

YouWasps: Towards Autonomous Multi-Robot Mobile Deposition for Construction

Julius Sustarevas¹, Benjamin K. X. Tan¹, David Gerber², Robert Stuart-Smith^{1,3} and Vijay M. Pawar^{1,3}

Abstract—Mobile multi-robot construction systems offer new ways to optimise the on-site construction process. In this paper we begin to investigate the functionality requirements for controlling a team of robots to build structures much greater than their individual workspace. To achieve these aims, we present a mobile extruder robot called YouWasp. We also begin to explore methods for collision aware printing and construction task decomposition and allocation. These are deployed via YouWasp and enable it to deposit material autonomously. In doing so, we are able to evaluate the potential for parallelization of tasks and printing autonomy in simulation as well as physical team of robots. Altogether, these results provide a foundation for future work that enable fleets of mobile construction systems to cooperate and help us shape our built environment in new ways.

I. INTRODUCTION

The US\$8.8 trillion global market value[1] of the construction sector, shows the unmatched effort that construction, as a human endeavour, receives. In fact, construction is often seen as the sector at the forefront of tackling global challenges like population growth, urbanisation and sustainability[2]. However, historically this sector has seen stagnating production efficiency [3] and industry reluctance to adopt new strategies and technologies [4]. Consequently, despite successful automation of additive manufacturing being on the rise[5], autonomous robotic construction remains under-explored. In particular, this includes techniques for controlling single or multiple robots that do not require a human operator or explicit instructions. By exploring how to manage and control multi-robot teams to autonomously perform construction tasks, in this paper we highlight the potential of these solutions to improve adaptability and scalability of production. Further, by developing and deploying these algorithms onto real autonomous multi-robot teams, we illuminate the diverse set of robotics challenges that on-site construction tasks raise.

Driving change, additive manufacturing technologies are demonstrating material efficiency and improved production, as well as enabling greater flexibility, speed and design expressiveness [6]. Emerging results include projects by Laing O’Rourke on off-site manufacturing, and additive

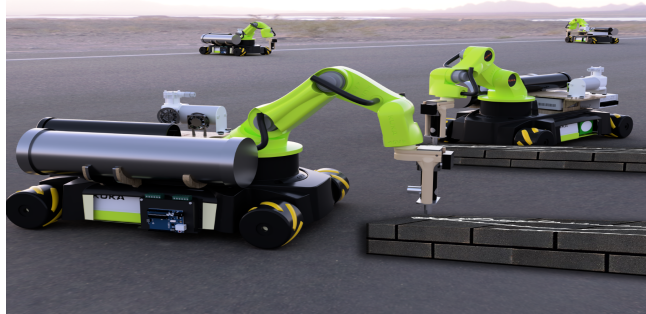


Fig. 1: Visualisation of YouWasp team of robots deployed in a construction scenario.

manufacturing companies such as Apis Cor in Russia, Winsun Decoration Engineering Company in China, NASA’s 3D printing in Zero-G and US Armed Forces on-site 3D printed barracks. Within this context, typical examples use either gantry-type solutions or industrial robots with 3-to-4 degrees of freedom to ensure precision and repeatability of motion for continuous material deposition [7]. However, these approaches are often limited by the reach of the robot, use of sequential production strategies and do not consider environmental constraints or concurrent activity occurring in the same space. As a result, for scenarios such as on-site construction in an urbanised or hazardous environment, current additive manufacturing approaches are impractical as they do not provide the form factors to easily scale production, dynamically manage operational activity nor flexibility to adapt to local events [8].

To consider the utility of mobile multi-robot systems for on-site construction, in this paper we: 1) designed and developed a mobile robotic extruder platform capable of performing autonomous material extrusion, called YouWasp. 2) developed approaches for on-board multi-robot-aware planning and multi-robot task allocation to perform deposition tasks. 3) tested the capacity of the YouWasp platform to operate as a multi-robot mobile extrusion system by evaluating the proposed planning and task allocation approaches. In doing so, we extend previous work in multi-terrain 3D printing mobile robots [9].

II. BACKGROUND AND RELATED WORK

This paper combines research from diverse range of fields from additive manufacturing to control and multi-robot systems. For a full review of additive manufacturing methods for large scale construction please see [10]. As the focus

¹Authors are members of the Autonomous Manufacturing Laboratory within the Department of Computer Science, University College London, Gower Street, WC1E 6BT, UK. julius.sustarevas.16@ucl.ac.uk benjamin.tan.17@ucl.ac.uk

²Arup: david.gerber@arup.com

³Corresponding authors: r.stuart-smith@ucl.ac.uk and v.pawar@ucl.ac.uk

of this paper is to investigate the utility of mobile multi-robot systems for construction, we explored 1) mechanisms for mobile multi-robot construction and 2) multi-robot task allocation strategies.

A. Mobile Multi-Robot Construction Systems

Recent progress has been made in construction robots that can assist in building large scale structures. Examples include: robot arms to construct brick [11] and steel reinforced walls [12], human-robot collaborative brick layers [13], and robot arms that can spin carbon-fibre pavilions on-site [14]. More widely, conventional industrial robots have been used to reduce labour costs by automating a variety of tasks including concrete spraying, surface finishing, installation of glass panels and joint welding tasks.

