2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Collaborative Manipulation and Transport of Passive Pieces using the Self-Reconfigurable Modular Robots Roombots

Stéphane Bonardi, Massimo Vespignani, Rico Moeckel, Auke Jan Ijspeert

***Abstract*— Manipulation and transport of objects using mobile robotic platforms is a well studied field with several successful approaches. The main difficulty while using such platforms is the lack of adaptation capabilities to changes in the environment and the restriction to flat working areas. In this paper, we present a novel manipulation and transport framework using the self-reconfigurable modular robots Roombots to collaboratively carry arbitrarily shaped passive elements in a non-regular 3D environment equipped with passive connectors. A hierarchical planner based on the notion of *virtual kinematic chain* is used to generate collision-free and hardware-friendly paths as well as sequences of collaborative manipulations. To the best of our knowledge, this is the first example of manipulation of fully passive elements in an arbitrary 3D environment using mobile self-reconfigurable robots. The simulated results show that the planner is robust to arbitrary complex environments with randomly distributed connectors. In addition to simulation results, a proof of concept of the manipulation of one passive element with two real Roombots meta-modules is described.**

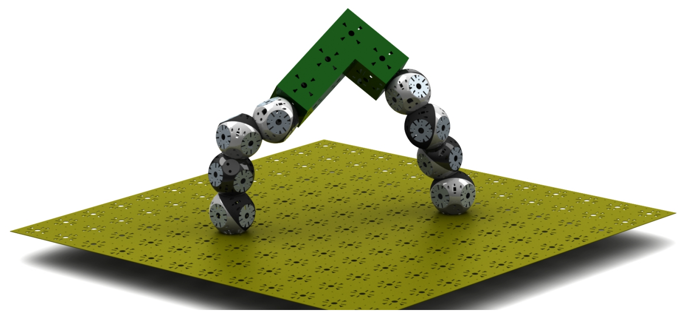


Fig. 1: Two meta-modules (two RB modules connected together) on a 2D grid collaboratively manipulate a L-shaped object (in green) equipped with passive connectors. The object is transported thanks to a sequence of manipulations and of meta-module on-grid locomotion.

I. INTRODUCTION

Modular robots, as opposed to monolithic ones, are composed of several homogeneous or heterogeneous units (often referred as modules) to improve the overall flexibility, adaptability and robustness of the structure to specific tasks in unknown environments. This modularity comes with the challenge of collaboration between the different modules to form the optimal configuration for a specific task.

Self-reconfigurable modular robots can create a large variety of kinematic structures depending on the applications. One possible use of this versatility is the creation of manipulators able to autonomously locomote in the environment using embedded connectors and to adapt to the object to be carried. Using their self-reconfiguration capabilities, these robots can efficiently move inside a structured environment and dynamically change shape to handle changes in the tasks (e.g. additional objects to be handled) or in the surroundings (e.g. new obstacles). Possible applications for such a system could be the automated construction of arbitrary structures or fully automated warehouses where modular robots are used to carry and store objects in shelves (for example, as a complement of the successful solution proposed by *KIVA systems* [1]).

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics. This project has also received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Future Emerging Technologies, Embodied Intelligence, under the grant agreement no. 231688 (Locomorph).

All authors are with the Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland. Corresponding authors: {stephane.bonardi, massimo.vespignani, rico.moeckel, auke.ijspeert}@epfl.ch

Our self-reconfigurable modular robot Roombots (RB) has been designed to be used as building blocks for adaptive pieces of furniture able to move, self-assemble and self-reconfigure. Using the reconfiguration capabilities of RB, we can study distributed locomotion control as well as self-organization and collaboration between modules [2].

A single RB module can autonomously travel to any position on a 2-dimensional grid by a sequence of connections and disconnections between the modules' active connection mechanism (ACM) and the grid structure (i.e. panels with regularly spaced connectors) and overcome concave edges in 3 dimensions.

