Design and Simulation of a Multi-robot Architecture for Large-scale Construction Projects

Vivek Thangavelu[1] and Nils Napp [1]

***Abstract*— Large-scale construction projects can benefit from having a team of heterogeneous building robots operating autonomously and cooperatively on unstructured environments. In this work, we propose a flexible system architecture, MARSala, that allows teams of distributed mobile robots to construct motion support structures in large and unstructured environments using purely local interactions. The paper primarily focuses on the deliberative layer of the architecture which provides a means for formulating a construction project as a motion support structure construction problem. We implemented the architecture in simulation and demonstrated the benefits of such a formulation in two different construction scenarios operating in large unstructured environments.**

I. INTRODUCTION

The advancement in construction automation holds promises to solve critical societal problems, allowing for safe, sustainable and inexpensive construction processes, as well as enabling new classes of applications, such as extraterrestrial construction [1], [2], [3]. A recent analysis of the technological gaps and advances in construction automation [1] suggests that, in order to reach a technological level where it is possible to achieve full on-site construction automation, further research is needed in automation of site preparation (e.g. leveling), construction of substructures (e.g. foundations) and auxiliary (temporary) structures, as well as in coordination of operations between robot systems. Multiple space agencies have revealed plans to establish human settlements in Mars and the Moon by 2050. This would entail large-scale construction tasks such as providing access to the site, preparing drainage, or building protective barriers, requiring long-term autonomy in modifying and operating in unstructured environments across vast areas of land. Such construction systems can also prove to be beneficial in many scenarios on earth such as disaster relief or in traditional construction efforts. Multi-robot systems (MRS) may play a vital role in large scale construction tasks, however, they pose additional challenges in coordination and planning [1], [3].

Though there has been considerable progress in using robots for construction [1], [3], [4], autonomous construction systems are predominantly designed for use in structured environments with standard construction infrastructure and require some level of human assistance. While some works concentrate on a centralized approach [5], [6], [7], many utilize purely local interactions. The TERMES system [8], [9] uses stigmergy to build a 3D structure inspired by mound-building termites, where a team of distributed climbing robots receive a set of low-level rules that collectively produce a specific structure using customized solid bricks. Stigmergy [10] is a biological phenomenon referring to environmentally mediated communication where information about building actions is encoded in the partially built structure. Allwright et al. [11] utilize stigmergic blocks that facilitate a multi-robot construction system. While such works operate on reactive decentralized building behaviors, they are restricted to specialized building materials and structured environments. Soleymani et al. [12] and Wawerla et al. [13] propose multi-robot systems that build protective barriers. The final shape is specified by a template and stigmergy allows for coordination and execution. Werfel [14] describes a decentralized coordination algorithm to build arbitrary 2D shapes by using individual robots as templates. These works operate in structured environments and are unable to erect large, tall structures.

While traditional construction requires a detailed blueprint, many builders in nature utilize a functional specification, where the utility of the structure is more important than its exact shape. This type of function-driven construction is robust and reactive to uncertainties in unstructured environments. This is particularly useful in remote locations such as disaster areas or extra terrestrial bodies for building utility structures like shelters, ramps or protective barriers. In biological swarm systems, the structures built are often many times the size of an individual and the building actions are coordinated without centralized control. For example, Leptothorax tubero-interruptus and other ant species use their brood pile as a template to build surrounding walls and are guided by stigmergy to coordinate build actions [15]. In Apicotermitinae nests, termites collectively build large, tall mounds in a layered approach [16] in reaction to numerous environmental stimuli [17] and are required to construct (helical) access ramps [18] to facilitate movement between floors. As such, scaffolding or access structures are required to facilitate large construction projects for ground-based systems.

The utilization of in-situ material is a suitable solution for many constructions tasks where proper infrastructure is not available or is limited. Many recent works have developed methods to fabricate construction material for additive construction from regolith [19], [20]. Most 3D-printed habitat systems [21] focus on a single, immobile robot system capable of building a specific design whose size is limited to its workspace. For large scale operations, construction tasks need to be designed to operate across a multi-robot system and largely devoid of a centralized

[1]Authors are with Electrical and Computer Engineering, Cornell University, Ithaca, NY 14850, USA