Of those focused at developing robots that can construct at architectural scales, new systems, designed to work as multi-robot system, are emerging. These projects include: Minibuilders[15], the Digital Construction Platform (DCP) [16], Swarm-based robotic construction [17], TERMES [18] and IkeaBot [19]. Developed by the Institute for Advanced Architecture of Catalonia, Minibuilders is a family of three small-scale construction robots: the Foundation robot, the Grip robot and the Vacuum robot. Each robot is capable of fabricating a structure large than itself and designed to operate as group of 3 heterogeneous robots using a pre-defined sequenced of tasks. In contrast, DCP developed by Keating et al is a much larger system consisting of a compound robotic arm with a mobile base (Altec AT40GW track-driven carrier). Physical demonstration of this system evaluated the ability to fabricate a 14.6-metre-diameter and 3.7-metre-tall hemispherical dome using a single robot. Further, whilst DCP was equipped with a mobile base, the study did not demonstrate freedom to print whilst moving. The Harvards TERMES project demonstrated mobile construction capabilities utilising bespoke robots and building blocks, displaying a capacity to assemble structures using a collective of simple independent autonomous robots. And more recently, extending work that produced the IkeaBot, Zhang et al demonstrated large scale 3D printing system using 2 mobile robots to build a concrete structure concurrently[7]. Besides the work by Zhang et al, there is limited research investigating the computer science challenges of multi-robot construction that have also been implemented on physical systems.

B. Multi-Robot Task Allocation

To date, the robotics community have realised several projects investigating different types of coordination strategies to enable efficient operations in time and working space. Examples include: ACTRESS, GOFER, CEBOT, ALLIANCE, M+, MURDOCH and ASyMTRe (for a full review of MRS and coordination to go [20]). Of those thought to be appropriate for performing construction tasks, Gerkey et al developed an auction based task allocation system called MURDOCH [21]. This system was designed to allocate and coordinate manipulation tasks in a fault-tolerant manner,

showing a tightly coupled cooperation among the robots. Specifically, the auction protocol comprises of the following steps: task announcement from a tree of tasks with hierarchical inter-relationship, metrics evaluation, anonymous bid submission, close of auction, progress monitoring and contract renewal. The reassignment of a task is also permitted by the auctioneer if insufficient task execution progress is found within the time limit. Distinguishing this study from other approaches, MURDOCH has also been validated through a series of experiments using real robots. Within the context of pushing a box collaboratively, the researchers demonstrated the viability of coordinating 3 heterogeneous robots to autonomously perform a set of physical interactions.

Gerkey and Mataric progressed this research by developing a taxonomy for evaluating multi-robot task allocation systems. They described the Multi Robot Task Allocation (MRTA) problem into 3 sub categories: single-task and multi-task robots, single-robot and multi-robot tasks, instantaneous assignment and time-extended assignment [22]. Based upon theoretical analysis and mathematical formulation, the MRTA problem was considered to be strongly NP-hard, except for the single-task robots (whereby single-robot tasks and instantaneous assignment can be solved in polynomial time). Since then, several studies have attempted to improve Gerkey and Mataric's taxonomy. One of the studies is the auction-based approach to complex task allocation for multi-robot teams. Within this study, Zlot et al introduced a list of definitions (e.g. inter-task constraints, inter-robot constraints, task decomposition, scheduling and etc.) to support optimality-oriented solution (for a full analysis of this approach see [23]). They also demonstrated a distributed auction protocol for the complex task in the context of multi-robot object pushing and area monitoring. Other approaches include Korsah et al. , introduced a new comprehensive taxonomy, iTax, that considers task constraints explicitly by mapping the mathematical models from combinatorial optimisation and operations research [24].

To evaluate the YouWasp platform and, in turn, capacity of mobile multi-robot systems for construction, we developed a task allocation system based upon an auctioning principle. This is described in chapter 3.

III. YOUWASP PRINTING PIPELINE

In this section we discuss the main components of the YouWasp printing pipeline. These are 1) hardware and software elements that make up a YouWasp robot and 2) the centralised multi-robot software that performs task decomposition and allocation functions. Development of these methods took into account construction sites usually being well defined and equipped areas. Hence few application-informed assumptions were made: 1) it is reasonable to assume 5GHz wireless network over the scale of a regular construction site 2) construction robots ought to make decisions autonomously and independently, but it is reasonable for a central ground station to perform task decomposition

and act as an auctioneer. Hence task decomposition and allocation itself need not be distributed.

As the focus of this paper is the YouWasp platform embodying the proposed software methods, a number of placeholder systems and artificial limitations were introduced to narrow and isolate the challenges at hand from other open questions in robotics or manufacturing. Firstly, rather than use a Simultaneous Localisation and Mapping (SLAM) solution, an OptiTrack motion capture system (calibrated to 0.3mm error) was used to provide mobile base pose estimation for closed-loop base control. OptiTrack was also used for ground truth end-effector position tracking. Secondly, We limited the evaluation of the printing pipeline algorithms to deposition of a single layer of material. YouWasps being ground robots, a single layer is sufficient to create untraversable obstacles in the environment and hence explore the challenges of multi-robot coordination and deposition in a dynamic environment. However, the deposition system was shown to be able to deposit few sequential layers even if their quality was not pursued.

A. YouWasp: Mobile Printing Robot

YouWasp platform consists of a modified KUKA Youbot robot [25] together with an on-board extruder system. These hardware components, seen in Fig.2 were integrated in order to enable untethered mobile extrusion and capacity for autonomous motion planning.

The Kuka Youbot operating time was extended from 90min to about 5 hours by addition of a 20Ah capacity Lithium-Polymer battery and regulating circuitry as seen in Fig.2. The Youbot internal computer was upgraded to an Intel Nuc i7 mini PC to increase the computational capacity and enable 5GHz wireless communication.

