



# Simulating Multi-robot Construction in ARGoS

Michael Allwright<sup>1</sup>✉, Navneet Bhalla<sup>2</sup>, Carlo Pincioli<sup>3</sup>,  
and Marco Dorigo<sup>1</sup>✉

<sup>1</sup> IRIDIA, Université Libre de Bruxelles, Brussels, Belgium  
[{mallwrig,mdorigo}@ulb.ac.be](mailto:{mallwrig,mdorigo}@ulb.ac.be)

<sup>2</sup> Department of Computer Science, University College London, London, UK  
[n.bhalla@cs.ucl.ac.uk](mailto:n.bhalla@cs.ucl.ac.uk)

<sup>3</sup> Department of Computer Science/Robotics Engineering,  
Worcester Polytechnic Institute, Worcester, MA, USA  
[cpincioli@wpi.edu](mailto:cpincioli@wpi.edu)

**Abstract.** Running hardware-based experiments in multi-robot construction is an expensive and time-consuming endeavor. Furthermore, it is difficult to disseminate the results from hardware-based experiments in a way that other researchers can build upon. In this paper, we present a number of plug-ins for a multi-robot simulator that we have developed to enable a high-fidelity simulation of the multi-robot construction systems typically found in laboratory settings. We validate these plug-ins qualitatively by repeating a hardware-based experiment in simulation where a single robot assembles a staircase from blocks [1]. We then show how we can use the plug-ins to scale up the complexity of the construction scenario and demonstrate multi-robot construction in simulation. To enable other researchers to replicate our experiments and to promote collaboration, we have made our plug-ins open source.

## 1 Introduction

In our research, we study how the coordination mechanisms used by social insects can be applied to a swarm of robots whose task is to collectively assemble structures. This research direction is motivated by the robustness, parallelism, and adaptivity to the environment that social insects exhibit as they construct their nests [4,5]. Construction by social insects is decentralized and is coordinated through stigmergic communication [7], where the probability of an insect performing a construction action is a function of the previously performed construction actions as perceived by that insect [19]. These construction actions may have been performed by the same insect or by other insects.

Using a single robot, we previously studied how stigmergic communication could be used to guide the construction of a staircase [1]. The robot in this study located unused blocks in the environment before attaching them to the staircase according to the arrangement of the blocks already in the structure. The development of this hardware and the running of the experiments, however,

were both expensive and time-consuming endeavors. Furthermore, since other researchers do not have access to our hardware, it is difficult for them to replicate our experiments and to build upon our results. These challenges make it difficult to answer a number of open research questions regarding, for example, whether it is possible to encode arbitrary structures as sets of assembly rules for swarms of robots and how these encodings can be compared in terms of their robustness and their ability to be executed in parallel by multiple robots.

To address these challenges and to answer these open research questions, we present in this paper a number of plug-ins for ARGoS (Autonomous Robots Go Swarming), an open-source, multi-robot simulator [14]. We show in this paper how we can use these plug-ins to qualitatively reproduce our hardware-based results in simulation and how we can use the plug-ins to scale up to and to study multi-robot construction in simulation. To enable other researchers in the swarm robotics community to build upon our results, we have made these plug-ins open source. In this paper, the presented validation of the plug-ins is strictly qualitative and further quantitative evaluation is required to identify and to mitigate any reality gap found between our hardware and our simulation.

This paper is organized as follows. In the next section, we discuss current practices regarding the use of simulation in the multi-robot construction literature. In Sect. 3, we introduce ARGoS and summarize the plug-ins which we have developed to enable the simulation of multi-robot construction systems. Section 4 presents a case study in which we demonstrate how the plug-ins can be used to simulate multi-robot construction. In the final section, we conclude this paper by discussing the open research questions that we would like to answer in our future work.

## 2 Background

While previous work in multi-robot construction has used simulation to mitigate some of the challenges of hardware, simulations are often limited to lattice-based two- or three-dimensional worlds where the interaction dynamics is unrealistic as are the perceptual capabilities of the robots [17, 18, 21, 22].

