Locomotion through Reconfiguration based on Motor Primitives for Roombots Self-Reconfigurable Modular Robots.

Stéphane Bonardi, Rico Moeckel, Alexander Sproewitz, Massimo Vespignani, Auke Jan Ijspeert
Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

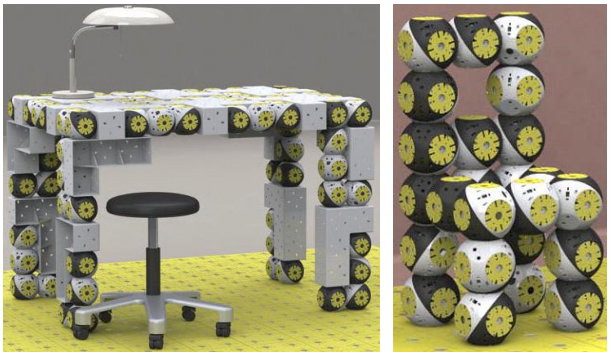
Abstract

We present the hardware and reconfiguration experiments for an autonomous self-reconfigurable modular robot called Roombots (RB). RB were designed to form the basis for self-reconfigurable furniture. Each RB module contains three degrees of freedom that have been carefully selected to allow a single module to reach any position on a 2-dimensional grid and to overcome concave corners in a 3-dimensional grid. For the first time we demonstrate locomotion capabilities of single RB modules through reconfiguration with real hardware. The locomotion through reconfiguration is controlled by a planner combining the well-known $D^\star$ algorithm and composed motor primitives. The novelty of our approach is the use of an online running hierarchical planner closely linked to the real hardware.

1 Introduction

The field of modular robotics is constantly evolving and comes with a variety of unique properties and challenges. In comparison to monolithic robots, modular robots are composed of a set of homogeneous or heterogeneous modules with the goal of making modular robots more flexible and adaptable to specific tasks in unknown environments.

Our modular robot Roombots (RB) has been designed to study three major challenges: (1) When being configured in chain or lattice structures we use RB modules as a rapid prototyping set for studies of distributed locomotion control in unknown terrains. (2) The self-reconfiguration capabilities of RB support the exploration of algorithms for self-organization, self-optimization and collaboration between modules. (3) The name "Roombots" refers to our goal of creating self-reconfigurable adaptive furniture, i.e. furniture that can move and change shape thanks to reconfiguration using dynamic connection mechanisms. RB are made for building reconfigurable living and working environments that adapt to the current needs of human beings. CAD drawings of furniture examples composed of RB modules and additional passive elements are presented in Fig. 1. Possible future scenarios include the application of adaptive furniture made of RB for assistive living, for example for elder care. A chair made of RB modules, like the one shown in Fig. 1b, can help someone getting up which can become a challenge in daily life of elder people. The same chair can follow a person or can be called if needed to assist somebody fallen to get up again or to support a person feeling tired. Roombots tables can assist by carrying objects like food or drinks. Beds made of RB modules can monitor a person while sleeping using force sensors. The reconfiguration capabilities of RB can furthermore improve the quality of life of people living in limited space. Furniture that become useless at a certain day time can decompose themselves into individual modules that can be stored in a compact way or that generate new pieces of furniture that are more suited to the current needs of the user. For examples modules forming a bed at night can create chairs and a table during the day. Thanks to the self-reconfiguration capabilities of RB these reconfigurations processes could later be performed with only little or even no human interaction.



(a) (b)

Figure 1: Rendered pictures of a table (a) and chair (b) composed of Roombots modules and passive elements.

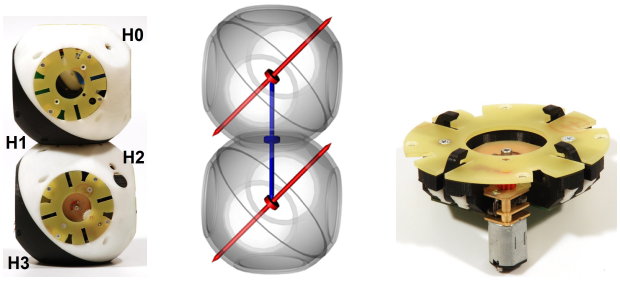
Through an extensive literature review we have identified more than 60 modular robot systems [1], designed in the last two decades. The most advanced systems include the M-TRAN series [2], PolyBot [3], ATRON [4], Molecubes [5] and SuperBot [6]. Only a small subset of those more than 60 modular robots incorporates a mechanism for self-reconfiguration that allows modules to autonomously connect and disconnect like RB since the design of a mechanism for self-reconfiguration in a compact