Youbot arm has a payload of 0.5Kg, which is not sufficient for most material deposition solutions. The WASP extruder[26], consisting of a material cylinder driven by air pressure and an screw based flow regulator end-effector weighing 1Kg. Therefore the gripper and link 5 of the 5-DOF arm were removed and the extruder was mounted on joint 5 as shown in Fig.2. Unfortunately this caused severe limitations and robustness issues as high weight of the end-effector would lead to current spikes during motion. Hence, the WASP end-effector was replaced by a passive revolute joint holder of the extrusion hose. The angle of the passive joint was governed by forces applied by the hose. An electric ball-valve was introduced after the material cylinder in order to control the flow to the nozzle. The system was made mobile by an on-board air compressor.

Motion Control: To enable autonomous printing by a single YouWasp, all software components were hosted on-board. Software was built using Robot Operating System (ROS) [27] and MoveIt! [28] motion planning framework. This let us use infrastructure tools such as robot description languages, networking and state-of-the-art libraries like the efficient inverse kinematics solver Trac-IK[29].

Continuous mobile printing in an area greater than the immediate working volume of a robot arm requires a com-

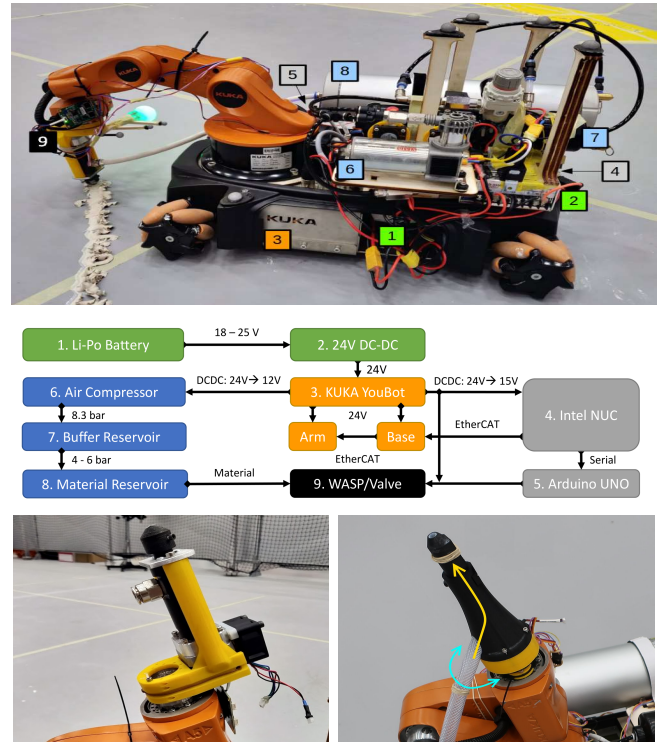


Fig. 2: **Top:** YouWasp mobile printing robot, hardware components labelled. **Middle:** System diagram of YouWasp hardware components. **Bottom Left:** WASP extruder replacing the 5th Youbot arm link. **Bottom Right:** Passive-joint(light-blue) hose-nozzle end-effector

bined base-arm control. Figure 3, shows how the existing Youbot interface [30], which provides a 5-DoF arm trajectory controller, had to be expanded to a base-arm 8-DoF trajectory controller. This was achieved using virtual planar and revolute joints and writing a closed-loop PID controller for base motion. The base controller was calibrated for quick response and used the OptiTrack motion tracking for loop closure. Optionally wheel odometry could also be used. This controller, together with the arm controller, was wrapped into a 8-DoF interface. Thus allowing planning to be performed on the entire robot.

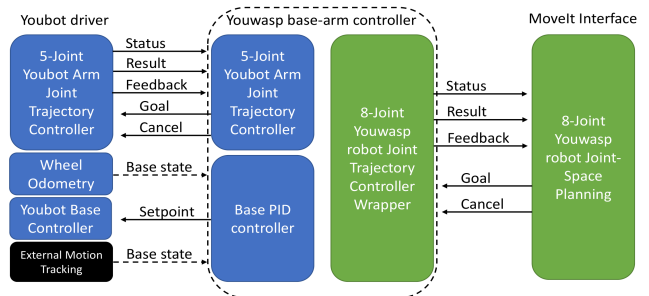


Fig. 3: 8-DoF base-arm joint trajectory controller implementation

Furthermore, the motion planning for printing functionality must avoid other YouWasps and any static obstacles in the environment e.g. previously deposited material. The MoveIt! framework was used by each robot to maintain independent

occupancy octomap and to perform collision aware motion planning. Collisions with printed material were accounted for by the depositing robot creating collision objects and adding them to its own occupancy map as well as broadcasting them to all other robots on the network once deposition stops. Hence, collision with material/trajectory that has already been deposited is taken into account by all robots, but a trajectory of material that is still not finished, is not avoided. Furthermore, every robot was made to populate and update the occupancy maps of all other robots with its own location.

Whilst maintaining an up to date occupancy map would allow collision free movement from pose to pose, the MoveIt! framework would not check for changes in occupancy map during an execution of a Cartesian trajectory. Therefore an algorithm had to be developed to allow planning and replanning a Cartesian trajectory in order to deposit material. The *PrintPointList* algorithm (simplified for readability) prints a point list (*pointList*) of x,y,z coordinates whilst replanning with updated occupancy map. The algorithm starts by picking a good robot state to start printing. This is done in the *GoToPrint* function which samples IK solutions of the first point and plans Cartesian trajectories from those states. Pose is chosen if the plan was successful for at least 20% ($thresA = 0.2$) of the trajectory. Then in *Print* function after the robot arrived to the pose, Cartesian trajectory is replanned and executed whilst constantly replanning from current state for the remainder of the points. Since these points were have been successfully planned before, $percent_successful(plan) < 1$ implies a new obstruction. To make the algorithm less sensitive, $percent_successful(plan) < thresB = 0.95$ was used, as it gives time for the obstruction, likely another robot, to move.

