Autonomous assembly of structures using pinning control and formation algorithms

Kléber M. Cabral*, Sidney N. Givigi, Jr.† Peter T. Jardine*
* Department of Electrical and Computer Engineering, Royal Military College of Canada
Kingston, ON, Canada
Email: cabral@rmc.ca, peter.jardine@rmc.ca
† School of Computing, Queen's University
Kingston, ON, Canada
Email: sidney.givigi@queensu.ca

***Abstract*—In this paper, we propose an autonomous construction system for building structures composed of multiple mobile robots. We use formation control techniques to guide the mobile robots to the desired positions in the structure, and pinning control methodologies to guide the group of robots to desired mounting locations. We also present here a technique for modelling the system based on network control theory. We demonstrate how our network-based approach for autonomous assembly must communicate with the robots in a system architecture diagram. In order to validate our approach, we show a simulation trial of the assembly of a pyramidal structure with nine robots.**

I. Introduction

Some systems are distributed by their nature, such as biological and transport systems, social relationships or swarms of robots. These systems are consider distributed because the behaviour of the larger system depends on the actions of multiple individual agents. The agents interact with, affect and are affected by others [1]. Examples of such interactions include synapses between neurons, detection of neighbouring robots using sensors, communication messages between computers, etc [2], [3].

Systems with multiple agents are often interpreted as networks, with the agents as the network nodes and their connections (i.e., communication, interaction) as the network edges [1], [2]. The ability to control a network means the ability to drive the states of all agents toward desired values. Feasibility, controllability and stability of operation are important aspects that must be considered when controlling a network. Such analysis often relies on network theory and graphs to derive the control laws [4].

For mobile robots, network control methodologies are usually applied to maintain formation [3], [5], [6], area coverage [5], [7], flocking [6], [8], or rendezvous [3], [9].

In this work, we apply network control methodologies to autonomous construction with multi-robot systems. Specifically, we propose the use of network control theory and formation techniques for assembling structures composed of multiple robots.

We follow the idea of an autonomous construction system presented by Cabral *et al.* [10]. The authors provide a high-level description of a construction system that uses quadrotors inside of cubic frames as the building blocks of the structure. The construction system architecture presented describes the information flow between the system's components.

Inspired by the system architecture in [10], we show a new system diagram where the proposed network-based algorithms are placed in the high-level system description. The contributions of this paper are as follows, first, it formulates a multi-robot formation algorithm to position many robots in a desired structure. Second, it demonstrates that pinning control can be used to drive the group of robots into a specific mounting position in space. Lastly, it investigates the change in desired inter-agent distance that allows for assembly of the structure.

Formation algorithms provide control inputs to each robot in a multi-agent system guiding the vehicles to a desired relative position within a pattern or shape [11]. The term *pinning control* defines the placement of local feedback controllers on a small fraction of the network nodes. Such nodes are called pins or pinned nodes. The robots not directly actuated will be influenced indirectly through their connections with the pins [12], [13]. Therefore, pinning control allows one to change the state of all nodes (controlling the network) by actuating on a subset of the nodes.

We execute a simulation trial to demonstrate the ability of the proposed approach to perform autonomous construction of structures using actuated parts.

The remaining of this work is organized as follows Section II shows related works and the mathematical formulation of the construction problem. Section III presents our solution, with the system architecture and control algorithms. Section IV shows the simulated results. Lastly, our conclusions are presented in Section V.

II. Preliminaries

In this section, we show previous related works in the literature for autonomous construction using robotic systems, control of swarms of vehicles, and pinning control of networks. Also, we present the problem formulation for autonomous construction with automated blocks using a formation algorithm.

A. Previous Work

Robotic systems that are capable of performing autonomous construction have been studied using various control methodologies and different robotic platforms. For example, in [14] the authors use a simulated 2-D environment to build desired structures using behaviour-based agents. Similarly, in [15] and [16] the authors use mobile robots inspired by termites, with limited sensing and communication, and were able to build a structure with simple control rules.

Wheeled land vehicles with a manipulator are also a common robotic platform to execute autonomous construction [17], [18]. Those robots are particularly interesting due to their stability, the weight carrying capability, and the ability to place the parts in a specific position or orientation in the structure.

Multicopters are another possible platform to be used for picking, transporting and positioning parts. Among other advantages, one can cite the capability of vertical take-off and landing, payload carrying, and the possibility of operating at different heights [19], [20].