In order to achieve our goal of furniture that can change shape to adapt to the user's needs, we have to be able to design efficient structures in terms of physical properties and cost. That is the main reason why we envision robotic furniture composed not only of active RB modules but also of passive elements, with the RB modules acting both as manipulators and as components of the structure (an example of the manipulation and transport phase of a passive element is presented in Fig. 1). In this application, a set of RB modules needs to perform on-grid locomotion to pass along passive objects. In order to build a heterogeneous structure using RB, we design a manipulation and transport framework that can be generalized to different self-modular reconfigurable robots able to use passive connectors to locomote. The requirement for environments equipped with connectors can be partially relaxed considering the off-grid locomotion capabilities of the RB platform [3]. In this paper, our goal is to find the sequence of servo movements and

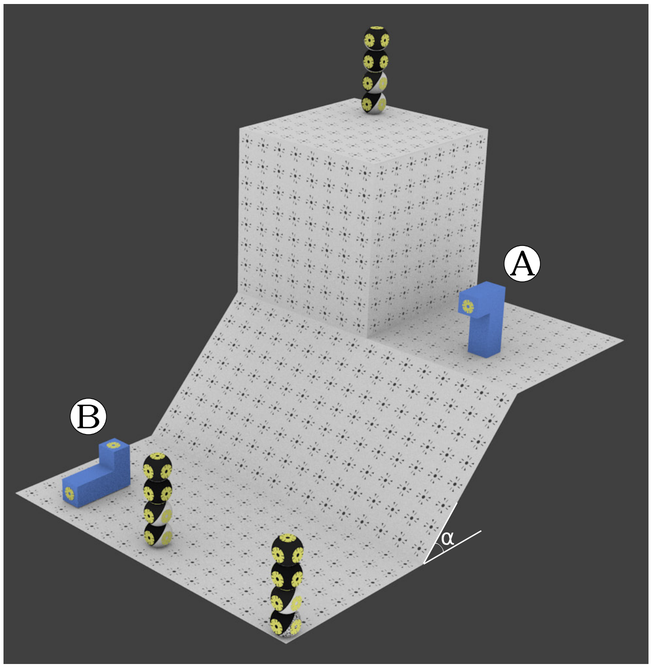


Fig. 2: Example of manipulation and transport scenario: three meta-modules have to carry an L-shaped object (in blue) with connectors, from a point *A* to a point *B*. This requires (i) that the meta-modules move by sequentially attaching and detaching to and from connectors in the environment (represented as black circle, randomly made available on the grid plates), (ii) that they attach to and manipulate the object, and (iii) that they collaborate to bring the object to the target position *B*

.

connections/disconnections for a group of active units to collaboratively carry a set of passive objects from an initial position to a final one in an arbitrary 3D non regular grid with obstacles (illustrated in Fig. 2).

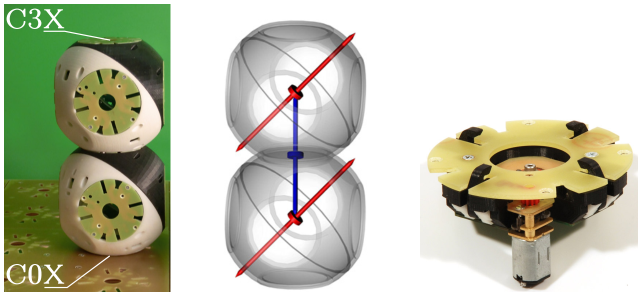
In section II we review some successful approaches in the field of objects manipulation and structures building using mobile and modular robots. We then briefly describe the RB platform (section III) and explain our manipulation architecture in section IV. We test our approach in simulation and describe afterwards a proof of concept using the RB hardware (section V). Finally, after discussing our results (section VI), we conclude and describe possible future works.