way is already a challenge on its own [7]. RB are designed with the property that a single module can autonomously travel through self-reconfiguration to any position on a 2-dimensional grid by a sequence of attachments and detachments between the modules' connection mechanism and the grid structure (i.e. panels with regularly spaced connectors) and to overcome concave edges in 3 dimensions with a minimum number of three degrees of freedom (DOFs). To the best of our knowledge this has never been demonstrated in hardware by any other modular robot before using fully autonomous modules with only 3 DOFs and including the mechanism for self-reconfiguration. Contrary to RB, M-TRAN [2], Molecubes [5], and ATRON [4] robots require more than a single module to change their direction of motion on a 2D grid. More than one RB module is only needed to overcome convex edges in 3-D configurations. In the presented experiments we control RB through a hierarchical, online running locomotion-through-reconfiguration planner that is based on the $D^\star$ algorithm [8] and *composed motor primitives*. A state-of-the-art reconfiguration planner that takes the kinematics of modular robots into account has been presented by Fitch et al. [9] in simulation. However, this planner has not yet been demonstrated on real hardware. In fact, so far most hardware experiments have been using pre-computed reconfiguration sequences [2, 10]. The major advantage of our online planner over these techniques is that we can take into account online changes of the environment and in the future autonomously incorporate readings of sensors.

In section 2 we briefly present the RB module configuration while section 3 presents the RB movement planner allowing a RB module to reach any position on a 2-D grid. Section 4 discusses experimental results and section 5 concludes and describes future work.

2 Roombots hardware specification

A single RB module is composed of two cube-like elements each with an edge length of 110 mm (Fig. 2a). Each module has three continuous rotational DOFs (Fig. 2b) and up to ten four-way symmetric, hermaphrodite Active Connection Mechanisms (ACM, Fig. 2c) allowing RB modules to autonomously connect to and disconnect from other RB modules and grid structures like the one shown in Fig. 4. A RB module weights about 1.4 kg. Any joints of a RB module can deliver sufficient torque to lift an additional RB module. A detailed description of the hardware can be found in [1].



(a) (b) (c)

Figure 2: (a) Single RB module. In (b) the three degrees of freedom of a RB module are depicted. (c) shows the current ACM design.

3 Planner

The goal of our planner is to compute a path (not necessarily optimal) on a 2-D grid from a start (S) to a goal (G) position. We built a hierarchical planner based on $D^\star$, a well established low-level path planning algorithm [8]. On top of it we added a high level planner which transforms the path computed by $D^\star$ into a sequence of basic "movement primitives". In section 3.1 we describe the algorithm used to compute the shortest path inside our 2-D grid. In section 3.2 we define the motor primitives alphabet on which we base our high level planner presented in section 3.3.

3.1 Low level planner

The problem of finding a path on a 2-D grid can be viewed as a path-finding problem in a graph. We consider a grid in which regular obstacles can be inserted but we assume that the dimensions of the grid are constant and the positions of the obstacles are fixed. One popular algorithm for solving path planning in 2-dimensional grids is the $A^\star$ algorithm [11]. Unfortunately, this algorithm is not really efficient to manage movable obstacles: a re-planning of the path is executed each time an obstacle is added or moved. Since we would like to allow obstacles moving (e.g. to represent other moving modules), we decided to use the $D^\star$ algorithm [8] which is based on an evolved version of the $A^\star$ [12]. A more detailed comparison between these algorithms can be found in [13].

3.2 Motor primitives

In order to ease the control of a RB module, we introduce a set of basic moves called *motor primitive*. Our goal is to perform locomotion through reconfiguration by using a sequence of attachments and detachments of the module on a 2-D grid. Although a RB module can contain up to 10 ACMs, in our experiments we consider only one ACM per outer hemisphere ($H0$ and $H3$, represented in Fig. 2a). When a RB module is connected to a grid using the ACM in $H0$ or $H3$, it can only move in one of two directions

(Fig. 4b). These two orthogonal directions form a coordinate system, the *relative coordinate system* (R^i_{rel}), where i is either 0 or 3 and corresponds to the connected hemisphere.