Taking into account collisions with static and dynamic environment, allows printing trajectories that require the robot to discontinue the print and reposition itself. To simulate this challenge, we used an artificial test scenario consisting of a straight line that is situated between 3 boxes as in Fig.4 and, by design, requiring at least two relocations to print. Using this test case, we found that trajectories were always executed successfully. However, without using look-ahead IK sampling-planning and simply moving to the first point on the list, the average repositioning count among seven trials was 11.6. A further test involving a straight line trajectory and two moving obstacles on either side as in Fig.4 found all 7 trials successful at printing the trajectory with on average 8.8 reposition events.

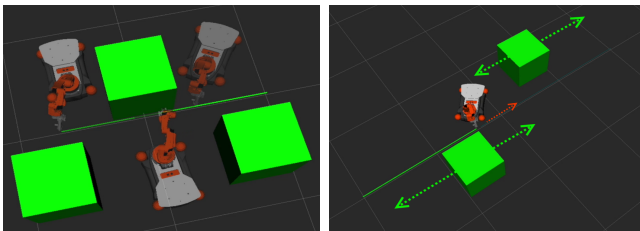


Fig. 4: **Left:** YouWasp discontinuing the print and repositioning around static obstacles. **Right:** YouWasp repositioning and printing in presence of dynamic obstacles

PrintPointList

```

procedure PRINTPOINTLIST(pointList)                                ▷ Print a segment
  while length(pointList) > 0 do
    GoToPrint(pointList)
    ▷ Plan printing from current state
    plan ← PlanCart(pointList)
    if percent_successful(plan) > thresA then
      ▷ Print only successfully planned points
      Print(succeeded(pointList), plan)
    else
      continue                                                    ▷ Reposition to a better pose
    pointList ← deletePrinted(pointList)
  procedure GOTOPRINT(pointList)                                    ▷ Move to first point
  while true do
    ▷ Sample IK for first point, plan path assuming that pose
    start_pose ← IK(pointList[0])
    plan ← PlanCart(pointList, start_pose)
    if percent_successful(plan) > thresA then
      break
    goToPose(start_pose)                                          ▷ Waits for motion to complete
  procedure PRINT(pointList, plan)
  ▷ Execute and do not wait
  execute(plan, wait = False)
  while true do                                                ▷ Replan during execution
    ▷ Plan for remaining points from current state
    plan ← PlanCart(remaining(pointList))
    ▷ percent_successful(plan) < 1 indicates new obstacle
    if percent_successful(plan) < thresB or isNotPrinting then
      ▷ Execution succeeded or encountered new obstacle
      stop()
      break

```

Deposition: Depositing material requires great fineness and most importantly - velocity control for the extruder. As printing quality was not the focus of YouWasp design, this was not accounted for and no explicit end-effector velocity control is implemented. Instead the aim was system completeness and enabling non-artificial or hypothetical feasibility testing. Therefore no quantitative metrics were used to assess the printing quality. This system completeness is enabling as effects of algorithms throughout the pipeline can then be seen at material deposition level.

The WASP extruder could only be tested for a single layer deposition due to robustness issues when the Youbot arm attempted to move the extruder. A trajectory was printed consisting of two 120deg, 3m radius arcs with 10cm offset and a sinusoidal curve in between as seen in Fig.5. Material overflow is seen as the material flow is not calibrated. Small oscillations due to base motion discussed in later chapters can be seen too.

The custom hose holder end-effector, being more agile, showed ability to stack few layers as it printed 1.6m length and 0.1m width ellipsoids with a sin wave inside. Blobs could be seen due to large delay between shutting the ball-valve and material ceasing to flow. Nonetheless, the trials were deemed successful as the tasks were completed and no collisions with material occurred.

B. Multi-robot Task Allocation Pipeline

Printing of a 2D geometry can be viewed as a *Decomposable simple task* that can be divided into subtasks i.e. printing segments that in turn are *Single-robot tasks* that are achievable by exactly one *Single-task robot*, a YouWasp,

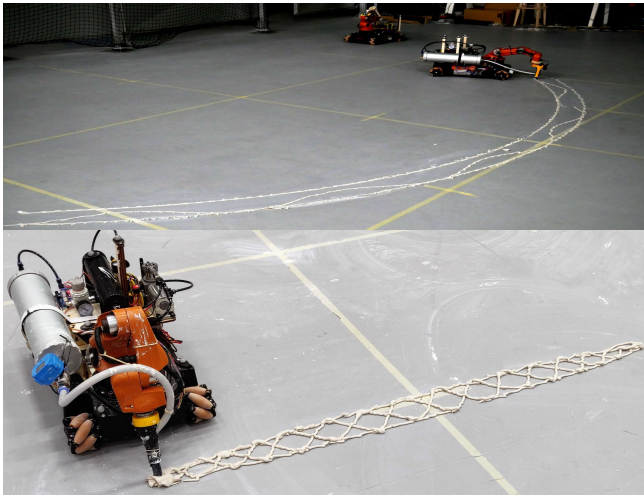


Fig. 5: **Top, middle left:** Physical deposition of 10mm thick wall along a 3m radius, 120deg arc using WASP extruder. **Bottom, middle right:** Physical deposition of $1.6m \times 0.1m$ episodes using no flow regulator.

capable of printing only one trajectory at a time. These classifications, described by Gerkey and Mataric[21] and expanded by Zlot[23] suit the system very well. Gerkey’s further notion of *Instantaneous assignment*[22] was chosen as a basis when designing the YouWasp Task allocation system, which uses auctioning and allows only immediate task allocation with no planning for future allocations, concerning the available information of the robots, tasks and environment.