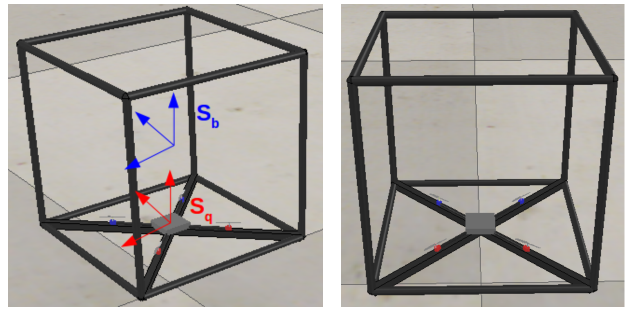
Instead of transporting parts, mobile robots can also be used as pieces of a structure. In [21], [22] the authors used small autonomous boats to build structures on the surface of the water and in [10] the authors considered small quadrotors within cubic frames as structure parts.

The autonomous construction task requires the planning of the movement of each part (automated or not). The simplest approach is using deterministic trajectories manually designed for each part [10], [20]. Moreover, learning and heuristic search algorithms can be used to derive the sequence of maneuvers, the sequence of assembly and the set of trajectories [19].

In this work, we use formation algorithms to drive the vehicles to the correct place in the structure. Formation is a particular case of swarms of multi-robot systems where the agents must assume a predefined position in a pattern. In [23] the authors define the control laws that can be used to actuate in swarms and navigate through an environment.

In [11] the authors show three approaches for obtaining formations with multiple vehicles: a) position-based, where each agent knows the global position of its neighbours; b) displacement-based, where each agent measures the relative position of its neighbours but describes the calculated position in the global frame; and c) distance-based, where each agent describes the relative position of neighbours on its own local frame. The authors describe the characteristics of each approach for specific scenarios, such as the needlessness of communication between agents in a local representation approach.

Pinning controllers can be used to control swarms of vehicles by describing the system as networks. In [24] the authors consider that only a fraction of the swarm agents has access to the desired reference. With this approach, all nodes have a control law component that tries to maintain the swarm cohesion, but only part of the agents are aware of the mission objective (reference). This contrasts with the idea presented in [23], where all agents were able to access the reference. In real systems, one presumable advantage of pinning control is the reduction of bandwidth needed for communication, given that the reference information is not needed in all agents anymore.



(a) (b)

Fig. 1. Quadrotor inside the construction block.

In this paper, we look at the autonomous assembly problem as a formation problem, and therefore, positioning multiple blocks in a structure is modelled as arranging robots in a formation. Also, given that pinning controllers can be applied to change the state of a group of agents, in our approach pinning controllers drive the robots to the mounting area. To best of our knowledge, this modelling method constitutes a novel approach to solve the autonomous construction problem with actuated parts.

B. Problem Formulation

In this work, we study the problem of autonomous assembling of structures using actuated blocks. The blocks have a cubic frame with quadrotors attached (Fig. 1). Also, the three dimensions (height, width, length) of each block are equal to d.

Let us describe three coordinate frames needed for our mathematical formulation hereafter. First, consider an inertial frame fixed in the environment, S_I . The axes X_I and Y_I describe the horizontal plane, and Z_I points upwards. Second, we assume that each block has a coordinate frame S_b located on its geometric center, with axes X_b, Y_b, and Z_b. Lastly, we can describe S_q as a reference frame fixed at the center of gravity of the quadrotor. The axes X_q and Y_q are coplanar with the plane formed by the four motors. X_q points toward the front of the quadrotor between the two front motors. The Y_q axis is directed 90º to the left of X_q. Z_q points upwards (towards the origin of S_b), perpendicular to the plane of the motors. Both S_b and S_q are depicted in Fig. 1a.

The relationship between the inertial frame (S_I) and the block frame (S_b) is given by the angles of pitch (rotation about Y_b), roll (rotation about X_b), and yaw (rotation about Z_b). We can assume that there is no rotation between S_q and S_b, but S_b is translated with respect to S_q along $Z_q = Z_b$ axis (see Fig. 1a). Thus, the angles of (roll=ϕ, pitch=θ, yaw=ψ) describe the rotation between quadrotor frame (S_q) and inertial (S_I), as well as between block frame (S_b) and inertial (S_I).