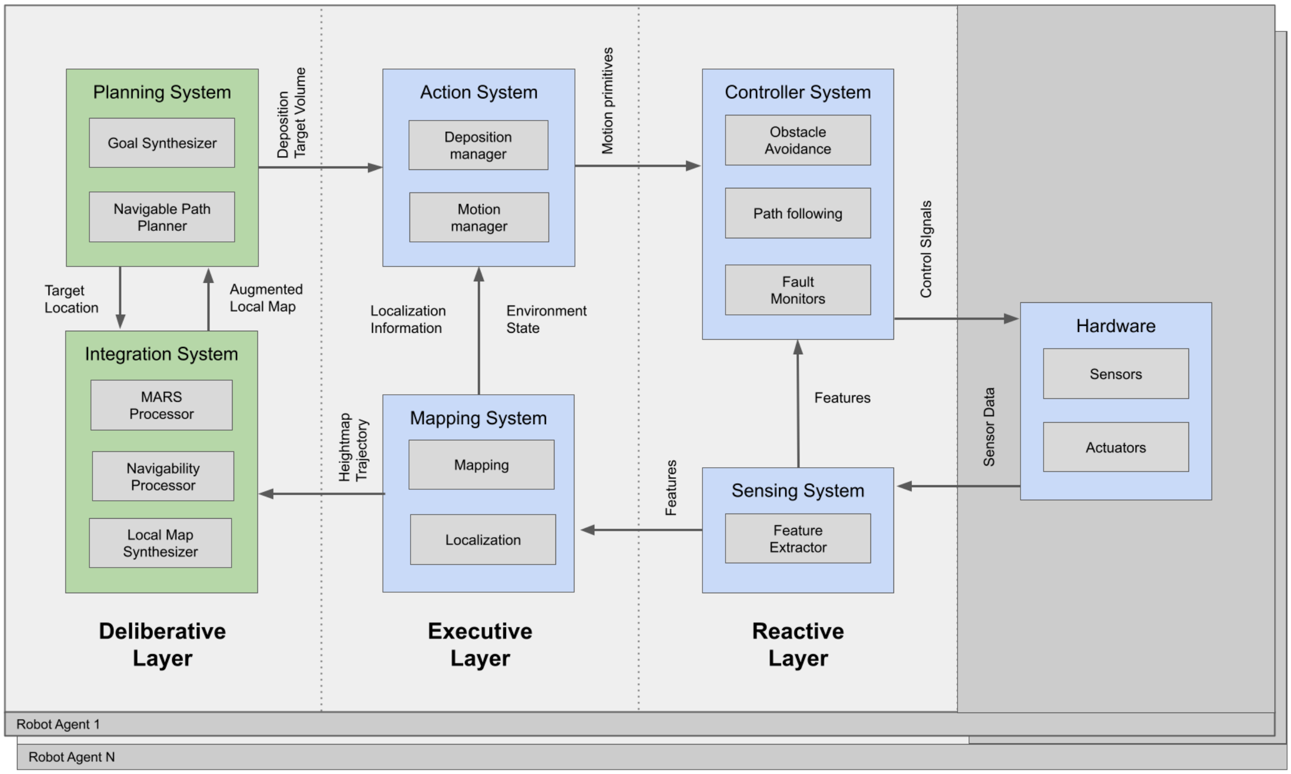

Fig. 1. The MARSala system architecture

architecture.

Simulation has been a useful tool in advancing the field of robotics. They offer a fast, safe and cost-effective alternative to real robot systems. Many types of general purpose robotic simulators exist such as Gazebo, V-Rep, WEBOTS and Pybullet for simulating various types of complex robotic systems, while simulators such as MuRoSimF and Argos are designed primarily for multi-robot systems in 3D space. A comprehensive review of various robot simulators is beyond the scope of this paper and the reader may refer to [22] for more details. Software frameworks that simulate or abstract the behavior of system components are thus indispensable tools to enable research and development of MRS techniques. They can help mitigate issues faced during the design and deployment of multi-robot systems. Deploying multi-robot systems in the real world is a challenging task; researchers would often need to develop and validate various coordination and collaborative robot mechanisms before deployment.

In this paper we propose a system architecture for the collective construction of motion support structures in large-scale unstructured environments called MARSala. A motion support structure is a type of structure whose purpose is to allow mobile agents to navigate from one location to another, where the exact shape of the structure is irrelevant. The architecture allows for decentralized coordination between various building agents based on purely local interactions. The construction approach is based on the Minimal Additive Ramp Structure (MARS) model [23] that allows building agents to determine the set of legal environment modifications that turns the environment into a navigable structure. Besides the construction model, we describe the architecture in terms of abstract components that can be adapted to any ground-based mobile agent. The MARSala architecture provides a means to translate various large-scale construction projects into a motion support structure problem. We demonstrate the benefits of this formulation using two different construction scenarios implemented in a simulated environment.

Simulation allows us to deliberately increase the number of building agents and the size of the operation area with fewer constraints than in a physical system. The construction of motion support structures with the MARS model has been demonstrated in limited physical scenarios, with one [24], [25] and two building robots [23] that use different types of building materials. In each case, the constructed structures are approximately the same size as the building agents.

A. Contribution

Our contribution is twofold. First, we proposed a flexible construction architecture, to enable large-scale, decentralized construction projects in unstructured environments by allowing mobile ground-based construction agents to build motion support structures using MARS. Second, we developed a simulation framework that implements the proposed architecture and allows for the design and evaluation of decentralized coordination algorithms for ground-based multi-agent construction systems.

The main features of our solution are as follows:

- MARSala is highly modular; each individual building agent is composed of modules that represent its actuation and sensing. The low level motion primitives such as material selection and handling, manipulation and motion are defined based on the user's requirement. The generalized notion of navigability (refer to § II) used is adaptable to any ground-based robot.
- The reactive building approach is based on the Minimal Additive Ramp Structure (MARS) model, which is a function of the environment and the robot's kinematic constraints. It calculates the amount of construction volume needed to make a given area navigable by the robot. This reactive construction approach allows the robot to respond to changes in the environments made by itself or other agents, and imperfect assembly even in unstructured environments.
- Rather than designating the deposition model, construction materials or methods, MARSala evaluates the validity of each construction action using MARS. This allows for heterogeneous robots to work cooperatively in the same structure.
- The architecture allows for construction volumes that are many times the size of an individual agent. The final structure is the composition of several intermediate structures, each of which can be traversed by the building agent. This principle allows for the continuous expansion of the built structure.
- Local construction rules allow for a decentralized coordination. Compiling target locations (refer to § III-A) as a function of the environment state allows for translating various construction problems into motion support structure construction problems that are inherently coordinated through the environment.

The remainder of the paper is structured as follows: §II discusses the various methods used to describe the abstract construction algorithm. §III details the system architecture, §IV describes the simulation framework, and §V describes the applications we used to demonstrate our proposed architecture. Finally, §VI concludes the paper.

II. Theoretical Background

We model the environment as continuous functions to effectively express arbitrary terrains. Consider the construction area Q, as a compact, simply connected, and finite subset of $\mathbb{R}^2$. It is the domain of a bounded, non-negative height function $h : Q \to \mathbb{R}^+$ which describes a *structure*.

