Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Droplet: Towards Autonomous Underwater Assembly of Modular Structures

Samuel Lensgraf*, Amy Sniffen*, Evan Honnold†, Jennifer Jain‡,
Zachary Zitzewitz§, Weifu Wang¶, Alberto Quattrini Li*, Devin Balkcom*
Dartmouth College*, Aurora Innovation†, Amazon Prime Air‡, Proof School§, Baidu Research¶

***Abstract*—This paper presents a first low-cost autonomous robotic system for underwater assembly of mortarless structures. The long-term goal is to enable the construction of large-scale underwater structures, such as retaining walls and artificial reefs. The approach follows the principle of co-design; the 2-DOF manipulator and blocks are designed to complement the localization and control strategies. The blocks and gripper are designed with a connector geometry that removes error during pickup of blocks and drop assembly. This error correction feature allows a simplification of localization and control, which are based on fiducial markers on custom platforms. We developed the proposed system on a low-cost heavily modified BlueROV2 autonomous vehicle – which we call *Droplet* – with a two-degree of freedom hand that can open and close a gripper and rotate over the yaw. We performed extensive experiments in the pool to evaluate each component and the system as a whole. Results showed a $100\,\%$ success rate in dropping blocks in the presence of some localization and control errors as well as the assembly of several different 3D structures composed of up to eight blocks.**

I. Introduction

This paper proposes a first low-cost underwater autonomous robotic system for assembly of mortarless structures as a first step towards construction of human-scale underwater structures, including retaining walls [17] or artificial reefs [34]. General underwater construction and assembly capabilities will enable the practical solution of many timely problems, e.g., support for marine agriculture, ecological repair, and mitigation of coastal weather events. While some robotic solutions exist for on-land construction [30], underwater assembly is still in its infancy: so far, only tele-operated robots have been proposed for such a task [4]. There are many challenges posed by the underwater environment [57], including, but not limited to, the absence of reliable localization systems (e.g., GPS) and communication infrastructure (e.g., WiFi).

To cope with such challenges, we tightly co-designed the manipulator and blocks with sensing and control strategies and extensively evaluated each component in the pool. The blocks and manipulator are designed using conical shapes that remove error as blocks compliantly contact the gripper or other blocks. This allowed a simplification of the localization strategy, based on fiducial markers, and of the control system based on a hierarchical PID controller. The implemented system is based on a low-cost heavily modified BlueROV2 autonomous vehicle with a two-degree of freedom hand that can open and close a gripper and rotate over the yaw. We call the robot *Droplet*. We designed two platforms: a source platform where blocks are initially stored, and a target platform where blocks are dropped to assemble the structure. Fast prototyping of the blocks is enabled through 3D printing. The robot is able to achieve an accuracy in localization and control on the order of a few centimeters. The designed blocks and gripper can correct a positional error of $3.5\,\mathrm{cm}$ and 20 degrees of yaw. We performed several experiments, including a 100-repetition trial. The robot was able to drop a block in the correct place in every repeatability trial. The robot built a number of 3D structures (L-shaped, wall, pyramid) composed of up to eight blocks stacked non-adjacently in a pool. Fig. 1 shows the complete system, while assembling an L-shaped 3D structure.

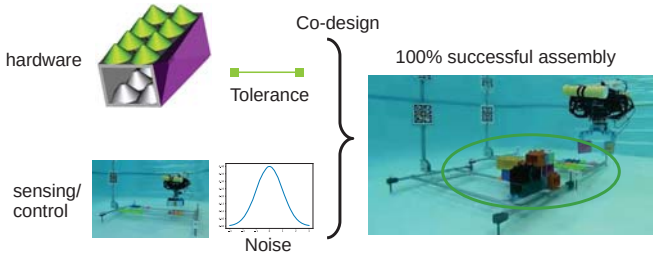


Figure 1. The blocks and gripper are designed to complement sensing and control, to achieve successful underwater robot assembly with a low-cost robot.

This work represents a foundation for low-cost robotic underwater assembly and real construction. There are a number of challenges that will be studied in the immediate future to scale up the construction to practical structures. On the design side, the challenge is to find ways of connecting blocks and grippers that allow simple construction of sturdy, large structures, requiring a deeper understanding of the interplay of gravity, buoyancy, and current. We will study the use of other materials, such as concrete, to enable the construction of useful structures. On the robot side, robust sensing and control are necessary to operate in turbid waters and in presence of currents.

II. Related Work

Our work begins to explore the possibility of automated underwater construction. Under the umbrella of this topic, work has been done on underwater manipulation and robotic construction on land, underwater, and block designs.

A. Underwater Manipulation

Underwater manipulation often focuses on enabling underwater vehicles to perform intervention activities, like manipulating an interface on a panel underwater or helping to perform sample collection [12]. Underwater manipulation systems in practice are generally tele-operated, but autonomous underwater manipulation has recently started to receive some attention. Typically, underwater manipulation systems use high degree of freedom manipulators meant to provide similar capabilities to robot arms on land. Underwater manipulators are expensive, with prices that range from a few thousand for the very simplest manipulators, up to millions of US dollars [40]. In water, the coupling between the arm's motion and the vehicle's requires complex control laws to overcome. The TRIDENT project in the EU [38] is an example of a project specifically designed to enable autonomous intervention with high degree of freedom manipulators mounted on a large Autonomous Underwater Vehicle (AUV). In our work, we take a nearly opposite approach. We design our system around making the manipulation task as simple as possible, which allows us to disregard the complex dynamic coupling between the manipulator and the robot's body. Our 2-DOF manipulator requires no special control, making our system much cheaper to deploy.

One of the closest prior works to ours in spirit is the system developed by Palomeras et al. [31] which features a docking station custom-designed to mate with features on an intervention AUV. The AUV pushes itself with a controllable force into cone-shaped features on the docking station, allowing it to have a stable base from which to use its on-board manipulator. While the idea of passive error correction is present in this work, we apply the idea in a more drastic way to our system, eliminating the need for complex control and manipulation behaviors entirely.

B. Robotic Construction

Underwater robotic construction is still in its infancy [4]: it has been directly addressed in the literature primarily focusing on tele-operated approaches. Akizono et al. [1] proposed using a combination of a conventional camera system and haptic feedback to help a human operator control an underwater construction robot. The experiments were conducted on land but suggested success underwater. Utsumi, Hirabayashi, and Yoshie [46] developed an augmented reality application for a grasping system that picks up and releases rocks underwater. Seo, Yim, and Kumar [39] presented some results on water surface block assembly. A few techniques for self-assembly of underwater robots have been explored in simulation [20, 48, 45]. While tele-operation can be effective for completing small-scale manipulation tasks, it requires specially trained human operators and high bandwidth connections between the robot and operator. Underwater communication of sufficient bandwidth still requires a physical tether in practice which drastically limits the range and duration possible for a deployment of a robotic system. In addition, the design of an intuitive interface for a human operator is often a task-specific process, making the deployment of new kinds of construction technologies challenging.

