IAC-16.D3.1.10x35852

PRELIMINARY FINDINGS FROM A MULTI-ROBOT SYSTEM FOR LARGE-SCALE EXTRA-PLANETARY ADDITIVE CONSTRUCTION

Samuel Wilkinson[a]* Josef Musil[a]*, Jan Dierckx[a], Richard Maddock[a], Xiaoming Yang[a], Miriam Dall'Igna[a], Octavian Gheorghiu[a], Xavier De Kestelier[a]

[a] Foster+Partners, Specialist Modelling Group, Riverside, 22 Hester Road, London SW114AN, UK
* Corresponding authors: swilkinson@fosterandpartners.com, jmusil@fosterandpartners.com

We present our findings from a 4-day workshop at SmartGeometry2016, during which the authors conducted an open-ended experiment to ascertain the viability of a multi-robotic system capable of large-scale additive construction with sand. The study and results pertain to future robotic systems operating in extreme environments, such as in lunar or Martian conditions, where there is a need for autonomous construction of regolith structures for infrastructure or human habitat protection. The purposes of this study were to: i) implement and document the practical knowledge of multi- or swarm-robot systems in physically realistic environments; ii) ascertain the feasibility of additive construction with many-simple rather than few-complex robots; and iii) explore individual behavioural rules for the robots which, although indirectly controlled themselves, can result in a controlled outcome. Behaviour is understood as the interactions between an individual and its environment where the behaviour of the individual affects its own perceptions, and thus its future actions and perceptions. Applying this concept to robots results in a field of autonomous behavioural machines capable of operating in partially unknown and changing environments without human intervention. In the paper we describe the design and mechanical operation of the multi-robot system, the local positioning and communication system, ending with a discussion of the programmed behaviours and the final workshop outcomes. The results of this workshop will be used to inform the next stage of the technology demonstration in which the process will be scaled up.

I. ADDITIVE REGOLITH CONSTRUCTION

As designs for human expeditions to Mars evolve over the next few decades, it is necessary to consider what type of a home can be provided for people on the surface of the planet. There is an opportunity of going beyond a purely functional scientific base, as has been the norm [8, 11], to build a place for sustained living, suitable for continued multi-mission occupation and development. With each successive expedition, feedback will be incorporated from the last into an evolving design and so any future colony is likely to be a mixed system of technologies and habitats of increasingly advanced design.

Surface habitats can be one of three classes: Class I are single pre-fabricated purely Earth-made pressurised modules, such as the Apollo Lunar lander; Class II are multiple such modules connected together; and Class III combine multiple modules plus structures built from in-situ materials [13]. Extending this taxonomy, Class IV would be constructed solely from local materials using imported machinery, and Class V the construction machines would themselves be fabricated in-situ. This last class marks a strong point of complete Earth-independence and may be someway off in the future

Here, the proposed Class III outpost is a composite of light-weight inflatable pressurised modules [14] and a multi-robot additive construction system to compile local regolith into a protective heavy-weight shield. Although the pressurised elements are better suited to high accuracy fabrication possible under tightly controlled conditions on Earth, the compressive heavy-weight protective shield can be more heterogeneous. Therefore delivery of construction robots is lighter, more compact and flexible than sending an equivalent item from Earth. Others have also considered additive regolith construction processes for extra-planetary missions [6, 13, 15], although their focus has been on centralised robotic systems.

The Smartgeometry [17] 2016 workshop originated from the design concept for a habitat on Mars [23, 24] (Figure 1) in which inflatable modules are covered by a protective regolith shield. The primary component of the concept design that we consider is security, in terms of both crew safety, peace of mind, and mission success in relation to the environmental conditions. Safety is arguably the most fundamental and primitive aspect of a home. Shelter from wild animals, storms, enemies, and extreme temperatures has for thousands of years defined our homes. Protection from environmental conditions on Mars leads us to consider the necessity of a heavy-weight regolith shield. A shield built from surface regolith has many advantages: protection from cosmic radiation, solar flares, dust storms, meteorites, and temperature variations; as well as increasing the durability of the light-weight habitat modules. Additionally, protective structures can be built, such as berms, and infrastructure like landing pads and roads.

Figure 1: Operational Mars habitat [23, 24].

Figure 3: Regolith excavation, transportation, and in-situ melting [23, 24].

An autonomous and generative additive construction method relies on rules and objectives, meaning that the outcome is more adaptive and open-ended. The regolith additive construction (RAC) approach is adapted for the low accuracy likely to be achieved from using variable materials at an uncertain site with autonomous robots in the field. The reliability of the AC is in its simplicity, implemented by the three classes of robot: the strategy is to 'dig, move, and melt' regolith.

Firstly, the site is prepared by excavating a shallow pit for the modules to rest within. Once the three habitat modules are positioned, each inflates and connects together (Figure 2). The large digging robots will extract loose regolith in close proximity for the medium-sized mover robots to transport to the habitat. This is the element of the construction process that is investigated further in the following workshop description.

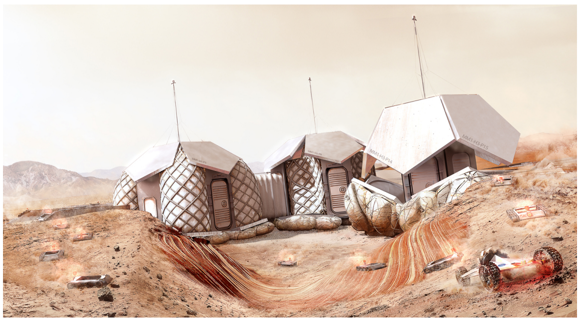

Figure 2: Preliminary site preparation, module deployment, inflation, and connection [23, 24].