A point $\mathbf{p} \in Q$ is termed **occupiable** if it satisfies the constraint:

$$|h(\mathbf{r}) - h(\mathbf{q})| \;\leq \epsilon + \kappa|\mathbf{r} - \mathbf{q}|,\ \forall(\mathbf{r}, \mathbf{q}) \in B_{\delta/2}(\mathbf{p}) \qquad (1)$$

where $|\cdot|$ represents the Euclidean distance between two points, $\kappa \in \mathbb{R}^+$ is the maximum climbable slope the robot can drive up or down, $\delta \in \mathbb{R}^+$ is the robot body length and $\epsilon \in \mathbb{R}^+$ is the maximum discontinuity that a robot can drive over per body length. $B_{\delta/2}(\mathbf{p})$ is the set of all points that are at a distance of at most $\delta/2$ from $\mathbf{p} \in Q$ i.e. $B_{\delta/2}(\mathbf{p}) = \{\mathbf{s} \in Q\ ;\ |\mathbf{s}-\mathbf{p}| \leq \delta/2\}$. A point $\mathbf{q} \in Q$ is termed **navigable** if it has a connected path $P \subseteq Q$ to the robot origin $\mathbf{o} \in Q$ such that each $\mathbf{p} \in P$ is *occupiable*. Formally, a structure is *navigable* if it is locally (parameter δ) close (parameter ϵ) to K-Lipschitz continuous. The operator $P_\kappa[h]$ projects any structure Q to the smallest function in $\mathcal{L}_K$, the space of K-Lipschitz functions on Q, that is at least as large as h, called the Minimal Additive Ramp Structure (MARS):

$$P_\kappa[h](\mathbf{p}) = \max_{\mathbf{q} \in Q} \{h(\mathbf{q}) - \kappa|\mathbf{q} - \mathbf{p}|\}, \forall \mathbf{p} \in Q. \qquad (2)$$

MARS provides the upper bounds for the least additive deposition required for building *Perfect Motion Support Structures*. Mobile robots are typically equipped with sensors that only provide local/partial information about their environment. In order to operate on local sensor information, we compute a local MARS bound by applying the operator $P_\kappa[h](\mathbf{p})$ to subsets of Q i.e. if the observation subset $Q_r \subseteq Q$, the partial function of the perfect structure is given by $P_\kappa[h](\mathbf{p})|_{Q_r}$. Modifications based on the local MARS bound are consistent with the global MARS bound [23].

The MARS gap, $\Delta P_\kappa[h]$, is the difference between the MARS bounds and the height function h that describes the structure, defined as:

$$\Delta P_\kappa[h] = P_\kappa[h] - h. \qquad (3)$$

The *Deposition Target Volume* is the MARS gap restricted to a chosen construction path $P \subseteq Q$. If the target volume is filled with construction material, P will represent a navigable path.

III. Architecture for Distributed Large Scale Construction

The proposed architecture, MARSala, is based on the well-known three-layer architecture [26] that integrates deliberative and reactive approaches for controlling mobile autonomous agents. MARSala is designed to deliberate each building agent to build a motion support structure to a target location. It allows for a multi-agent system to consist of heterogeneous agents that may be organized into a centralized, decentralized or a hybrid system with direct and/or indirect communication among them. Each agent comprises three functional layers, as shown in Fig. 1.

The *Deliberative Layer* consists of modules that generate high-level goals to fulfill construction and coordination tasks. Our contribution focuses on the various modules in this layer, which are described in Section III-A.

The *Executive Layer* is responsible to carry out the high-level goals proposed by the deliberative layer by sequencing them into behavior sequences for the reactive layer. More specifically, the action system consisting of the deposition and action managers compose high-level motion plans to fulfill the deposition target volume proposed by the deliberative layer, taking into account the agent's kinematic constraints and the deposition material used. The mapping system is tasked to build a dense global map and track the agent's trajectory which are required for executing the high-level goals.