1) *Task Decomposition:* The YouWasp printing pipeline starts with a binary image, created by a human. It is then loaded into the decomposition/auctioneer software and transformed into an undirected graph. The vertices and edges representing pixel occupancy and adjacency, respectively. This graph is then split up into its connected components. Cartesian coordinates of a print point and the weight (i.e. physical distance between two vertices/points) are then associated with the graph.

The algorithm traverses each connected component of the graphs with starting vertex randomly chosen from a set of 1-degree and 2-degree vertices depending on graph being open and closed shape. Then the Depth-First-Search (DFS) path is decomposed into small segments based on two criteria: 1) the cut-off threshold (i.e. accumulated physical distance in metres) that was drawn from a Gaussian distribution with mean *seg.mean* for each segment or 2) the detection of DFS path discontinuity produced by DFS. Altogether, this criteria decomposed the 2D geometry into several tasks with each task comprising of only continuous trajectory as depicted in Fig. 6.

To compactly model and represent the task dependency, a directed graph was used as shown in Fig. 6. In order to aid robot distribution in space, hence decreasing congestion and replanning time, adjacency in space was used as a task dependency. This was considered in the task sequencing method by rearranging segments of the directed graph alternatively into two groupings of different priority. The first

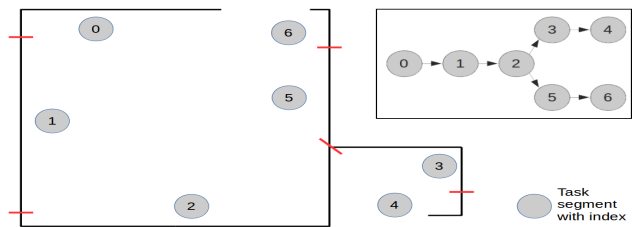


Fig. 6: Example of task decomposition and sequencing

grouping of Fig. 6 consists of vertices $\{0, 2, 4, 6\}$ whereas the second grouping contains $\{1, 3, 5\}$. This resulted in every connected component of a 2D geometry being separated into two groupings of segments each. The task segments within the same grouping were considered independent and passed to task allocation for parallel execution.

2) *Task allocation:* A first-price sealed-bid auction was used to allocate the decomposed task segments efficiently among ‘selfish’ robots in the system. Bidding was based on the Euclidean distance from the end-effector position to the starting point of the task. This was regularised by the printing capacity of the bidding robot which decreased when printing. The lowest bidding robot was allocated a task and did not participate in further bidding until the task was completed. All participating robots, bid for one task at a time and the winner was announced once all bids were received. Lastly, robots were made to only interact with centralised auction and not reallocate tasks in a peer-to-peer manner.

The auction mechanism was implemented as a ROS service server with robots querying the service as they bid for the latest task on auction. If the bidding was successful, robot invokes the PrintPointList algorithm and executes the task. Upon completion, the robot reports the finished task to the task server. Whenever it holds no task, it will return home to avoid causing congestion to other robots.

A test case was designed to investigate the important parameters of this task decomposition, sequencing and allocation approach. Teams of 1, 3, and 5 robots in simulation, were set to print a $3 \times 3m$ square with an opening as seen in Fig.7. The *seg.mean* guiding length of individual segments was varied to be 3, 5 and 7, and the two groupings of alternate and distinct segments were sequenced as follows. Sequencing A, for each connected component randomly allocated tasks from grouping one first and then grouping two. Sequencing B, for each connected component randomly allocated segments from group one whilst appending group one with segments from group two which had their parent segments in group one completed.

Five trials were carried out for each parameter configuration and total printing time and number of robots with a task (Robot engagement) were tracked. As seen in Fig.8 making the topological children of already printed segments to be immediately available for printing (Sequencing method B) has lead to higher printing time for teams of 3 and 5, and consistently lower amount of utilised robots. Therefore sequencing method B was discarded as it hindered parallelization. The mean length of print segments influence the amount

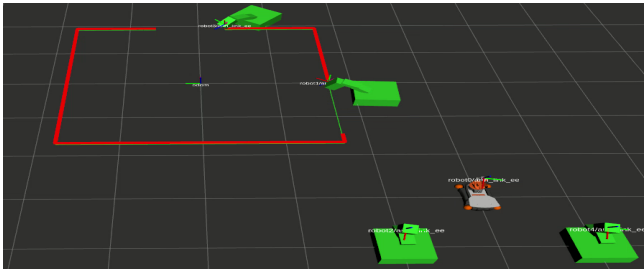


Fig. 7: Simulation test setup to explore task decomposition, sequencing and allocation parameters

of elements in each grouping and hence the total amount of tasks available for parallelization. Robots repositioning and travelling to new locations is a time-wise expensive process. This is reflected in data as short segments of mean 3 achieve full robot utilisation however the printing time is higher than other means and has high variance. Making the auctioning cost function aware of congestion, could address this. Alternatively, three-way relationship between segment mean, number of robots and size of the whole 2D geometry could be modelled.