The rotation matrix from S_q to S_I is given by R_q^I (1).

$$R_q^I = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (1)$$

where $c(\cdot)$ and $s(\cdot)$ denote $cosine\,(.)$ and $sine\,(.)$, respectively.

Each motor of a quadrotor generates a force F_p in the direction of Z_q. These parameters can be expressed in the coordinate system S_q as:

$$\begin{aligned} T &= F_1 + F_2 + F_3 + F_4 \\ \tau_x &= \frac{\sqrt{2}}{2} L \left(F_1 + F_4 - F_2 + F_3\right) \\ \tau_y &= \frac{\sqrt{2}}{2} L \left(-F_1 - F_2 + F_3 + F_4\right) \\ \tau_z &= \frac{k_M}{k_F} \left(F_1 - F_2 + F_3 - F_4\right) \end{aligned} \tag{2}$$

where k_F and k_M are the propulsion and moment coefficients produced by the p^{th} rotor/propeller set. L is the distance between the center of mass of the quadrotor and the axis of rotation of the rotors. τ_x, τ_y, τ_z, are torques around the axes X_q, Y_q, and Z_q respectively. T is the total exerted force in the Z_q axis.

We consider that the rotors numbers 1, 2, 3, and 4 are on the front-left, front-right, back-right, back-left positions respectively.

The linear acceleration vector $\begin{bmatrix}\dot{u} & \dot{v} & \dot{w}\end{bmatrix}^T$ is measured at the center of mass of the quadrotor (origin of S_q) and it is expressed in S_q. Note that u, v, and w are linear velocities in the axes X_q, Y_q, and Z_q respectively. We can express the linear acceleration vector $\ddot{p_q} = \begin{bmatrix}\ddot{x} & \ddot{y} & \ddot{z}\end{bmatrix}^T$ in the inertial reference system S_I as:

$$\ddot{p_q} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R_q^I \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \tag{3}$$

where g is the gravitational constant. The linear acceleration in Z_q is computed as $\dot{w} = \frac{T}{m}$, where m is the mass of the quadrotor.

Let us define the the rotational dynamics of the vehicle as:

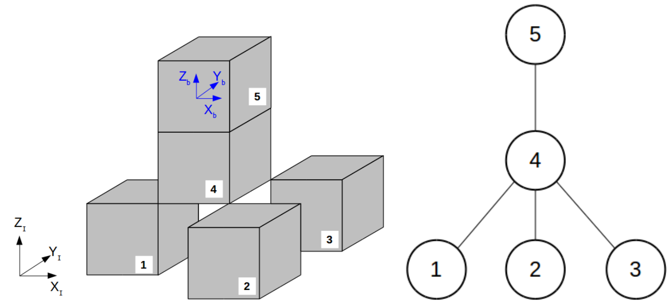
$$\begin{aligned} \dot{p} &= \left[\left(I_{yy} - I_{zz}\right) qr/I_{xx}\right] + \tau_x/I_{xx} \\ \dot{q} &= \left[\left(I_{zz} - I_{xx}\right) pr/I_{yy}\right] + \tau_y/I_{yy} \\ \dot{r} &= \left[\left(I_{xx} - I_{yy}\right) pq/I_{zz}\right] + \tau_z/I_{zz} \end{aligned} \tag{4}$$

where p, q, and r are the angular velocities of roll, pitch, and yaw (around axes X_q, Y_q, and Z_q). Also, I_{xx}, I_{yy}, and I_{zz} are the inertia moments of the quadrotor with respect to the axis X_q, Y_q, and Z_q respectively.

The angular velocity vector (described in S_q) is $\begin{bmatrix}p & q & r\end{bmatrix}^T$ and the angular acceleration vector (also in S_q) is $\begin{bmatrix}\dot{p} & \dot{q} & \dot{r}\end{bmatrix}^T$. We can obtain the angular acceleration vector $\begin{bmatrix}\ddot{\phi} & \ddot{\theta} & \ddot{\psi}\end{bmatrix}^T$ in the inertial coordinate frame (S_I) by applying:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\theta \\ 0 & s\phi sec\theta & c\phi sec\theta \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{5}$$

where $t(\cdot)$ and $sec(\cdot)$ denote $tangent\,(.)$ and $secant\,(.)$, respectively.



(a) Structure composed of multiple blocks. (b) Network topology.