The regolith is positioned into rough layers by the transporter robots, with the thickness continuously measured. Once a thin layer of regolith is in place, the third class of smallest robots selectively melts patches into a hard crystalline material [2] (Figure 3 and 4).

To protect the crew from radiation over long-term periods, rather than transporting heavy shielding from Earth, the construction of a regolith shield is a logical alternative [16]. The largest reduction in dose equivalent (rem) occurs in the first $20g/cm^2$, so assuming a regolith density of $1.5g/cm^3$ the regolith depth should be at least 15cm [19]. Whilst this is a minimum depth to ensure the crew does not receive 'career limiting' doses, regolith is compiled to a depth of 1.5m above the work/sleep modules for general protection from cosmic radiation, and 2.5m above the communal space for protection from solar flares during periods of increased solar activity.

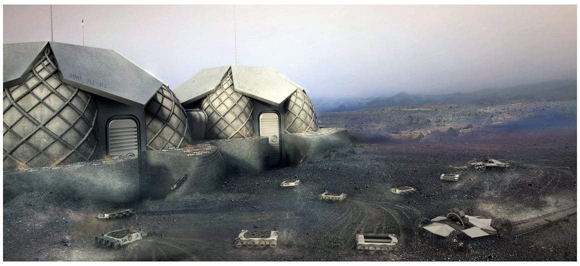

Figure 4: Digging and transportation of regolith [23, 24].

The form of the regolith shield is driven by two key criteria, which become the operational rules for the individual robots. The first criteria is the minimal thickness of regolith needed to protect the inhabitants from radiation. The second criteria is the ability of all construction robots to transfer themselves to the highest layer printed so far during the construction process. As a result, multiple ramp structures blended into the overall form are introduced next to every opening of each module (airlocks, windows and suit-ports). Because of their location, they also serve as an extra protection of these openings. The construction sequence is shown in Figure 5.

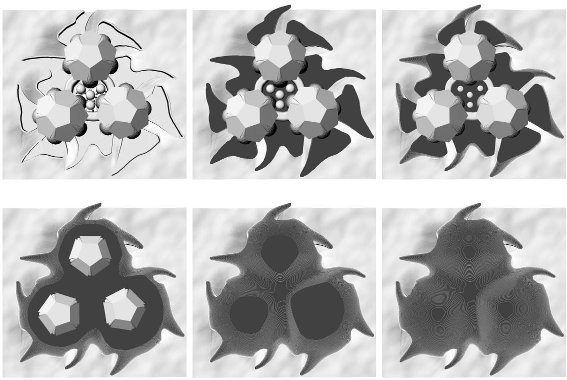

Figure 5: Construction sequence [23, 24]

In order to robotically construct a protective regolith shield in such an environment, a central element of the proposal was the idea that many simple robots would be more robust and therefore more likely to succeed than a single complex robot. The focus of the workshop was on developing the autonomous robotic construction system.

II. SWARM ROBOTICS

The proposed system is comprised of three classes of robot, each specialising in either excavation, transportation, or melting. The traditional approach to mission robotics is to consolidate risk and complexity into a single fate-sharing machine, with built in redundancy (e.g. over-engineering and system duplication). However, distributing risk across multiple specialised, simpler robots has the advantage of isolating risk to individual units. Given enough robots, there is also the potential for emergent behaviour, i.e. group behaviours greater than the sum of the individual behaviours [22].

Due to communications delays of up to 40 minutes from Earth, the robot control must be autonomous, with each individual capable of decision-making and following rules. In the near future, it is predicted that computational intelligence and robotic technology will be sufficiently advanced to allow for a distributed system of autonomous intelligent machines. The system is capable of adapting to uncertain operating environments (autonomy), of self-management and system awareness (autonomicity), and of following high-level commands such as 'explore', 'gather materials' or 'construct habitat'.

A number of questions arise from this design exploration around the operation of the robotic construction system. How many individual units are required? What is their basic functionality? How would they operate together in a remote harsh environment?

There is a broad range of on-going research topics that explore algorithms to control robotic swarm systems, such as self-assembly [1, 20, 18], collective construction [10, 9], and exploration [12, 5]. For comprehensive reviews of the state-of-the-art in robotic swarm engineering see [4, 21].

II.I From Concept to Product

In the original concept proposal, three different classes of robots were envisaged: a large digging robot; a medium transporter and depositor; and a small melting robot.

This concept was simplified into focusing on the medium class of robot for the workshop. This decision was influenced by several factors: i) the design of one robot class is less complex than the design of three robot types; ii) condensing to one class meant that more behavioural studies were possible with a limited number of robots; and iii) the robot design could be more easily replicated. This approach was reinforced by the idea that the robot had to be able to run continuously for a whole day, which required a significant battery pack. This would not have been feasible for the smaller robots.

Flying drones were dismissed in the concept since the atmospheric density of the Martian atmosphere is 0.6% of Earth's mean sea level pressure. This means, although objects on Mars are lighter (since Mars has 0.38g), the drone would require a lot of energy to achieve lift. Drones are in general also more vulnerable than land bound machines, and flying payload is usually more energy intensive than transporting over ground. This said, prospects for flying drones on Mars are more suited to light-weight science missions.

For the workshop, it was decided that melting or binding of the loose sand was not feasible and that the objectives of the experiment could be achieved without this additional process, thus excluding the smallest class of robot. Binding sand, or regolith, without any adhesive requires considerable energy for either melting or sintering; this was considered too costly, complex, and dangerous to investigate here.