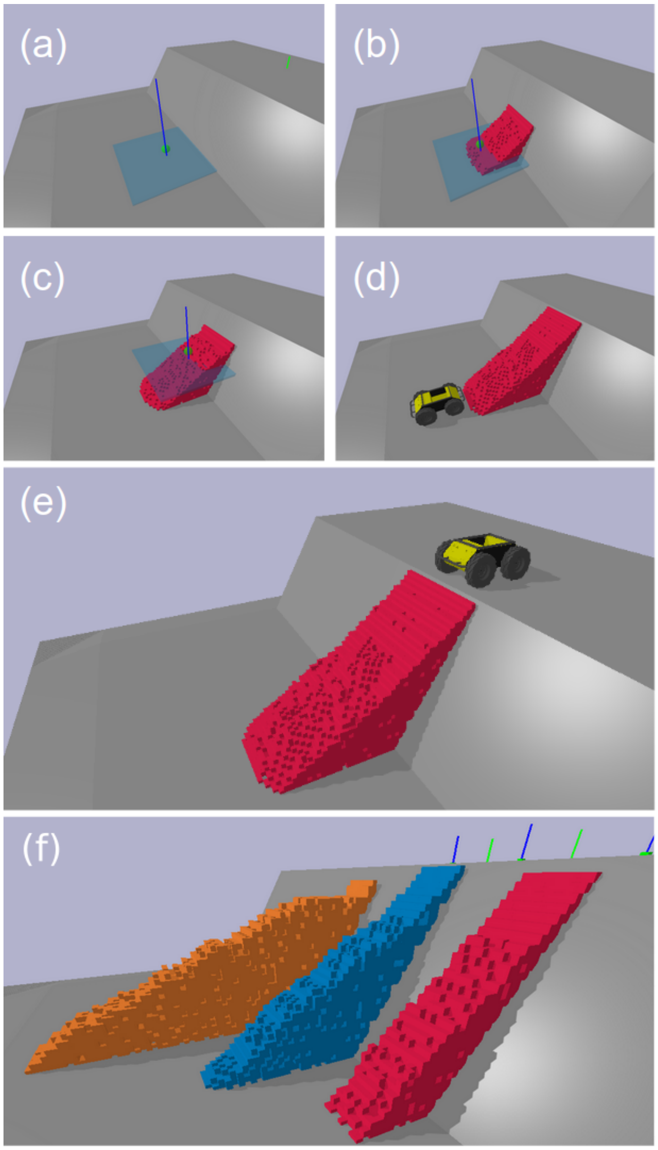

Fig. 2. Illustration of some simulated components of the MARSala architecture. (a) the perception field is shown as a blue shaded area, and the green block depicts the projection of the center of mass of the robot onto the ground. The construction process is depicted in (b-e) in a simple/regular environment. (f) depicts the differences in the final shape of the structures for different slope values used in the construction model.

The *Reactive Layer* consists of reactive components of the system that make decisions at run-time, using integrated sensor information and applying situation-action rules to control the *hardware system* and execute the high-level motion plans.

A. Deliberative Activities

The building agent is tasked to build a navigable path to a target location. The modules in this layer synthesize a tractable solution that is inherently coordinated through the environment state and local interactions. At each building step, the deliberative layer operates on a discretized heightmap of the agent's local environment and outputs the target deposition volume.

`Navigability Processor`: It calculates the occupiability of a cell in a discretized heightmap using (1). All cells that have an occupiable path to the agent's current position are termed navigable.

`MARS Processor`: It estimates the MARS gap (3) for a discretized heightmap. The MARS gap is the minimum deposition height for each cell in the heightmap in order to make the entire local map occupiable.

`Local Map Synthesizer`: The computation complexity of MARS and Navigability is cubic (from (1) and (2)) in the size of the map, and hence do not scale well in large environments. In order to keep the problem tractable, the global map is partitioned into smaller *augmented local maps*. A discretized heightmap of the agent's local environment is derived from the mapping system, and the dimensions of this local map depends on the agent's sensing capabilities and computational resources. The occupiable cell in the agent's global trajectory closest to the boundary of the local map is termed as the *local origin*. The intersection of the boundary of the local map with the line joining the agent's current position and the target location is determined as the *local goal*. The augmented local map consists of the local heightmap, MARS gap map, navigability map, local goal and local origin. The Local Map Synthesis provides consistency for the local construction planner in the:

- Global deposition bounds: The local MARS bounds computed on partial information is consistent with the global MARS bounds [23, §3].
- Target location: The local goal drives the building agent towards the global target location.
- Return path: The local origin maintains a consistent return path to the agent's global origin, as the navigable region may shrink during the construction process.

This approach allows for breaking down the global construction planning problem in large environments into smaller local planning problems. By definition, the augmented local map is agnostic to the larger global terrain. A building agent initialized with such a map and with no access to the history of the construction process (executed by itself or other agents) will still make predictable modifications to its environment purely based on its location in the local environment state.

`Navigable Path Planner`: The module composes a plan to build a navigable path given the augmented local map. The Navigable Path Planner (NPP) chooses a deposition path to reach the local goal and computes the required *deposition target volume*. As the building agent deposits material in the environment, the navigable region may expand and shrink; the planner guarantees the agent has at least one navigable position (local origin) in the local map after deposition. NPP is designed as an informed graph search algorithm with the following evaluation function for a node's cost:

$$f(\mathbf{n}) = g(\mathbf{n}) + \alpha.c(\mathbf{n}) + \beta.l(\mathbf{n}) \quad (4)$$

Here, $g(\mathbf{n})$ is the total movement cost from the start node to node $\mathbf{n}$, $l(\mathbf{n})$ is the heuristic estimate of the cheapest cost