In the work by Jones and Matarić, a high-fidelity simulator was used, but the simulation was tweaked so that the robots simply requested the simulation engine to add construction material directly to the structure, sidestepping several challenges present in a multi-robot construction system such as manipulating the construction materials and avoiding collisions with other robots during construction [8]. In two other cases, results from simulation were presented but not discussed. Werfel et al., for example, included two videos from simulation in their supplementary material but did not discuss these results in either the main article or in the supplementary report [23]. Lindsey et al. also presented results from simulation without detailing their nature [10].

While the aforementioned works significantly contributed to the state of the art in multi-robot construction, the means to replicate the experiments is problematic since other researchers are unable to build upon the presented results.

This lack of a means to replicate experiments and to build upon previous results is a significant barrier to pushing the state of the art in multi-robot construction further forward. One way to remove this barrier is by using an open-source, high-fidelity simulator that can run multi-robot construction experiments. This simulator would need to support the range of sensors, actuators, and mechanisms that are commonly used by multi-robot construction systems in laboratory settings. For example, the simulator would need to support camera systems that can detect tags or light-emitting diodes (LEDs), rangefinders, prismatic and revolute actuators, and magnetism. Furthermore, the simulator would ideally provide good performance when running experiments with large numbers of robots.

### 3 Contributions to the ARGoS Simulator

ARGoS (Autonomous Robots Go Swarming) is a multi-robot simulator built around the principle of tunable accuracy. In contrast to general-purpose simulators such as USARSim [6], Gazebo [9], and Webots [11] whose performance quickly degrades with increasing numbers of robots, benchmarks for ARGoS have demonstrated the simulation of 10,000 robots running faster than real time [14]. ARGoS is also widely used in the swarm robotics community by more than 21 robotics laboratories in 13 different countries<sup>1</sup>.

At the time of writing, however, the latest release of ARGoS (3.0.0-beta49) only comes with plug-ins for two-dimensional dynamics and a limited number of generic sensors and actuators. The implementation of new robots in ARGoS also requires a significant amount of coding and the compilation of new classes that represent a robot's sensors, actuators, physics model, and visualization. However, ARGoS has been purposefully designed to be modular. Most aspects of a simulation can be overridden and developers can add functionality to the simulator in the form of new sensors, actuators, visualizations, physics engines.

We have enhanced three of the plug-ins from the ARGoS core package and have implemented two completely new plug-ins that enable the simulation of three-dimensional dynamics and the rapid prototyping of new robots. The plug-ins are designed to support a high-fidelity simulation of the dynamics, sensors, actuators, and other mechanisms commonly used by the multi-robot construction systems found in robotics laboratories. By making these plug-ins open source and by including them in the ARGoS core package (from 3.0.0-beta50), we hope to increase the pace of the research in multi-robot construction by enabling researchers to implement their own multi-robot construction systems in simulation and by enabling other researchers to replicate the presented results.

In the remainder of this section, we summarize our contributions to ARGoS in the context of multi-robot construction. The plug-ins that we have developed, however, are highly flexible and their potential applications go significantly beyond multi-robot construction. For a more in-depth discussion of the plug-ins including examples on how to use them, we refer the reader to the supplementary technical report for this paper [2].

---

<sup>1</sup> Users of ARGoS – <http://www.argos-sim.info/users.php>.

### 3.1 Enhancements to the Entities Plug-in

In the ARGoS simulator, entities represent the components of a robot in the simulation. This representation allows plug-ins such as visualizations, physics engines, and controllers to read and to write the state of these components. We have added four new entities to the entities plug-in. The simulation of magnetism is enabled by introducing the magnet entity. Magnetism can be used in multi-robot construction systems for both manipulating the building materials [15, 16] and for the self-alignment of the building materials inside a structure [13, 24]. A magnet entity is capable of representing a permanent magnet, an electromagnet, or a semi-permanent electromagnet. We have also implemented a radio entity which represents a simple omnidirectional radio that can broadcast messages to other radios within a given range. This radio entity is suitable for simulating communication between robots and for simulating communication between robots and a building material [22, 25]. In contrast to the range-and-bearing entity in ARGoS, the radio entity is not based on infrared light and its messages are not obstructed by other entities in the simulation. A tag and a directional light-emitting diode (LED) entity were also implemented and can be attached to any object in the simulation. The directional LED entity is based on the standard LED entity in ARGoS and includes two additional attributes, namely an orientation and an observable angle. These entities can be detected by camera sensors and can be used by robots to locate building materials.