II.II Design Requirements

The final robot design used during the workshop evolved over a number of weeks in a small test sandbox. The original concept was to rely as much as possible on available technology and components, and enrich them with custom components where necessary.

The robot had three requirements: i) to pick up, transport and deposit material in an organised fashion; ii) be powered by a rechargeable energy source with enough charge for an entire working day; and iii) can communicate wirelessly and receive behavioural updates on the fly.

Different frameworks were looked at for the electrical and mechanical components of the robot, including Lego Mindstorms, and resulted in the choice of the Makeblock platform because it fulfilled all the requirements: i) a large number of highly robust mechanical components which can easily be assembled; ii) a wide range of electrical components including motors, servos, drivers, etc.; iii) built-in compatibility with Arduino and Raspberry Pi; and iv) can run on 5V power supply. This platform was combined with juice packs which are designed to recharge small electronic devices and had sufficient energy (about 28,000 mAh) to power the robot the entire day. The custom components which executed the specific functions of digging, transporting and depositing were 3D printed using Selective Laser Sintering (SLS).

III. ROBOT DESIGN DEVELOPMENT

Robot design development went on for a few weeks prior to the workshop with numerous iterations. Here we highlight four key iterations and describe the final design.

III.I Iteration 1: Auger Deposition

It was established that controlled deposition of material was the hardest of the three requirements, and was therefore addressed first. The first iteration (Figure 6) focussed on this requirement and was a deposition device consisting of an auger and nozzle. The material is simply stored in a bin above the auger.

This iteration also included rubberised tracks to move around on loose sand. This decision reduced the required number of motors from four to two compared with not using tracks. It needs to be noted that for an actual space-ready prototype it would be preferable to avoid tracks as they are a complex element which can break or jam easily.

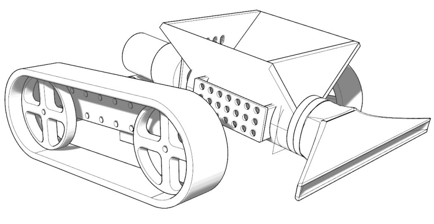

Figure 6: Robot development iteration 1.

The rubberised tracks functioned adequately for moving around on loose sand, even when running the motors at a lower voltage. The auger proved only successful without the nozzle but worked less consistently with a wetter sand mix. This option was dismissed because this deposited the sand in a narrow line parallel with the robot path, rather than a wide band following the robot. This iteration also made it clear that a sufficient clearance from the ground is required for a robot to be able to navigate hilly terrain. Therefore, the chassis of the robot had to be mounted as high as possible.

III.II Iteration 2: Triangular Tracks

One of the concerns of a robotic rover on a hilly terrain is that it might flip. The second iteration addressed this by use of a triangular configuration of the tracks (Figure 7), which allowed the robot to flip over and continue operation.

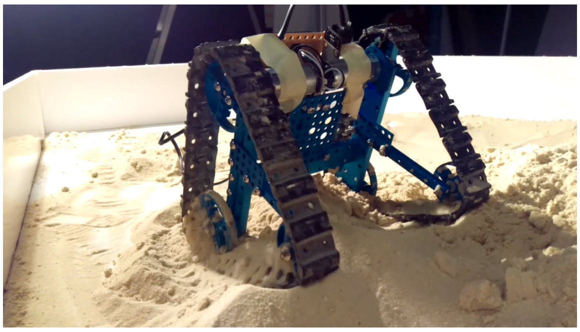

Figure 7: Robot development iteration 2.

Although this configuration allowed the robot to continue operating when flipped, it proved to be more unstable as the higher centre of gravity made it flip more often. The internal triangular volume available for hardware was less efficient and it did not provide an optimal positioning of the motors. Therefore this option was dismissed and the results integrated in the next iteration, using a non-flippable design with a low centre of gravity. This meant mounting the battery as close to the ground as possible, leaving sufficient clearance to avoid getting beached on rough ground.

III.III Iteration 3: Excavation and Deposition

Reverting to the simpler track configuration of the first design, the back depositor was redesigned and an angled excavator bucket was added (Figure 8). The deposition auger was rotated to be perpendicular with the movement of the robot and transformed into a flat-bladed rotor to distribute sand in a band. The inspiration came from inverting the function of a harvester. This iteration also included a scoop to dig up the sand and transfer it into the excavator bucket. Gravity would feed it into the rotor which could then deposit it. The speed of the robot relative to the robot speed will define the amount of sand deposited.

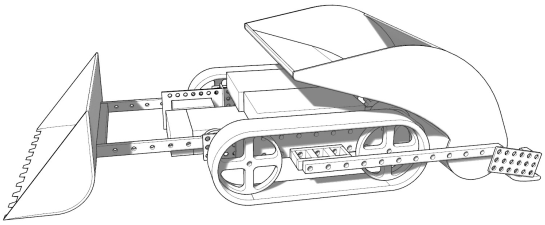

Figure 8: Robot development iteration 3.

The scoop proved very successful but was positioned too low to be efficient. It was best for the scoop to be pointing slightly down when excavating so it can really ‘dig’ itself into the sand with the weight of the robot behind it. The excavator bucket also proved effective to store the sand, but the amount was too small for gravity to push the sand to the rotor. The rotor worked perfectly and enabled a precise deposition of sand.

III.IV Iteration 4: Excavation, Transfer, and Deposition

In addition to the excavator bucket and the depositor, a middle transfer bucket was required (Figure 9). This meant that the excavator bucket arm could be shortened, as the servo motor struggled with a fully-loaded bucket on a long arm.