Ground robotic construction is more mature [36], with solutions that propose the use of a ground robot with a manipulator, e.g., wheeled robots [35, 19, 22] and tracked robots [15, 24]. The first construction projects involved building brick walls [3, 7] and only few years ago, Sam100 robot became the first commercially viable bricklaying robot [14]. More recently, Sustarevas et al. [43] proposed a mobile robot for 3D printing large structures. Several construction systems work by deploying multiple robots that tightly cooperate, e.g., to carry long rods [41, 11, 56, 9, 21] or assemble parts [26]. Other robotic construction methods used swarms of robots that assemble simple passive blocks that robots can easily pick up and assemble into 2- and 3-D structures [50, 44, 53, 32, 54, 2, 13, 49, 23, 27]. A set of work is based on multiple simple robots and programmable blocks that can interact with each other and self-assemble into complex structures, e.g., [42, 52, 51, 33]. The use of smart blocks would be too expensive for a large scale construction underwater.

Aerial robotic construction, although not as mature as ground robotics construction, has seen a rise [6], due to advances in aerial grasping and manipulation [55]. Some work proposed the construction of truss-like structures assembled by a team of quadrotors [37, 29]. Braithwaite et al. [10] demonstrated constructions of a multi-element tensile structure between anchor points through small UAVs. Hunt et al. [25] presented a prototype for 3D printing with a flying vehicle. The idea of adjusting construction elements to suit the aerial vehicle was recently explored. Augugliaro et al. [6] designed 90-gram foam blocks that small quadcopters were able to pick up to construct a 6-m tower. The UAVs were able to place the blocks with millimeter of accuracy. Similarly, Latteur et al. [28] showed drones that were able to build a tower of hollow concrete cones, the design of which allowed for $6.5\,\mathrm{cm}$ of inaccuracy from the controller.

Our approach to underwater construction is similar in spirit to the latter type of work: we design blocks with geometries that maximize the capture region of a joint and reduce error during manipulation, resulting in a lower demand on both sensing and control.

III. Underwater robotic construction system

We are particularly motivated by mortarless construction techniques, as a simple first approach to underwater assembly. Fig. 2 shows some motivating examples: a human-assembled modular reef structure, and an Incan wall which is still sturdy half a millennium after its construction.

While some of the design choices are independent of the specific robot used, we ground our presentation to *Droplet*, a low-cost BlueROV2 heavy configuration[1], which we heavily modified to make it autonomous and tetherless. The robot has an 8-thruster vectored configuration allowing for 6 degree-of-

[1] https://bluerobotics.com/store/rov/bluerov2-upgrade-kits/brov2-heavy-retrofit-r1-rp/

(a) The MARS modular reef structure (Reef Design Lab; Alex Goad)

(b) Mortarless construction of an Incan wall (Bcasterline; Wikimedia).

Figure 2. Sources of inspiration: modular artificial reefs and mortarless construction.

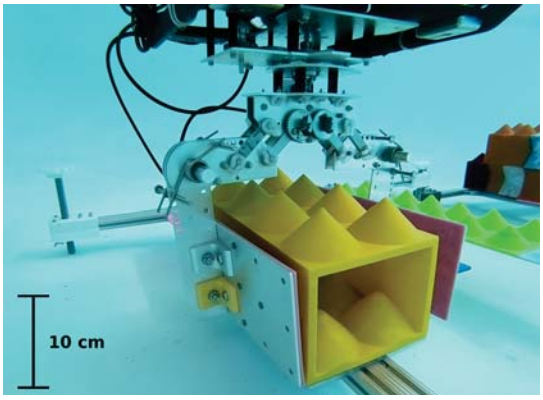
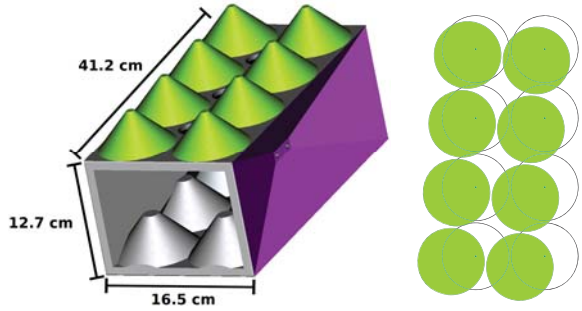


Figure 4. 2DOF manipulator grasping a block.



(a)

(b)

Figure 3. (a) Rendering of error correcting block. The gripper slot is magenta and drop error correction features are green. (b) Idealized model used for acceptance area. Black and white circles represent receiver. Green represent successfully dropped block.

fredom control, and a number of sensors, including a camera and an IMU.

A. Passive Error Correcting Blocks

We exploit passive error correction both to absorb noise in the system and to provide a convenient simplification of the assembly process. Figure 3a shows a rendering of the block design. The sides are fitted with large $3.175\,\mathrm{cm}$ deep pyramid-shaped recesses that mirror the pyramids on the robot's finger tips. During pickup, the robot's positional noise actually helps the block reliably slide into the desired grasp position as the fingers are gradually closed. The top and bottom of the block are fitted with mirrored cone slots. The cones allow the block to slide into place from a variety of pre-drop positions. The cones also help keep the block in place despite gentle bumps from the robot as it grasps the block.

Our design is inspired by Eckenstein and Yim's work on *acceptance area* for docking connectors [16]. The acceptance area, $\mathcal{A}$, is the set of all deviations from the *ideal dropoff position*, $\mathbf{I}$, from which the blocks can successfully align by sliding along the mirrored cones. Figure 8 shows the acceptance area and ideal dropoff location for our blocks in $2D$. To determine whether the robot is in the acceptance area of a block we use a conservative tolerance region $\mathbf{t}$ specified as a box in $6D$ configuration space.

In an ideal case, disregarding friction and bouncing effects, we can think of any pre-drop position of the block as being successful if the projected circles of the slots on the bottom of the block contain the tips of the cones in the slots the block is being dropped onto – see Fig. 3b. By using this low dimensional idea of the acceptance area of a block pair, we can create an abstraction of the robot's workspace in which we can plan assembly behaviors. We consider the acceptance area of each possible block position to be a single point in our discrete task space.

The size of the acceptance area of the blocks can be designed to fit estimated localization quality and external effects such as currents, making the system robust in a variety of circumstances. Section IV discusses how the design can be adapted based on known sensor noise.

B. Custom 2DOF Gripper

In the spirit of co-designing hardware and software to achieve a simple system in aggregate, we designed a 2 degree-of-freedom gripper specifically for the task of stacking our blocks (Fig. 4). The gripper has two large fingers fitted with 3D-printed pyramid-shaped pads as large as the side of the blocks. The fingers are actuated with a parallel linkage driven by a high torque underwater servo[2]. The gripper has a "wrist" joint which can rotate the fingers relative to the robot using another servo. The wrist joint allows improved design freedom without increasing the complexity of how Droplet localizes.