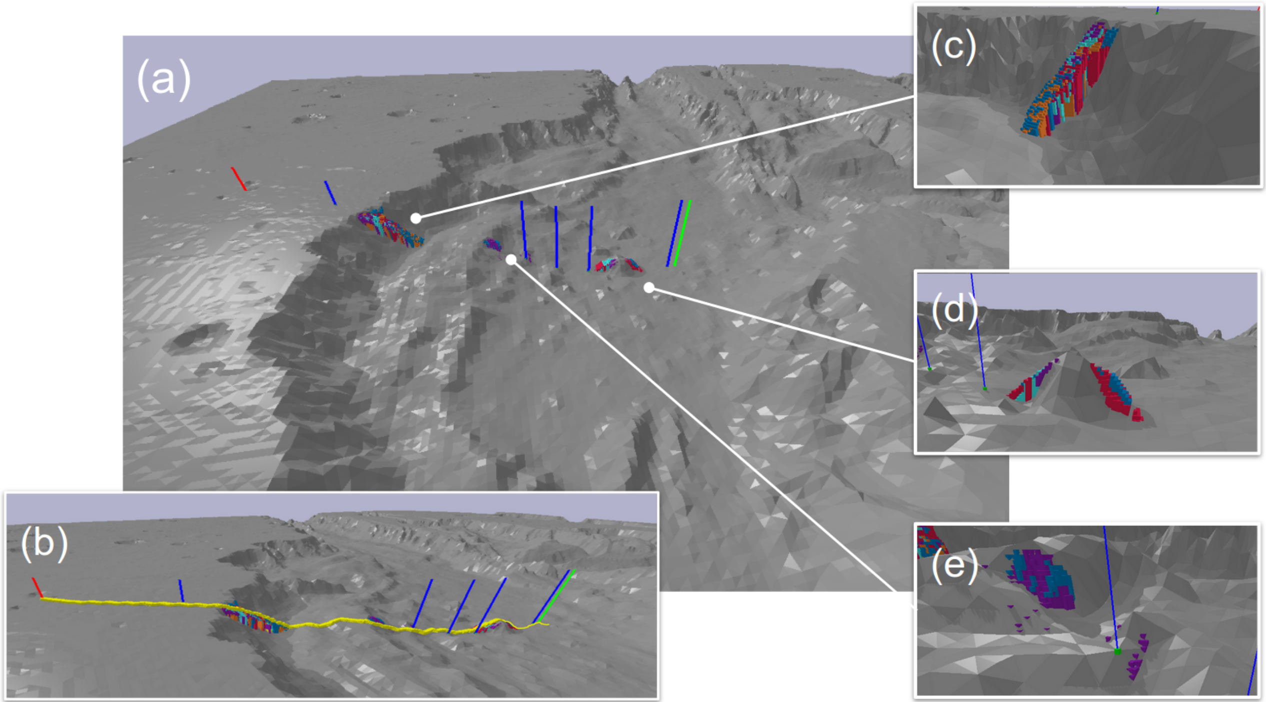

Fig. 3. Five robots are tasked to collectively construct a navigable path for a larger mobile payload system from the start location (red line) to the target location (green line). The blue lines indicate the positions of the robots. The yellow path in (b) is the final constructed navigable path. (c), (d) and (e) show zoomed in views of the three major construction sites along the path. The initial terrain is gray and each color of deposited material corresponds to a specific robot.

from node $\mathbf{n}$ to the goal node, and $c(\mathbf{n})$ is the construction cost from the start node to node $\mathbf{n}$ given by:

$$c(\mathbf{n}) = c(\mathbf{m}) + \sum_{\mathbf{i} \in B_{\delta/2}(\mathbf{n})} \Delta P_\kappa[h](\mathbf{i})|_{Q_r}$$
$$\textit{where } \mathbf{m} = \arg\min_{\mathbf{j} \in N(\mathbf{n})} f(\mathbf{j}) \tag{5}$$

α and β are scale parameters of the cost function, and $N(\mathbf{n})$ is the set of neighboring nodes of node $\mathbf{n}$.

`Goal Synthesizer`: The goal synthesizer is responsible to effectively guide the building agent to solve a high level construction problem by compiling target locations and motion parameters based on the agent's state. This is demonstrated in Section V.

IV. Simulation Framework

MARSala DOSA (Minimal Additive Ramp Structure à la Distributed Operation of Simulated Agents) is a simulation framework that implements the proposed architecture. The framework is written in C++ and Python, and interfaced with the ROS ecosystem. The integration system is implemented in C++ and takes advantage of multi-threading and SIMD instruction sets. A grid map implementation based on an octree (Octomap) [27] is used to efficiently represent the robot's environment and probabilistically incorporate sensor measurements. The mapping system is built using the GridMap [28], Octomap and Eigen [29] libraries. The Navigable Path Planner is implemented in both C++ and python. A web-based user interface is designed using Jupyter and Tmux. The simulator has various robot modules implemented, including a 3D laser scanner, RGBD sensor, ground truth sensor, robot proximity sensor, contour crafting depositor, and a cone-based depositor. The physics simulation engine is provided by Bullet through the Pybullet [30] python bindings. It supports rigid and deformable objects, and allows for GPU acceleration. The simulator framework abstracts the low level primitives in the controller system, however, they may be implemented based on the user's requirements. The simulator framework is packaged with a terrain generation tool that provides different methods to generate random terrains for evaluation purposes. The source code for the simulator framework is available on GitHub[1].

For the demonstrations presented in this work, we used the simulator framework with the following specifications regarding the building agent and building material.

A. Building Agent

A building agent is considered to be a self-contained robot system capable of motion, perception, estimation, deposition and material replenishment. The robot system is designed as a mobile manipulator with on-board sensors; an RGBD sensor to build dense maps and a proximity sensor to sense other robots. Without loss of generality, we have abstracted certain components of the robot system, particularly the low level motion planners for movement and deposition as they depend on the type of the mobile robot system and the

[1] https://github.com/napp-lab/marsaladosa

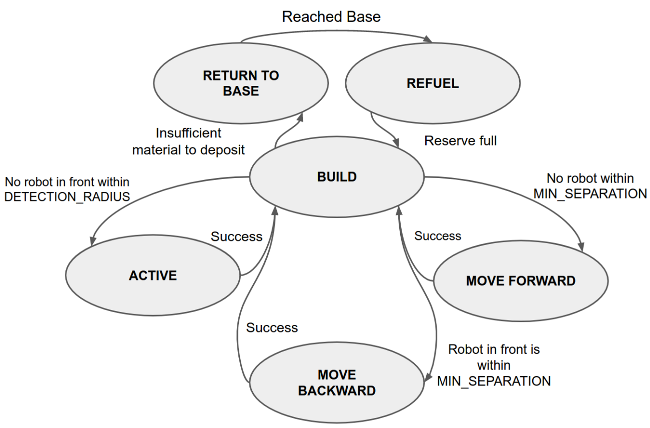

Fig. 4. State diagram for a decentralized cooperative construction of a navigable path.

deposition material used. However, the high-level motion and deposition planners respect the kinematics and physical constraints of the robot system. Since the efficacy of MARS has been showcased using multiple materials and robots in the presence of noise in the system and the environment [24], [23] our abstraction does not violate the real world efficacy of the architecture. The last few decades of research in SLAM has led to successful applications in many areas of robotics. However, SLAM in large, unknown and complex environments is still a challenging problem and an active area of research [31], [32]. For the experiments described in this work, we abstract the SLAM problem by incorporating the sensor measurements at a well-localized robot pose. Octomap is used to represent the probability of occupancy of the voxels in 3D space. The 3D grid map is converted into the relevant height maps for the integration and planning systems.