II. RELATED WORK

Manipulation and transport of objects using mobile platforms equipped with robotic arms is a well studied research area. However using reconfigurable modular robots for manipulation of passive objects have been scarcely explored so far. Terada et al [4] proposed a complete framework to build arbitrarily layered structures using a specialized manipulator with four Degrees Of Freedom (DOF) and specific building blocks. The robot uses inch-worm locomotion on the structure and occasionally rotation to change direction. The sequence of moves of the robot is controlled using a gradient approach and a local negotiation via blackboard to avoid collisions between several manipulators. One of the main limitation of this approach is the need for active connection mechanisms on the external faces of the elements being carried around, as opposed to the arbitrarily shaped fully passive elements we are considering. Additionally, the limited degrees of freedom of the manipulator constrains the structure to be built in a layered fashion as opposed to the fully 3D manipulation problem we are tackling. Another very successful approach has been proposed by Petersen et al [5]. The authors use mobile units to grab specially designed elements to built an arbitrary structure from a high level representation. The path chosen by the robots to go from the supply spot for passive elements to the goal position is determined using a depth-first search algorithm coupled with a set of rules to prevent inaccessible positions. The task of manipulation is simple since it mainly consists in deposing the piece in the given spot with a rotation of a one DOF actuator. In this case the complexity of the manipulation is shared between the manipulator and the design of the element. The main difference with our method is that we can easily transform everyday life objects into moveable objects simply by adding passive connector plates to them and connect active units in a plug and play fashion. This aspect brings more flexibility in the type of structures that can be built using the manipulation and transport method we present. Groß et al [6] presented a framework in which several Swarm Bots modular robots [7] collaborate to move an object from one position to another on a flat terrain. The modular aspect comes from the fact that the wheeled robots used can dynamically connect between each other using a gripper based mechanism, to form larger chains able to move (using traction) bigger objects. The main strength of the approach used in this paper is the careful experimental validation of the transportation task. The main limitation of the proposed approach is the difficulty for the platform used to locomote in irregular 3D environment (limitation to almost 2D terrain) as well as the use of pure traction to move the object. A large number of successful approaches have been developed to achieve displacement of modular robots to form arbitrary structures [8], [9], [10] but they only consider active units as building blocks. To the best of our knowledge, no mobile self-reconfigurable robots have been used before to manipulate fully passive elements in an arbitrary 3D environment equipped with connectors.

III. ROOMBOTS MODULE CONFIGURATION

A Roombots (RB) module is composed of four half-spheres (see Fig. 3a for precise shape) linked together using revolute joints with continuous rotation capabilities (depicted on Fig. 3b). Using four-way symmetric compact Active Connection Mechanisms (ACMs, up to 10 per module, illustrated in Fig. 3c) each RB module can autonomously connect and disconnect from another module or from a passive connector embedded in the environment. The ACM is genderless and non-back-drivable. In the remaining parts of this paper, we consider that only the most external connectors of a module



(a) (b) (c)

Fig. 3: (a) Single RB module. In (b) the three degrees of freedom of a RB module are depicted. (c) shows the current ACM design.

(C0X and C3X) are equipped with an ACM, the remaining eight being completely passive. A RB module is controlled through wireless communication and contains two Li-Po battery packs ensuring more than one hour of autonomy in full charge. Each module is driven by a set of distributed embedded electronics. A single module weights around 1.4kg and any of its joints can provide sufficient torque to lift at least an additional RB module. Two modules connected together form what we call a meta-module (MM). A MM has a payload of around 500g on the most external connector (C3X, described in Fig. 3a). The upper limit for the nominal torque of the two external DOFs of the RB module is around 4.9Nm whereas for the middle DOF, it is around 3.6Nm. A detailed description of the hardware can be found in [11].

IV. HIERARCHICAL PLANNER

The task that we are solving is to find the complete sequence of servo angles and connections/disconnections for a set of active elements to collaboratively bring a set of passive elements from an initial to a final position. We assume that the passive elements are not actuated and that they have to be always connected to at least one active unit during the transportation task. The former requirement arises from the need of maintaining the element at a given position before the next active unit connects to it. An equivalent solution would be to design holders at predefined points to store the pieces between two handling actions. Nevertheless we prefer to consider solutions that would require the least amount of extra facilities to solve the task we defined. The world (i.e. the available connectors and the obstacles, their position and orientation) is supposed to be known. The information about the shape and available connectors of the passive elements are also assumed to be known beforehand. No parallel motion with multiple active units is considered. As a consequence, the weight of the passive element should not exceed the possible payload of one active unit.

We decomposed the handling task into four main elements, (i) a low level kinematic planner, (ii) a motion planner, (iii) a path planning algorithm, and (iv) a handling method. Each of these components is incrementally added into the next one. This decomposition brings flexibility in terms of hardware platforms by decoupling the kinematic constraints from the high level planning.

A. Level 1: kinematic planner

Any assembly of Roombots modules and passive elements can be viewed as a set of kinematic chains. Despite the restriction on the actual version of the RB hardware (the fact that one meta-module can only lift one passive element), we used a very general representation of the kinematic chain of the structure to allow future generalizations. One module is represented by a 3 rotational DOF chain with 10 connection points. We derive the inverse kinematic solution using the iterative damped Levenberg-Marquardt algorithm [12] provided in the Rigid Body Dynamic library [13]. This algorithm, also called damped least-square (DLS) method, is an iterative minimization method close to the Gauss-Newton algorithm and the gradient method, but generally more stable. Using this technique, we can impose a complete final posture for any chain or tree configurations. Passive elements can be easily integrated into the structure as pure sets of connection points.