The fingers feature a slightly acute angle held in place by tensioned torsion springs between the actuated link and the link that interfaces with the block. When the fingers are closed around a block, tension is increased due to the torsion springs. The torsion springs help prevent the gripper from pulling itself apart on failed grasps and add a natural end stop for the servo, ensuring the block is tightly grasped when the servo reaches its stall current. No position or force sensing is required by the high-level planner to reliably grasp blocks using these fingers.

After several iterations on placement choices for the gripper, we settled on mounting the gripper directly below Droplet's center of gravity. In this position, roll and pitch noise in Droplet's position translate to position error in the manipulator.

[2]https://www.bluetrailengineering.com/servos

The extent of this error increases linearly with the distance that the fingers hang below the chassis. Because of this we adopted the design goal of making the gripper as short as possible. The gear box that drives the wrist joint hangs only 3.5 cm below the chassis and the gear box driving the finger linkage adds 9 cm. Because Droplet grasps blocks from the side, the primary contributor to the height of the manipulator is the large set of fingers which must be tall enough to surround the block.

While picking up a block, the manipulator controller gradually closes its fingers. Gradually closing the fingers increases the total time required to manipulate each block, but prevents the fingers from jamming before they slide into place. While closing the fingers slowly, the position noise of the robot helps the block slide into the correct position more reliably.

While dropping a block, Droplet opens its fingers as quickly as possible. Opening the fingers rapidly allows Droplet to take advantage of opportunities to drop the block at the correct position amid positioning noise. By limiting the amount of time the block is in contact with the fingers, Droplet limits the effect of changing velocity and position on the block's trajectory as it falls.

C. Assembly Planning

Fiducial markers on two build platforms provide the frames of reference assembly behaviors take place in. Each platform is fitted with a bundle of fiducial markers which are used to determine Droplet's pose R in the platform's frame of reference P_i, ${}^{P_i}\mathbf{T}_R$, with Z pointing upward, X towards the marker along the long aluminum struts, and Y to the right facing the marker.

To simplify the assembly process, we fitted the platforms with 3D printed slots which provide error correction for the first layer of dropped blocks and hold blocks in position reliably before they are used during assembly (Fig. 5). Both of the platforms are levelled before assembly occurs using bolts that serve as adjustable height legs. Because Droplet holds itself orthogonal to the gravity vector, it is assumed there is no angular difference between the build platforms and the robot on the roll and pitch axes.

Assembly sequences for Droplet are specified in a high level GCode like format used in CNC machines. Currently, there are three possible commands: PICKUP, DROP and ROTATE_WRIST. The pickup and drop commands take as arguments the platform ID and the 3-dimensional indices of the target slot for the behavior. For example, PICKUP 0 1 2 3 instructs the robot to pick up the block on platform ID 0 at slot number 1 2 3. The index of a slot is translated to a waypoint relative to the marker mounted to the build platform by displacing the known location of slot number 0 0 0 by the dimensions of each slot. The system is sensitive to the proper calibration of the location of slot 0 0 0. Measurements during the platform's fabrication process proved to be enough for proper calibration in our experiments. The specification format is simple enough that an assembly sequence could easily be generated using a higher level layout planning method. Currently, the sequences are specified by hand.

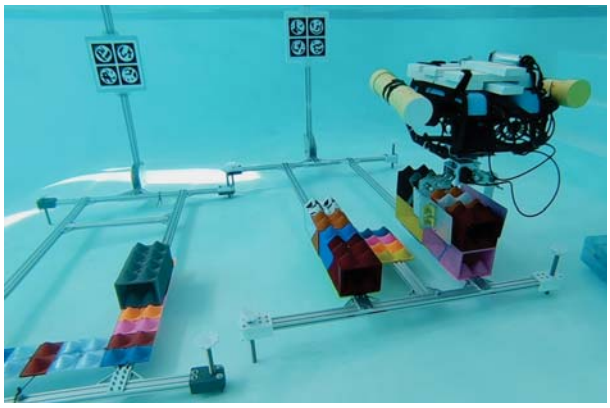


Figure 5. Build platforms with rigidly attached fiducial markers; pickup and drop platforms are on the left and right, respectively.

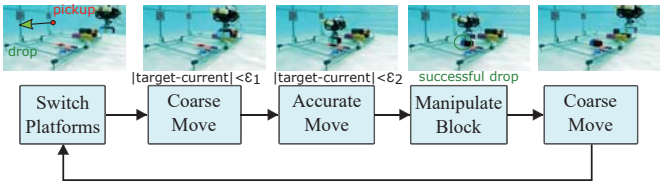


Figure 6. State machine phases to manipulate a block. For simplicity, we identified "Manipulate Block" for CLOSEGRIPPER, OPENGRIPPER; ROTATEWRIST can happen just after the last COARSEMOVE, when Droplet is far from the platforms.

Given a high-level build sequence specification, the robot compiles the sequence into a state machine that controls the robot through the assembly process. The state machine has 6 possible states: SWITCHPLATFORMS, COARSEMOVE, ACCURATEMOVE, CLOSEGRIPPER, OPENGRIPPER, ROTATEWRIST. Each state has a paired PID controller – described in the next subsection – and a termination condition. Each state specifies a waypoint in the frame of reference of one of the build platforms which becomes the target for the PID controller. The SWITCHPLATFORMS state moves the robot on an open loop rotation then engages a P controller to navigate to a point far away from the blocks. The ROTATEWRIST, OPENGRIPPER and CLOSEGRIPPER states use a PID controller to hold the robot steady while moving the gripper.

The ACCURATEMOVE and COARSEMOVE states move the robot using a PID controller until it receives a location reading within the tolerance region $\mathbf{t}$ of its target slot. The user may specify the size of $\mathbf{t}$ to trade accuracy for speed. ACCURATEMOVE is given a smaller $\mathbf{t}$ than COARSEMOVE to ensure blocks are dropped in the proper position. Figure 6 shows the states the robot moves through for each pickup and drop action.

D. AUV Control

Droplet's thruster configuration is as follows: four of the thrusters face downwards on a square about Droplet's chassis and are used to control the θ (roll), ϕ (pitch) and Z axes. Four thrusters below the main electronics tube control the X, Y and

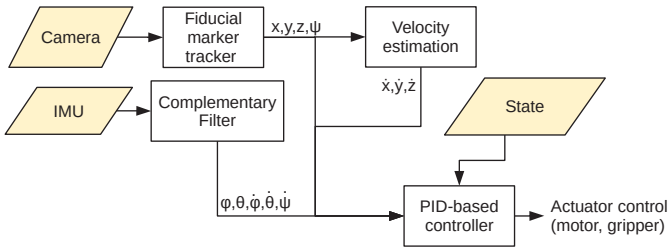


Figure 7. Sensing/control diagram of Droplet.

ψ (yaw) axes. Though there is coupling between Droplet's roll and pitch and its X and Y velocity (the top four motors are like a drone), we disregard the coupling of the two sets of motors because, in practice, Droplet is effective at staying level.

