2017, edited by T. Nagakura, S. Tibbits, M. Ibañez, and C. Mueller, 248–259. CUMINCAD.

Goessens, Sébastien, Caitlin Mueller, and Pierre Latteur. 2018. "Feasibility Study for Drone-Based Masonry Construction of Real-Scale Structures." *Automation in Construction* 94: 458–480.

Hunt, Graham, Faidon Mitzalis, Talib Alhinai, Paul A. Hooper, and Mirko Kovac. 2014. "3D Printing with Flying Robots." In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 31 May–7 June 2014, 4493–4499. IEEE.

Intel. n.d. "Intel Drone Light Show Fact Sheet." Accessed November 10, 2020. https://www.intel.com/content/www/us/en/technology-innovation/drone-light-show-fact-sheet.html.

Lindsey, Quentin, Daniel Mellinger, and Vijay Kumar. 2011. "Construction of Cubic Structures with Quadrotor Teams." In *Proceedings of Robotics: Science and Systems (RSS). Vol. VII*. Cambridge, MA: MIT Press.

Long, Pinxin, Tingxiang Fan, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. 2018. "Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning." ArXiv:1709.10082 [Cs], May. http://arxiv.org/abs/1709.10082.

Lowe, Ryan, Yi I. Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments." *Advances in Neural Information Processing Systems* 30: 6379–6390.

Lucieer, Arko, Darren Turner, Diana H. King, and Sharon A. Robinson. 2014. "Using an Unmanned Aerial Vehicle (UAV) to Capture Micro-Topography of Antarctic Moss Beds." *International Journal of Applied Earth Observation and Geoinformation*, *Special Issue on Polar Remote Sensing* 27 (part A): 53–62. https://doi.org/10.1016/j.jag.2013.05.011.

Mademlis, Ioannis, Nikos Nikolaidis, Anastasios Tefas, Ioannis Pitas, Tilman Wagner, and Alberto Messina. 2018. "Autonomous Unmanned Aerial Vehicles Filming in Dynamic Unstructured Outdoor Environments." *IEEE Signal Processing Magazine* 36 (December): 147–153. https://doi.org/10.1109/MSP.2018.2875190.

Maes, Wouter H., and Kathy Steppe. 2019. "Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture." *Trends in Plant Science* 24 (2): 152–164. https://doi.org/10.1016/j.tplants.2018.11.007.

Marris, Emma. 2013. "Fly, and Bring Me Data: Unmanned Aerial Vehicles Are Poised to Take Off as Popular Tools for Scientific Research." *Nature* 498 (7453): 156–159.

Nägeli, Tobias, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. 2017. "Real-Time Planning for Automated Multi-View Drone Cinematography." *ACM Transactions on Graphics* 36 (4): 132:1–132:10. https://doi.org/10.1145/3072959.3073712.

Nishar, Abdul, Steve Richards, Dan Breen, John Robertson, and Barbara Breen. 2016. "Thermal Infrared Imaging of Geothermal Environments and by an Unmanned Aerial Vehicle (UAV): A Case Study of the Wairakei – Tauhara Geothermal Field, Taupo, New Zealand." *Renewable Energy* 86 (February): 1256–1264. https://doi.org/10.1016/j.renene.2015.09.042.

Qie, H., D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang. 2019. "Joint Optimization of Multi-UAV Target Assignment and Path Planning Based on Multi-Agent Reinforcement Learning." *IEEE Access* 7: 146264–146272. https://doi.org/10.1109/ACCESS.2019.2943253.

Romano, Matthew, Yuxin Chen, Owen Marshall, and Ella Atkins. 2019. "Nailed It: Autonomous Roofing with a Nailgun-Equipped Octocopter." ArXiv:1909.08162 [Cs], September. http://arxiv.org/abs/1909.08162.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. "Proximal Policy Optimization Algorithms." ArXiv:1707.06347 [Cs], August. http://arxiv.org/abs/1707.06347.

Shahmoradi, Javad, Elahe Talebi, Pedram Roghanchi, Mostafa Hassanalian. 2020 "A Comprehensive Review of Applications of Drone Technology in the Mining Industry." *Drones* 4 (3): 34.

Silver, David. 2005. "Cooperative Pathfinding." *Aiide* 1: 117–122.

Wang, Ko-Hsin Cindy, and Adi Boeta. 2008. "Fast and Memory-Efficient Multi-Agent Pathfinding." In *ICAPS'08 [Proceedings of the Eighteenth International Conference on International Conference on Automated Planning and Scheduling]*, Sydney, Australia, 14–18 September 2008, edited by Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, 380–387. AAAI Press.

Wang, Ko-Hsin Cindy, and Adi Boeta. 2011. "MAPP: a Scalable Multi-Agent Path Planning Algorithm with Tractability and Completeness Guarantees." *Journal of Artificial Intelligence Research* 24 (1): 55–90. https://dl.acm.org/doi/10.5555/2208436.2208439

Wood, Dylan, Maria Yablonina, Miguel Aflalo, Jingcheng Chen, Behrooz Tahanzadeh, and Achim Menges. 2018. "Cyber Physical Macro Material as a UAV [Re]Configurable Architectural System." In *Robotic Fabrication in Architecture, Art and Design 2018. ROBARCH 2018*, edited by J. Willmann, P. Block, M. Hutter, K. Byrne, and T. Schork. Springer: Cham. https://doi.org/10.1007/978-3-319-92294-2_25.

Zhang, Chunhua, and John M. Kovacs. 2012. "The Application of Small Unmanned Aerial Systems for Precision Agriculture: A Review." *Precision Agriculture* 13 (6): 693–712. https://doi.org/10.1007/s11119-012-9274-5.

## IMAGE CREDITS



**Zhihao Fang** is a recent graduate of Carnegie Mellon's Master's in Computational Design program.

**Yuning Wu** is a current PhD student in the Computational Design program, CoDe LAB, and a Master's candidate in Machine Learning at Carnegie Mellon University.

**Ammar Hassonjee** is a current fourth-year student at Carnegie Mellon University in the Bachelor of Architecture program also minoring in Human-Computer Interaction.

**Ardavan Bidgoli** is a current PhD candidate and Robotics Fellow in Carnegie Mellon's Computational Design Program, CoDe LAB.

**Daniel Cardoso-Llach PhD** is an Associate Professor of Computational Design at Carnegie Mellon and the track chair of the Computational Design program as well as a codirector of CoDe LAB.