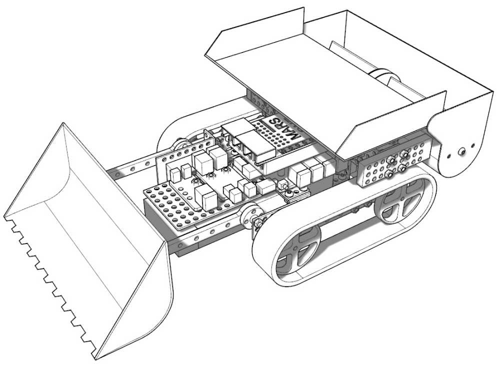

Figure 9: Robot development iteration 4.

III.V Final Design

The final iteration was similar to the last except with a number of small improvements (Figure 10). The bucket size was increased, the internal hardware was reconfigured and rationalised to make better use of space, the size of the middle transfer bucket was increased, and a protective shield was wrapped around the sensitive electronics to stop sand from collecting.

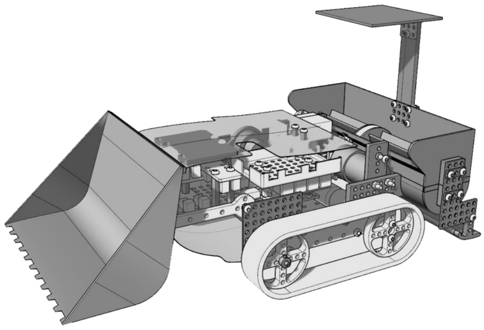

Figure 10: Schematic of individual robot parts.

The final design consisted of a mixture of basic Makeblock components and SLS 3d printed custom parts (Figures 11 and 12).

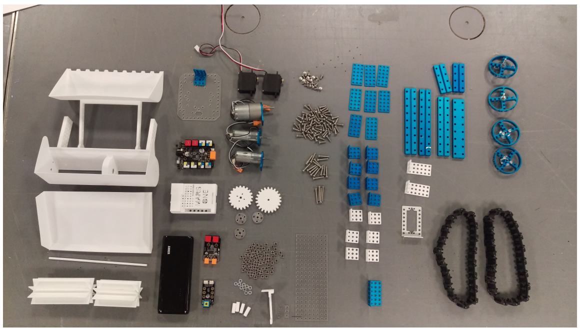

Figure 11: Individual robot parts.

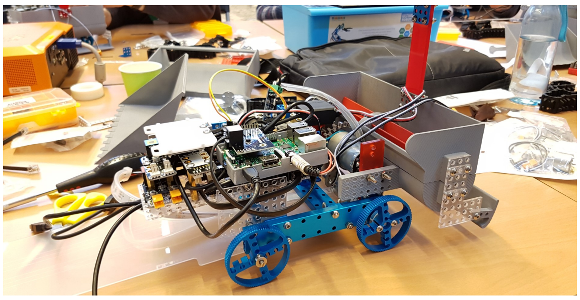

Figure 12: Final robot design built during workshop.

For the workshop participants constructed their own robots and a logical construction sequence was documented (Figure 13).

IV. WORKSHOP: COMMUNICATION

The overall system and setup of communication developed for the workshop is logically split into communication within each robot itself (internal, Figure 15) and communication between each robot and its environment (external, Figure 14). This is also aligned

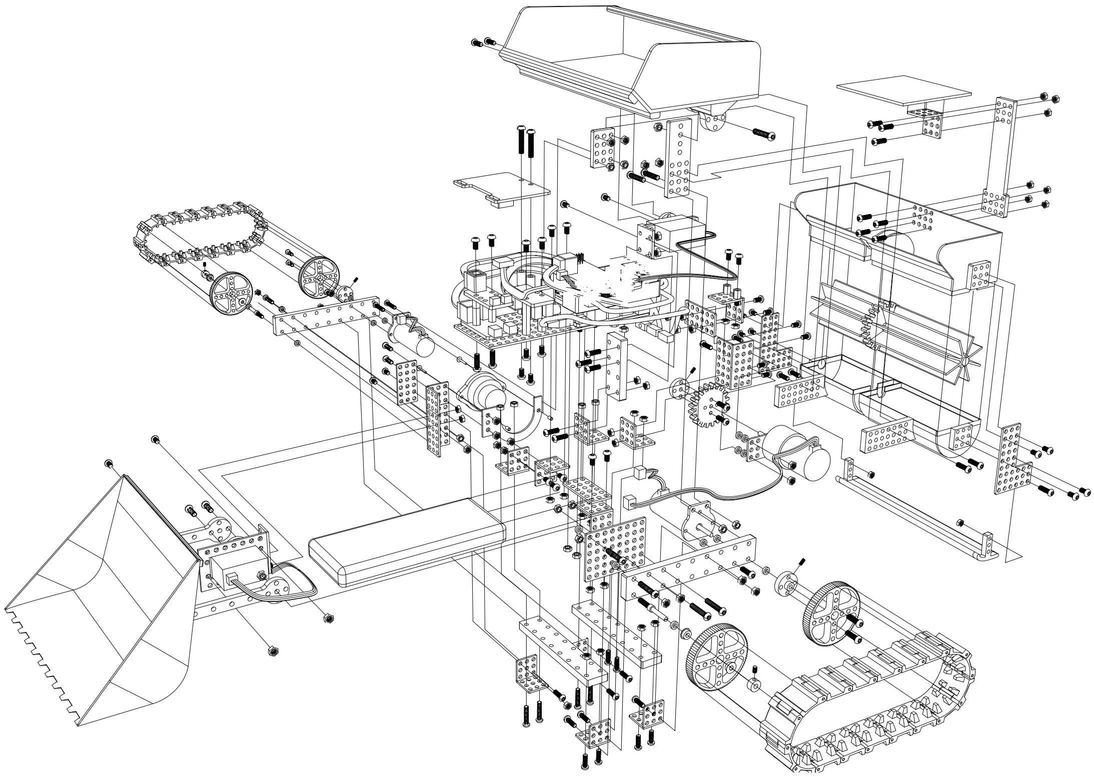

Figure 13: Exploded diagram of final robot construction sequence.