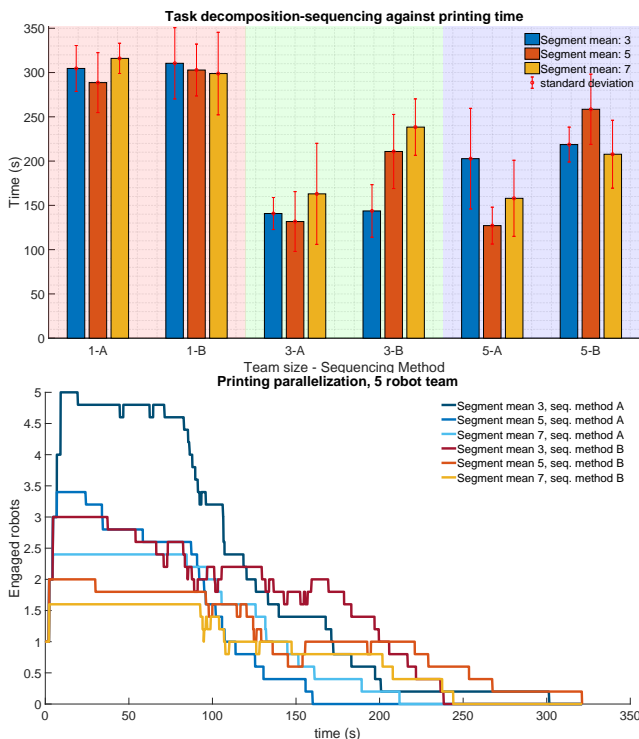


Fig. 8: Test results Printing time (Top) and Robot engagement (Bottom) (i.e. number of robots with a task in team of 5) for different team sizes, print segment lengths (*segmentmean*) and sequencing methods.

IV. SYSTEM EVALUATION

Two core experiments were carried out to showcase the validity of the YouWasp robot and printing pipeline: 1) Feasibility test in simulation to show the adaptability and scalability of the task allocation and motion planning pipeline

on a large team of robots 2) Evaluation of these pipelines via deploying them on-board a team of physical robots.

A. Scaling in Simulation

A simulation experiment was setup using a team of 10 robots and a 12.8 x 22.86m floor-plan inspired geometry as seen in Fig.9. The total printable path was 100.8m As per earlier investigation the printing segment mean was chosen to be 7 determined by taking into account the size of the team relatively to the size of the geometry. To support simultaneous running of 10 YouWasp software stacks (that also involved 3D collision aware motion planning) we used a 3.20GHz Intel Core i7-6900K CPU. All Five trials carried out were successful with average time taken of 745s (std. 53s). Note that the robot bases were limited to 0.8m/s whilst real robots are limited to 0.15m/s. The floor-plans three connected components were consistently (std. 0.7) decomposed to 25 subtasks in total.

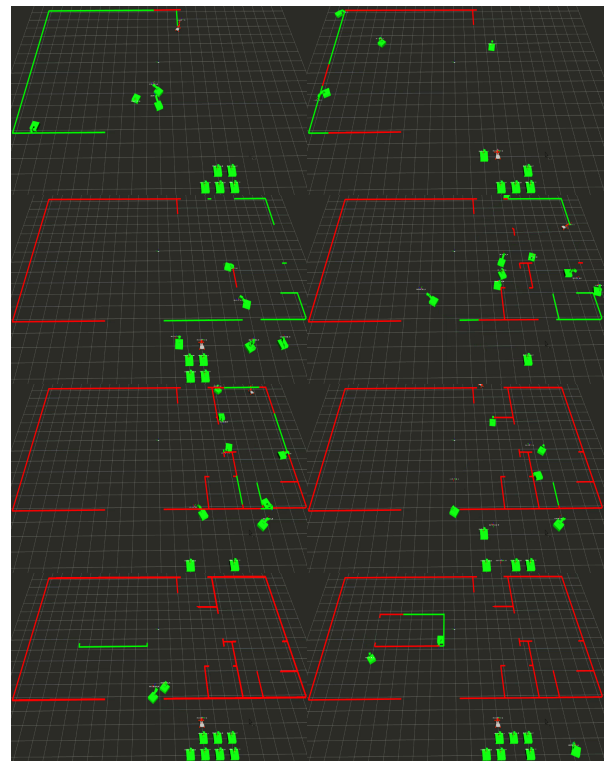


Fig. 9: 10 robot team experiment printing a 100.8m long geometry. Frames taken at 60, 240, 420, 600, 780, 960, 1140, 1320 seconds. Green trajectories are currently allocated tasks, Red are already printed points.

Peaks in robot utilisation data Fig.10 highlights the under-utilisation of robots due to connected components being printed sequentially. This is a major flaw as unnecessary limitation drastically reduced utilisation of robots. Merging groupings 1 and 2 between connected components and forming single large sets of *parent* and *child* tasks could potentially address the issue. Importantly, consistent and successful completion of the task helps build confidence in scalability and adaptability of the task decomposition, sequencing and allocation approaches taken.

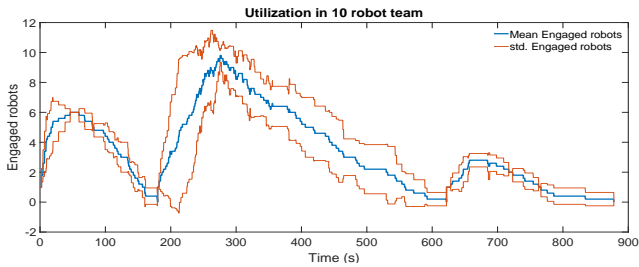


Fig. 10: Robot utilisation (in engaged robots i.e. robots with allocated task) throughout the large-scale printing simulation

B. Autonomous Multi-Robot Construction

Whilst only one YouWasp platform was built, a second platform lacking the extruder system was modified in the same fashion. To showcase the completeness of YouWasp printing pipeline and the embodiment of software on a physical team of robots, we performed a series of *virtual printing* (i.e. printing without depositing material). The team of two robots was formed consisting of one fully equipped YouWasp robot and one that lacked the extrusion system. Both robots used a *virtual printing* end-effector that was simply an RGB LED bulb designed for reliable motion tracking and path visualisation. As the tracked arena and hence robot workspace was 8x8m, a 5x5m studio flat inspired floor-plan was chosen for printing. Software was deployed as described in Chapter 3, the robots planning autonomously, but using OptiTrack localisation. The Task decomposition and allocation software was deployed on a ground station PC. A total of 4 trials were conducted using this 2-robot team that autonomously navigated the space and tracked print segment trajectories whilst visually indicating printing. The setup of these trials can be seen in Fig.11. Data from one trial was discarded as an expected collision of not modelled (no collision boxes) elements of end-effector occurred. As this limitation is known, data was discarded.