B. Level 2: motion planner

In order to find a collision free path between two postures of a given structure given by the previously mentioned kinematic planner, we use a variation of the classical Rapidly-exploring Random Trees (RRT-Connect [14]) motion planning algorithm available in the Open Motion Planning Library [15]. The search for a possible path is done discretely and the validity of every intermediate posture is evaluated. A posture is valid if two conditions are fulfilled:

1) The posture is collision free: we use the exact model of the hardware module and passive elements to compute the collision manifold of any structure.
2) The posture does not lead to impractical stress constraints on the servos. We compute for every posture candidate an approximation of the resulting torque on each servo and check whether this value is inferior to the nominal torque of the motor. To compute the torque estimate, we project each pivot point (corresponding to each motor) on the plane perpendicular to the gravity force and multiply this value by the distance L between this projected point and the projection of the center of mass of the remaining segments on the same plane: $T_{motor_i} = m_i * g * L_i$ (m_i corresponding to the mass of the remaining segments in the direction of the lever). This computation gives a crude upper-bound estimate of the real torque applied to the motor and neglects both the friction and the dynamics during the movement, since we consider a completely rigid structure and a fine grain discretisation of the movement of the robot leading to an almost static analysis. This overestimate of the torque will favour moves that prevent overstressing the hardware.

Additional constraints on the posture, such as orientation constraints for a carried object, can easily be added.

C. Level 3: path planner

The goal of this planner is to find the complete sequence of moves and connection/disconnection to go from one initial structure state (i.e. position, orientation, type of connection and posture) to a final one. The problem of finding a path on a 2-D grid can be viewed as a path-finding problem in a graph. The sequence of grid positions to go from the initial position to the final one is generated using the $A^\star$ algorithm, a popular algorithm for solving path planning in 2-dimensional grids [16]. This algorithm is based on the evaluation of a cost function f which takes into account the distance from the start position and a heuristic estimate of the distance to the goal position (often chosen to be the distance to the goal along a straight line).

$$\forall s = (x,y) \in Grid \quad f(s) = g(s) + h(s) \tag{1}$$

where $g(s)$ corresponds to the distance from the start position to the current position and $h(s)$ corresponds to an *estimate* of the distance to the final goal. h is defined in our case as the Euclidean distance from the current position to the goal position, in order to favour path with fewer and longer moves.

The search space S is composed of connector position p and orientation o as well as type of connection c (there are four main types of connections since the ACM is four ways symmetric):

$$\forall s \in S \quad s = (p,o,c) \text{ with } p \text{ and } o \in \Re^3 \text{ and } c \in [0..3] \tag{2}$$

For each state space in S, a neighbourhood of reachable states is computed based on the previous motion planner. This computation is done inside a sub-routine which can be modified to integrate further constraints.

D. Level 4: handling planner

In order to handle a passive element, we need to define two main parameters: (1) the connection points between the element and the handling active structures, (2) the postures of the active structures and their connection type to the grid. We use the notion of *virtual chain* (VC) to tackle this problem. A virtual chain is defined as a moveable structure composed of at least one active unit and one passive element. A structure is said to be moveable if it is not blocked (i.e. with elements around that would prevent movement) and if it possesses at least one active unit. We define four basic types of virtual chains (illustrated in Fig. 4) depending on the number of active units they are composed of. We assume that the passive element is at first not connected to the active units.

The displacement of one passive element e from a state $A \in S$ to a state $B \in S$ is planned as follows:

1) Depending on the number of active units available, we form the widest (in the sense of the wider kinematic space) virtual chain by connecting virtual active units to e. For example, virtual chain of type 3 would be favoured over virtual chain of type 2. The choice of the connectors on the passive element is made so that the length of the total virtual chain is maximized.

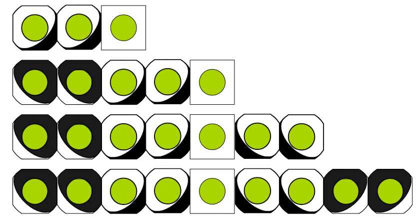


Fig. 4: The four different virtual chains. The passive element is represented by a square and the green circles correspond to the passive connectors.