### 3.2 Enhancements to the Media Plug-in

Media are used by the simulator to manage entities during a simulation. At a minimum, the implementation of a medium consists of a data structure which is typically queried by a sensor and modified by an actuator. We have defined three new mediums, namely the radio, tag, and directional LED medium. These mediums are required to support the sensors and actuators for a robot, so that it can interact with the radio, tag, and directional LED entities in a simulation.

### 3.3 Enhancements to the Generic Robot Plug-in

The generic robot plug-in defines generic sensors and actuators that can be used by any robot. We have enhanced this plug-in by implementing a generic camera framework that allows any number of cameras to be attached to a robot. For performance reasons, we have chosen to simulate computer vision algorithms rather than rendering the simulated environment from the perspective of each robot. For example, instead of rendering the tags in a simulation onto a virtual buffer and passing these pixels to a tag detection algorithm, we use a simulated tag detection algorithm that simply queries the tag medium and directly calculates the pixel coordinates of a tag's corners. We have provided three simulated computer vision algorithms for detecting tags, directional LEDs, and the standard LEDs in ARGoS. The output from the simulated tag detection algorithm is consistent with the output from the AprilTag algorithm and enables a robot

to estimate the pose of a detected tag [12]. In addition to the generic camera framework, we have also added a generic radio sensor and actuator. In conjunction with the radio entity, this sensor and actuator allow robots to send messages and to receive messages from nearby robots or building material.

### 3.4 The Three-Dimensional Dynamics Plug-in

To simulate the interaction dynamics that occur during multi-robot construction, we have created a new physics engine plug-in for ARGoS, which is a wrapper around Bullet Physics<sup>2</sup>. This plug-in replaces the deprecated three-dimensional dynamics plug-in from previous versions of ARGoS, which was based on ODE.

The three-dimensional dynamics plug-in provides a number of helper classes for quickly creating new robots. The plug-in also provides a configurable floor, and tunable gravity and magnetism. Magnetic forces and torques are applied to bodies in the simulation where a corresponding magnet entity has been defined. The forces and torques are calculated using a variant of Thomaszewski's algorithm [20], where each magnet is approximated by a single dipole.

### 3.5 The Prototyping Plug-in

The prototyping plug-in enables the implementation of new robots and building materials in ARGoS without the need to manually code and compile new classes. The plug-in defines a new robot that is entirely described by ARGoS's experiment configuration file.

The plug-in provides new entities for describing links and joints. Links are defined by their mass and geometry, which currently can be either a box, a cylinder, or a sphere. Joints specify how two links are connected to each other. There are four joint types currently supported: fixed, spherical, prismatic, and revolute. For prismatic and revolute joints, it is possible to limit the range of a joint's motion and to assign a sensor and an actuator to the joint. The default sensor can measure either joint position or joint velocity and the default actuator can be configured to use either position control or velocity control. The plug-in also provides a physics model for the three-dimensional dynamics plug-in and a visualization model for the Qt-OpenGL plug-in.

Sensors, actuators, and other robot components such as tags, directional LEDs, standard LEDs, radios, and cameras are also defined in the experiment configuration file and can be attached to any link in the robot.

## 4 Case Study: Multi-robot Construction

In this section, we demonstrate how our plug-ins for the ARGoS simulator enable us to qualitatively reproduce our hardware-based experiment, in which a single robot constructed a staircase [1]. In addition to reproducing our hardware-based

---

<sup>2</sup> Bullet Physics – <http://bulletphysics.org/>.

experiment, we show in simulation a more complicated construction scenario involving four robots that collectively assemble a stepped pyramid.

We have used the prototyping plug-in to model the robot and the building material in simulation. The models and sample controllers for the robot and the building material are available on the supplementary material website for this paper [3].