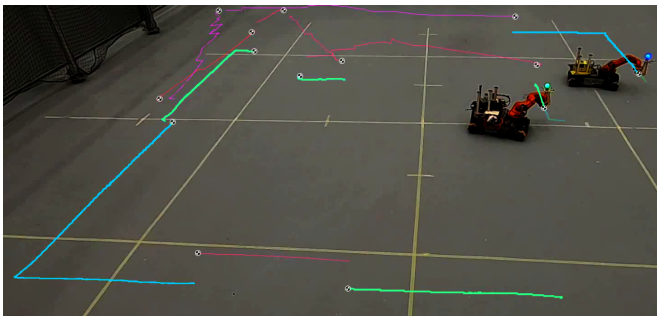


Fig. 11: *Virtual printing* with a team of 2 robots. Grid of 2m increments and 1 m markings seen on the floor. The coloured end-effector tracking visualised using visual post-processing and only roughly correspond to true trajectories followed. Green and Blue colours are end-effector positions whilst printing corresponding to different robots. Thin red and purple lines are travel-time positions.

To assess our multi-robot printing approach, we used printing accuracy. As seen in bottom part of Fig.12 the nominal accuracy for printed segments was less than 10mm. The average error, across 3 trials and all trajectories, of the end-

effector 3D position whilst printing was 9.1mm (std. 8.7mm). However, as top part of Fig.12 indicates a consistent presence of errors as high as 50mm around all major corners in the geometry. Note that this plot was generated by superimposing data across trials based on the error. Thus it is a plot of maximum printing error across trials. This error pattern was found to be consistent with property of DFS decomposition. The starting points for printed segments were likely to be where lines of the geometry join or end. This pattern can then be explained by noticing from print segment error plot that the end-effector consistently fails to position itself accurately to start the printed segment. This is likely due to the base controller PID loop not tracking commanded trajectory well enough. The data also showed an oscillatory behaviour of the error. This was due to the Youbot omni-wheels having a parasitic compliance at low velocities. Both of these sources of

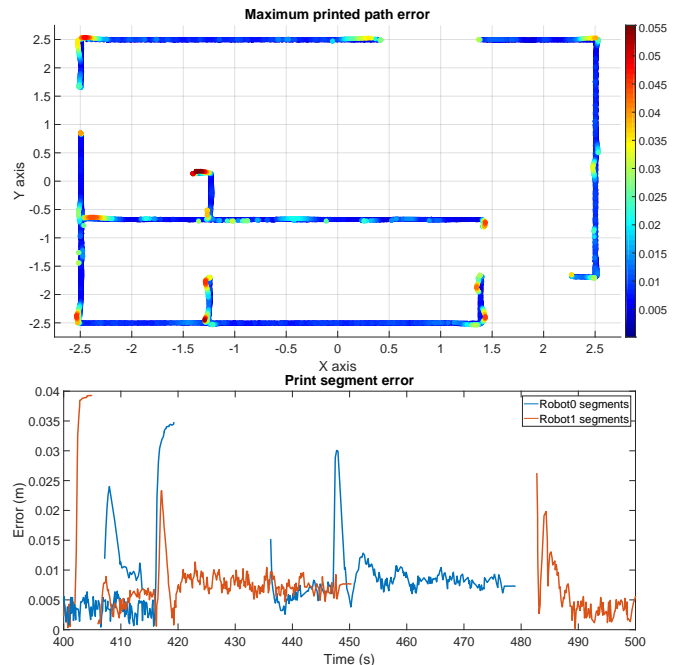


Fig. 12: **Top:** Cartesian error of the printed geometry. 3 sets of data superimposed with points of highest error value on top. **Bottom:** A 100s time segment throughout 3 trials of end effector position error whilst printing.

error showcase the need for multi-scale sensing and layered-control as described by Dogar[31]. Such *relative printing* (i.e. localising printing in absolute frame as well as frame relative to already deposited material), as well as base and arm motion being layered and prioritised differently, pose a great immediate area of investigation and improvement. Lastly, on average over trials the robots had to reposition themselves 8.5 times including travel between segments. As no unexpected collisions between robots or already deposited (virtually) material occurred, the collision aware motion solution was proved to be feasible and sufficiently adaptable to static and dynamic environment.

V. CONCLUSIONS

The need for further work in the application of layered control and multi-scale localisation were highlighted by shortcomings of the system. Further, thanks to an anonymous reviewer, an opportunity to adapt well established parallel computing metrics like speed up and efficiency can be added to near-term future work. However, by completing a physical print, in addition to *virtual printing* in a team, the YouWasp system have demonstrated competency as a research platform for mobile multi-robot material deposition. These contributions will provide a foundation for future work that incorporates adaptive autonomous strategies for planning and optimising construction tasks.

ACKNOWLEDGMENT

The presented work was supported by the Engineering and Physical Sciences Research Council (EP/N018494/1). We also thank Zhu Wu, UCL MSc Robotics and Computation class 2018, for his initial work on collision aware printing.