3.2.1 Atomic motor primitives

We define an *atomic* motor primitive (AMP) as a set of servo motor angles allowing the module to translate by a distance of one unit of the grid. The translation is represented by a vector τ. The direction of this translation is parallel to one of the axis of R^i_{rel}. During an AMP the two relative coordinate systems are inverted (R^0_{rel} becomes R^3_{rel} and vice versa), and an absolute rotation ρ between them takes place. An AMP is fully characterized by the couple (τ, ρ). We define four different AMPs valid when H_0 is connected to the grid and their equivalent when H_3 is connected. The main difference between P^0_i and P^3_i is the order in which the servo angles are sent to the module. The AMPs will be denoted by P^j_i with $i \in [1..4]$ and $j \in \{0, 3\}$. The four AMPs for H_0 connected to the grid are summarized in Table 1.

	Atomic motor primitives			
	P^0_1	P^0_2	P^0_3	P^0_4
$\boldsymbol{\tau}$	$(0, 1)$	$(0, 1)$	$(1, 0)$	$(1, 0)$
$\boldsymbol{\rho}$	π	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	$-\pi$

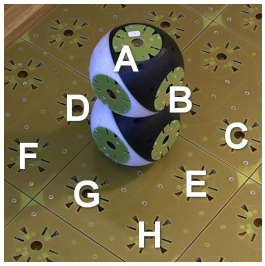
Table 1: The four different atomic motor primitives for H_0 connected to the grid.

3.2.2 Composed motor primitives

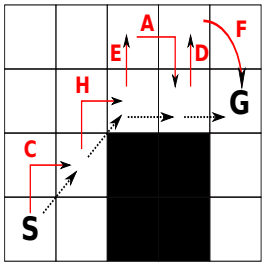
We introduce a set of *composed* motor primitives (CMP), defined as the concatenation of one or more AMP, to simplify the planning of the sequence of motor primitives to move from the start to the goal position. We define eight CMPs, that represent the motor primitive *alphabet*, to cover the eight direct neighbour cells (A-H shown in Fig. 3a) on a flat grid. The sequence of AMP composing the CMP alternate between P^0_i and P^3_i, following the connection and disconnection of the two RB hemispheres. For each CMP we defined the notion of a *spanning area* (SA), which corresponds to the grid positions crossed by the module during the execution of this CMP. This notion will be used later for checking obstacle avoidance.

3.3 High-level planner

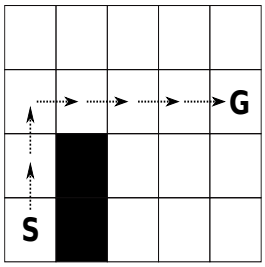
We introduce a high level planner to find the sequence of CMPs required to follow the path found by the low-level planner. From the set of grid positions found by the low level planner, the high level planner computes the sequence of CMPs allowing the module to follow the path toward the final position. Using the low level planner, we are able to find the set of grid positions that needs to be crossed to reach the final position.



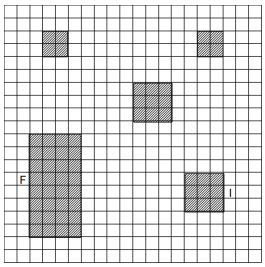
(a) The AMP alphabet.



(b) One planner trial.



(c) A world with no path.



(d) World with obstacles.

Figure 3: (a) shows the grid with a Roombots module and the eight nearest grid positions (labelled from A to H). In (b) an example of planning result is depicted: the goal position, G, is reached from the start position S using the CMPs C, H, E, A, D, and F (in red) based on the path found by the $D^\star$ algorithm (represented by dotted black arrows). The black area represents an obstacle in the grid. (c) illustrates a world in which no CMPs path can be found due to the kinematic constraints of the RB modules (the obstacle prevents the module to get to the grid positions above the starting position S). The dotted arrows indicates the path found by the $D^\star$ algorithm. (d) Fixed world with obstacles used in experiment 3.

The main steps of this high-level planner can be summarized as follows:

1. Find the shortest path $C = (c_0, c_1, ..., c_n)$ to the goal position using the low-level planner (c_0 is the initial position, c_n the final position and $c_{i\in[1..n-1]}$ are the intermediate positions).
2. For every point $c_i \in C$ starting from the initial position, use the CMP that follows direction $c_i c_{i+1}$.
3. For every selected CMP, check whether its spanning area intersects with an obstacle.
   - If *yes*, compute the set of neighbouring positions of c_{i+1} and select the reachable one (denoted as N_r). For every point in N_r:
     - Modify the initial path to incorporate this new point.
     - Find the new CMPs matching this new path.