with the concept of agency, where each agent is following its own decision based on its local environment. For practical and cost reasons we replace local environmental sensing with a global sensor. Our software processes and splits this data into separate data unique for each robot. In the last subsection, we talk further about constraints and advantages of communication in a system that is low-tech and with low precision of hardware sensing and actuation.

IV.I External Communication with Environment

All communication, including robots with remote computers and robots with global sensors as well as in-between global sensors, runs over a locally setup Wi-Fi network. Each robot or device is identified by its unique IP address. The main format for data exchange is a JSON file shared over an HTTP server. The JSON format is a human readable text file and can be open and read in a web-browser on any device on the same Wi-Fi network. This significantly improved our debugging abilities as we had real time access to all data exchanged.

We used Python 3.5 programming language to create and run an HTTP server. Raspberry Pi can also run a Python HTTP server, so we could create a direct communication between any two devices on the network. Moreover any number of devices can read data from the same server simultaneously. Further more custom Python classes can be converted to JSON files and directly read by a Python client, giving us an opportunity to effectively save and access any data with any structure. Keeping data structure also helps to keep names short and makes transfer speeds more efficient.

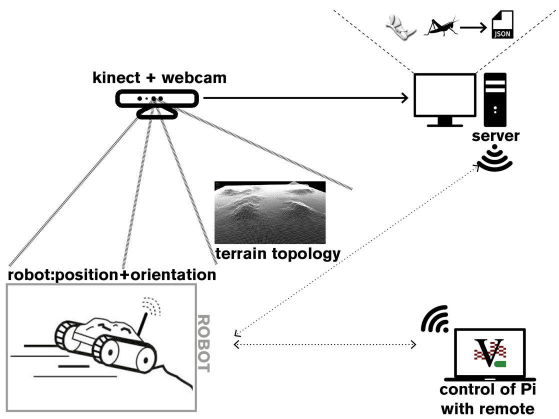

Figure 14: Schematic of whole system communication.

As stated previously, we replace a local hardware sensing of 3D environment by each robot with global 3D sensing by Kinect. Because this device is placed above the whole landscape, it gives us real time information about the 3D shape of the manipulated terrain. This data is further processed, trimmed and split into separate local data for each robot to simulate what a hardware sensor on a robot would see. Because of the low precision of the overall system, we down-sampled the point cloud and uploaded all height values onto another custom HTTP server. As these points are on a fixed grid, we do not need to upload their x and y coordinates and thus minimize the data transferred. We used Processing to get Kinect data as our server-client setup naturally allows for use of multiple platforms.

We used a full HD web-cam for tracking positions of all robots. This again follows the concept of trimming and splitting global data into separate positions for each robot. The TUIO library we use gives a good performance in low level lighting and outputs marker's x and y position and orientation in the x-y plane. These data are saved as a structured JSON files and shared through another Python HTTP server.

We used a Python application based on the PyGame library to read the two servers containing global positioning and 3D landscape information. This application trims the data into each robot's local view and calculates the next necessary movement for each robot as described in the next section. It defines the movement by a simple string command that is saved as a JSON file and shared over its own HTTP server. Available commands are 'go forward', 'go backward', 'turn left', 'turn sharp left', 'turn right', 'turn sharp right', 'stop' and 'execute custom function'. Custom functions are described later.

IV.II Internal Communication

Each robot's main processing unit is a special Makeblock board based on Arduino Uno. Makeblock's modifications include universal ports replacing standard Arduino pins while keeping all functionality. Universal ports designed for Makeblock's modular hardware is the reason to keep this board in our robot's design as well as low power consumption. The drawback is difficult wireless communication and debugging of code while the robots are in operation.

For this reason we used Raspberry Pi on top of the Arduino board. The Arduino board listens to various string commands sent from the Raspberry Pi, processes them and then directly controls the robot's motors and servos over a PWM signal. The communication with Pi is wired and through a serial port.

Because the Pi is running on a Linux based Raspbian system, it naturally allows us to use some key features, keeping our Wi-Fi connection alive. Because Pi has its own power supply, even in case of a failure on the Arduino board (for example due to a power shortage when a robot's motors draw too much current), Pi does not restart. This together with the ability to use wireless remote desktop on the Pi, gives us an efficient way of controlling and debugging each robot's code. Additionally we can directly see what data is being sent to each robot in a web-browser on the Pi, which significantly helps with debugging communication issues.

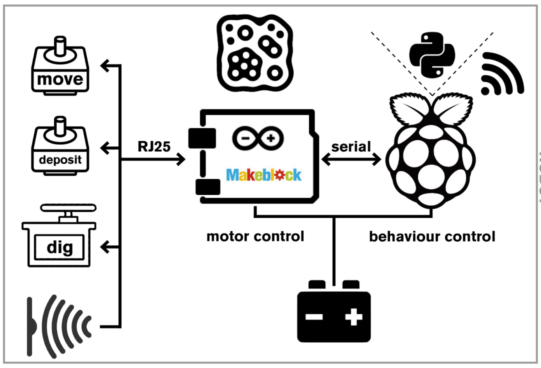

Figure 15: Schematic of on-board robot communication.