#### 4.1 Summary of the Hardware and the Control Software

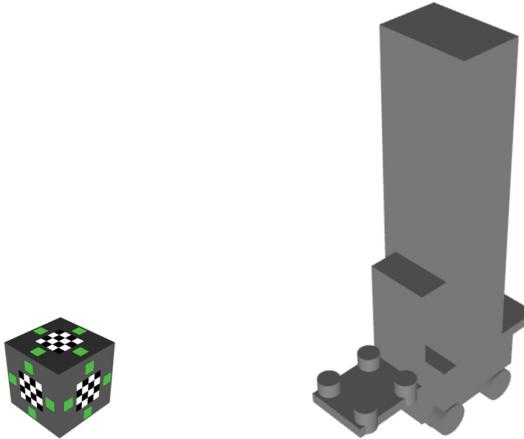
The hardware of our autonomous construction system consists of a building material known as stigmeric blocks and an autonomous robot, which assembles the blocks into structures [1]. The blocks are cubes that contain on each face: (i) a tag, (ii) a near-field communication (NFC) interface, and (iii) four multi-color light-emitting diodes (LEDs). The tags are used by the robot to estimate the pose of a block. Spherical magnets are assembled into the corners of the blocks to provide self-alignment and so that the blocks can be picked up by a robot.

The robot consists of two tracks (treads) that form a differential drive, allowing the robot to move around its environment. Using its camera, the robot can locate blocks by detecting their tags and can identify the colors of the LEDs on a block. Semi-permanent electromagnets are attached to an end effector, which enable the robot to pick up a block and to attach it to a structure.

The control software for the robot is implemented using a finite state machine (FSM). In this FSM, a robot starts by locating an unused block (a single block whose LEDs are not illuminated). The robot then searches for, approaches, and inspects structures in its environment (where a structure is defined by one or more blocks whose LEDs are illuminated). If the arrangement and LED colors of the blocks in a structure match a predefined rule, the robot uses its NFC interface to configure the LED colors on the unused block and attaches the unused block to the structure with respect to the matched rule. Since the attachment of this block modifies the arrangement and LED colors of the blocks in the structure, a feedback loop emerges where other predefined rules can now be matched and can continue to coordinate the assembly of a structure. At the time of writing, the robot can only modify the color of the block that is attached to its end effector. However, we are currently enhancing the block's software to allow block-to-block communication that will enable a recently placed block to update the LED colors of adjacent blocks in a structure.

#### 4.2 Modeling the Hardware in Simulation

For the simulation work presented in this paper, both the stigmeric block and the autonomous robot are implemented using the prototyping plug-in. The main body of a block is modeled as a box-shaped link with side lengths of 55 mm. Eight additional sphere-shaped links are defined inside the block to simulate the freely-rotating spherical magnets. Since the magnetism provided by the three-dimensional dynamics plug-in uses a single-dipole approximation, we have tuned



**Fig. 1.** Visualization of a stigmergic block and an autonomous robot in the ARGoS simulator: these visualizations are generated automatically by the Qt-OpenGL visualization model from the prototyping plug-in