2) Once we decide on a given VC, we consider the passive element as a fixed point and we use the motion planner previously defined to find the possible grid connection states (called $S_{available}$) for the active unit closer to the final position of the passive element.
3) We sort the available active units based on the distance from their current connection point to the center of the passive element. The available units configuration is fixed, meaning a meta-module cannot split to form two single modules.
4) We compute the path from the current position of the active units to the closest grid connection state in $S_{available}$ to determine if the structure is reachable using any of the active units. We iterate over the states in $S_{available}$ until we find a path or we switch to another active unit. If no solution is found, we change the type of virtual chain and restart the process from step 1.
5) If the passive element is reachable we can now compute the set of connector states towards the final state B. We use the path planner described in subsection IV-C without the torque limit constraint to provide servos angles and connection states from A to B. The validation function contains an extra constraint to ensure that any selected state is reachable by at least one active unit, tested in sorted order according to their euclidean distance to this grid state. This validation is based on the path planner from subsection IV-C including the complete set of constraints on the collision and the torque limit. The final state of the connected active unit is integrated as an obstacle to the collision world to ensure a collision free path for the second moving unit.
6) If the final state is not reachable using the current VC we switch to a smaller type and repeat from point 1.

The main steps of this manipulation routine are illustrated in Fig. 5.

V. EXPERIMENTAL RESULTS

Since we do not allow on the fly disconnection and reconnection of active units, we propose to test our manipulation method using a single passive object, two meta-modules and two single modules as active units. We consider a centralized implementation of the above method but a fully distributed version could be achieved, if we still consider that the map of the environment is known by every active