If no points lead to a valid sequence, go back to the previous location c_{i-1} and repeat the process.

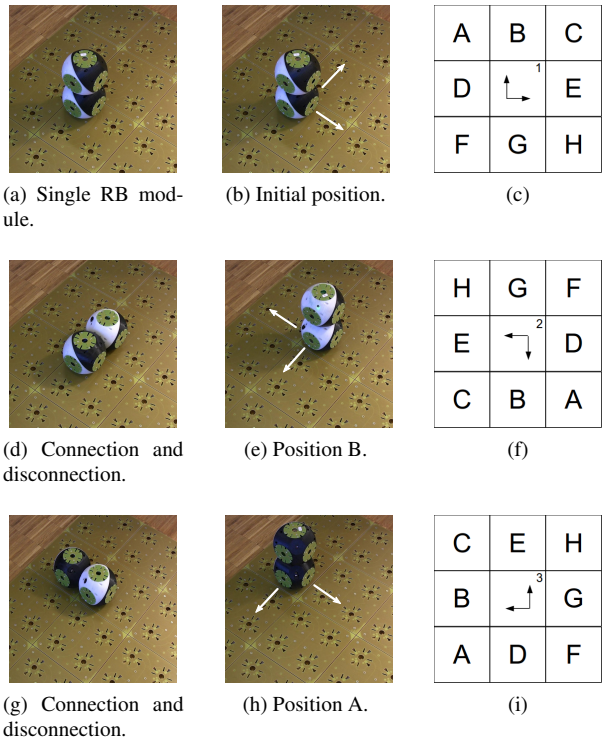
- Otherwise, continue.

The process ends when the goal position has been reached.

4 Experimental results

4.1 Hardware experiments

We extensively tested the RB hardware performing both locomotion through reconfiguration experiments on horizontal and vertical grids as well as RB modules performing a transition between horizontal and vertical grids (see Fig. 5 for an illustration). Movies can be found at the Roombots website [14]. We concentrated first on open-loop experiments: RB modules are PID position-controlled through relative position sensors at each joint but no additional sensors for example for sensing the alignment of a module with the grid have been used.



(a) Single RB module. (b) Initial position. (c)

(d) Connection and disconnection. (e) Position B. (f)

(g) Connection and disconnection. (h) Position A. (i)

Figure 4: Composed motor primitive A (white arrows indicate the 2 possible directions of movements): (a) RB module on a 2-D grid. (b), (d), (e), (g), and (h) illustrate the RB module following CMP A. (c),(f), and (i) show the relative referential and CMP alphabet for RB module configuration in (b), (e), and (h), respectively.

We tested all sequences to reach all neighbouring positions on a horizontal 2-dimensional grid shown in Fig. 3a. Fig. 4 shows snapshots from the CMP sequence A. For locomotion on a horizontal 2-D grid the RB hardware is sufficiently reliable. The design of the ACM supports the modules during reconfiguration to overcome elasticity in the joints and connectors as well as backlash in the gear boxes and supports self-alignment of a module with the grid without the need of additional control or sensing. We had only 1 out of 20 connection trials failed (success rate of 95%) where the gripping range of the ACM was not sufficient to overcome elasticity in the module's joints. This was typically because the ACM that was supposed to form a connection with the grid was slightly rotated with respect to the grid so that only one or two of the four ACM grippers could grip into a hole on the grid while the other grippers collided with the grid.
When climbing on vertical surfaces the connection process fails more often since gravity is bending the modules due to the elasticity in the joints, RB shells and connectors. To improve the performance and further increase the success rate, we are currently working on an improved RB design featuring less elasticity in the joints and a bigger ACM gripping range. We are also working on bending detection using infrared distance sensors and active compensation.