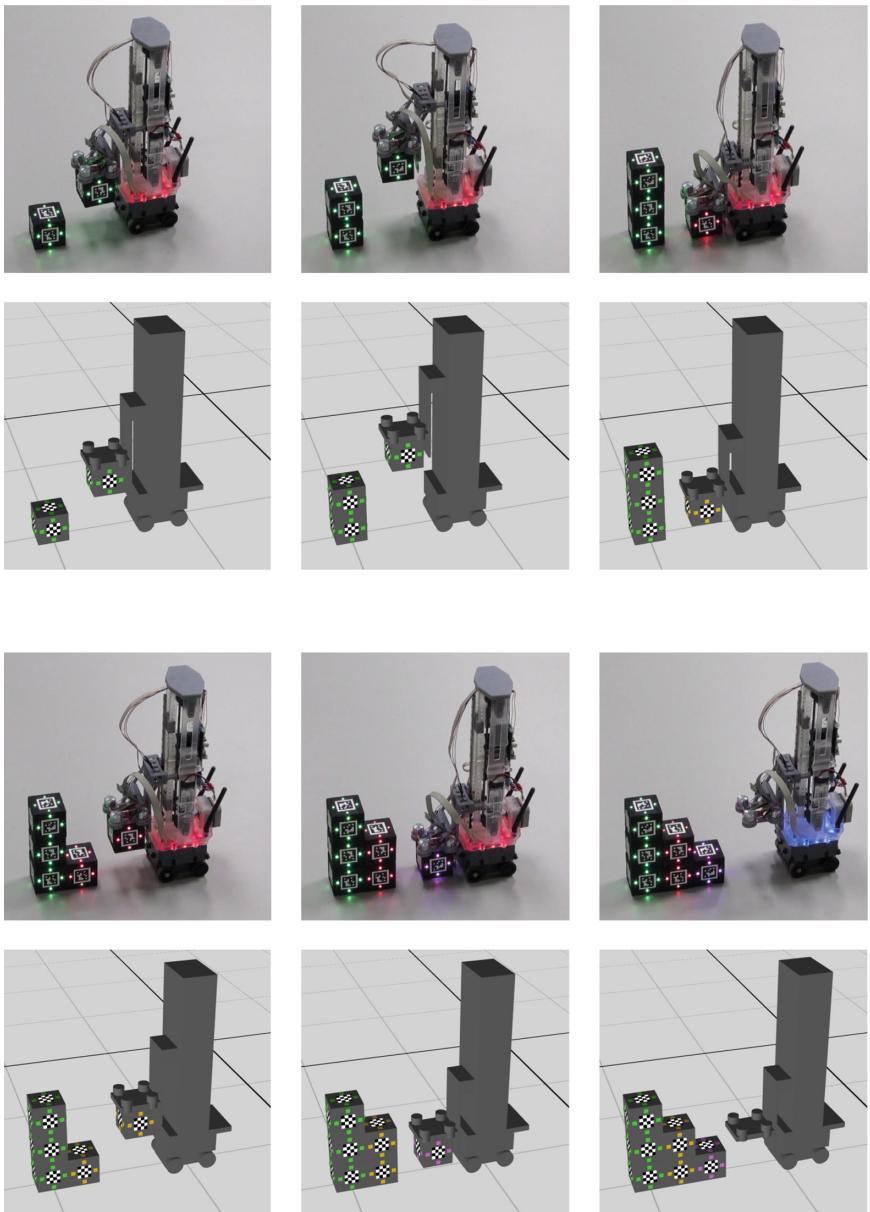
the strength of the magnets based on empirical testing to match the characteristics of the hardware. A tag, radio, and four directional LEDs are added to each face of the block to complete its model in simulation.

The model of the robot consists of 15 links and 15 joints. Four of the links and four of the joints are used to simulate the tracks (treads) of the robot. An additional joint represents the end effector of a robot which is lowered and raised in order to pick up and to assemble unused blocks into a structure. The remaining links are used to simulate the geometry of the robot and the joints between them are fixed. The robot is configured with a camera, rangefinders, radios, semi-permanent electromagnets, and joint sensors and actuators to match the capabilities of the hardware.

These descriptions of the stigmergic block and the autonomous robot are provided to the prototyping plug-in via the experiment configuration file. The prototyping plug-in parses these descriptions and creates models of these objects using the three-dimensional dynamics plug-in and the Qt-OpenGL visualization plug-in (see Fig. 1).

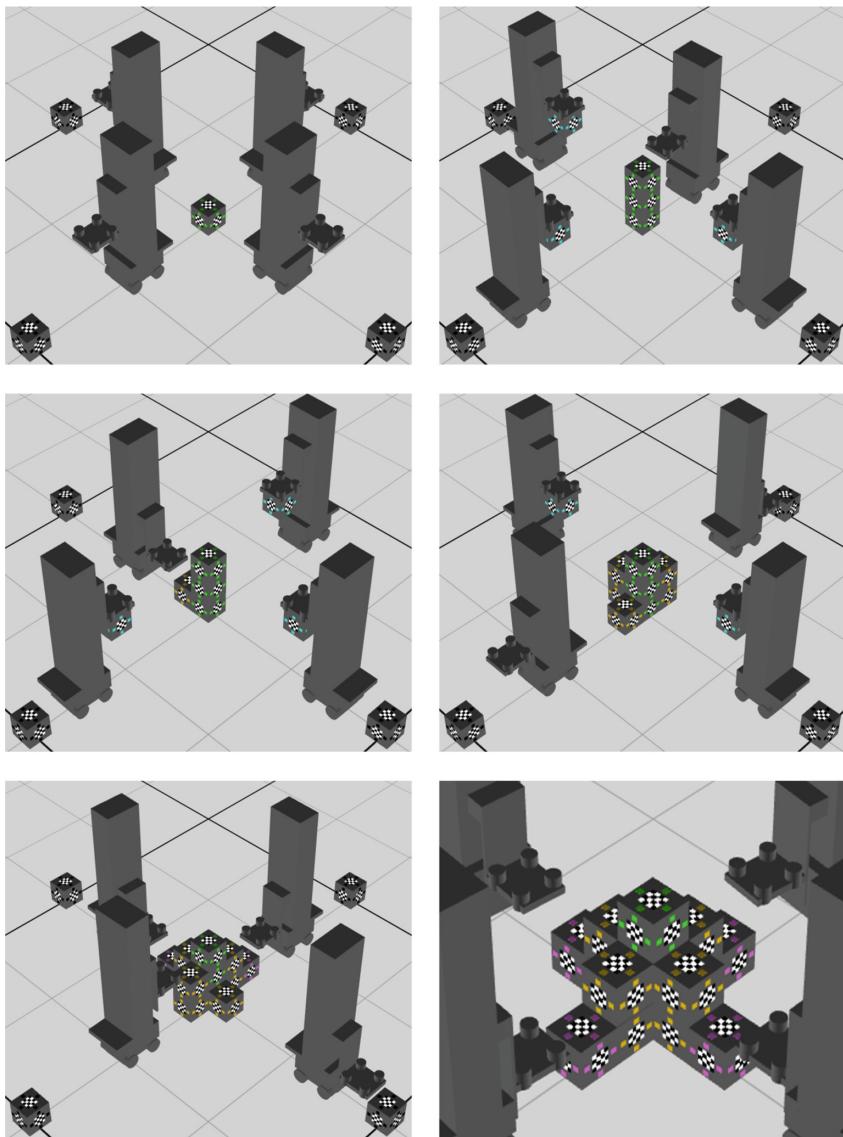
### 4.3 Reproducing the Hardware Results