Pi reads commands sent out by the HTTP server created with the PyGame tool. These commands are processed and, further more, low-level commands with direct speed for motors or position angle for servos are sent over serial port to the Arduino.

Pi also enabled the robot to listen to a special command 'do' function 1, 2 etc. These functions are stored directly on the Pi and were created by human manual control of the robot while this activity was recorded. Thus each function is a sequence of speed values for all motors and position angles for all servos with a constant time frequency. We used these functions to teach a robot how to dig sand and how to deposit sand. These functions need to control all motors and servos simultaneously in a specific order to overcome hardware constraints and thus are efficiently created with manual guidance.

IV.III Constraints

Because we purposefully used low-tech hardware, the data we deal with naturally has lower precision, as well as execution of data sent is with lower precision. An HTTP server has data ready on request, e.g. whenever a client needs the data. This helps us to overcome any communication delays caused by different execution times of parts of the system.

Because each robot's ability to follow only a limited range of curvature along its path, we approximate each robot's path by a sequence of straight movements and turns.

Communication plays a key role in navigation as each robot's next move is always based on its actual position to deal with any imprecision in physical movement.

V. WORKSHOP: NAVIGATION

A frame crossing the width of the sandbox was supported at a height of 2.7m above the centre to hold the Kinect scanner and a webcam. The Kinect is used to gather a height-map of the sand terrain in real-time (Figure 16), resulting in a point cloud of a uniformly spaced horizontal rectangular grid (x,y), and corresponding height values (z). The grid is down-sampled from the native resolution of 800:600 points (5mm horizontal and 1mm vertical resolution) to 80:60 points to allow for real-time processing.

Figure 16: Kinect height map of sand.

This resolution is good enough for obstacle avoidance, including other robots. As is further described in the behaviour section, the robots navigate through space by following the flattest path while avoiding bumps in their surroundings. These bumps or height differences are measured by the Kinect.

The webcam tracks the reacTIVision fiducial markers (TUIO tracking toolkit [3]) attached to each individual robot to get their position and orientation which is then stored as a vector in a custom Python class (Figure 17). Because we use hardware motors that have limited precision in terms of speed and torque, especially at low speed, robots move along their paths following the next rules. Each path is considered the target path and each robot tries to follow this as closely as possible, but it is allowed to only move forward, turn left or right or turn sharp left or right on spot.

Because of that each robot starts moving forward and has its position continuously tracked and once it is far enough from its target path, it stops going forward and turn left or right to steer towards its path. A point on the robots path, which is also offset from the actual robot's position forward by the robot's length is the robot's target orientation. Once the angle between the robot's forward vector and the vector given by the target point on its path and the robot's centre point is less then our setup threshold, the robot moves forward again. This continuous loop of moving and correcting eventually ends up in following the calculated path with certain precision.

Figure 17: TUIO tracking algorithm.

Combination of the terrain height-map and the robot positions gives the final positioning system (Figure 18). As is described previously in the communication section, communication speed plays an important role, because the robots listen to commands, which might force them to change direction, but keeps executing current commands before new commands are available. Thus the speed of communication directly influences how precisely the robot follows its path.

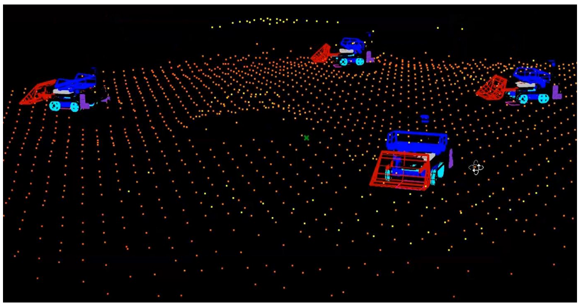

Figure 18: Terrain navigation analysis.

VI. WORKSHOP: BEHAVIOUR

Each of the individual robots has encoded a set of behaviours meant to enable them as individuals to achieve the swarm goals. Based on the tasks the robots need to accomplish we can break the behaviours down into three categories: i) path planning and obstacle avoidance; ii) environment mapping and knowledge sharing; and iii) task related behaviours.

VI.I Path Planning and Obstacle Avoidance

The basic task that each type of robot needs to accomplish is moving around the environment. In order to simplify the task we breakdown the navigation space into a grid of tiles, representing connected nodes in a mesh-type graph. This allows us to transform the navigation problem into a path-finding and graph-traversal problem: the process of mapping an efficient traversable path between multiple nodes.

We define a start point and an end point as nodes in the graph. The start point is always the position of the agent and the end point varies on the task that the agent needs to accomplish. Possible end points are either regolith gathering or depositing areas (Figure 19).

We decided to use the A* search algorithm, recognised by the computer science community [7] for performance and accuracy [25]. A* is an informed search algorithm, solving the problem by querying all possible paths to the solution (goal) for the smallest incurred cost (least distance travelled, shortest time, etc.), considering first the ones that potentially could lead most quickly to the solution. The algorithm is formulated in terms of weighted graphs, starting from a specific node of a graph, it creates a tree of paths starting from that node, expanding paths one step at a time, until one of its paths ends at the predetermined goal node.