B. Building Material

Since the material type and deposition planning are highly correlated, we generalize the deposition mechanism by considering a material extrusion system on the end of the manipulator, such that the smallest, indivisible deposition that can be made is a cube of a predefined size. The use of the MARS model for construction tasks have been demonstrated using single or a combination of building materials of different physical properties, such as compliant bags, rectangular building blocks of varying size and irregular stones [24], [23]. Our focus in this work is to showcase different large scale scenarios and evaluate the benefits of a decentralized coordination mechanism between building agents.

Figure 2 demonstrates simulated building agents using the MARSala architecture, where the robots are tasked to climb a platform at a height of $1.5\mathrm{m}$. Figure 2 (a) showcases the simple, regular terrain before any construction activity; the blue shaded area depicts the local map (perception field) synthesized from the executive layer and the green block depicts the projection of the center of mass of the robot onto the ground. The blue vertical line helps to locate the robot in large environments. Figure 2 (b-d) depict snapshots along the progress of a simple ramp construction. The motion parameters of the constructed structure conforms to the motion parameters of the Husky robot [33]. In Figure 2 (e), the robot has successfully used the built structure to reach its target destination.

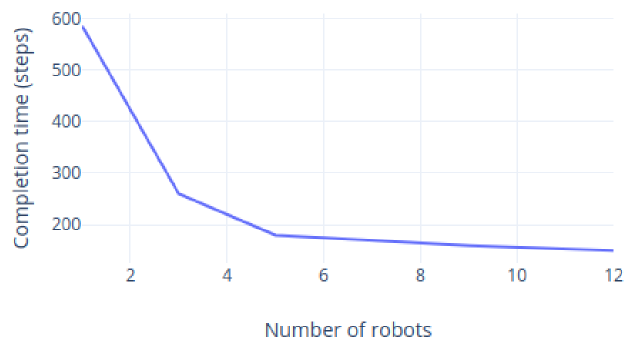

Fig. 5. Scalability curve for a decentralized cooperative construction of a navigable path. The completion time reduces exponentially with the number of robots and saturates at a team size of 9 robots.

Figure 2 (f) shows the final structures for three different slope parameters (parameter κ in (2)) used for the MARS processor. The final structures do not have smooth surfaces; this is a direct result of the navigability property which models small discontinuities that the robot can climb over (parameter ϵ in (1)).

V. Applications

A. Scenario: Cooperative Construction of a Navigable Path

Problem Overview: A large, mobile payload system with conservative motion parameters is to be transported to an inaccessible target location in a large unstructured environment. A team of agile builder robots capable of depositing material in their environment are deployed to build a navigable path to the target location for the mobile payload.

Implementation: The builder robots are implemented in the simulation framework based on the description in Section III. Each builder robot is equipped with an additional visual sensor that can detect the presence of its neighbors within a distance. Builder robots are spawned at a base station location. They have a limited material reserve and will periodically return to the base station for replenishment. The width of the path built is such that it accommodates the larger mobile payload and is wide enough to support two lanes of motion for the builder robots; a forward lane for motion from the base station to a construction site and a backward lane to move back to the base station for material replenishment. The path also conforms to the more conservative motion parameters of the mobile payload system. Each robot is aware of the target location and uses the NPP module on the augmented local map to get the target deposition volume. An abstract motion and deposition manager that respects the kinematic constraints of the robot is employed to move and deposit material in the environment.

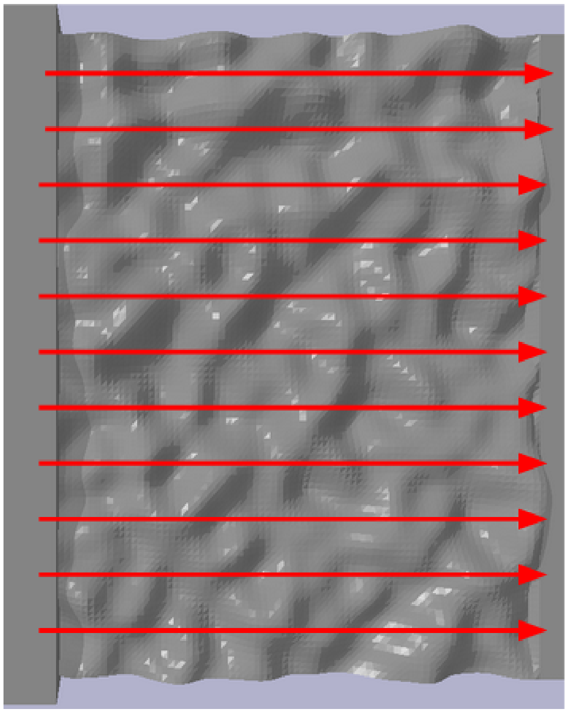

Fig. 6. A vector map depicts the target locations formulated by the goal synthesizer for the terrain levelling problem. The pointed end of an arrow depicts the target location of a robot positioned at its rear end.

The robots form a *construction pipeline* from the base station and the robot at the front of the pipeline is termed the *active* robot, which is allowed to make depositions. Other robots are not allowed to make depositions as they may trap the robots in front of them. Once an active robot does not have enough material to complete a deposition task, it turns around and retraces its trajectory along the construction path in the opposite lane. The robot next in the pipeline is now termed as the *active* robot. The robots maintain a minimum separation distance between themselves based on their front neighbor. When a robot reaches the base station, it replenishes its material reserve and joins the end of the construction pipeline in the forward lane. As such, a construction pipeline across many robots is maintained merely through local interactions and the decentralized MARSala architecture since the depositions depend only on the environment state and the target location.