In our previous work, we demonstrated the construction of a staircase using a single robot [1]. The staircase consisted of three columns of blocks descending in height with the blocks in each column illuminated with a distinct LED color. The highest column contained three blocks with the LEDs set to green, the middle column contained two blocks with the LEDs set to red, and the last column contained a single block with its LEDs set to violet. To replicate this demonstration in simulation, we create an instance of the finite state machine used by



**Fig. 2.** Snapshots comparing hardware and simulation results

the hardware inside an ARGoS controller. The controller acts as a wrapper that synchronizes the state machine's data with the sensors and actuators provided by the prototyping and generic robot plug-ins. Snapshots of this demonstration on the hardware and in simulation are shown side-by-side in Fig. 2. We consider



**Fig. 3.** Snapshots of four robots building a stepped pyramid in simulation

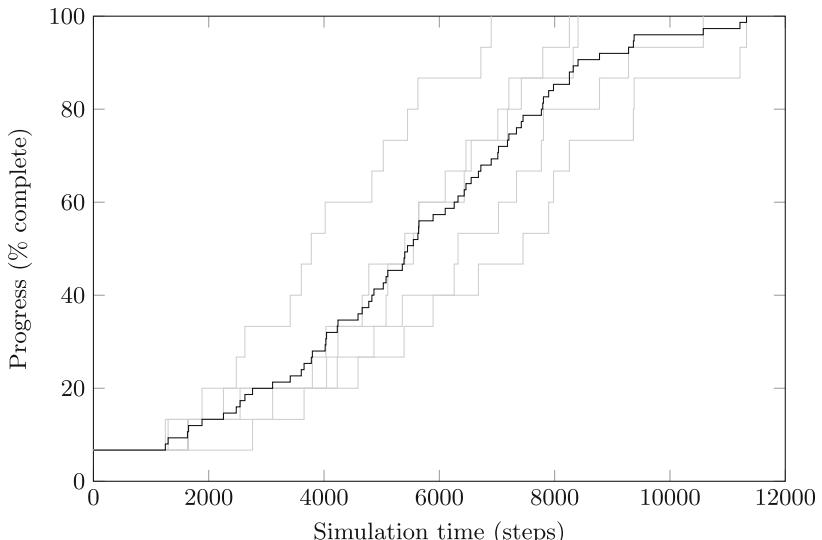
the ability to reproduce the hardware results in simulation as a qualitative validation of the presented plug-ins and of the models of the stigmergic block and the autonomous robot. However, further quantitative testing will be required to identify the extent of any reality gap and to mitigate it. A video of this demonstration is available on the supplementary material website for this paper [3].