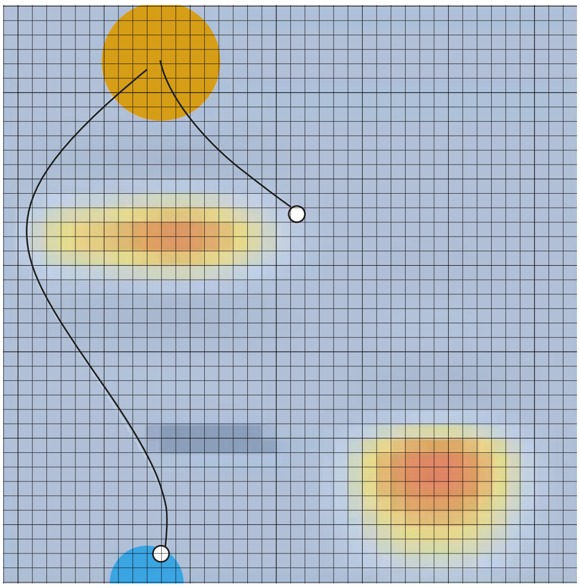

Figure 19: Path planning and obstacle avoidance.

Obstacle avoidance is achieved by adding the height value of the terrain, with zones that are more difficult to navigate having a higher cost to traverse. We use the navigation and obstacle avoidance algorithm to also control collision between agents. In the height-map data, the position of the agents is represented as a wall surrounding the robot that cannot be navigated. This is done by adding an infinite weight value around the neighbouring tiles surrounding the agents positions.

VI.II Environment Mapping and Knowledge Sharing

We are constantly monitoring the agents' environment for topographical changes using the Kinect sensor. Data from the sensor is used to map the complex topographical sand surface as a simplified 3d point cloud.

For the workshop we decided to develop and test an algorithm that would add resolution to the topography information by using the agents' actions as another layer of data. As each agent performs the task of digging or depositing, we record their navigation paths and endpoints, mapping where sand was collected and deposited. Each agent thus creates an individual local environment map. The individual maps are aggregated when two or more agents are within a distance that allows them to communicate with each other (Figure 20).

We believe giving each agent a personal memory would enable emergent behaviours and allow for a top-down goal achievement using a bottom-up individual resolve. This is very similar to ants' stigmergic behaviours where pheromones are used to communicate data about the ants' environment.

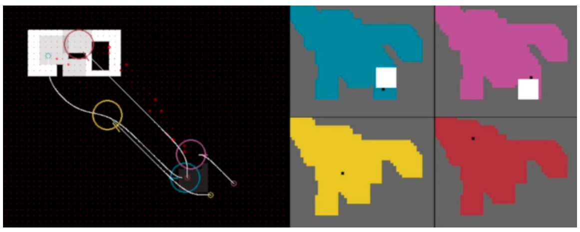

Figure 20: Environment mapping and sharing.

VI.III Task-related Behaviours

The robots have three task related behaviours encoded: moving, digging and depositing. The digging and depositing algorithms are very simple and relate to the robots servos being triggered in specific sequences, activating different mechanical parts.

Moving is a more complex algorithm. After the navigation path has been calculated, we de-construct the path as a series of commands to activate the driving servos in specific commands such as move left, move right, move forward and in some cases move backwards.

Not knowing the workshop participants' coding levels we decided to build a record function that allowed them to record a sequence of commands and replay that sequence. We hoped in the first days of the workshop, participants would play with the recorded functions and develop strategies for accomplishing the bigger goal of constructing the Martian habitat.

VII. IMPLEMENTATION/WORKSHOP RESULTS

An artificial terrain landscape was created measuring 3.5m x 6m, with a depth of 0.15m. This was filled with $3m^3$ of washed beach sand (97% quartz, 0 - 0.2mm grain size) to create a sandbox (Figure 21).

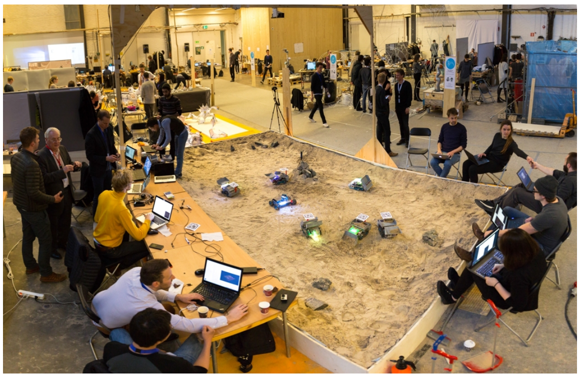

Figure 21: Final demonstration.

The workshop was held at Chalmers University over four days in April 2016. Ten participants with various backgrounds and levels of experience were involved in putting the systems together, exploring their own developmental ideas, and testing. The final demonstration involved a public exhibition on 8th April 2016 where we received feedback and further ideas from the other nine workshop clusters (Figure 22).

Figure 22: Final demonstration.

During the workshop, the participants succeeded in understanding the requirements necessary for the robot system, constructing their own individual robot following the pre-workshop plans, and integrating them with the global communication system. This enabled us to run a series of basic tests, such as navigating between ‘dig’ and ‘deposit’ regions whilst avoiding one another and environmental obstacles. Given the short period of time during the workshop, there was only sufficient time for the participants to become acquainted with building and operating the robots, rather than developing their own improvements.

VII.I Limitations

Due to the overall complexity and interdependency of separate components, it was difficult for the robots to dig useful quantities of sand and therefore to construct anything. Mechanical issues in the robot design, also because of using low-cost hardware, are balanced with software sophistication taking the hardware limitation into account. The root cause of this problem was not the scoop bucket, but rather the low weight of the robot, which meant that this traditional method was not efficient. The same issue would arise in a low-gravity environment.