Each set of motors is controlled by a PID controller for each axis which are combined using a tweaked version of the algorithm from the BlueROV2 low-level controller [5]. The T200 thrusters have a signal deadband around their stopped position which makes using them at low speeds difficult. Our version of the motor mixing algorithm adds the offset values (M_f, M_b) for each motor to eliminate the need to reason about the signal deadband. In practice, the value required for each motor differs drastically and is different in different directions.

The data used to drive Droplet's PID controllers come from both the visual fiducial markers and the on-board IMU. The on-board accelerometers are processed using a complementary filter [47] to give a high accuracy, high rate estimate of Droplet's θ and ψ angles. Droplet's internal gyroscope provides θ, ϕ, and ψ velocities. Because the data for the θ and ϕ angles is high quality and high rate, Droplet is effective at holding itself steady. In addition, we ballasted Droplet specifically to increase stability, on its θ and ϕ angles.

Droplet's X, Y, Z, ψ are recovered from the visual fiducial markers. Droplet's linear velocity is computed by differentiating the reading from the visual fiducial markers and applying a low pass filter. The ψ angle receives high quality velocity estimates, making it very stable as well. The X, Y and Z axes receive the relatively low quality velocity data making them the largest source of position noise.

IV. Adapting Block Design to Localization Noise

By modifying the size of the acceptance area, $\mathcal{A}$, of the blocks and slots, we can adapt the system to different levels of noise, localization, or control quality. Because Droplet makes use of an opportune dropping strategy, it is paramount to limit the number of false positive location readings. We define a false positive location reading, $\mathcal{E}$, as the event in which the robot's true location, $\mathbf{l}$, is outside $\mathcal{A}$ but it receives a sensor reading, p, which is within the tolerance region $\mathbf{t}$ about the ideal dropoff position $\mathbf{I}$. If the block is dropped from $\mathbf{l}$ it will fail to align as intended.

Misplacing a single block causes the entire assembly process to fail, so reliably determining when the robot is within the acceptance area of a slot is key. We can achieve this reliability in two ways: by either reducing the noise in our localization data, or by increasing the size of the acceptance area. Decreasing the size of the tolerance region $\mathbf{t}$ reduces the probability of getting a false positive reading but makes it more difficult to get a true positive reading. Increasing the size of the acceptance area requires increasing the size of the blocks, eventually making them unwieldy to manipulate. Reducing localization noise is challenging especially in underwater environments. Thus, there is a trade off between three design dimensions of the system: the size of the acceptance area, the size of the tolerance region, and the noise level in the localization data.

As discussed in the previous section, the least reliable location information available to Droplet is its global x, y position, so we limit our analysis to a $2D$ tolerance region $\mathbf{t} = (t_x, t_y)$ specified as a box of side lengths $2t_x, 2t_y$ centered on $\mathbf{I}$. Figure 8 shows the tolerance region, acceptance area and ideal dropoff position, for our blocks. Let the radius of the base of the female cones on the bottom of the blocks be r. In the $2D$ case, $\mathcal{A}$, is a circle of radius r centered on $\mathbf{I}$. We can parametrize block designs in one dimension using r.

Given a known covariance of sensor readings, Σ, we can compute the probability of getting a false positive at location $\mathbf{l}$, $\mathbb{P}(\mathcal{E}|\mathbf{l}, \Sigma)$ by integrating the multivariate Gaussian distribution, $\mathcal{N}$, with mean $\mathbf{l}$ over the tolerance region $\mathbf{t} = (t_x, t_y)$:

$$\mathbb{P}(\mathcal{E}|\mathbf{l}, \Sigma) = \int_{y=\mathbf{I}_y - t_y}^{\mathbf{I}_y + t_y} \int_{x=\mathbf{I}_x - t_x}^{\mathbf{I}_x + t_x} \mathcal{N}([x, y], \mathbf{l}, \Sigma) dx dy \quad (1)$$

To use $\mathbb{P}(\mathcal{E}|\mathbf{l}, \Sigma)$ as a general design aid, we can compute a worst case $\mathbb{P}(\mathcal{E}|\Sigma)$ by finding the point $\mathbf{l}$ on the border of $\mathcal{A}$ that maximizes $\mathbb{P}(\mathcal{E}|\mathbf{l}, \Sigma)$:

$$\mathbb{P}(\mathcal{E}|\Sigma) = \max_{\mathbf{l} \in \partial\mathcal{A}} \mathbb{P}(\mathcal{E}|\mathbf{l}, \Sigma) \quad (2)$$

Figure 8 shows how $\mathbb{P}(\mathcal{E}|\Sigma)$ changes with r using the worst location covariance observed in the experiment in Section V-B and $\mathbf{t} = (0.012, 0.012)$ m, the value used by AccurateMove before dropping a block. By conservatively selecting r, we can ensure reliable assembly. For our implementation, we chose $r = 0.042$ m to balance error probability and $3D$ printing speed.

V. Experimental Validation

To achieve low-cost autonomous underwater assembly, a significant amount of re-engineering on the robot and hardware was required. Because of a latency of 0.1 seconds of the stock camera, making visual-based autonomous control at cm-level impossible, we used a FLIR Blackfly S 1440 x 1080 monochrome camera at 30 FPS with a Stardot Technologies 4 mm fixed focal length lens. For autonomous onboard operations, we removed the tether and we exchanged the stock Raspberry Pi computer for a more powerful Intel Upboard UpSquared featuring 8 GB RAM and a 2.5 GHz quad core processor which runs the localization and main control software. We replaced the stock firmware of the low-level controller with our implementation – *SimpleSub* – to have a

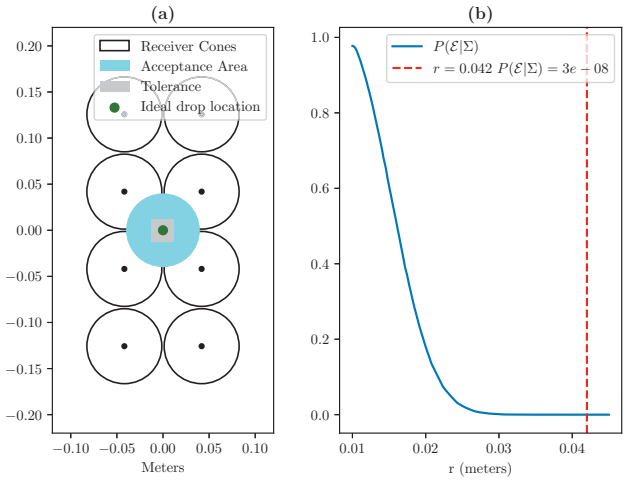


Figure 8. (a) $\mathcal{A}$ (cyan), $\mathbf{I}$ (green) and $\mathbf{t}$ (gray) for our block design. (b) Numerical approximation of $\mathbb{P}(\mathcal{E}|\Sigma)$ for increasing r with the value of $r = 0.042$ used for our blocks marked in red.

lower latency to directly control each thruster and get IMU measurements at $500\,\mathrm{Hz}$ (instead of $10\,\mathrm{Hz}$) from the main computer. To achieve stability over roll and pitch, we ballast the robot using a raft made of PVC trim fixed to the top of its chassis and a pair of ballast tubes filled with marine buoyancy foam fixed to the sides of its chassis. Designs for the mounting hardware and a parts list to replicate Droplet are available on Github[3].