#### 4.4 Scaling up to Multi-robot Construction

Following the qualitative validation of our plug-ins and models, we are now able to scale up to a more complex construction scenario where four robots are used to assemble a stepped pyramid in simulation. This demonstration was only possible in simulation due to reliability issues with the robot's drive system.

The stepped pyramid is, in essence, four of the staircases from the previous demonstration which share a common central column. We leverage this symmetry and make only two minor modifications to the control software used by a robot, namely (i) we introduce a random delay state so that the robots do not all approach the structure at the same time and (ii) we introduce a basic collision avoidance mechanism that detects if another robot is attempting to attach a block to the central column of the stepped pyramid. Snapshots of this demonstration running in simulation are shown in Fig. 3. A video of the complete demonstration is available on the supplementary material website [3].

We ran this experiment in ARGoS and gathered data on the construction throughput for five runs. The plot in Fig. 4 shows the construction progress during each of these runs in gray and the average of the five runs in black. From the data, we observe that the construction throughput is initially low due to the central construction site being saturated as all robots attempt to assemble the central column. Following the construction of the central column, however, the construction throughput increases as the robots start building the wings of the pyramid in parallel. The rate of construction then decreases again towards the end of the construction task since the robots no longer have any work to do.



**Fig. 4.** Construction progress for the stepped pyramid from five runs (shown in gray) and the average progress across all runs (shown in black)

## 5 Conclusions

In this paper, we have discussed and demonstrated the use of five plug-ins that enable the simulation of our autonomous construction system in ARGoS, a modular, multi-robot simulator. These plug-ins have allowed us to qualitatively reproduce our hardware-based results in simulation and to investigate a more complicated construction scenario.

The plug-ins presented in this paper have been made open source and have been integrated into the ARGoS core package (from 3.0.0-beta50). These plug-ins aim to enable a high-fidelity simulation of the multi-robot construction systems that are commonly found in laboratory settings. We hope that our contribution increases the pace of multi-robot construction research by enabling researchers to disseminate their results and the means to reproduce them more effectively.

In future work, we will use these plug-ins as part of a workflow that aims to answer a number of open research questions regarding the representation of arbitrary structures as assembly rule sets for swarms of robots. We will also investigate and compare different representations of a given structure in terms of its robustness and its ability to be built in parallel by multiple robots.

**Acknowledgments.** Michael Allwright was supported by the Australian Government through the Endeavour Scholarships and Fellowships Program. Navneet Bhalla was partially supported by a postdoctoral fellowship from the Natural Sciences and Engineering Research Council of Canada. Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Director. The work presented in this paper was partially supported by the FLAG-ERA project RoboCom++ and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement number 681872). We would like to thank Haitham El-faham and Weixu Zhu for their help with implementing and testing the magnetism code in the three-dimensional dynamics plug-in.

## References

1. Allwright, M., Bhalla, N., Dorigo, M.: Structure and markings as stimuli for autonomous construction. In: Proceedings of the Eighteenth International Conference on Advanced Robotics, pp. 296–302. IEEE (2017). <https://doi.org/10.1109/icar.2017.8023623>
2. Allwright, M., Bhalla, N., Pincioli, C., Dorigo, M.: ARGoS plug-ins for experiments in autonomous construction. Technical report TR/IRIDIA/2018-007, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium (2018)
3. Allwright, M., Bhalla, N., Pincioli, C., Dorigo, M.: Simulating multi-robot construction in ARGoS (supplementary material website) (2018). <http://iridia.ulb.ac.be/supp/IridiaSupp2017-004/index.html>
4. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1999)
5. Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in Biological Systems. Princeton University Press, Princeton (2001)

6. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: USARSim: a robot simulator for research and education. In: 2007 IEEE International Conference on Robotics and Automation, pp. 1400–1405. IEEE (2007). <https://doi.org/10.1109/robot.2007.363180>
7. Grassé, P.P.: Reconstruction of the nest and coordination between individuals in terms. *Bellicositermes Natalensis* and *Cubitermes* sp. the theory of stigmergy: test interpretation of termite constructions. *Insectes Soc.* **6**(1), 41–80 (1959). <https://doi.org/10.1007/bf02223791>
8. Jones, C., Matarić, M.J.: Automatic synthesis of communication-based coordinated multi-robot systems. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 381–387. IEEE (2004). <https://doi.org/10.1109/iros.2004.1389382>
9. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2149–2154. IEEE (2004). <https://doi.org/10.1109/iros.2004.1389727>
10. Lindsey, Q., Mellinger, D., Kumar, V.: Construction with quadrotor teams. *Auton. Robots* **33**(3), 323–336 (2012). <https://doi.org/10.1007/s10514-012-9305-0>
11. Michel, O.: Cyberbotics Ltd., Webots<sup>TM</sup>: professional mobile robot simulation. *Int. J. Adv. Robot. Syst.* **1**(1), 39–42 (2004). <https://doi.org/10.5772/5618>
12. Olson, E.: AprilTag: a robust and flexible visual fiducial system. In: 2011 IEEE International Conference on Robotics and Automation, pp. 3400–3407. IEEE (2011). <https://doi.org/10.1109/icra.2011.5979561>
13. Petersen, K., Nagpal, R., Werfel, J.: TERMES: an autonomous robotic system for three-dimensional collective construction. In: Proceedings of Robotics: Science and Systems, pp. 257–264. RSS Foundation (2011). <https://doi.org/10.15607/rss.2011.vii.035>
14. Pincioli, C., et al.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* **6**(4), 271–295 (2012). <https://doi.org/10.1007/s11721-012-0072-5>
15. Soleymani, T., Trianni, V., Bonani, M., Mondada, F., Dorigo, M.: Autonomous construction with compliant building material. In: Menegatti, E., Michael, N., Berns, K., Yamaguchi, H. (eds.) Intelligent Autonomous Systems. AISC, vol. 302, pp. 1371–1388. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-08338-4\\_99](https://doi.org/10.1007/978-3-319-08338-4_99)
16. Sugawara, K., Doi, Y.: Collective construction by cooperation of simple robots and intelligent blocks. In: Kubota, N., Kiguchi, K., Liu, H., Obo, T. (eds.) ICIRA 2016. LNCS (LNAI), vol. 9834, pp. 452–461. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-43506-0\\_40](https://doi.org/10.1007/978-3-319-43506-0_40)
17. Sugawara, K., Doi, Y.: Collective construction of dynamic equilibrium structure through interaction of simple robots with semi-active blocks. In: Chong, N.-Y., Cho, Y.-J. (eds.) Distributed Autonomous Robotic Systems. STAR, vol. 112, pp. 165–176. Springer, Tokyo (2016). [https://doi.org/10.1007/978-4-431-55879-8\\_12](https://doi.org/10.1007/978-4-431-55879-8_12)
18. Theraulaz, G., Bonabeau, E.: Coordination in distributed building. *Science* **269**(5224), 686–688 (1995). <https://doi.org/10.1126/science.269.5224.686>
19. Theraulaz, G., Bonabeau, E.: A brief history of stigmergy. *Artif. Life* **5**(2), 97–116 (1999). <https://doi.org/10.1162/106454699568700>
20. Thomaszewski, B., Gumann, A., Pabst, S., Straßer, W.: Magnets in motion. *ACM Trans. Graph.* **27**(5), 162:1–162:9 (2008). <https://doi.org/10.1145/1409060.1409115>

21. Werfel, J., Nagpal, R.: Extended stigmergy in collective construction. *IEEE Intell. Syst.* **21**(2), 20–28 (2006). <https://doi.org/10.1109/mis.2006.25>
22. Werfel, J., Nagpal, R.: Three-dimensional construction with mobile robots and modular blocks. *Int. J. Robot. Res.* **27**(3–4), 463–479 (2008). <https://doi.org/10.1177/0278364907084984>
23. Werfel, J., Petersen, K., Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. *Science* **343**(6172), 754–758 (2014). <https://doi.org/10.1126/science.1245842>
24. Wismér, S., Hitz, G., Bonani, M., Gribovskiy, A., Magnenat, S.: Autonomous construction of a roofed structure: synthesizing planning and stigmergy on a mobile robot. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5436–5437. IEEE (2012). <https://doi.org/10.1109/iros.2012.6386278>
25. Worcester, J., Ani Hsieh, M., Lakaemper, R.: Distributed assembly with online workload balancing and visual error detection and correction. *Int. J. Robot. Res.* **33**(4), 534–546 (2014). <https://doi.org/10.1177/0278364913509125>