4.2 Planner results

In order to test our planner, we performed three different types of experiments in a simulated environment representing a 20×20 regular 2-D grid. The initial condition of the module (orientation, values of the degrees of freedom,...) is the same in all the experiments.
In the *first experiment*, we exhaustively tested the planner by trying to reach all the positions around the initial position of the module, within a range of 2 grid units. We did not include any obstacles on the grid. The success rate for this experiment was 100%.
In the *second experiment*, we generated a single squared obstacle of a dimension randomly chosen between 1 and 15 grid units that we randomly placed on the grid. The start and goal position of the module were randomly chosen as well. We generated and tested 300 worlds. The success rate of the planner is 100% for worlds that contain at least a solution. The worlds for which the planner was not able to find a path are in fact worlds with no existing path between the initial and the final position (see Fig. 3c for an example of such a world).
In the *third experiment*, we fixed the number of obstacles, their dimensions and their position (as illustrated in Fig. 3d) and we randomly chose the start and goal position of the module. We performed 300 trials. The success rate of the planner was 70% on average. The worlds in which no paths were found correspond to those where either the goal position or the start position of the module were at the boundaries of an obstacle and/or of the world so that the module could not leave or reach this position due to kinematic constraints of the RB module. Although we successfully presented climbing experiments with the

(a) Initial position. (b) (c) (d)

(e) (f) (g) (h) Final position.

Figure 5: By a well coordinated sequence of DOF movements that we calculated using inverse kinematics a single RB module can approach a concave corner and climb a wall. This experiment was done in open loop. We could not climb further than position (h) because of elasticity effects in the RB hardware.

real hardware, the planner is currently not capable of using motor primitives for approaching or getting away from walls. Thus all initial and final positions with a distance of less than one cell from an obstacle or world boundary cannot be reached or left with the current state of the planner. We will be able to increase the success rate once we include the same motor primitives in the planning process that we use to approach and leave obstacles for overcoming concave corners. Other world configurations where the planner failed to find a path included those where the initial position of the RB module was placed so that its two possible directions of motion where blocked by an obstacle and/or by the border of the grid, as illustrated in Fig. 3c.

5 Conclusion and future work

We presented the autonomous self-reconfigurable modular robot Roombots (RB). RB are designed with the property that a single module can fully autonomously travel through self-reconfiguration to any position on a 2-dimensional grid with a minimum number of three degrees of freedom. We presented a simple but effective online locomotion-through-reconfiguration planner based on the $D^\star$ algorithm and composed motor primitives that is closely linked to the real hardware and allows steering RB modules on a grid by simply giving a goal position. We presented experimental results illustrating the reliability of RB moving on 2-dimensional grids.

Although RB are already working reliably on a horizontal 2-D grid we are currently optimizing the RB hardware to improve its climbing capabilities. This work includes the increase of the gripping range of the RB ACMs, the removal of unwanted elasticities and the addition of further sensors allowing closed-loop reconfiguration experiments. This will help RB to more reliably climb on vertical planes and ceilings. We are also working on optimizing the sequence of CMPs by removing unnecessary rotations. Furthermore we are working on a more elaborated reconfiguration planner supporting collaboration between multiple RB modules.

6 Acknowledgements

The authors gratefully acknowledge the technical support of Christophe Chariot, Andre Guignard, and Andre Badertscher. This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics. This project has 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).

References

[1] A. Sproewitz, *Roombots: Design and Implementation of a Modular Robot for Reconfiguration and Locomotion*. PhD thesis, EPFL, Switzerland, 2010.

[2] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 431–441, 2002.

[3] M. Yim, D. Duff, and K. Roufas, "PolyBot: a modular reconfigurable robot," in *Proc IEEE Int Conf Rob Autom.*, vol. 1, pp. 514–520, 2000.

[4] E. H. Ostergaard, K. Kassow, R. Beck, and H. H. Lund, "ATRON lattice-based self-reconfigurable robot," *Autonomous Robots*, vol. 21, no. 2, pp. 165–183, 2006.

[5] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and Designed Self-Reproducing Modular Robotics," *IEEE Trans. Robotics*, vol. 23, pp. 308–319, April 2007.

[6] W. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh, "Multimode locomotion via SuperBot reconfigurable robots," *Autonomous Robots*, vol. 20, no. 2, pp. 165–177, 2006.

[7] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular Self-Reconfigurable Robot Systems," *IEEE Robot. Automat. Mag.*, vol. 14, pp. 43–52, March 2007.

[8] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *ICRA*, pp. 3310–3317, IEEE, 1994.

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

[10] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji, "Distributed adaptive locomotion by a modular robotic system, M-TRAN II," in *IROS*, pp. 2370–2377, 2004.

[11] 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.

[12] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004.

[13] D. Mackay, "Path planning with d*-lite," *DRDC Suffield TM*, vol. 242, 2005.

[14] Roombots website, www.roombots.org, 2011.