Since the simulator is designed as a discrete time system and abstracts the low level motion primitives for deposition and movement, we propose an evaluation metric for this scenario. In each time step, a robot's motion and deposition are limited by a *distance threshold* and *deposition threshold*, respectively. The robots are deployed on a topographic model[2] (Fig. 3) of a portion of *Valles Marineris* on the planet Mars with vertical exaggeration, derived from data collected by the Mars Orbiter Laser Altimeter (MOLA) onboard NASA's Mars Global Surveyor mission.

[2]https://github.com/nasa/NASA-3D-Resources/tree/master/3D%20Models/Valles%20Marineris

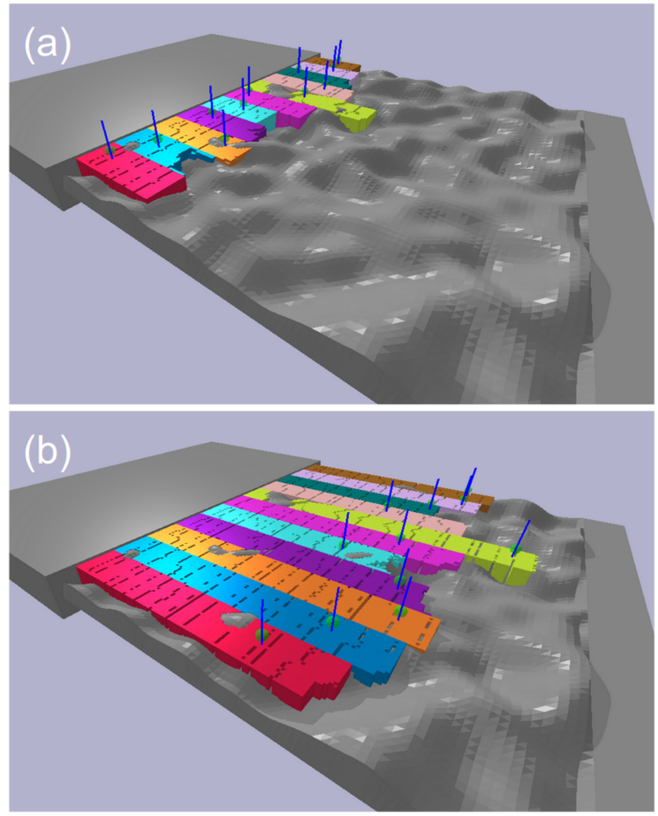

Fig. 7. Terrain levelling problem. Eleven robots, each depositing material with a different color, are tasked to level a terrain.

The terrain in simulation spans an approximate size of $(160 \times 160 \times 14\mathrm{m}^3)$. The robots are deployed at the location highlighted by the red line $(-26.725\mathrm{m}, 2.50\mathrm{m})$ and the target location is highlighted by the green line $(-5.225\mathrm{m}, -5.225\mathrm{m})$. Experiment runs are executed by a team of building agents with sizes 1, 3, 5, 9, and 12. Fig. 3 shows the completed construction path for the experiment run with 5 robots, where each deposit material of a different color on the initial gray terrain. The average total deposition volume across all runs was $(580.4 \pm 16.8\mathrm{m}^3)$. The final construction path spans a length of $21.3\mathrm{m}$ across all runs. The three major construction sites where most of the material was deposited by the robots are shown in Fig. 3 (c-e) at distances $7.3\mathrm{m}$ $15.88\mathrm{m}$ and $19.20\mathrm{m}$ from the base station, respectively.

Fig. 5 showcases the scalability curve for the scenario. The completion time reduces exponentially with the number of robots, however, it saturates at a team size of 9 robots. As the number of robots increases, the total duration where no robots deposit material (idle time) decreases. For smaller team sizes, there is often no active construction taking place while the robots travel back to the base station for material replenishment. As the number of robots increases, the construction pipeline steadily releases robots into the construction site until it reaches full capacity. The curve saturates as the number of robots increases due to the physical limitation in the construction pipeline approach. The saturation point depends on the minimum separation distance between the robots, the speed of motion primitives, and the length of the construction pipeline which is subsequently influenced by the exact path taken.

By employing a simple state machine (depicted in Fig. 4) for local interactions along with the MARSala architecture, the robots coordinated intrinsically using the environment state and the target location, regardless of the number of robots. This scenario showcases an approach to cooperatively build a navigable path in a completely decentralized manner using a simple communication principle and a static target location.

B. Scenario: Site Preparation

Problem Overview: A team of robots is tasked to level an uneven terrain as part of a site preparation project.

Implementation: The goal synthesizer compiles goals based on the location of the robot and instructs them to use a small slope parameter ($\kappa = 0.1$) throughout the mission. The simulation was run with 11 robots to cover the entire area as shown in Fig. 7. The robots were deployed at the larger flat area on one end of the terrain. The goal vector map shown in Fig. 6 represents the target location of a robot based on its location. The pointed end of a red arrow represents the target location for a robot located at its read end. When a robot is deployed, it first finds the closest rear end of the available goal vectors, and positions itself to it. It then begins planning a construction path to the target location given by the pointed end of the arrow. The robots were able to successfully level the terrain as shown in Fig. 7.

If the deposition material is modelled to be loose gravel or sand, a Discrete Element Method (DEM) based simulation may be used to determine the ground reaction when a robot moves over it. In such a scenario, if the robots are allowed to traverse the region multiple times, they would repair and maintain the flat terrain since the reactive approach in the MARSala architecture responds to any changes in the environment.