We built the pickup/drop platforms out of $5\,\mathrm{cm}$ by $2.5\,\mathrm{cm}$ aluminum extrusions. One platform is designated as a palette which holds the blocks in known positions before assembly. The other is designated as the dropoff platform and serves as the base for the built structure. Each platform is fitted with a set of cone-shaped slots that mate with the blocks. The slot cones are fastened to the build platform and held securely in known locations relative to the fiducial marker. The build platforms also feature threaded legs which are used before each experiment to level the platform, eliminating the need to reason about the platform's relationship to the gravity vector.

The fiducial markers were printed on aluminum with laser cut mounting holes that allow it to be precisely mounted on a white Delrin back plate, fixed onto the aluminum extrusion. We chose STags [8] as fiducial markers, due to their detection robustness underwater in our preliminary experiments compared to other markers commonly used, e.g., ARTag [18]. We printed our blocks in PETG (material commonly used in water bottles) using a Pegasus 12 and a Creality CR-10 3D printer. Each block was ballasted to be slightly negatively buoyant, using stainless steel washers suspended on bolts through the center of the block.

A. Validating ideal acceptance area

To understand whether our ideal model of the valid pre-drop locations is valid amid hydrodynamic and frictional effects, we built a rig that fits onto one of the build platforms and allows x,y,z and yaw placement of the hand (Fig. 9a). To position the hand relative to a drop slot, we hang a plumb bob from the center of the hand which points to a $2.5\,\mathrm{cm}$ grid of dots, painted on the slot using a 3D printed stencil. While this setup allows some manual positioning error, it gives us a higher level of certainty about the true position of the fingers relative to the slot than would be possible using our visual positioning system alone. For each drop trial, a block is placed in the gripper and manually released by an operator.

As a first experiment, we dropped a block with the fingers elevated $13\,\mathrm{cm}$ above the slot with the yaw of the fingers aligned correctly with the slot. We completed five trials at each of 49 locations, for a total of 245 drops, and recorded whether or not the block fell into the proper alignment with the slot. At $13\,\mathrm{cm}$, the bottom of the block is resting about $3\,\mathrm{cm}$ above the top of the cones of the slot. At this height, hydrodynamic effects proved to have minimal effect on the trajectory of the block as the block fell and slid into place properly for each of the five trials. In this configuration, each of the 245 drops slid into the predicted alignment.

When we moved the gripper $25\,\mathrm{cm}$ to above the slots, the results of dropping differed significantly from the ideal model of acceptance area. We dropped the block twice from each location in a 7×5 subset of the grid. In Figure 10 we compare the results of dropping at $13\,\mathrm{cm}$ and $25\,\mathrm{cm}$. Small perturbations in the block's initial configuration as it is dropped cause it to catch the water as it falls in different ways. The center void that we left out to speed print time and allow the mounting ballasts acts like a sail. At $25\,\mathrm{cm}$, the block did not align as expected in 14 of the 70 drops.

We also tested the yaw of the gripper relative to the slot when held over the ideal position. In this case, the block was able to slide into the correct position from up to ten degrees of misalignment. With higher yaw misalignment, the block lands on top of the cones but without sliding into place on them.

When the block failed to land in proper alignment with the slot, the most common problem was that it would be either one set of cones too far forward or backward. When the block lands in a stable, but wrong, location, the error is more difficult for the system to recover from: the robot must understand where it dropped the block and how it might remove the block. If the block completely falls off the side, it is more simple to correct: the robot needs simply understand the block didn't land and try again.

B. Localization noise testing

To understand the probability of getting a false positive localization reading, we mounted the robot on the same rig used for the dropping experiments and positioned it at 49 different xy locations relative to the desired slot. We recorded the position readings received from the visual localization system over 10 seconds at each location. Figure 9b shows the robot mounted on the rig and Fig. 11 shows the results of this experiment. We mark the tolerance region as a $2.4\,\mathrm{cm}$ square about the center of the slot and the ideal acceptance

[3]https://github.com/dartmouthrobotics/underwater-assembly-auv

(a) (b)

Figure 9. Manual positioning rig used for dropping (a) and localization (b) experiments.

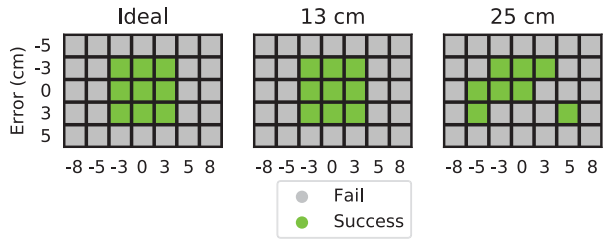


Figure 10. Expected (left) vs actual successful pre-drop locations sampled on X,Y plane at 13 cm (center) and of 25 cm (right) above the slots sampled on a grid of X, Y pre-drop position errors.

area of the block as a cyan region.

Upon inspection of the noise at each location, we found that the noise is roughly Gaussian distributed for each location with a standard deviation of 0.0004 on the x axis and 0.0033 on the y axis when the robot was positioned just above the ideal dropoff location. We received no false positive readings for the duration of the experiment which supports the low probability predicted using Equation (2) in Figure 8.

C. Effect of tolerance on time

When reducing the size of the tolerance region for a waypoint, reaching a waypoint becomes more challenging for the robot, increasing the time required for the robot to complete a build sequence. To understand the tradeoff between runtime and accuracy, We searched over a set of 81 combinations of tolerance values for the X and Y axes of the robot. For each combination of X and Y tolerances, the robot navigated a triangle of three waypoints spaced about $10\,\text{cm}$ apart and recorded the time to complete the entire circuit.

Figure 12 shows the results of this experiment. While the measurement process involved a large amount of noise arising from slight differences in how the robot approaches each waypoint, it is clear that larger tolerance regions allow the robot to complete the circuit faster. Based on these results, we use a large tolerance $4.0\,\text{cm}$ for repositioning maneuvers near the blocks and then a smaller tolerance for dropping and pickup actions: $1.2\,\text{cm}$ on the X, Y axes and $2.5\,\text{cm}$ on the Z axis.