Fig. 2. Structure composed of multiple blocks

During the structure assembly stage, we can describe each construction block as an agent in a network. Also, we can consider that, logically (in the network topology), each robot is connected to its adjacent neighbours (Fig. 2). An edge between an agent i and j exists if, in the final structure, both blocks are adjacent. Note that, during the assembly stage studied in this work the blocks are not physically connected yet. However, the robots are logically connected, i.e., they are able to measure each other's position to compute the formation algorithm. For example, in Fig. 2 robot number 1 is always able to measure the position of robot 4, but not the position of robot 2.

A network can be visually represented as a graph, with the agents being the nodes and their relationships represented by the edges [4]. A graph is a set $G = \{V, E, \mathbf{A}\}$ that contains:

- a finite set V with N nodes;
- the finite edge set $E \subseteq V \times V$, such that $e_{ij} = \{v_i, v_j\} \in E$ describes and edge from v_i to v_j. Note that for undirected graphs, $e_{ij} = e_{ji}$;
- the adjacency matrix $\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times N}$ (positive semidefinite matrix with N columns and N rows), that encodes the adjacency relationships in the graph. The elements of $\mathbf{A}$, a_{ij} are equal to 1 if the node v_i can receive information from a node v_j. For undirected graphs, $\mathbf{A}$ is symmetric.

For example, the structure in Fig. 2b can be represented as

$$\begin{aligned} G = \{\ & V = \{v_1, v_2, v_3, v_4, v_5\}, \\ & E = \{e_{14}, e_{24}, e_{34}, e_{45}\}, \\ & \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}\ \} \end{aligned}$$

We represent the assembly process as a formation-reaching problem. Thus, guiding each construction block to the structure is the same as driving all agents from a disorganized arrangement to a specific pattern.

First let us describe the position of each agent (i) as $p_i = \begin{bmatrix}x_i & y_i & z_i\end{bmatrix}^T$, expressed in S_I. Also, the dynamics of the agent takes the form of a single integrator:

$$\dot{p}_i = u_i \tag{6}$$

where $i = 1, 2, ..., N$, and N is the number of agents. u_i is the desired velocity of each agent.

In [11], the authors present the following control law designed to achieve formations:

$$u_i = -k_f \sum_{j=1, j\neq i}^{N} [(p_i - p_j) - (p_i^* - p_j^*)] \tag{7}$$

where $p^* = \begin{bmatrix} x^* & y^* & z^* \end{bmatrix}^T$ is the desired position of a node in the structure. Also, k_f is a positive gain for the formation controller.

Note that, p_i, p_j, p_i^*, and p_j^* are described in the inertial frame. Moreover, the magnitude of the vector $(p_i^* - p_j^*)$ describes the desired inter-agent distance between i, j and the magnitude of $(p_i - p_j)$ is the actual distance.

The assembly process ends when all parts are positioned at the desired final position in the formation. At the end of the construction process, all blocks are physically attached to any adjacency block.

III. Autonomous construction architecture

During this section, we describe the control algorithm that performs autonomous assembly of a structure and demonstrate how it can be integrated with the system structure proposed in [10].

A. Control Algorithm

The control law described in equation (7) drives the states of all robots to a predefined location in a formation. However, the agents try to achieve a relative position between neighbours, but this control law will not necessarily guide the whole formation to a desired position in space. For example, for a system with two agents at the positions ($p_1 = 0$, $p_2 = 0$), the control signal u_1 is equal to -2 for both ($p_1^* = 1$, $p_2^* = -1$) or ($p_1^* = 10$, $p_2^* = 8$). Therefore, we must add another term that will guide the whole group of robots to a desired location in space, called mounting area.

First, let us categorize the autonomous blocks into two classes, *normal robots* and *pin robots*. Pinning control strategy describes the placement of feedback controllers on a small fraction of nodes (pin robots). That is, only a small portion of the robots have the information about the desired position in space.

We apply the pinning control strategy to drive all agents to the mounting area by only controlling a few robots. The full control signal of each robot is composed of a formation term and a pinning control term:

$$u_i = -h_i k_p (p_i - p_r) - k_f \sum_{j=1, j\neq i}^{N} a_{ij} [(p_i - p_j) - (p_i^* - p_j^*)] \tag{8}$$

and a_{ij} is a element of the adjacency matrix $\mathbf{A}$. h_i is equal to 1 only if i is a pin robot in the network, and 0 otherwise. p_r is the reference position, that is, the physical position of the node i in the mounting area, written in S_I. $k_p > 0$ is the pinning controller gain.