REFERENCES

- [1] Timetric, "KEY HIGHLIGHTS FROM GLOBAL CONSTRUCTION OUTLOOK 2021," Construction Intelligence Center, Tech. Rep., 2017.
- [2] HM Government, "Construction 2025. Industrial Strategy: Government and industry in partnership," *UK Government*, no. July, p. 78, 2013. [Online]. Available: <https://www.gov.uk/government/publications/construction-2025-strategy>
- [3] N. Labonnote, A. Rönquist, B. Manum, and P. Rüter, "Additive construction: State-of-the-art, challenges and opportunities," *Automation in Construction*, vol. 72, pp. 347–366, dec 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.autcon.2016.08.026>
- [4] T. Bock, "The future of construction automation: Technological disruption and the upcoming ubiquity of robotics," *Automation in Construction*, vol. 59, pp. 113–121, nov 2015. [Online]. Available: <https://doi.org/10.1016/j.autcon.2015.07.022>
- [5] Y. W. D. Tay, B. Panda, S. C. Paul, N. A. Noor Mohamed, M. J. Tan, and K. F. Leong, "3D printing trends in building and construction industry: a review," *Virtual and Physical Prototyping*, vol. 12, no. 3, pp. 261–276, jul 2017. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/17452759.2017.1326724>
- [6] M. Despeisse and S. Ford, "The role of additive manufacturing in improving resource efficiency and sustainability," in *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*, S. Umeda, M. Nakano, H. Mizuyama, H. Hibino, D. Kiritsis, and G. von Cieminski, Eds. Cham: Springer International Publishing, 2015, pp. 129–136.
- [7] X. Zhang, M. Li, J. H. Lim, Y. Weng, Y. W. D. Tay, H. Pham, and Q.-C. Pham, "Large-scale 3D printing by a team of mobile robots," *Automation in Construction*, vol. 95, pp. 98–106, nov 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0926580518304011>
- [8] V. M. Pawar, R. Stuart-Smith, and P. Scully, "Toward autonomous architecture: The convergence of digital design, robotics, and the built environment," *Science Robotics*, vol. 2, no. 5, p. eaan3686, 2017.
- [9] J. Sustarevas, D. Butters, M. Hammid, G. Dwyer, R. Stuart-Smith, and V. M. Pawar, "MAP - A Mobile Agile Printer Robot for on-site Construction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018, pp. 2441–2448. [Online]. Available: <https://ieeexplore.ieee.org/document/8593815/>
- [10] F. Bos, R. Wolfs, Z. Ahmed, and T. Salet, "Additive manufacturing of concrete in construction: potentials and challenges of 3d concrete printing," *Virtual and Physical Prototyping*, vol. 11, no. 3, pp. 209–225, 2016.
- [11] V. Helm, S. Ercan, F. Gramazio, and M. Kohler, "Mobile robotic fabrication on construction sites: Dimrob," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4335–4341.
- [12] M. Lussi, T. Sandy, K. Dorfler, N. Hack, F. Gramazio, M. Kohler, and J. Buchli, "Accurate and Adaptive in Situ Fabrication of an Undulated Wall Using an on-Board Visual Sensing System," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018, pp. 1–8. [Online]. Available: <https://ieeexplore.ieee.org/document/8460480/>
- [13] Z. Dakhli and Z. Lafhaj, "Robotic mechanical design for brick-laying automation," *Cogent Engineering*, vol. 4, no. 1, p. 1361600, 2017.
- [14] M. Yablonina, M. Prado, E. Baharlou, and T. Schwinn, *Mobile Robotic Fabrication System for Filament Structures*. UCL Press, 2017, pp. 202–209. [Online]. Available: <http://www.jstor.org/stable/j.ctt1n7qkg7.32>
- [15] I. Jokic, Sasa, I. Novikov, Petr, I. Shihui, Jin, I. Maggs, Stuart, I. Nan, Cristina, and D. I. Sadan, "Mini Builders," 2017. [Online]. Available: <https://iaac.net/project/minibuilders/>
- [16] S. J. Keating, J. C. Leland, L. Cai, and N. Oxman, "Toward site-specific and self-sufficient robotic fabrication on architectural scales," *Science Robotics*, vol. 2, no. 5, p. eaam8986, apr 2017. [Online]. Available: <http://robotics.sciencemag.org/lookup/doi/10.1126/scirobotics.aam8986>
- [17] N. Oxman, J. Duro-Royo, S. Keating, B. Peters, and E. Tsai, "Towards robotic swarm printing," *Architectural Design*, vol. 84, no. 3, pp. 108–115, 2014.
- [18] K. Petersen, R. Nagpal, and J. Werfel, "TERMES: An Autonomous Robotic System for Three-Dimensional Collective Construction," in *Robotics: Science and Systems VII*, 2011.
- [19] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "Ikeabot: An autonomous multi-robot coordinated furniture assembly system," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 855–862.
- [20] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013. [Online]. Available: <https://doi.org/10.5772/57313>
- [21] B. P. Gerkey and M. J. Mataric, "Sold!: Auction methods for multi-robot coordination," *IEEE transactions on robotics and automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [22] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [23] R. M. Zlot and A. Stentz, "An auction-based approach to complex task allocation for multirobot teams," Ph.D. dissertation, Carnegie Mellon University, 2006.
- [24] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.
- [25] "Kuka youbot," 2018. [Online]. Available: http://www.youbot-store.com/wiki/index.php/YouBot_Detailed_Specifications
- [26] "Wasp extruder system," 2018. [Online]. Available: <http://www.wasproject.it/w/en/wasp-launches-the-new-professional-clay-extruder/>
- [27] "Robot operating system," 2018. [Online]. Available: <http://www.ros.org/>
- [28] I. A. Sucas and S. Chitta, "Moveit motion planning framework," *ROS*, 2018. [Online]. Available: <http://moveit.ros.org/about/>
- [29] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *Proceedings of the IEEE RAS Humanoids Conference*, Seoul, Korea, November 2015.
- [30] "Youbot ros driver," 2018. [Online]. Available: https://github.com/youbot/youbot_driver_ros_interface
- [31] 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, no. 13, pp. 1645–1659, 2015.