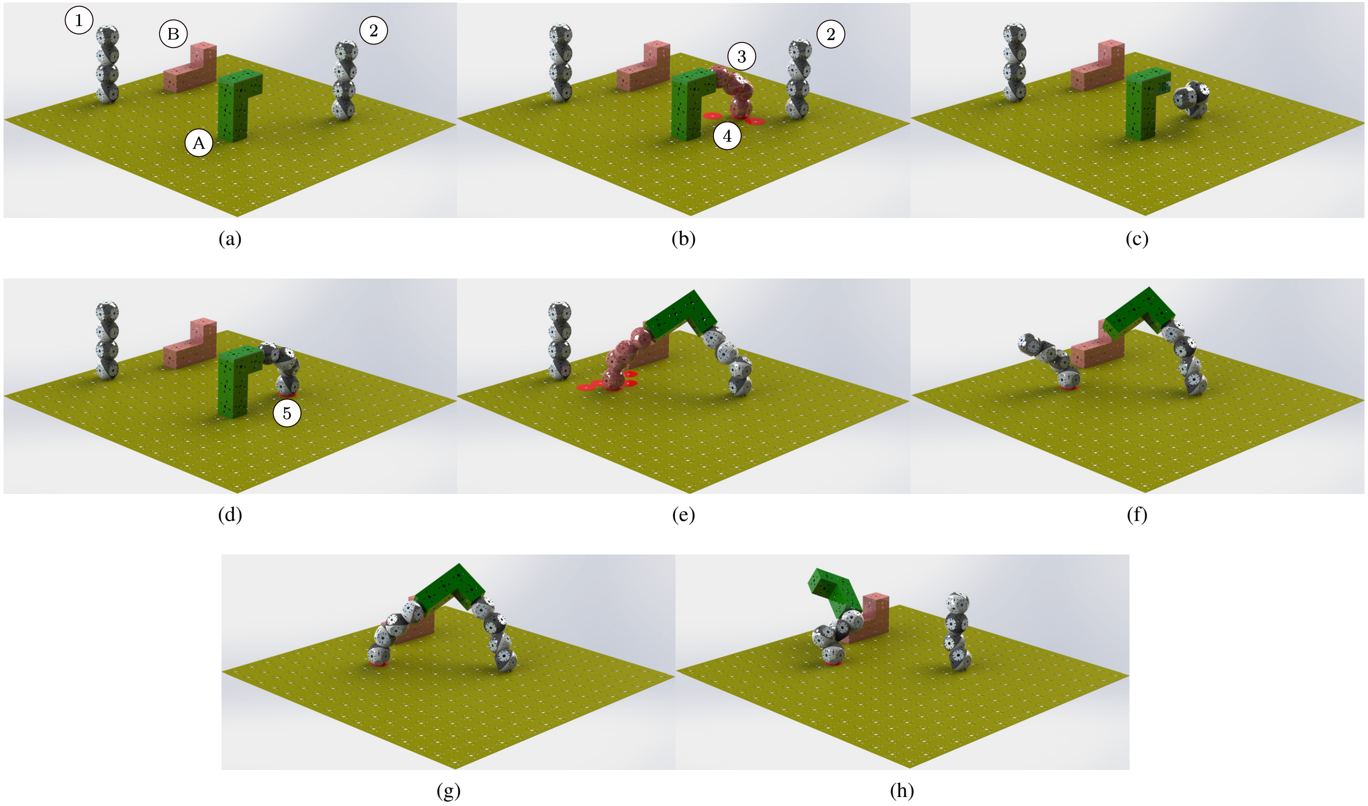


Fig. 5: The main steps of the manipulation routine, with one passive element and two meta-modules (labelled 1 and 2 in (a)). The connector are indicated by small white circles. The passive object is modelled as a green L-shaped element (labelled A in (a) in its initial position). The red transparent element in the different images represent the desired final state of the passive object and the virtual state of the active units (for example labelled 3 in (b)). In (a) we present the initial configuration of the terrain. The final position of the passive element is displayed in transparent red (labelled B). In (b) the red connectors correspond to the $S_{available}$ set (labelled 4 and mentioned at step 2 in the previous description) determined using the closest meta-module as active unit (labelled 2). (c) and (d) show respectively an intermediate state to get to the chosen connector (in red, labelled 5) by the first meta-module and the connection of the first meta-module to the passive element. (e) depicts the position of the virtual chain when checking the available connection point (red connectors) for the second meta-module (step 5). (f) and (g) show respectively an intermediate state to get to the chosen connector (in red) by the second meta-module and the connection of the second meta-module to the passive element. Finally, (h) represents the final move of the second meta-module to place the passive element into its final position and orientation.

unit beforehand. Using the real hardware, we illustrate one step of the handling algorithm we presented earlier using one passive element and two meta-modules.

A. Simulated results

We test our approach using our own simulation environment based on Open Scene Graph [17] and Bullet Physics [18]. We consider a terrain template (depicted in Fig. 2) composed of several initially perpendicular planes and a maximum of 458 connectors. We vary the number of connectors per terrain by introducing a probability p which determines whether a connector in the regular grid is available or not. We choose four values for p (0.2, 0.4, 0.6 or 0.8) and we randomly generate a set of 50 terrains per value of p by varying the angle of two of the main planes of the the terrain (angle α depicted in Fig. 2) in the range $[-pi/4; pi/4]$ radians as well as the final object position and orientation.

We consider the following quantities as an evaluation of the efficiency of the algorithm:

- Successful reaching or not of the final position.
- Number of moves needed to reach the final position: a move is considered as the sequence of servo positions between two connections. The number of moves corresponds to the number of connections.
- Average angular displacement of the active units for the completion of the task, computed using an estimate of the real time needed to perform a move assuming a constant angular velocity.

The results are summarized in Fig. 6 and Fig. 7.

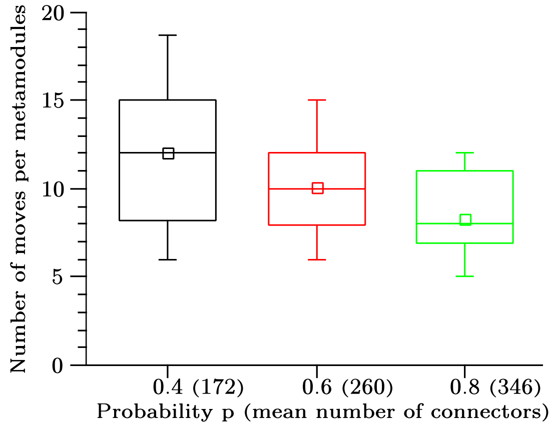


Fig. 6: Box-plot representing the number of moves for one meta-module during the successful runs of the algorithm for the different values of p.

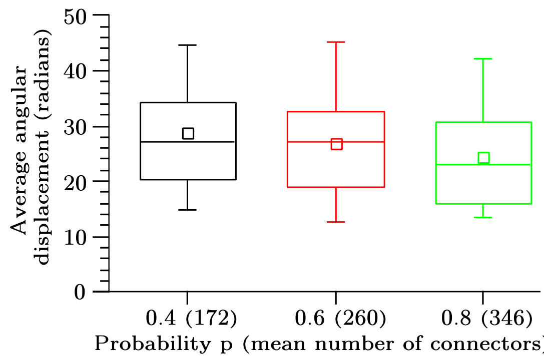


Fig. 7: Box-plot corresponding to the average of the modulus of the angular displacement for one meta-module during the successful runs of the algorithm for the different values of p.