Note that p_r describes the final position of the node i in space, and it is written in S_I. As mentioned, p^* is used to describe relative position between modules, and not the absolute position of the formation.

We propose two algorithms for the autonomous construction problem. First, algorithm 1 is used to achieve formation with robots at the mounting area. Second, algorithm 2 defines the desired relative position between all robots (inter-agent distance), as well as the pin position in space.

Algorithm 1 Formation

Get the actual position of the node (p_i)
Get the actual position of the adjacent nodes (p_j)
Get the desired position of the node (p_i^*)
Get the desired position of the adjacent nodes (p_j^*)
if i is a pin node **then**
    Get the desired final position of node i, p_r
    $h_i \leftarrow 1$
else
    $h_i \leftarrow 0$
end if
Compute the desired velocity using equation (8)

Algorithm 2 Structure Assembly

Define the desired position of all nodes in the structure $p^* = \{p_1^*, p_2^*, ..., p_N^*\}$;
Define the final position of node i, p_r
$p^* = kp^*$ */* multiply the value of position in structure by a positive constant $k > 1$ to obtain the desired positions in the formation */*
while Structure not assembled **do**
    if Pinned node i reached desired position (p_r) **then**
        $p^* = kp^*$ */* multiply the desired position in the formation by a positive constant $k < 1$ to reduce inter-agent distance */*
        Wait for nodes to get to the position
    end if
end while

The desired behaviour of the autonomous construction process using the two proposed algorithms is that first the blocks are driven to a pattern that is a scaled-up version of the structure. That is, the robots are driven to a formation where the inter-agent distance is bigger than in the assembled structure. Also, the pin robot drives the whole group to the mounting area. As time passes, inter-agent distance is reduced and the parts get closer until the final structure is assembled.

B. High-Level System Architecture.

The high-level system architecture used here was originally proposed by Cabral *et al.* [10]. The authors describe the system architecture needed for the autonomous constriction task with actuated parts, showing the information flow between system modules.

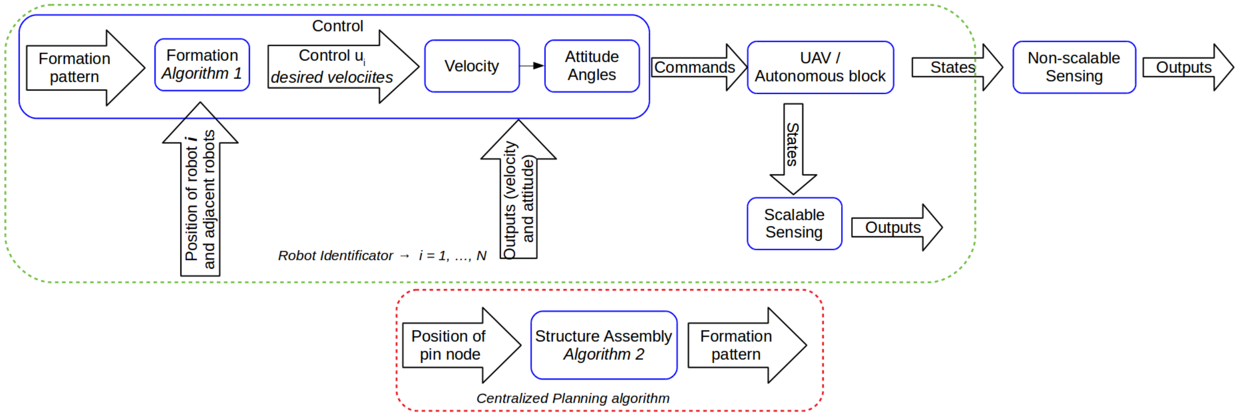


Fig. 3. System architecture structure assembly using network control.

The construction algorithms presented in Section III-A are responsible for coordinating the movement of all agents, and they can be classified as:

- Algorithm 1 is a *position control algorithm*, being executed locally at each robot in order to achieve the desired location in a formation.
- Algorithm 2 is seen as a *centralized planner*, it has only one instance of execution during system operation, and it affects the desired position of all nodes.

Fig. 3 shows the system architecture. Note that the centralized planning stage and the cascaded controllers described above. The structure is considered to be built once the formation size (inter-agent distance), is the same as the real structure size. Also, in contrast with the architecture presented by Cabral *et al.* [10], our approach does not use any learning module applied to the structure assembly task.