D. Repeatability testing

To validate our model of how often a false positive reading will occur, we completed two sets of repeatability trials in which the robot picked up and dropped a single block from the center location in Figure 11 a total of 100 times. For the first set of 50 trials, Droplet approached the block, picked it up, moved backwards and up $30\,\text{cm}$ then replaced the block. To make the second set of trials more challenging, droplet moved to a randomly selected 3D point behind and above the slots between every pickup and drop. Before each drop action, Droplet descended to a point where the bottom of the block was just above the cones of the slot. In all 100 trials, Droplet successfully placed the block in the correct location. Figure 13 shows the location readings that triggered each drop. The reliability Droplet achieved during this experiment shows that our simplified analysis of the acceptance area and false

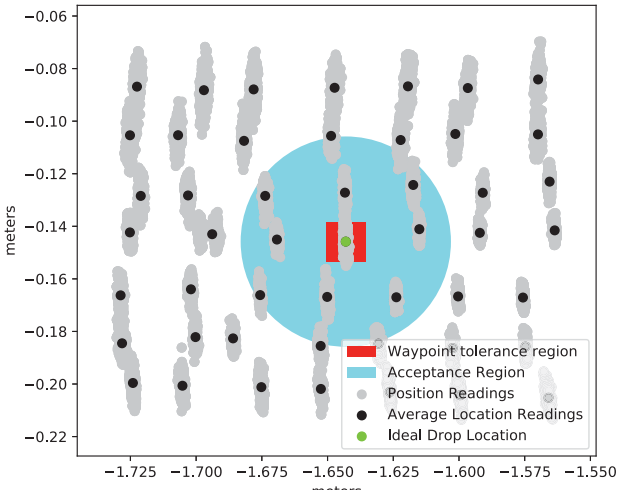


Figure 11. Locations measured using plumb bob relative to the tolerance zone and ideal acceptance area.

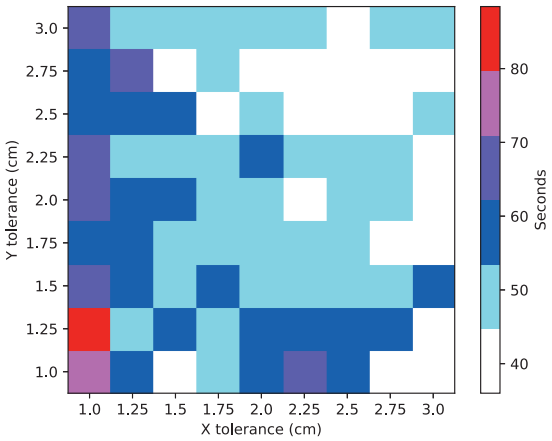


Figure 12. Time to reach a waypoint using the first-hit, pause, second-hit method of deciding on when a waypoint is reached. Standard deviation: $9.26\,\text{s}$.

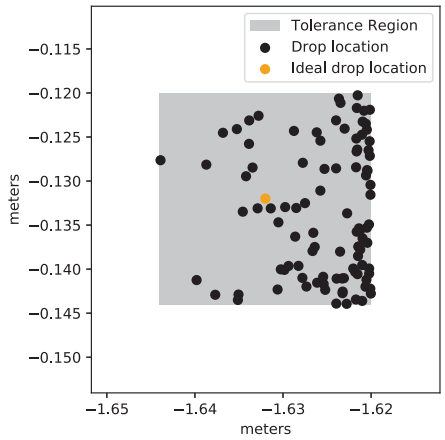


Figure 13. Successful pre-drop position readings for 100 repeatability trials.

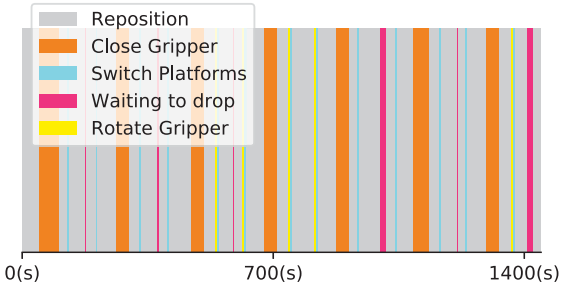


Figure 14. Timeline of robot's assembly states during a seven block trial of the system.

positive probability works well despite unpredictable noise from control and slight tipping on the roll and pitch axes.

On average, each drop took 147 seconds to complete, making the entire set of 100 trials take over four hours of runtime to complete. Reducing the time spent per block could have significant impact on allowing Droplet to build larger structures on a single charge. In the next subsection, we show a breakdown of the time required for each drop.

E. System Trials

To validate our system as a whole, we specified a seven block build plan in which all blocks are picked up from a pallet and placed on a build platform, forming an L-shaped wall. Once both platforms are properly levelled, the robot was able to successfully complete the build plan in about 30 minutes. This build plan showcases all of the design aspects of our system. We designed the structure to have one-slot spaces between blocks to allow the robot to have a clean area to approach to prevent bumps with the structure from knocking the whole structure over. In addition, we specified two straight structures: a pyramid of six blocks and a wall of eight blocks. See Figure 15 for examples of the structures built by Droplet.

We had the robot record the time each phase of the assembly process took during the seven block L-wall build. Figure 14 shows a visualization of the phases the robot moves through to complete the assembly process of the seven block L-wall. The majority of the time was spent repositioning the robot between grasp and drop locations (1025 seconds). The second most time consuming action was closing the gripper (278 seconds). Finally, switching platforms took 60 seconds and waiting to drop the block took 44 seconds.

VI. Discussion and Future Work

This paper establishes a solid first step towards achieving more general autonomous underwater construction. Our robotic system is capable of creating structures out of mortarless error correcting blocks and achieve $100\,\%$ success rate.

Starting from this foundational work, we plan to pursue research directions that will enable Droplet to build more practical structures in more realistic field settings.

Design Flexibility. Our current system is limited to a single block geometry placed in one of two angles on a rectangular grid. By exploring different shapes of blocks, we can allow Droplet to build more interesting and useful structures using the same assembly behaviors. As a simple first step, we plan to explore the addition of double length blocks which could be used to make roofs or to more quickly build large structures.

Block material. We currently 3D print the building blocks from PETG which makes it easy to iterate on designs and keeps the blocks light weight. Because of the amount of time required to print each block and the cost of material, it is infeasible to scale the production of PETG building blocks to realistic construction levels. In previous work, we made several blocks using a concrete / perlite mix which were buoyant enough to easily manipulate. While the concrete blocks had increased friction, they were still able to slide into place. We plan to explore ways to design concrete blocks that can still slide into place while having features large enough that they do not break after repeated drops.

Block hydrodynamics. By tuning the weighting and outer shape of our blocks, it would be possible to design them to fall in ways that helps increase the reliability of drop-based assembly. We purposefully limited the time that the blocks spend falling through the water because our design disregarded hydrodynamic effects. By using hydrodynamics as a design opportunity, we could speed the assembly process by allowing Droplet to drop blocks while moving quickly high above the structure. Paying attention to the hydrodynamics and weighting of the blocks combined with an opportune dropping strategy could also allow Droplet to build structures in waters with currents.

Error detection. While we currently achieved a perfect success rate, in more complex structures blocks might fall out of place. To solve this problem, we plan to enable our robot to perform basic error detection. Because the blocks are designed to slide into place if they are stacked within a given acceptance area, we can start by using the vehicle's front mounted camera to detect whether a structure matches the desired structure. If not, the misplaced block can be knocked off of the top with

the gripper or a new block can be fetched and added to the top of the structure.