B. Hardware results

We tested our framework using RB hardware modules[1]: we use a meta-module placed on a 2D grid to grab a passive cube of 110*mm* edge-length with connection plates on every side and hand it over to another meta-module attached to a grid of connectors placed above the first one (the setup is depicted on Fig. 8). The passive element is placed at its initial position in a holder that allows easy picking and avoid sliding of the passive element. The element is also maintained in position using small magnets. The positions of the meta-modules have been computed using the planner described in section IV. The experiment is performed in open-loop and the environment is fully known. In order to facilitate the alignment between the active connection mechanism and the connectors on the passive object and on the grid, we equipped every ACM and passive connector with small magnets. The magnets are used for guidance only and the connection/disconnection sequence is performed using the grippers of the ACM.

[1]This experiment is shown in the multimedia attachment. It should be noted that the two meta-modules were remotely controlled at low speed to avoid dynamic effects.

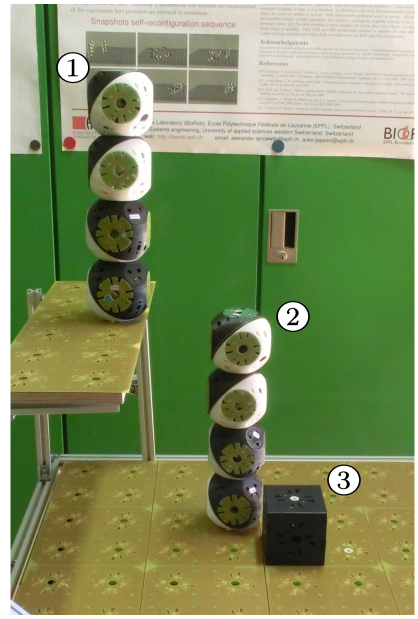


Fig. 8: The experimental setup: one meta-module is connected to the grid (number 1) above the second meta-module (number 2). A passive element (black cube, labelled as 3) is maintained in position using a magnet. Meta-module 2 will grab the passive element and hand it to the first meta-module.

VI. DISCUSSION

Throughout the simulated experiments described in subsection V-A, we observed that the chosen VC was always of type 3. This can be explained by the significantly bigger working space offered by the meta-module in comparison with the single module. We postulate that the use of the chain of lower type would arise only when considering a transport task in which the active units would be allowed to let the passive object on the ground, disconnect and reconnect to reach a previously inaccessible position (due for example to low hanging obstacles) and take back again the object.

Since the number of impossible worlds generated when the probability p was equal to 0.2 was too high to compare it to the other cases, we chose to discard the results related to this value. Some unsolvable worlds include those with no existing path to the final position (too spaced connectors) or with a passive object placed below the slope (when α is negative), in such a way that the passive element is inaccessible without creating collisions. Similarly, when the passive element is situated closed to one of the inclined surfaces, the complexity of the manipulation task increases. Given the degrees of freedom of the RB platform considered for the tests, we can discard some of the generated terrains considering they cannot be solved using the kinematic chains involved. Finally, the overall success rate of the algorithm was around 97% for all the solvable worlds generated. The reason for failure in some of the solvable worlds is due to the robot kinematic that prevents some moves in the given configuration of the terrain.

We can see on Fig. 6 that the number of moves required to reach the final position increases with the decrease in the number of connectors. This can be explained by the need for the active units to go back and forth on some positions before being able to reach a position with the correct orientation for the next step. When considering a sufficiently large number of connectors the effect of the heuristic function selected for the path planning (described in subsection IV-C) can be observed, with a significant decrease in the number of moves needed to reach the goal (the average number of moves to perform the manipulation task was 11 per meta-module). Nevertheless, the average angular displacement per meta-module (depicted in Fig. 7) remains almost constant for varying p. This can be related to the previous observation about the number of moves, since more small moves will be equivalent to less large moves in terms of angular displacement.

We also observed that the value of the angle α has no significant effect on the success rate or on the number of moves required. A possible explanation would be that the variation of the angle α does not induce a fundamental change in the topology of the terrain when moved. This topological stability of the terrain coupled with the randomness of the final position of the object, does not favour a given strategy in terms of number of moves or movement amplitude.