VI. Conclusion and Future Work

In this work, we proposed a decentralized multi-robot construction architecture for building motion support structures in large unstructured environments. The deliberative layer of the architecture is modelled such that it provides a means for translating different constructions projects into a motion support structure problem. This was demonstrated using two construction scenarios in a simulated environment, one where robots were tasked to collectively build a navigable structure to a target location and the other where robots were tasked to level an uneven terrain. We also developed a modular multi-robot simulation framework for ground-based mobile building agents that allows us to design and analyze large scale construction problems. In future works, we aim to develop and study automatic goal compilers for abstract plans, similar to the one presented in [34], which should convert a goal structure into a set of target locations that a team of distributed robots can follow to perform decentralized construction. This framework can also be used to validate and inspire the development of new planning and scheduling techniques to handle situations in which robots are tasked maintain a structure in predictable ways.

Acnowledgements

This work is supported by the National Science Foundation under Grant NSF#184634. We would also like to thank Dr. Maira Saboia for her helpful feedback.

References

[1] N. Melenbrink, J. Werfel, and A. Menges, "On-site autonomous construction robots: Towards unsupervised building," *Automation in Construction*, vol. 119, p. 103312, 2020.

[2] L. Vasey, B. Felbrich, M. Prado, B. Tahanzadeh, and A. Menges, "Physically distributed multi-robot coordination and collaboration in construction," *Construction Robotics*, vol. 4, no. 1, pp. 3–18, 2020.

[3] K. H. Petersen, N. Napp, R. Stuart-Smith, D. Rus, and M. Kovac, "A review of collective robotic construction," *Science Robotics*, vol. 4, no. 28, p. eaau8479, 2019.

[4] H. Ardiny, S. Witwicki, and F. Mondada, "Are autonomous mobile robots able to take over construction? a review," 2015.

[5] A. Hsu, A. Wong-Foy, B. McCoy, C. Cowan, J. Marlow, B. Chavez, T. Kobayashi, D. Shockey, and R. Pelrine, "Application of micro-robots for building carbon fiber trusses," in *2016 international conference on manipulation, automation and robotics at small scales (MARSS)*. IEEE, 2016, pp. 1–6.

[6] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, 2014.

[7] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D'Andrea, "Building tensile structures with flying machines," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3487–3492.

[8] K. H. Petersen, R. Nagpal, and J. K. Werfel, "Termes: An autonomous robotic system for three-dimensional collective construction," *Robotics: science and systems VII*, 2011.

[9] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[10] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial life*, vol. 5, no. 2, pp. 97–116, 1999.

[11] M. Allwright, N. Bhalla, C. Pinciroli, and M. Dorigo, "Towards autonomous construction using stigmergic blocks," *Université Libre de Bruxelles*, 2017.

[12] T. Soleymani, V. Trianni, M. Bonani, F. Mondada, and M. Dorigo, "Bio-inspired construction with mobile robots and compliant pockets," *Robotics and Autonomous Systems*, vol. 74, pp. 340–350, 2015.

[13] J. Wawerla, G. S. Sukhatme, and M. J. Mataric, "Collective construction with multiple robots," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 3. IEEE, 2002, pp. 2696–2701.

[14] J. Werfel, "Building blocks for multi-robot construction," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 285–294.

[15] N. R. Franks, A. Wilby, B. W. Silvermant, and C. Tofts, "Self-organizing nest construction in ants : sophisticated building by blind bulldozing," Tech. Rep., 1992.

[16] J. F. Genise, "Blueprints of termite and ant nests," in *Ichnoentomology*. Springer, 2017, pp. 247–284.

[17] L. Guo, G. Theraulaz, C. Jost, and A. Perna, "Self-organized linear and helicoidal ramps in insect nests," in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*. IEEE, 2016, pp. 148–153.

[18] G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg, "The origin of nest complexity in social insects," *Complexity*, vol. 3, no. 6, pp. 15–25, 1998.

[19] M. Roman, M. Yashar, M. Fiske, S. Nazarian, A. Adams, P. Boyd, M. Bentley, and J. Ballard, "3d-printing lunar and martian habitats and the potential applications for additive construction." 2020 International Conference on Environmental Systems, 2020.

[20] R. P. Mueller, M. C. Roman, and H. S. Kim, "Nasa centennial challenge: Three dimensional (3d) printed habitat, phase 2," 2017.

[21] A. Paolini, S. Kollmannsberger, and E. Rank, "Additive manufacturing in construction: A review on processes, applications, and digital planning methods," *Additive manufacturing*, vol. 30, p. 100894, 2019.

[22] J. Kramer and M. Scheutz, "Development environments for autonomous mobile robots: A survey," *Autonomous Robots*, vol. 22, no. 2, pp. 101–132, 2007.

[23] M. Saboia, V. Thangavelu, and N. Napp, "Autonomous multi-material construction with a heterogeneous robot team," *Robotics and Autonomous Systems*, 2019.

[24] V. Thangavelu, M. S. da Silva, J. Choi, and N. Napp, "Autonomous modification of unstructured environments with found material," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7798–7804.

[25] M. Saboia, V. Thangavelu, W. Gosrich, and N. Napp, "Autonomous adaptive modification of unstructured environments," *Proc. Robotics: Science & Systems XIV (RSS 2018)*, 2018.

[26] E. Gat, R. P. Bonnasso, R. Murphy *et al.*, "On three-layer architectures," *Artificial intelligence and mobile robots*, vol. 195, p. 210, 1998.

[27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[28] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5. [Online]. Available: http://www.springer.com/de/book/9783319260525

[29] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[30] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[31] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood *et al.*, "Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 80–86.

[32] Z. Hong, Y. Petillot, and S. Wang, "Radarslam: Radar based large-scale slam in all weathers," *arXiv preprint arXiv:2005.02198*, 2020.

[33] "Husky unmanned ground vehicle," https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/, accessed: 2021-05-18.

[34] Y. Deng, Y. Hua, N. Napp, and K. Petersen, "Scalable compiler for the termes distributed assembly system," in *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 125–138.