It is important to notice that, it is desired that the convergence of algorithm 1 is faster than algorithm 2. That is, the construction blocks must assume their places in the formation before the inter-agent distance is reduced (agents move to the final destination).

IV. Simulation and Results

We performed a simulation trial of the autonomous construction of a pyramidal structure to evaluate the algorithms proposed. The simulation environment is composed of a) *MATLAB/Simulink* - velocity controllers, Formation algorithm and Structure Assembly algorithm; b) *Virtual Robot Experimentation Platform* (V-REP) - physical simulation of quadrotors and low-level controllers (attitude); c) *Robot Operating System* (ROS) communication between the previous software.

It is desired that the final attitude $\begin{bmatrix}\phi & \theta & \psi\end{bmatrix}$ is equal to zero for all blocks, and the desired yaw angle (ψ^i) is equal to zero during the entire simulation.

As shown in Fig. 3, the *Control* stage consists of a set of Formation algorithm and cascaded PID controllers empirically adjusted. Fig. 4 shows the location of each controller in our simulation environment. Each robot has its own set of controllers, as well as a Formation algorithm.

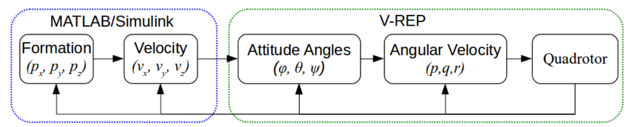
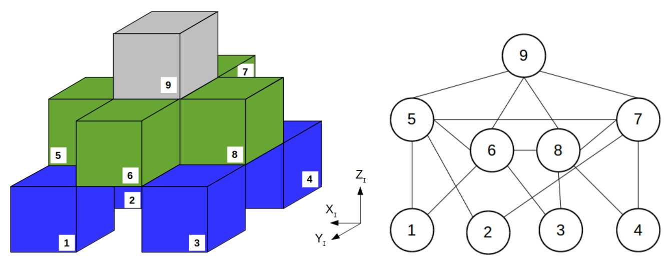


Fig. 4. Set of controllers for each quadrotor.



(a) Physical view. (b) Graph.

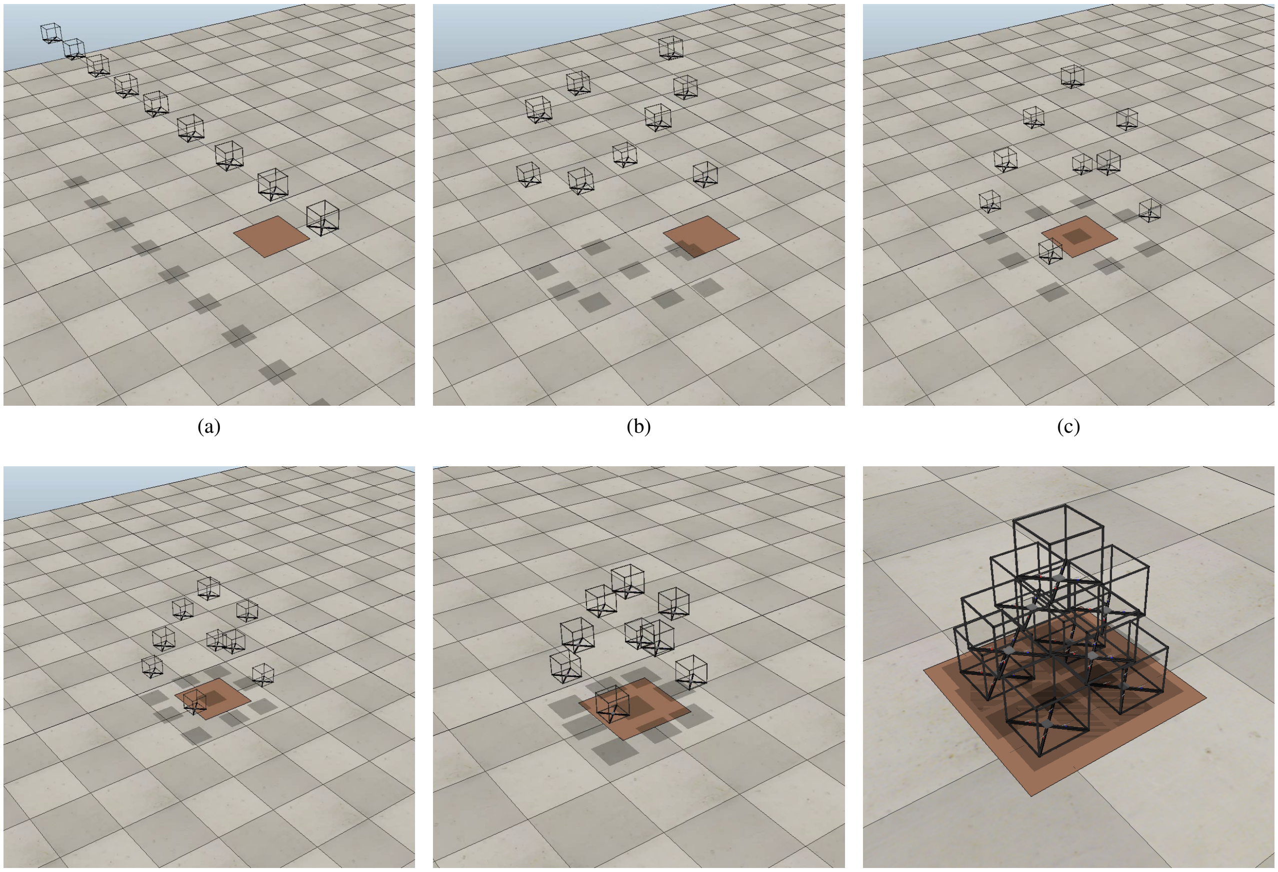
Fig. 5. Desired structure - pyramid.