To develop the system further, more complex behavioural rules can be added to the swarm, along with improved positioning resolution, on-board distributed sensing to remove the need for the proxy global positioning, and the addition of the capability for material bonding.

VIII. CONCLUSION

Although the objective of constructing a prototype habitat regolith shield was not possible within the workshop period, a number of useful lessons were learned during the pre-workshop development phase and the workshop itself. The workshop study allowed us an empirical understanding of the challenges involved and made clear the areas that required further work.

In terms of the concept design, the next developmental steps can be broken down into four parts: i) operation of the robotic system (detailed robot specification, autonomous control, communication, power beaming); ii) the additive construction process (material characterisation, filtering, melting process, layering); iii) further design of the habitat module itself; and iv) integration of the previous three components in the field.

IX. ACKNOWLEDGEMENTS

We would like to thank Smartgeometry, Chalmers university and Foster + Partners for supporting this investigation, Makeblock for sponsoring our cluster by supplying equipment, and the cluster participants for their involvement during the workshop (Daniel Akesson, Angelo Figliola, Max Haney, Johanna Isaksson, Daniel Lamm, Scott Leinweber, Carmen Matiz, Teodor Michalski, Katja Virta, Konrad Zaremba, Reinier Zeldenrust, Ahmad Abdul Sater, Oscar Borgstrom).

References

[1] D. Arbuckle and A. A. Requicha. Self-assembly and self-repair of arbitrary shapes by a swarm of reactive robots: algorithms and simulations. *Autonomous Robots*, 28(2):197–211, 2010.

[2] M. Barmatz, D. Steinfeld, M. Anderson, and D. Winterhalter. 3d microwave print head approach for processing lunar and mars regolith. In *Lunar and Planetary Science Conference*, volume 45, page 1137, 2014.

[3] R. Bencina and M. Kaltenbrunner. The design and evolution of fiducials for the reactivision system. In *Proc. 3rd International Conference on Generative Systems in the Electronic Arts*, 2005.

[4] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[5] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 476–481. IEEE, 2000.

[6] G. Cesaretti, E. Dini, X. De Kestelier, V. Colla, and L. Pambaguian. Building components for an outpost on the lunar soil by means of a novel 3d printing technology. *Acta Astronautica*, 93:430–450, 2014.

[7] D. Delling, P. Sanders, D. Schultes, and D. Wagner. Engineering route planning algorithms. In *Algorithmics of large and complex networks*, pages 117–139. Springer, 2009.

[8] B. G. Drake, S. J. Hoffman, and D. W. Beaty. Human exploration of mars, design reference architecture 5.0. In *Aerospace Conference, 2010 IEEE*, pages 1–24. IEEE, 2010.

[9] J. Everist, K. Mogharei, H. Suri, N. Ranasinghe, B. Khoshnevis, P. Will, and W.-M. Shen. A system for in-space assembly. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2356–2361. IEEE, 2004.

[10] K. C. Galloway, R. Jois, and M. Yim. Factory floor: A robotically reconfigurable construction platform. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2467–2472. IEEE, 2010.

[11] S. J. Hoffman. *The mars surface reference mission: a description of human and robotic surface activities*. National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, 2001.

[12] A. Howard, L. E. Parker, and G. S. Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6):431–447, 2006.

[13] A. S. Howe, B. Wilcox, C. McQuin, J. Townsend, R. Rieber, M. Barmatz, and J. Leichty. Faxing structures to the moon: Freeform additive construction system (facs). *AIAA*, 5437:10–12, 2013.

[14] K. J. Kennedy, J. Raboin, G. Spexarth, and G. Valle. Inflatable structures technology handbook. chapter 21; inflatable habitats. 2000.

[15] B. Khoshnevis, M. P. Bodiford, K. H. Burks, E. Ethridge, D. Tucker, W. Kim, H. Toutanji, and M. R. Fiske. Lunar contour craftinga novel technique for isru-based habitat development. In *43rd AIAA Aerospace Sciences Meeting and ExhibitMeeting Papers*, pages 7397–7409, 2005.

[16] J. Kozicki and J. Kozicka. Human friendly architectural design for a small martian base. *Advances in Space Research*, 48(12):1997–2004, 2011.

[17] T. Peters and B. Peters. *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design*. John Wiley & Sons, 2013.

[18] M. Rubenstein and W.-M. Shen. Automatic scalable size selection for the shape of a distributed robotic collective. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 508–513. IEEE, 2010.

[19] L. C. Simonsen and J. E. Nealy. Radiation protection for human missions to the moon and mars. 1991.

[20] K. Stoy and R. Nagpal. Self-repair through scale independent self-reconfiguration. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 2062–2067. IEEE, 2004.

[21] Y. Tan and Z.-y. Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.

[22] W. Truszkowski, H. Hallock, C. Rouff, J. Karlin, J. Rash, M. Hinchey, and R. Sterritt. *Autonomous and autonomic systems: With applications to NASA intelligent spacecraft operations and exploration systems*. Springer Science & Business Media, 2009.

[23] S. Wilkinson, J. Musil, J. Dierckx, I. Gallou, and X. de Kestelier. Autonomous additive construction on mars. In *ASCE Earth and Space*, 2016.

[24] S. Wilkinson, J. Musil, J. Dierckx, I. Gallou, and X. de Kestelier. Concept design of an outpost for mars using autonomous additive swarm construction. *ESA Acta Futura special issue*, 2016.

[25] C. R. L. Zeng W. Finding shortest paths on real road networks: the case for a*. *International Journal of Geographical Information Science*, pages 531–543, 2009.