Time. One key limitation of our system is that the size of a structure which can be built on a single battery charge is limited by the amount of time it takes to place each block. Our current manipulation strategy consisting of a set of phases with increasingly tight tolerances ensures that the robot remains in control throughout the execution of a trajectory, limiting overshoot for important maneuvers. The trade we make is that it takes several minutes to place each block. In many instances, the robot will be visibly in the right location to move its gripper but will take several seconds before actuating the gripper. We will study improved opportunistic drops and automated layout planning to speed assembly time.

Robust localization. To make our system applicable in live field settings, it will need to be robust to a variety of water and lighting conditions. Our dependence on fiducial markers limits us to deploying in clear waters with ample light. Reduced light or cloudy waters would result in unacceptably inaccurate positioning information. In future work we plan on exploring other forms of rapidly deployable localization infrastructure such as acoustic-based systems or by introducing basic sensor fusion, allowing an IMU to aid in localization for short periods of low visibility.

VII. Acknowledgements

This work was supported in part by NSF GRFP, CNS-1919647, IIS-1813358, IIS-1813043.

References

[1] Junichi Akizono, Taketsugu Hirabayashi, Tomohide Yamamoto, Hiroshi Shibaura chome Sakai, Hiroaki Yano, and M Iwasaki. "Teleoperation of Construction Machines with Haptic Information for Underwater Applications." In: *Automation in Construction* (2004).

[2] Michael Allwright, Weixu Zhu, and Marco Dorigo. "An open-source multi-robot construction system." In: *HardwareX* (2018), e00050.

[3] Jurgen Andres, Thomas Bock, Friedrich Gebhard, and Steck Werner. "First Results of the Development of the Masonry Robot System ROCCO: a Fault Tolerant Assembly Tool." In: *International Symposium on Automation and Robotics in Construction*. 1994, pp. 87–93.

[4] Hadi Ardiny, Stefan Witwicki, and Francesco Mondada. "Construction automation with autonomous mobile robots: A review." In: *Proc. ICROM*. IEEE. 2015, pp. 418–424.

[5] *Ardupilot Firmware*. (Accessed on Feb 26, 2021). URL: http://ardupilot.org/.

[6] Federico Augugliaro, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W. Mueller, Jan Sebastian Willmann, and Fabio Gramazio. "Cooperative Construction with Flying Machines." In: *IEEE Control Syst. Mag.* 34.4 (Aug. 2014), pp. 46–64.

[7] C. Balaguer, E. Gambao, A. Barrientos, E. A. Puente, and R. Aracil. "Site Assembly in Construction Industry by means of a Large Range Advanced Robot." In: *International Symposium on Automation and Robotics in Construction*. 1996.

[8] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. "STag: A Stable Fiducial Marker System." In: *CoRR* abs/1707.06292 (2017).

[9] Adrienne Bolger, Matt Faulkner, David Stein, Lauren White, Seung-kook Yun, and Daniela Rus. "Experiments in decentralized robot construction with tool delivery and assembly robots." In: *Proc. IROS*. IEEE. 2010, pp. 5085–5092.

[10] Adam Braithwaite, Talib Alhinai, Maximilian Haas-Heger, Edward McFarlane, and Mirko Kovač. "Tensile web construction and perching with nano aerial vehicles." In: *Robotics Research*. Springer, 2018, pp. 71–88.

[11] Jonathan Brookshire, Sanjiv Singh, and Reid Simmons. "Preliminary results in sliding autonomy for assembly by coordinated teams." In: *Proc. IROS*. Vol. 1. IEEE. 2004, pp. 706–711.

[12] Romano Capocci, Gerard Dooly, Edin Omerdić, Joseph Coleman, Thomas Newe, and Daniel Toal. "Inspection-class remotely operated vehicles—A review." In: *Journal of Marine Science and Engineering* 5.1 (2017), p. 13.

[13] Lucian Cucu, Michael Rubenstein, and Radhika Nagpal. "Towards self-assembled structures with mobile climbing robots." In: *Proc. ICRA*. 2015, pp. 1955–1961.

[14] Zakaria Dakhli and Zoubeir Lafhaj. "Robotic mechanical design for brick-laying automation." In: *Cogent Engineering* (2017).

[15] Kathrin Dörfler, Timothy Sandy, Markus Giftthaler, Fabio Gramazio, Matthias Kohler, and Jonas Buchli. "Mobile robotic brickwork." In: *Robotic Fabrication in Architecture, Art and Design 2016*. Springer, 2016, pp. 204–217.

[16] N. Eckenstein and M. Yim. "Modular robot connector area of acceptance from configuration space obstacles." In: *Proc. IROS*. 2017, pp. 3550–3555.

[17] Wahid Ferdous, Yu Bai, Ahmed D Almutairi, Sindu Satasivam, and Juri Jeske. "Modular assembly of water-retaining walls using GFRP hollow profiles: Components and connection performance." In: *Composite Structures* 194 (2018), pp. 1–11.

[18] Mark Fiala. "ARtag revision 1, a fiducial marker system using digital techniques." In: *National Research Council Publication* 47419 (2004).

[19] Ernesto Gambao, Carlos Balaguer, Antonio Barrientos, R Saltaren, and Eugenio Andrés Puente. "Robot assembly system for the construction process automation." In: *Proc. ICRA*. Vol. 1. IEEE. 1997, pp. 46–51.

[20] Varadarajan Ganesan and Mandar Chitre. "On Stochastic Self-Assembly of Underwater Robots." In: *IEEE Robot. Autom. Lett.* 1.1 (2016), pp. 251–258.

Figure 15. Structures built by Droplet.

[21] Roderich Groß, Elio Tuci, Marco Dorigo, Michael Bonani, and Francesco Mondada. "Object transport by modular robots that self-assemble." In: *Proc. ICRA*. IEEE. 2006, pp. 2558–2564.

[22] Brad Hamner, Seth Koterba, Jane Shi, Reid Simmons, and Sanjiv Singh. "An autonomous mobile manipulator for assembly tasks." In: *Auton. Robot.* 28.1 (2010), p. 131.

[23] Mary Katherine Heinrich, Payam Zahadat, John Harding, Mads Nikolaj Brandt, Emil Fabritius Buchwald, Leonardo Castaman, Kit Wai Chan, Malte Harrig, Stian Vestly Holte, Lynn Kieffer, et al. "Using interactive evolution to design behaviors for non-deterministic self-organized construction." In: *Proceedings of the Symposium on Simulation for Architecture and Urban Design*. Society for Computer Simulation International. 2018, p. 21.

[24] Volker Helm, Selen Ercan, Fabio Gramazio, and Matthias Kohler. "Mobile robotic fabrication on construction sites: DimRob." In: *Proc. IROS*. IEEE. 2012, pp. 4335–4341.