While the Formation algorithm provides the desired velocities for each part in the system (described in S_I), the attitude controllers output the angular speed of each one of the quadrotor propellers. All quadrotors are simulated as if they had ideal sensors (no noises and uncertainties are considered in the measurements).

At the beginning of the operation, each robot is displaced to a *waiting position* situated above the desired mounting area. Note that, when two blocks get sufficiently close, they become physically and rigidly attached in simulation. That is, the blocks are not able to move independently after attached. Also, since all parts arrive at the same time in the structure, the motors of each vehicle stay active until the structure is fully assembled.

The pyramidal structure showed in Fig. 5 is composed of nine actuated parts. Blocks from 1 to 4 are closer to the ground on the bottom level, blocks from 5 to 8 are on the middle level, and block number 9 is on the top level with the highest height (Z_I value). It is possible to see that each node is connected to an adjacent block in the graph representation of the structure.

Fig. 6 shows the autonomous assembly process of the



(a) (b) (c)

(d) (e) (f)

Fig. 6. Assembly of a pyramidal structure.

pyramid as well as the final structure. It is possible to see that each actuated part moves first to the waiting position, then they take place on the formation configuration in the mounting area, and as the inter-agent distance is reduced, the structure is assembled.

We selected the autonomous block number 9 (robot on top of the pyramidal structure) as the pin node. That is, given the Formation algorithm, $h_i = 1$ in equation (8) only for this node. That means that only the robot 9 is "aware" of the desired position in the mounting area. Fig. 7 shows position of the robot 9. The figures show the actual position of the part $p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$, expressed in the inertial frame, S_I. In the figure it is highlighted which control action was predominant at each time interval.

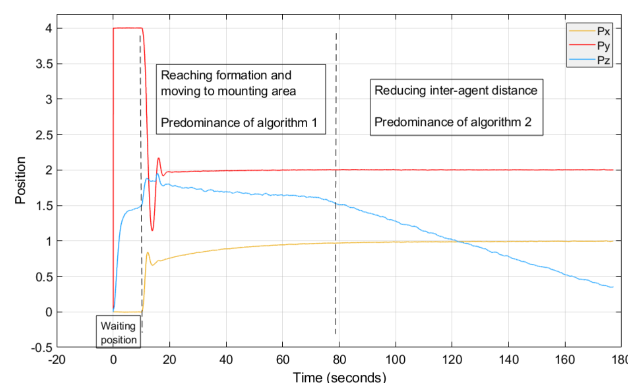


Fig. 7. Position of node 9 showing the action of the algorithms proposed

V. Conclusions and Future Work

In this paper, we proposed control and planning algorithms that enable the autonomous construction process using actuated parts. We modelled the multi-agent system as a network and the assembly of a structure as a formation problem.

We started the assembly process by driving the blocks to a similar shape of the final structure, but with a bigger inter-agent distance. After the robots reached their position in the formation, we gradually reduced the inter-agent distance in order to assemble the structure (shrinking the formation size).