During the hardware experiment, we observed that the elasticity in the meta-module structures (at the level of the joints and the level of the ACM connected to the grid) induced a significant error in the final position of the connecting surface. That is the reason why we added magnets to provide the compensation needed to achieve a successful connection between the passive element and the meta-module ACM.

VII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We presented in this paper a complete collaborative manipulation and transport framework using self-reconfigurable modular robots to handle passive elements in a structured environment equipped with connectors. Our method is based on a hierarchical planner that uses the notion of virtual kinematic chain to compute waypoints and collision free paths. We also included an online computation of the applied torque to the different motors of the active units to favour hardware friendly moves. Our approach proved to be robust and efficient in arbitrary simulated environments, with a success rate of almost 100%. An example of a manipulation step using two RB meta-modules and one cube-shaped passive element has been successfully demonstrated.

B. Future Work

The manipulation of passive elements presented in this article is one of the main building blocks of a complete framework to allow the construction and deconstruction of arbitrary heterogeneous structures made of passive elements and active units. We are currently working on incorporating other collaborative behaviours such as parallel manipulation as well as dynamic assembly and disassembly of virtual chains to widen the range of objects that can be manipulated and transported. We are also working on optimizing the sharing of active units between handling tasks. An extension of our work would also be to use the active units to dynamically change their own environment by adding or removing connectors according to their needs.

VIII. ACKNOWLEDGMENT

The authors gratefully acknowledge the help of Dr Alexander Sproewitz, Emiliano Ramelli, André Guignard, André Badertscher, Peter Bruhlmeier, Philippe Voessler, and Manuel Leitos in the construction of the robot modules as well as Christophe Chariot for his help with the first firmware version.

REFERENCES

[1] K. Systems, "Kiva systems," http://www.kivasystems.com/.

[2] A. Spröwitz, S. Pouya, S. Bonardi, J. van den Kieboom, R. Möckel, A. Billard, P. Dillenbourg, and A. Ijspeert, "Roombots: Reconfigurable Robots for Adaptive Furniture," *IEEE Computational Intelligence Magazine, special issue on "Evolutionary and developmental approaches to robotics"*, vol. 5, no. 3, pp. 20–32, 2010.

[3] S. Pouya, J. Van Den Kieboom, A. Spröwitz, and A. Ijspeert, "Automatic gait generation in modular robots: to oscillate or to rotate? that is the question," *Proceedings of IEEE/RSJ IROS 2010, Taipei, Taiwan, October 18*, vol. 22, 2010.

[4] Y. Terada and S. Murata, "Automatic modular assembly system and its distributed control," *IJRR*, vol. 27, no. 3-4, pp. 445–462, March 2008.

[5] K. Petersen, R. Nagpal, and J. Werfel, "Termes: An autonomous robotic system for three-dimensional collective construction," *Proc. Robotics: Science & Systems VII*, 2011.

[6] R. Groß, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, "Object transport by modular robots that self-assemble," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2558–2564.

[7] F. Mondada, A. Guignard, A. Colot, D. Floreano, J.-L. Deneubourg, L. Gambardella, S. Nolfi, and M. Dorigo, "Swarm-bot: A new concept of robust all-terrain mobile robotic system," 2002.

[8] K. Stoy, "Using cellular automata and gradients to control self-reconfiguration," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 135–141, Feb. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889005001521

[9] K. Golestan, M. Asadpour, and H. Moradi, "A New Graph Signature Calculation Method Based on Power Centrality for Modular Robots."

[10] R. Fitch and R. McAllister, "Hierarchical Planning for Self-Reconfiguring Robots Using Module Kinematics," 2010.

[11] A. Sproewitz, "Roombots: Design and Implementation of a Modular Robot for Reconfiguration and Locomotion," Ph.D. dissertation, EPFL, Switzerland, 2010.

[12] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[13] M. Felis, "Rigid body dynamics library," http://rbdl.bitbucket.org/index.html.

[14] J. J. Kuffner Jr and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

[15] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December, http://ompl.kavrakilab.org.

[16] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[17] D. Burns and R. Osfield, "Open scene graph," in *Proceedings of the IEEE Virtual Reality*, 2004, p. 265.

[18] E. Coumans *et al.*, "Bullet physics library," *Open source: bulletphysics.org*, 2006.