[25] Graham Hunt, Faidon Mitzalis, Talib Alhinai, Paul A Hooper, and Mirko Kovač. "3D printing with flying robots." In: *Proc. ICRA*. IEEE. 2014, pp. 4493–4499.

[26] Ross A Knepper, Todd Layton, John Romanishin, and Daniela Rus. "Ikeabot: An autonomous multi-robot coordinated furniture assembly system." In: *Proc. ICRA*. IEEE. 2013, pp. 855–862.

[27] Sven Koenig and S Kumar. "A case for collaborative construction as testbed for cooperative multi-agent planning." In: *ICAPS-17 Scheduling and Planning Applications Workshop*. 2017.

[28] Pierre Latteur, Sebastian Goessens, Marie-Francoise Reniers, Ma Zhao, and Caitlin V Mueller. "Masonry Construction with Drones." In: *IASS Annual Symposium* (Sept. 2016).

[29] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. "Construction with quadrotor teams." In: *Auton. Robot.* 33.3 (2012), pp. 323–336.

[30] Nathan Melenbrink, Justin Werfel, and Achim Menges. "On-site autonomous construction robots: Towards unsupervised building." In: *Automation in Construction* 119 (2020), p. 103312.

[31] Narcís Palomeras, Pere Ridao, David Ribas, and Guillem Vallicrosa. "Autonomous I-AUV docking for fixed-base manipulation." In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 12160–12165.

[32] Kirstin Petersen, Radhika Nagpal, and Justin Werfel. "Termes: An autonomous robotic system for three-dimensional collective construction." In: *Proc. RSS* (2011).

[33] Benoit Piranda and Julien Bourgeois. "Designing a quasi-spherical module for a huge modular robot to create programmable matter." In: *Auton. Robot.* (2018), pp. 1–15.

[34] Reef Design Lab. *MARS: Modular Artificial Reef Structure*. (Accessed on Feb 26, 2021). URL: https://www.reefdesignlab.com/mars1.

[35] Maira Saboia, Vivek Thangavelu, and Nils Napp. "Autonomous Multi-material Construction with a Heterogeneous Robot Team." In: *Proc. DARS*. Springer, 2019, pp. 385–399.

[36] Kamel S. Saidi, Thomas Bock, and Christos Georgoulas. "Robotics in Construction." In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 1493–1520.

[37] Sérgio R Barros dos Santos, Sidney Givigi, Cairo L Nascimento, Jose M Fernandes, Luciano Buonocore, and Areolino de Almeida Neto. "Iterative Decentralized Planning for Collective Construction Tasks with Quadrotors." In: *Journal of Intelligent & Robotic Systems* 90.1-2 (2018), pp. 217–234.

[38] Pedro J Sanz, Pere Ridao, Gabriel Oliver, Giuseppe Casalino, Carlos Insaurralde, Carlos Silvestre, Claudio Melchiorri, and Alessio Turetta. "TRIDENT: Recent improvements about autonomous underwater intervention missions." In: *IFAC Proceedings Volumes* 45.5 (2012), pp. 355–360.

[39] Jungwon Seo, Mark Yim, and Vijay Kumar. "Assembly sequence planning for constructing planar structures with rectangular modules." In: *Proc. ICRA*. IEEE. 2016, pp. 5477–5482.

[40] Satja Sivčev, Joseph Coleman, Edin Omerdić, Gerard Dooly, and Daniel Toal. "Underwater manipulators: A review." In: *Ocean Engineering* 163 (2018), pp. 431–450.

[41] Ashley Stroupe, Terry Huntsberger, Avi Okon, Hrand Aghazarian, and Matthew Robinson. "Behavior-based multi-robot collaboration for autonomous construction tasks." In: *Proc. IROS*. IEEE. 2005, pp. 1495–1500.

[42] Ken Sugawara and Yohei Doi. "Collective construction of dynamic structure initiated by semi-active blocks." In: *Proc. IROS*. IEEE. 2015, pp. 428–433.

[43] Julius Sustarevas, Daniel Butters, Mohammad Hammid, George Dwyer, Robert Stuart-Smith, and Vijay M Pawar. "MAP-A Mobile Agile Printer Robot for on-site Construction." In: *Proc. IROS*. IEEE. 2018, pp. 2441–2448.

[44] Yuzuru Terada and Satoshi Murata. "Automatic modular assembly system and its distributed control." In: *Int. J. Robot. Res.* 27.3-4 (2008), pp. 445–462.

[45] Michael T Tolley and Hod Lipson. "Programmable 3d stochastic fluidic assembly of cm-scale modules." In: *Proc. IROS*. IEEE. 2011, pp. 4366–4371.

[46] Makoto Utsumi, Taketsugu Hirabayashi, and Muneo Yoshie. "Development for teleoperation underwater grasping system in unclear environment." In: *International Symposium on Underwater Technology* (2002).

[47] Roberto G Valenti, Ivan Dryanovski, and Jizhong Xiao. "Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs." In: *Sensors* 15.8 (2015), pp. 19302–19330.

[48] Iuliu Vasilescu, Paulina Varshavskaya, Keith Kotay, and Daniela Rus. "Autonomous Modular Optical Underwater Robot (AMOUR) design, prototype and feasibility study." In: *Proc. ICRA*. Vol. 2. IEEE; 1999. 2005, p. 1603.

[49] Todd Wareham and Andrew Vardy. "Putting it together: The computational complexity of designing robot controllers and environments for distributed construction." In: *Swarm Intelligence* 12.2 (2018), pp. 111–128.

[50] Justin Werfel. "Building Blocks for Multi-robot Construction." In: *Proc. DARS*. Springer, 2004.

[51] Justin Werfel. "Collective construction with robot swarms." In: *Morphogenetic Engineering*. Springer, 2012, pp. 115–140.

[52] Justin Werfel, Yaneer Bar-Yam, Daniela Rus, and Radhika Nagpal. "Distributed construction by mobile robots with enhanced building blocks." In: *Proc. ICRA* (2006), pp. 2787–2794.

[53] Justin Werfel and Radhika Nagpal. "Three-dimensional construction with mobile robots and modular blocks." In: *Int. J. Robot. Res.* 27.3-4 (2008), pp. 463–479.

[54] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. "Designing collective behavior in a termite-inspired robot construction team." In: *Science* 343.6172 (2014), pp. 754–758.

[55] Jan Willmann, Federico Augugliaro, Thomas Cadalbert, Raffaello D'Andrea, Fabio Gramazio, and Matthias Kohler. "Aerial robotic construction towards a new field of architectural research." In: *International Journal of Architectural Computing* 10.3 (2012), pp. 439–459.

[56] Seung-kook Yun, Mac Schwager, and Daniela Rus. "Coordinating construction of truss structures using distributed equal-mass partitioning." In: *Robotics Research*. Springer, 2011, pp. 607–623.

[57] Enrica Zereik, Marco Bibuli, Nikola Mišković, Pere Ridao, and António Pascoal. "Challenges and future trends in marine robotics." In: *Annual Reviews in Control* (2018).