Using a simulation trial, we demonstrated that the proposed algorithms are able to perform autonomous construction using actuated parts. In the future, we intend to explore priority on the mounting sequence using the proposed formulation, as well as control techniques that can be used after the structure is assembled.

References

[1] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[2] E. Nozari, F. Pasqualetti, and J. Cortés, "Heterogeneity of central nodes explains the benefits of time-varying control scheduling in complex dynamical networks," *Journal of Complex Networks*, Feb. 2019.

[3] J. Cortés and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[4] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, pp. 40–52, Mar. 2014.

[5] G. Notomista and M. Egerstedt, "Constraint-driven coordinated control of multi-robot systems," *arXiv:1811.02465 [cs]*, Nov. 2018.

[6] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, Jan. 2007.

[7] K. Laventall and J. Cortés, "Coverage control by multi-robot networks with limited-range anisotropic sensory," *International Journal of Control*, vol. 82, no. 6, pp. 1113–1121, 2009.

[8] X. Wang and H. Su, "Pinning control of complex networked systems: A decade after and beyond," *Annual Reviews in Control*, vol. 38, no. 1, pp. 103–111, 2014.

[9] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[10] K. M. Cabral, S. N. Givigi, S. R. B. d. Santos, and P. T. Jardine, "Design of a self-assembly system of three-dimensional structures using autonomous construction blocks," in *2019 IEEE International Systems Conference (SysCon)*, pp. 1–8, April 2019.

[11] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control: Position-, displacement-, and distance-based approaches," *Number: Gist DCASL TR*, vol. 2, 2012.

[12] Zhi-Hong Guan, Zhi-Wei Liu, Gang Feng, and Yan-Wu Wang, "Synchronization of complex dynamical networks with time-varying delays via impulsive distributed control," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, pp. 2182–2195, Aug. 2010.

[13] A. Adaldo, F. Alderisio, D. Liuzza, G. Shi, D. V. Dimarogonas, M. di Bernardo, and K. H. Johansson, "Event-triggered pinning control of switching networks," *IEEE Transactions on Control of Network Systems*, vol. 2, pp. 204–213, June 2015.

[14] A. Panangadan and M. G. Dyer, "Construction in a Simulated Environment Using Temporal Goal Sequencing and Reinforcement Learning," *Adaptive Behavior*, vol. 17, pp. 81–104, Feb. 2009.

[15] J. Werfel, K. Petersen, and R. Nagpal, "Designing Collective Behavior in a Termite-Inspired Robot Construction Team," *Science*, vol. 343, pp. 754–758, feb 2014.

[16] K. H. Petersen, R. Nagpal, and J. K. Werfel, "TERMES: An Autonomous Robotic System for Three-Dimensional Collective Construction," in *Robotics: Science and Systems VII*, MIT Press, 2011.

[17] M. Dogar, R. A. Knepper, A. Spielberg, C. Choi, H. I. Christensen, and D. Rus, "Multi-scale assembly with robot teams," *The International Journal of Robotics Research*, vol. 34, pp. 1645–1659, nov 2015.

[18] A. Bolger, M. Faulkner, D. Stein, L. White, S. Yun, and D. Rus, "Experiments in decentralized robot construction with tool delivery and assembly robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5085–5092, Oct 2010.

[19] S. R. B. d. Santos, C. L. N. Júnior, and S. N. Givigi, "Planning and learning for cooperative construction task with quadrotors," in *2014 IEEE International Systems Conference Proceedings*, pp. 57–64, mar 2014.

[20] K. M. Cabral, S. R. B. dos Santos, S. N. Givigi, and C. L. Nascimento, "Design of model predictive control via learning automata for a single uav load transportation," in *2017 Annual IEEE International Systems Conference (SysCon)*, pp. 1–7, April 2017.

[21] J. Seo, M. Yim, and V. Kumar, "Assembly planning for planar structures of a brick wall pattern with rectangular modular robots," in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, (Madison, WI, USA), pp. 1016–1021, IEEE, aug 2013.

[22] J. Seo, *Grasping and Assembling with Modular Robots*. PhD thesis, University of Pennsylvania, 2014.

[23] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, pp. 401–420, March 2006.

[24] X. Wang, X. Li, and J. Lu, "Control and flocking of networked systems via pinning," *IEEE Circuits and Systems Magazine*, vol. 10, no. 3, pp. 83–91, 2010.